EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Research in Computing Science

Vol. 152 No. 8
August 2023

Research in Computing Science

Research in Computing Science, Año 22, Volumen 152, No. 8, agosto de 2023, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de agosto de 2023.

Research in Computing Science, year 22, Volume 152, No. 8, August 2023, is published monthly by the Center for Computing Research of IPN.

Volume 152(8)

Advances in Artificial Intelligence

Nestor Velasco Bermeo (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2023

ISSN: in process

Formerly ISSNs: 1870-4069, 1665-9899

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

Esquema de aprendizaje híbrido de agentes colaborativos en videojuegos MOBA

José A. Torres León, Marco A. Moreno Armendáriz,
Francisco Hiram Calvo Castro

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

{jtorresl2019,mam_armendariz,hcalvo}@cic.ipn.mx

Resumen. En este artículo se presenta un esquema de aprendizaje híbrido, que busca dividir el problema de aprendizaje para un equipo de agentes jugadores de videojuegos del género Multiplayer Online Battle Arena (MOBA). Este esquema se basa en un análisis por módulos de la tarea colaborativa, la cual se descompone en situaciones de juego. Hacer el análisis a nivel de situación de juego permite dividir el problema de aprender un comportamiento colaborativo del equipo de agentes en distintas sub-tareas de aprendizaje, donde la solución a cada una de ellas requiere de un enfoque específico, ya sea supervisado, no supervisado o reforzado, que se adecue mejor a la estructura de los datos en dicha sub-tarea. Este nuevo esquema de aprendizaje híbrido debería de sentar las bases para entrenar un equipo de agentes jugadores de videojuegos MOBA capaces de resolver una misma situación de juego para múltiples videojuegos del género.

Palabras clave: Inteligencia artificial, aprendizaje automático, agentes inteligentes, tareas colaborativas.

Hybrid Learning Scheme of Collaborative Agents in MOBA Videogames

Abstract. In this paper a hybrid learning scheme is presented, which aims to divide the learning problem for a video games player agents team of the Multiplayer Online Battle Arena (MOBA) genre. This schema is based in a per module analysis of the collaborative task, which decomposes into game situations. By doing this analysis at game situation level, it is possible to divide the problem of learning a collaborative behavior of the agents team into different learning sub-tasks, where the solution to each one of them requires a specific approach, whether it's supervised, unsupervised or reinforced, which adequates better to the structure of the data of that sub-task. This new hybrid learning scheme should lay the foundation to train a team of MOBA video games player agents team capable of solving a game situation for multiple video games of the genre.

Keywords: Artificial intelligence, machine learning, intelligent agents, collaborative tasks.

1. Introducción

Actualmente, uno de los problemas abiertos de la inteligencia artificial (IA) son las tareas colaborativas, las cuales requieren de un alto grado de inteligencia, tal como lo mencionan los autores de [1] . Para estos problemas, un equipo de agentes inteligentes necesita colaborar para encontrar la solución a una tarea, coordinando sus acciones y haciendo predicciones a largo plazo. Como caso de estudio, se seleccionó el género de videojuegos Multiplayer Online Battle Arena (MOBA), que, como se explica en [5], presenta desafíos colaborativos interesantes, como la limitación de los agentes a disponer de información incompleta del juego, planear estrategias, entre otras.

La meta global de estos videojuegos es destruir la base enemiga. Para lograr esa meta global, el equipo debe coordinar sus acciones para resolver una variedad de sub-tareas, para alcanzar metas individuales que incrementarán la probabilidad de una victoria. La estrategia involucrada en estos videojuegos es dirigida por las particularidades de las sub-tareas, en otras palabras, el equipo que tiene un buen desempeño en cada tarea, tendrá una mayor probabilidad de ganar.

En este caso de estudio, dichas sub-tareas se conocen como "situaciones de juego". Los jugadores humanos detectan estas situaciones y se adaptan a ellas de manera natural, fluyendo con el ritmo del juego. Sin embargo, aún se investiga para lograr que equipos basados en algoritmos de IA logren un desempeño similar. En este trabajo, se presenta un esquema de aprendizaje automático (Machine Learning, ML) híbrido enfocado en aprender cómo lidiar con el ajuste dinámico a las situaciones para resolver las tareas colaborativas complejas presentes en los videojuegos del género MOBA.

La mayor parte de las propuestas existentes en el estado del arte que proponen equipos de jugadores artificiales de videojuegos MOBA, que se presenta en la sección 2, se basan principalmente en modelos de aprendizaje por refuerzo profundo para resolver las predicciones a largo plazo y el modelado de las sub-tareas, lo cual significa que no se hace un modelado explícito de estos fenómenos, sólo se infieren por las arquitecturas profundas reforzadas.

Por ello, se busca dividir el problema de aprendizaje en diferentes módulos, cada uno enfocado en resolver una parte específica de éste y luego, al acoplarlos, podrían solucionar la tarea completa. Al abordarlo así, se pretende partir la tarea principal de aprendizaje en diferentes sub-tareas. Además, estos modulos podrían trabajar con diferentes versiones de la tarea, ya que no estarán asociados a un videojuego específico, sino a un género entero, por lo tanto, esta arquitectura podría generalizar el procesamiento por sub-tareas para diferentes videojuegos.

1.1. Los videojuegos del género MOBA

Se trata de un género de videojuegos de estrategia en tiempo real en equipo, donde dos equipos, de entre 3 y 5 miembros cada uno, se enfrentan en un mapa para tratar de destruir la base enemiga. La base de cada equipo se compone por un conjunto de estructuras, típicamente torres y la base central. Las torres se distrubyen a lo largo del mapa, el cual se divide en tres líneas principales. En cada línea hay un conjunto de torres de cada equipo, por lo que, para ganar la partida, estas torres deben destruirse para llegar a la base central.

Tabla 1. Relación entre fases y situaciones de juego.

Fase de juego (por nombre)	**Situaciones de juego que pueden suceder (por número)**
Selección y prohibición de personajes	-
Apertura	1, 2, 3, 4, 5
Líneas (laning)	6, 7 ,8
Juego medio	3, 4, 9, 10, 12, 13
Juego tardío	3, 4, 9, 10, 11, 12, 13

En la zona entre líneas o jungla, se reparten diversos sub-objetivos, como enemigos neutrales y objetos, que otorgan beneficios al equipo que los consigue. Los miembros de cada equipo deben elegir un personaje o avatar, que usarán a lo largo de la partida. Los personajes pertencen a uno o varios roles, los cuales determinan el tipo de habilidades que tendrá cada personaje. En los MOBA existen entre 5 y 12 roles, a continuación, se definen los que permanecen constantes a lo largo de todos ellos:

1. Carry o atacante. Su papel es generar mucho daño en poco tiempo a los avatares enemigos.
2. Jungla o ágil. Su objetivo es recorrer la jungla consiguiendo los sub-objetivos presentes en ella y asistir a los demás jugadores en las líneas.
3. Tanque o defensivo. Su rol consiste en defender a las torres y a los jugadores aliados, absorbiendo la mayor cantidad de daño posible en las peleas.
4. Soporte o curador. Su tarea es ayudar al resto del equipo, haciéndolos más fuertes y/o resistentes.
5. Guerrero o todo terreno. Su meta es soportar suficiente daño en las peleas mientras que genera una buena cantidad de daño a los enemigos.

Además del objetivo global de destruir la base enemiga, cada jugador tiene la meta individual de maximizar la experiencia y recursos, los cuales le sirven para aumentar los efectos de sus habilidades. La partida inicia con cada jugador en la base central de su equipo y termina cuando una base central es destruida o cuando se acaba el tiempo límite.

Durante la partida, cada jugador recibe información parcial del mapa, ya que sólo saben lo que hay a un cierto radio de distancia de dónde se encuentra su personaje y lo que hay a cierto radio de las torres, el resto de objetos que permanecen fuera de esos radios, son desconocidos para los jugadores. Además de los avatares, típicamente cada equipo cuenta con un ejército de súbditos o creeps, que son unidades que aparecen cada cierto tiempo en la base principal y que se mueven y atacan al equipo contrario de forma automática.

Cada partida dura entre diez minutos y hora y media, dependiendo del videojuego en cuestión y de la habilidad de cada equipo. Dadas estas condiciones, se ha identificado a este género de videojuegos como una plataforma ideal para probar algoritmos de inteligencia colaborativa, puesto que es necesario hacer predicciones a largo plazo, con información incompleta, mientras que se coordinan las acciones de cada miembro del equipo para lograr la meta global, además, cada jugador busca desempeñarse de acuerdo al rol de su personaje y de cumplir con los objetivos individuales.

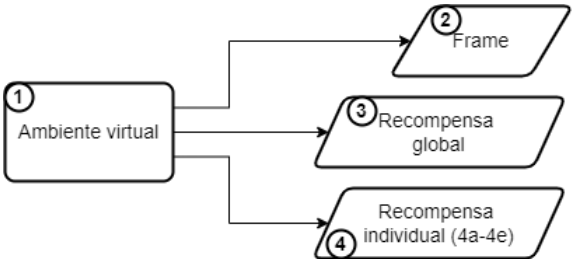


Fig. 1. Diagrama del ambiente virtual.

1.2. Fases y situaciones de juego

En esta sección se describen las fases de una partida estándar de un videojuego MOBA según lo descrito en [2], enlistando cuales son las situaciones de juego particulares que pueden presentarse en cada fase. A continuación, se definen las fases de juego:

1. Selección y prohibición de personajes. En esta fase cada miembro de cada equipo selecciona a su personaje, además, eligen un conjunto de personajes que los rivales no podrán usar en la partida. Es una fase previa a la partida o el juego como tal.
2. Apertura. Esta fase inicia cuando empieza la partida y termina cuando los creeps han llegado a la mitad de cada línea. Esta fase consiste de dos partes, la compra inicial de objetos (en caso de que el videojuego lo incluya) y el posicionamiento de los avatares en las líneas.
3. Líneas (Laning). Esta fase inicia cuando los creeps han llegado a la mitad de cada línea y consiste en quedarse dentro de los límites de la línea tratando de recolectar recursos. Esta fase termina cuando un avatar tiene suficientes recursos para obtener objetos valiosos o tiene una gran diferencia de niveles con respecto al resto de los avatares.
4. Medio juego. Se trata de la fase donde la estrategía toma forma después de la fase de líneas. Esta fase cuenta con desafíos muy diversos, sobresaliendo la cooperación y el razonamiento en tiempo real, donde la sinergia del equipo es vital para definir el rumbo de la partida. Esta fase termina cuando todos los avatareas están en nivel alto y, en caso de que el videojuego lo incluya, sólo les falta comprar uno o dos objetos para equipar.
5. Juego tardío. Es la última fase del juego. Inicia al finalizar el medio juego. En esta etapa, los combates y las estrategias se vuelven más complicados dado que todos los personajes han alcanzado su máximo potencial. Esta fase termina cuando un equipo es derrotado.

Por su parte, las situaciones de juego son los desafíos individuales que pueden presentarse en cada fase de juego. Nótese que las situaciones de juego no tienen un flujo predeterminado, así que aunque se presenten en un cierto orden en la lista, no necesariamente se presentan siempre las mismas en el mismo orden en todas las partidas. Las situaciones de juego son las siguientes:

1. Compra de objetos. Cada jugador interactúa con la tienda y compra objetos de acuerdo a los recursos con los que cuenta y con los beneficios que ese objeto le dará. No hay una definición de victoria o derrota en esta situación.
2. Posicionamiento de los roles. Cada jugador elige, de acuerdo a la estrategia del equipo, una línea (línea superior, línea media, junglas o línea inferior), donde defenderá las estructuras y recolectará recursos. No hay una definición de victoria o derrota en esta situación.
3. Persecución (pickoffs). En esta situación, un avatar es perseguido por uno o varios avatares del equipo contrario, con el fin de matarlo. Esta situación la ganan los perseguidores si logran matar al avatar perseguido y la gana el avatar perseguido si logra escapar de los perseguidores hacia una zona segura (detrás de sus estructuras).
4. Peleas de equipo (team fight). En esta situación dos o más avatares de un equipo pelean contra dos o más avatares del equipo contrario. La victoria la obtiene el equipo cuyos miembros involucrados en la pelea logran matar a todos los miembros del otro equipo involucrados en la pelea y la derrota es para el equipo cuyos miembros involucrados en la pelea son asesinados.
5. Invasión de jungla. Esta situación implica que un personaje, típicamente el rol ágil/jungla, pasa a las zonas del equipo enemigo y obtiene recursos en ella. La situación se gana si el invasor logra robar recursos (dar golpe final a enemigos neutrales o tomar objetos) y salir de la zona invadida. Esta situación se pierde si el invasor no logra robar recursos antes de escapar o si es asesinado mientras invade.
6. Farmeo (farming). En esta situación, los jugadores recopilan recursos a lo largo del mapa, el recurso principal para esto es dar el golpe final a enemigos neutrales o a creeps enemigos. Esta situación se gana cuando se logra dar el golpe final al enemigo seleccionado, de lo contrario, se pierde.
7. Gankeo (ganking). Esta situación se deriva de las persecuciones o de las peleas grupales y consiste en que un avatar que no estaba involucrado en la situación anterior, se una al combate para dar ventaja a su equipo. Esta situación se gana cuando el miembro que se une (ganker) mata al avatar perseguido o logra ganar la pelea en equipo, si este avatar es asesinado, entonces la situación se pierde.
8. Vagar (roaming). Esta situación implica que un personaje, típicamente el ágil/jungla, recorre diversas zonas del mapa para recolectar recursos o en busca de oportunidades de gankeo o para invadir jungla. No hay una definición de victoria o derrota en esta situación.
9. Empujar línea (lane push). En esta situación varios miembros de un equipo atacan una estructura enemiga para destruirla y reducir el espacio seguro del equipo contrario. Esta situación la gana el equipo atacante si logra destruir la estrcutura enemiga y la gana el equipo defensor si logra matar a todos los avatares involucrados en el ataque a la estructura.
10. Empuje dividido (split pushing). En esta situación un miembro de un equipo ataca una estructura enemiga para destruirla y reducir el espacio seguro del equipo contrario. Esta situación la gana el atacante si logra destruir la estrcutura enemiga y la gana el equipo defensor si logra matar al atacante.
11. Cambio de objetos. Esta situación es posterior a la compra de objetos y típicamente sucede sólo cuando un avatar ya no puede comprar más objetos y consiste en que un jugador decide cambiar alguno de los objetos que ya tiene equipados por otros.

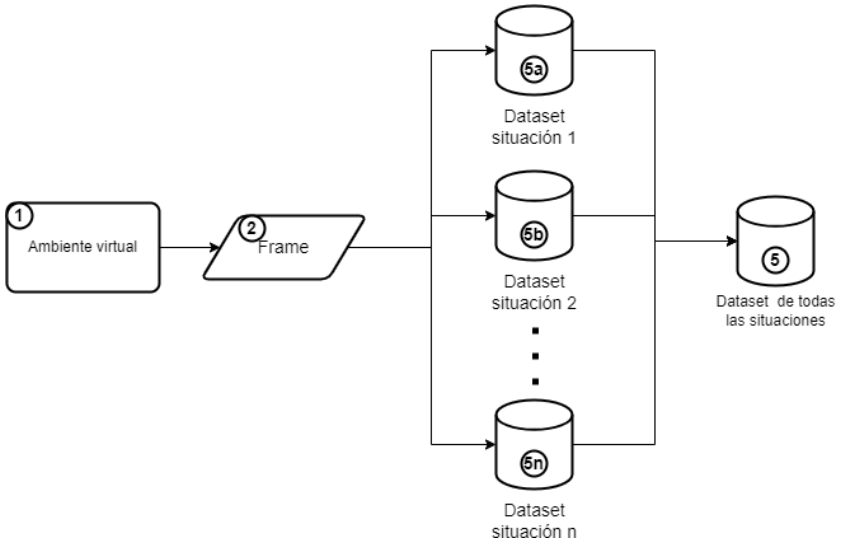


Fig. 2. Diagrama de la generación de datos.

Sólo sucede en videojuegos donde los avatares pueden comprar objetos. No hay una definición de victoria o derrota en esta situación.

12. Sitiar (sieging). Es una situación derivada de empujar líneas y consiste en que el equipo atacante haga retroceder al equipo defensor por detras de la estructura atacada, de modo que son incapaces de defenderla y el equipo atacante puede destruir la estructura más fácilmente. Esta situación la gana el equipo atacante si logra destruir la estrcutura enemiga y la gana el equipo defensor si logra matar a todos los avatares involucrados en el ataque a la estrcutura.
13. Emboscada. En esta situación uno o más miembros de un equipo se ocultan en zonas que bloquean la visión enemiga (arbustos, neblina, etc.) y esperan a que uno o varios avatares enemigos pasen cerca del escondite para atacarlos y matarlos. Esta situación la gana el equipo emboscante si logra matar a los avatares emboscados y la pierden si el equipo emboscado mata a los emboscantes o logra escapar a una zona segura (detrás de sus estructuras).

Finalmente, las situaciones de juego pueden suceder en una o varias fases de juego, esta relación se muestra en el Tabla 1.

2. Estado del arte

Los agentes jugadores son un tema de especial interés para la IA, en el caso de los videojuegos del género MOBA, éste se intensifica debido a la complejidad que presentan los desafíos de estos videojuegos, considerándose como un género que requiere de ciertos rasgos de inteligencia, como trabajo en equipo, coordinación, predicción a corto y largo plazo, inferencia con información incompleta, entre otros.

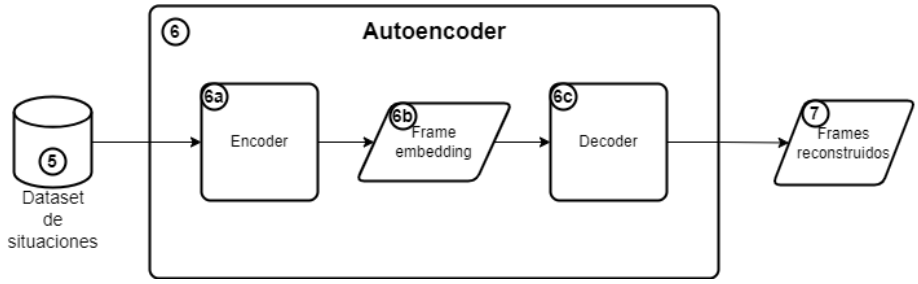


Fig. 3. Diagrama del entrenamiento del Autoencoder profundo.

Para fines de esta investigación, los trabajos más relevantes sobre agentes jugadores son precisamente aquellos sobre equipos jugadores de MOBA. De estos trabajos, sobresalen tres investigaciones, la primera es el trabajo [3], donde presentan una arquitectura basada en lógica difusa para agentes jugadores de MOBA, haciendo una simplificación del comportamiento del bot con una máquina de estados difusa con tres estados.

Este enfoque busca una respuesta sencilla, en cuanto a complejidad computacional, para resolver los MOBA, sin embargo, se probó en un bot único, no en un equipo. La segunda investigación, se encuentra en los trabajos [4, 5], cuyo trabajo derivó en el primer equipo de bots jugadores de DoTA2 capaz de derrotar equipos profesionales.

La relevancia de este trabajo recae en los hallazgos que tuvieron de la red una vez entrenada, puesto que, aunque no se programó específicamente para detectar situaciones, los estudios de interpretabilidad que hicieron a su arquitectura reforzada profunda, revelaron que el algoritmo era capaz de detectar algunas situaciones y reaccionar adecuadamente en consecuencia a ellas.

La tercera, es la investigación [6], cuyos autores desarrollaron un algoritmo de aprendizaje supervisado, donde se asocian estados del mapa con comportamientos que, dado el análisis que se hizo del conjunto de datos, estaba más asociado con victorias. Con este algoritmo lograron obtener un equipo de agentes jugadores de Honor of Kings con desempeño superior al de un equipo humano promedio.

Además de los agentes jugadores, es importante mencionar los avances en investigación sobre agentes jugadores generales, es decir, aquellos que juegan varios juegos una vez entrenados. En esta área, destaca el trabajo [7], en el cual buscan entrenar un agente reforzado que generalice su capacidad de jugar juegos (en inglés se le conoce como General Game Playing o GGP). Para este trabajo se experimenta la transferencia de conocimiento para que agentes inteligentes logren resolver videojuegos 1vs1, donde se evalúa al agente con la arquitectura que proponen contra el algoritmo del estado del arte Upper Confidence Bound on Trees (UCT).

En otros trabajos relacionados con el esquema propuesto en este artículo se usaron autoencoders como extractores de características profundas para agentes jugadores y generadores de contenido, como los presentados en [8, 9], respectivamente. Por otro lado, la idea de utilizar ambientes virtuales de entrenamiento en busca de la generalización se puede rescatar de las investigaciones presentadas en [10, 11, 12].

La diferencia principal de este trabajo con el estado del arte, se encuentra en el planteamiento modular del problema de aprendizaje, separando el comportamiento de equipo para jugar videojuegos MOBA en: análisis de la imágen del mapa para extracción de características; clasificación de estados de juego para el ajuste del algoritmo reforzado; el algoritmo reforzado no profundo para jugar MOBAs, que puede ser no profundo gracias a los dos módulos anteriores; el entrenamiento en ambientes virtuales, diferentes a los ambientes reales.

Además del planteamiento modular del problema, por medio del extractor de características, se lograría crear una arquitectura para un equipo que juegue diversos videojuegos MOBAs, puesto que aprendería a resolver las diferentes situaciones de juego independientemente del videojuego del género MOBA.

3. Desarrollo de la solución

3.1. Planteamiento del aprendizaje de un comportamiento colaborativo por situaciones de juego

Para resolver el problema global de aprender un comportamiento adecuado para jugar videojuegos MOBA se propone resolver cada una de sus partes, identificadas como situaciones de juego. Para ello, se plantea el uso de un ambiente virtual, en donde se presenten condiciones propias de cada situación de juego a un equipo de agentes inteligentes.

El ambiente virtual debe presentar al equipo condiciones que propicien el aprendizaje, por lo tanto, no será precisamente como algún ambiente real (un videojuego MOBA real), sino un pseudo-videojuego donde los desafíos sean lo suficientemente similares a los de los ambientes reales como para poder extrapolar el conocimiento adquirido a problemas desconocidos parecidos, pero lo suficientemente diferente para poner a prueba la capacidad de generalización del modelo entrenado. Por lo tanto, la primera parte del esquema de aprendizaje es el ambiente virtual, presentado en la Figura 1.

En el bloque 1 se ejecuta el motor del pseudo-videojuego, el cual debe calibrarse para presentar una situación de juego en particular a partir de una definición formal de las situaciones de juego, siendo este el espacio idóneo para aprender esa situación. Como resultado, este módulo produce tres datos, el primero de ellos, el bloque 2, es el Frame, que es la matriz asociada a lo que se mostraría en pantalla, este dato es la fuente principal de información sobre el pseudo-videojuego, ya que es el equivalente a la información disponible para los jugadores en los videojuegos MOBA.

Además de la matriz, este ambiente debe informar al equipo sobre su desempeño para ganar la situación de juego a la que se enfrente, esta información se proporciona mediante las recompensas en los bloques 3 y 4. El bloque 3 le indica al equipo completo qué tan bueno es su comportamiento para resolver la situación de juego y el bloque 4 se divide en cinco datos, una recompensa para cada rol y miembro de un equipo estándar de un videojuego MOBA, estas recompensas, de la 4a a la 4e, le indican a cada agente qué tan bien cumplen con su rol en la situación de juego a la que se enfrentan. Ambas recompensas, toman valores en el rango [0,1], donde 0 es el peor desempeño posible y 1 es el mejor.

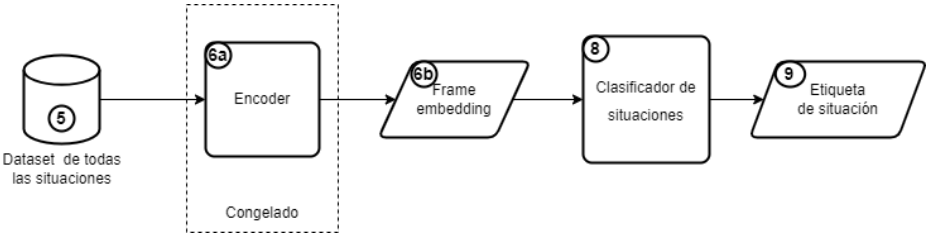


Fig. 4. Diagrama del entrenamiento del clasificador de sitaciones.

El ambiente será el módulo a través del cual se obtendrán los frames asociados a situaciones de juego específicas, por lo tanto, es posible aprovecharlo como mecanismo generador de datos etiquetados. Por ello, como se muestra en la Figura 2, los frames generados en el ambiente virtual se almacenarán, en conjuntos por cada situación, en los bloques 5a hasta el 5n, correspondiendo a las n situaciones que se hayan definido. Esta información se combinaría posteriormente en un conjunto que contenga todas las situaciones, en el bloque 5.

Para emplear sólo la información visual más importante del frame, se propone un mecanismo enfocado únicamente a la extracción de características. Esto es posible mediante el uso de un autoencoder profundo (Deep Autoencoder, DAE), cuyo entrenamiento no supervisado se presenta en la Figura 3. El DAE (bloque 6), puede entrenarse una vez que el bloque 5 tenga información de al menos una de las n situaciones de juego definidas.

El bloque 5 enviará frames de las situaciones de juego a la aqruitectura del DAE en el bloque 6. Este bloque se divide en tres partes, el primero es el encoder (6a), que es la parte que extrae características de los frames y produce un vector de dimensiones reducidas (frame embedding), este vector es el dato del bloque 6b. Este frame embedding se usa en el decoder (6c), para reconstruir la imagen de entrada, que corresponde al dato del bloque 7. Los frames reconstruidos se usan úncamente para evaluar el aprendizaje del DAE, entre más similares sean a los frames de entrada, mejor desempeño tiene el DAE.

Ya que el DAE ha sido entrenado, entonces es posible utilizar los frame embeddings que produce el bloque 6a en fases posteriores. Además de un mecanismo para determinar la información visual más relevante, el equipo requiere de un módulo que le ayude a determinar a qué situación de juego en particular se está enfrentando. Para ello, se propone un clasificador de situaciones, mismo que podría entrenarse de manera supervisada, dado que en el conjunto de datos del bloque 5 cada imágen estaría asociada con una situación de juego particular.

El entrenamiento de este clasificador se muestra en la Figura 4, donde el dataset completo (bloque 5), enviaría frames al bloque 6a, el encoder pre-entrenado y congelado, que generaría un frame embedding por cada imagen del dataset (bloque 6b). Estos embeddings serían la entrada al clasificador (bloque 8). Este bloque produce como salida una etiqueta de situación de juego (bloque 9), la cual, una vez que el clasificador haya sido entrenado, debería ser igual a la etiqueta original de las imágenes del dataset.

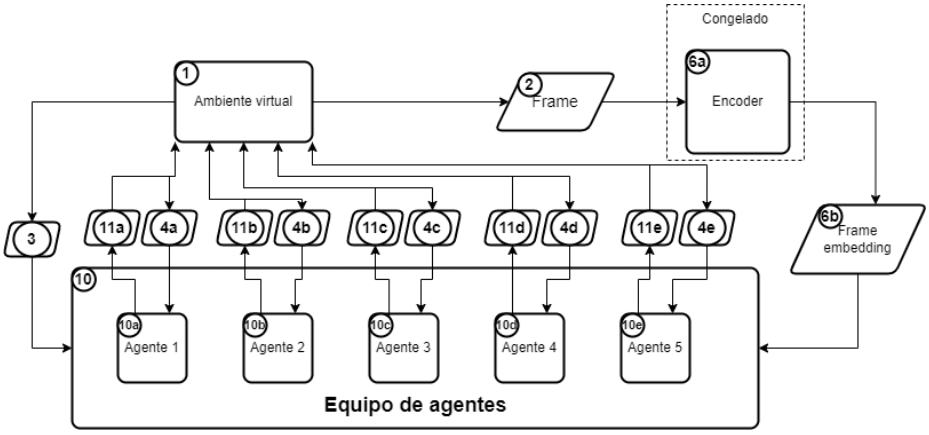


Fig. 5. Diagrama del entrenamiento de una política del equipo de agentes inteligentes para resolver una situación de juego.

En cuanto al entrenamiento del equipo de agentes para resolver una situación de juego en particular, se propone el esquema de la Figura 5. Este esquema inicia con el bloque 1, el ambiente virtual, el cual envía el frame actual del pseudo-videojuego (bloque 2), al codificador pre-entrenado (bloque 6a), éste produce un frame embedding (bloque 6b), que será la entrada para los agentes en el bloque 10, mismo que se divide en los bloques del 10a al 10e, uno por cada miembro y rol de un equipo estándar de un videojuego MOBA.

Todos los agentes reciben el bloque 6b y lo usan para elegir una acción de acuerdo a su política. Dada una situación i de juego, los agentes emplearán una política i correspondiente. En esta etapa, el ambiente generaría sólo espacios que presenten una situación de juego específica, por lo tanto, los agentes pueden ajustar únicamente la política que corresponde a dicha situación.

Cada agente produce como respuesta al frame embedding, una acción, con la que buscan obtener la victoria o terminar la situación de juego actual. Las acciones de los agentes se envían al ambiente por medio de los bloques del 11a al 11e, donde cada uno de ellos está asociado a un agente en particular.

En conjunto, todas las acciones producen un único cambio en el ambiente, que produce un nuevo frame en consecuencia. Para indicarle al equipo qué tan bueno fue su desempeño, el ambiente envía las recompensas, la global en el bloque 3 y la individual en los bloques del 4a al 4e. Los agentes usarán esas recompensas para ajustar su política, la cual debe ayudarlos a resolver la situación de juego que están aprendiendo en ese momento dadas las condiciones actuales del ambiente, presentes en el frame embedding.

En este esquema, los agentes están aprendiendo a resolver cada parte de la tarea global, es decir, ajustan una política que sirve para resolver una situación de juego, aprendiendo cada situación por separado en vez de aprender a jugar un videojuego MOBA, sin embargo, para poder resolver una partida completa se presenta el esquema de la Figura 6.

En éste, la ejecución comienza en el bloque 14, que sustituye al bloque 1, este bloque correponde a un videojuego MOBA, el cual únicamente produce el frame (bloque 2) (nótese que este bloque 2 es equivalente, pero no igual al bloque 2 de las figuras anteriores, las dimensiones y los valores en él pueden variar), que se envía hacia el codificador pre-entrenado (bloque 6a), que produce el frame embedding (6b), el cual llega al clasificador de situaciones pre-entrenado (bloque 8) y al equipo de agentes (bloque 10).

En este esquema, las situaciones irán cambiando, ya que se trata de una partida completa de un MOBA, por ello, para poder determinar cuál de todas las políticas pre-ajustadas deben usar los agentes, se utiliza el clasificador de situaciones, el cual emitirá una etiqueta de situación de juego (bloque 9), que será utilizada por un selector de políticas (bloque 12), el cual será un mecanismo que le indique a los agentes qué política usar, de acuerdo a la situación de juego identificada en la etiqueta del bloque 9.

El selector de políticas del bloque 12 enviará una señal (bloque 13), a cada agente, del 10a al 10e, para que empleen la política correspondiente a la situación de juego actual. Cada agente generará una acción de acuerdo a la política seleccionada y a la información actual de la partida contenida en el frame embedding.

Estas acciones, enviadas al videojuego MOBA a traves de los bloques del 11a al 11e, generarán un cambio en la partida y por lo tanto, el frame producido por el bloque 14 contiene este cambio producido a consecuencia de las acciones de los agentes.

En la Figura 6, todos los bloques basados en aprendizaje máquina han sido pre-entrenados, el codificador de manera no supervisada, el claisficador bajo el enfoque supervisado y el equipo de agentes bajo un enfoque reforzado, por lo que en este esquema no hay necesidad de hacer el ajuste de ningún modelo.

Sin embargo, para que el equipo de agentes logre resolver diferentes versiones de videojuegos MOBA, es necesario tener una definición general de todos los videojuegos que sea comprensible por los agentes. Para lograrlo, se propone entrenar diversos codificadores, uno por cada videojuego MOBA que deba resolver el equipo de agentes. Estos codificadores se entrenarían tomando como base un dataset de imágenes de un videojuego MOBA en particular, mismo que se obtendría de manera similar al dataset de situaciones de juego, presentado en la Figura 2.

Con estos datasets específicos para cada videojuego, es posible entrenar diversos codificadores, uno por cada juego, un encoder generaría frame embeddings de las mismas dimensiones que los del bloque 6b, puesto que serían la entrada del decoder 6c, el que se entrenó para reconstruir imágenes del ambiente virtual de entrenamiento, que quedan plasmadas en el bloque 7.

De este modo, todos los encoders se entrenarían para producir embeddings que sirvan para reconstruir imágenes del ambiente virtual, como si todos los ambientes se tradujeran a características del ambiente donde fue entrenado el equipo de agentes inteligentes. Este entrenamiento de encoders para cada MOBA y su correcto uso durante la fase de ejecución en el diagrama de la Figura 6 son el mecanismo de generalización que se propone para resolver diversos videojuegos del género MOBA.

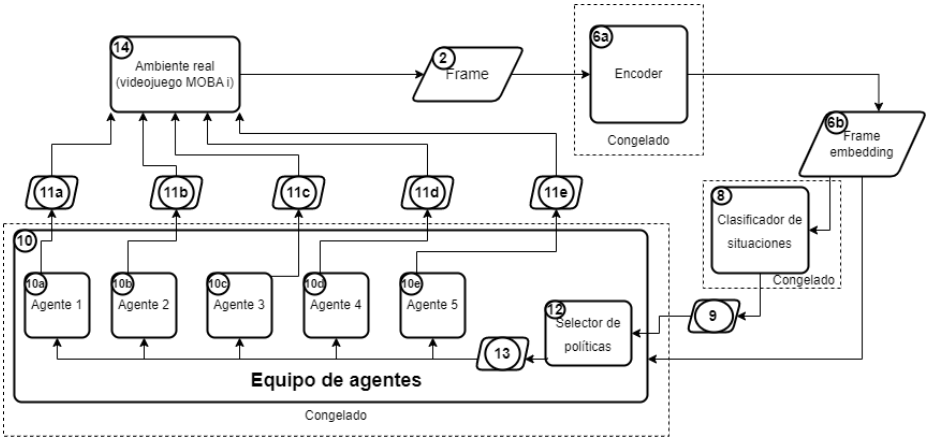


Fig. 6. Diagrama para una partida de un videojuego MOBA por situaciones de juego.

3.2. Situaciones complejas vs. situaciones triviales

El esquema propuesto en la sección anterior se propone para aquellas situaciones de juego que implican un proceso complejo de toma de decisiones individuales y de estrategia en equipo, sin embargo, existen situaciones de juego que no son tan complejas y no requieren de un procesamiento tan profundo de la información para tomar las decisiones.

Hay situaciones en las que, por ejemplo, los jugadores solo deben moverse hacia algún punto en particular, como al inicio de cada partida cuando cada avatar se posiciona en una línea cerca de alguna torre o estructura. Para este tipo de situaciones, se pueden usar políticas sencillas como instrucciones condicionales (if) en los agentes, sin necesidad de usar aprendizaje reforzado.

3.3. Novedad científica

El ensamble propuesto en este trabajo no se ha encontrado en el estado del arte, sin embargo, más que el esquema nuevo, la aportación viene desde el plantemiento modular con base en el aprendizaje de situaciones de juego. El esquema es la forma de materializar la propuesta de solución que se da con este análisis del problema. Además, el esquema propuesto busca generalizar su aprendizaje para resolver varios videojuegos del mismo género, problema que tampoco se ha resuelto en el estado del arte.

3.4. Evaluación

Aunque no existe una propuesta exactamente igual a la de este artículo, sí es posible hacer comparaciones del desempeño del equipo de agentes jugadores de MOBA. En primer lugar, existen equipos de agentes artificiales capaces de resolver juegos MOBA (principalmente DoTA 2, League of Legends y Honor of Kings), por lo que a nivel de una versión de la tarea, es posible comparar el desempeño del equipo propuesto.

Esto se mide a partir de variables bien definidas, como el índice de desempeño, el KDA o el porcentaje de victorias. Estas medidas de desempeño para cada caso servirían para comparar el desempeño del equipo de agentes jugadores propuesto con otros equipos que resuelven un único videojuego. Por otro lado, para medir qué tan bueno es el equipo para generalizar, se planea usar esas mismas métricas, aplicadas para los diferentes videojuegos.

Es decir, se tomarían las medidas de esas variables para el mismo equipo resolviendo diferentes videojuegos MOBA, una vez que ya ha sido entrenado en el ambiente virtual. De esto modo se corroboraría la capacidad del equipo para resolver situaciones similares a las que aprendió durante el entrenamiento pero que no son exactamente iguales, dado que algunas condiciones y/o reglas varían en los diferentes videojuegos.

4. Conclusiones y trabajo futuro

En este artículo se presenta un esquema de aprendizaje híbrido que busca definir una estrategia de aprendizaje para agentes jugadores capaces de resolver distintos videojuegos del género MOBA. Es necesario llevar a cabo los experimentos, para contrastar el planteamiento teórico de la propuesta de solución con los resultados en implementación para comprobar que realmente el esquema híbrido propuesto es capaz de resolver la tarea como se ha planteado que lo hará. El planteamiento presentado no contempla la posibilidad de que varias situaciones de juego puedan presentarse de manera simultánea, por lo que incluir esta condición al problema puede formar parte de una nueva investigación.

Agradecimientos. Este trabajo fue posible gracias al apoyo del gobierno mexicano a través del programa FORDECYT-PRONACES del Consejo Nacional de Ciencia y Tecnología (CONACYT) bajo la beca APN2017–5241; las becas de investigación de la SIP-IPN SIP-2259, SIP-20231198 Y SIP-20230140; la IPN-COFAA y la IPN-EDI.

Referencias

1. Lindqvist, K., Nilsson, D.: Developing a 5v5 framework for DOTA 2 bot competition (2020)
2. Silva, V., do N., Chaimowicz, L.: MOBA: A new arena for game artificial intelligence (2017) doi: 10.48550/ARXIV.1705.10443
3. Waltham, M., Moodley, D.: An analysis of artificial intelligence techniques in multiplayer online battle arena game environments. In: Annual Conference of the South African Institute of Computer Scientists and Information Technologists, no. 17, pp. 1–7 (2016) doi: 10.1145/2987491.2987513
4. Raiman, J., Zhang, S., Wolski, F.: Long-term planning and situational awareness in OpenAI five (2019) doi: 10.48550/ARXIV.1912.06721
5. Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., Pinto, H.P. d. O., Raiman, J., Salimans, T., Schlatter, J., Schneider, et al.: DOTA 2 with large scale deep reinforcement learning (2019) doi: 10.48550/ARXIV.1912.06680

6. Ye, D., Chen, G., Zhao, P., Qiu, F., Yuan, B., Zhang, W., Chen, S., Sun, M., Li, X., Li, S., Liang, J., Lian, Z., Shi, B., Wang, L., Shi, T., Fu, Q., Yang, W., Huang, L.: Supervised learning achieves human-level performance in MOBA games: A case study of honor of kings. In: IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 3, pp. 908–918 (2020) doi: 10.48550/ARXIV.2011.12582
7. McEwan, C., Thielscher, M.: Knowledge transfer for deep reinforcement agents in general game playing. In: Australasian Joint Conference on Artificial Intelligence, vol. 13151, pp. 53–66 (2022) doi: 10.1007/978-3-030-97546-3
8. Alvernaz, S., Togelius, J.: Autoencoder-augmented neuroevolution for visual doom playing. In: IEEE Conference on Computational Intelligence and Games, pp. 1–8 (2017) doi: 10.48550/ARXIV.1707.03902
9. Jadhav, M., Guzdial, M.: Tile embedding: A general representation for level generation. In: Conference on Artificial Intelligence and Interactive Digital Entertainment, vol. 17, no. 1, pp. 34–41 (2021) doi: 10.1609/aiide.v17i1.18888
10. Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., Efros, A. A.: Investigating human priors for playing video games (2018) doi: 10.48550/ARXIV.1802.10217
11. Open Ended Learning Team, Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M., McAleese, N., Bradley-Schmieg, N., Wong, N., Porcel, N., Raileanu, R., Hughes-Fitt, S., Dalibard, V., Czarnecki, W. M.: Open-ended learning leads to generally capable agents (2021) doi: 10.48550/ARXIV.2107.12808
12. Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., de Freitas, N.: A generalist agent (2022) doi: 10.48550/ARXIV.2205.06175

ISSN 1870-4069

Análisis comparativo de técnicas de aprendizaje automático y aprendizaje profundo para la detección de TDAH

Ismael Santarrosa-López[1], Giner Alor-Hernández[1], Maritza Bustos-López[1], Laura Nely Sánchez-Morales[2], José Luis Sánchez-Cervantes[2]

[1] Tecnológico Nacional de México, Instituto Tecnológico de Orizaba, México

[2] Consejo Nacional de Ciencia y Tecnología, Instituto Tecnológico de Orizaba, México

{m16011220, giner.ah, laura.sm, jose.sc}@orizaba.tecnm.mx, maritbustos@gmail.com

Resumen. El Trastorno por Déficit de Atención e Hiperactividad (TDAH) es un trastorno neurológico que afecta del 5-10% de los niños y 2-5% de los adultos en el mundo, causando maltrato infantil, rechazo escolar y aislamiento social. Por ello es importante detectar y tratar el TDAH de manera efectiva. En este trabajo, se presenta un análisis comparativo para determinar qué conjunto de datos y técnicas de Aprendizaje Automático y Aprendizaje Profundo ofrecen mejores resultados con base a diferentes métricas conocidas para la detección del TDAH. Los resultados indicaron que los algoritmos Máquina de Vectores de Soporte, Regresión Lineal, K-Vecinos más Cercano, Árbol de Decisiones y CatBoost son los de mejor clasificación con resultados de entre 66-97% de correcta clasificación de TDAH, superiores a los resultados obtenidos por los algoritmos de Aprendizaje Profundo, por lo que las técnicas de Aprendizaje Automático indican ser más efectivas en la detección de TDAH.

Palabras clave: Aprendizaje automático, aprendizaje profundo, biomarcadores, conjunto de datos, TDAH.

Comparative Analysis of Machine Learning and Deep Learning Techniques for ADHD Detection

Abstract. Attention Deficit Hyperactivity Disorder (ADHD) is a neurological disorder affecting 5-10% of children and 2-5% of adults worldwide, causing child abuse, school rejection, and social isolation. It is therefore important to detect and treat ADHD effectively. In this paper, a comparative analysis is presented to determine which Machine Learning and Deep Learning datasets and techniques provide better results based on different known metrics for ADHD detection. The

results indicated that the Support Vector Machine, Linear Regression, K-Nearest Neighbors, Decision Tree, and CatBoost algorithms are the best classifiers with results between 66-97% correct ADHD classification, superior to the results obtained by the Deep Learning algorithms so that Machine Learning techniques indicate to be more effective in ADHD detection.

Keywords: ADHD, biomarkers, dataset, deep learning, machine learning.

1. Introducción

El Trastorno por Déficit de Atención e Hiperactividad (TDAH) es un trastorno neurológico que afecta a cerca del 5% a 11% de los niños y 2% a 5% de los adultos en todo el mundo, según la Organización Mundial de la Salud (OMS) [1] y se estima que en México hay aproximadamente 1.5 millones de los 33 millones de niños y adolescentes en todo el país que tienen la probabilidad de padecer este trastorno según el CDC [2].

El TDAH tiene un impacto significativo en la vida de las personas, incluyendo dificultades en la escuela, en el trabajo y en las relaciones interpersonales [3]. Por lo tanto, es importante detectar y tratar el TDAH de manera efectiva. Este trastorno se caracteriza por dificultades en la atención, la concentración y el control de impulsos; estos síntomas llevan a problemas en los entornos ya mencionados.

Además, el TDAH también se relacionó con un mayor riesgo de abuso de sustancias y trastornos de salud mental, como la ansiedad y la depresión. Un estudio realizado por el Instituto Nacional sobre el Abuso de Drogas encontró que las personas con TDAH son más propensas a abusar de sustancias y tener problemas relacionados con las drogas [1].

La importancia del uso de técnicas de Inteligencia Artificial en la identificación de enfermedades ha sido reconocida por la comunidad científica y médica. Estas técnicas permiten la automatización y mejora de procesos de diagnóstico y terapéuticos, lo que resulta en una atención más eficiente y efectiva de la salud. El aprendizaje automático permite a las máquinas aprender a partir de datos y hacer predicciones sobre nuevos datos sin ser explícitamente programadas [4].

Por su parte, el aprendizaje profundo es un tipo de aprendizaje automático que imita la estructura y función de la corteza cerebral, lo que le permite a la IA aprender a partir de imágenes, audio y otras formas complejas de datos [4]. El uso de técnicas de aprendizaje automático y aprendizaje profundo en la identificación de enfermedades demostró ser efectivo en numerosos estudios y aplicaciones. Por ejemplo, el uso de técnicas de aprendizaje profundo para la detección de enfermedades cardiovasculares [5] o la detección temprana del Parkinson [6].

En la literatura existen trabajos que han reportado revisiones y análisis de conjuntos de datos para la detección del TDAH, como los realizados por Vikas Khullar et al. [7] y Hui Wen Loh et al. [8]. Sin embargo, estos trabajos tienen un enfoque basado en el análisis de un solo conjunto de datos o en un grupo pequeño de técnicas de inteligencia artificial, ya sea sólo de Aprendizaje Automático o sólo de Aprendizaje Profundo.

Como diferencia, en este trabajo se utilizan múltiples técnicas de inteligencia artificial para el análisis y evaluación de 4 conjuntos de datos diferentes para identificar

qué algoritmos ofrecen los mejores resultados, así como también los biomarcadores más relacionados para la identificación del TDAH.

La estructura de este artículo es la siguiente: en la sección 2 se presenta el estado del arte con un conjunto de investigaciones y trabajos relacionados con la identificación del TDAH, en la sección 3 se presentan los conjuntos de datos utilizados para este análisis y sus características, en la sección 4 se presenta los resultados del análisis y evaluación realizado a los conjuntos de datos con técnicas de inteligencia artificial previamente seleccionadas; en la sección 5 se presenta la discusión de los resultados y finalmente, en la sección 6, las conclusiones y el trabajo a futuro.

2. Trabajos relacionados

A continuación, se presentan algunos de los trabajos relacionados con respecto al diagnóstico del TDAH mediante técnicas de inteligencia artificial.

2.1. Enfoques basados en técnicas de aprendizaje profundo para la identificación del TDAH

Arthi y Tamilarasi [9], realizaron un sistema de red neuronal híbrido que consta de mapas autoorganizados de Kohonen seguidos de una función de base radial que utilizó valores de pertenencia difusos como entrada. El modelo se entrenó en dos fases sobre datos de TDAH.

Se observó que el modelo híbrido de redes SOM y RBF tiene una mejor clasificación en comparación con las redes neuronales de retropropagación. Kuang y He [10] propusieron el uso de una Red de Creencia Profunda (DBN), esta se compone por una pila de máquinas de Boltzmann (RBM).

DBN se aplicó para predecir los subtipos de TDAH, pero se necesitó un preprocesamiento para los datos específicos de fMRI. El método propuesto demostró ser eficaz para discriminar el TDAH del control y los subtipos. Con la idea de acelerar el proceso de clasificación clínico entre pacientes con TDAH y AOS, Chu et al.[11] realizaron tres modelos de aprendizaje automático para encontrar la mejor manera de ayudar al médico a realizar el diagnóstico.

Los resultados mostraron que los algoritmos de redes neuronales se adaptaron bien a la tarea de clasificación en cuestión, porque tenían una tasa de clasificación errónea menor que el modelo CART y CHAID.

Bo Miao & Yulin Zhang [12] utilizaron el proceso de Amplitud Fraccional Alterada de Fluctuación de Baja Frecuencia que sirvió para obtener la actividad cerebral en reposo, obteniendo un subconjunto de características para después complementar con el algoritmo Relief y obtener las características candidatas y así lograr la clasificación con la SVM, resultando en una clasificación viable de pacientes con TDAH en función a los datos de las neuroimágenes.

Shao L. et al. [13] utilizaron varias técnicas de aprendizaje automático, incluidas las redes neuronales profundas, para clasificar el TDAH. Se propuso como alternativa el bosque profundo, siendo un conjunto de conjuntos de árboles de decisión.

Tabla 1. Comparación de los conjuntos de datos para el diagnóstico de TDAH.

Conjunto de datos	Núm. de atributos	Núm. de registros	Descripción de los datos
HYPERAKTIV	+25	51 sujetos con TDAH y 51 sin TDAH	Datos de movimiento, ritmo cardiaco y concentración
ADHD200	23	776	Datos atómicos de Resonancia Magnética Multifuncional
Working Memory and Reward in Children with and without ADHD	8	79 niños	Imágenes por Resonancia Magnética Funcional 3D
Working Memory and Reward in Adults	8	24 adultos	Imágenes por Resonancia Magnética Funcional 3D
Eeg Data for ADHD	12	61 niños con TDAH y 60 sin TDAH	Datos atómicos de señales cerebrales (EEG)

Los resultados experimentales en los conjuntos de datos mostraron que este método logró un rendimiento superior que los métodos informados en la literatura sobre los conjuntos de datos de prueba de retención.

Los autores Mao, et al. [14] utilizaron la computación granular mediante el análisis de exploraciones de fMRI a través de perspectivas espaciales y temporales. Se utilizó CNN para conocer las características espaciales de cada cuadro de todo el escaneo.

Se utilizaron núcleos de convolución 3D y después se extendieron a la dimensión temporal, aprendiendo patrones espaciales y temporales simultáneamente con la operación de convolución 4D (este encapsula varias imágenes a la vez y aprender el movimiento del cerebro). Los resultados arrojaron una precisión del 71,3 % y un AUC del 0,80.

En el estudio realizado por Tosun [15], se reveló el canal más efectivo y el estado de registro más efectivo para el diagnóstico de TDAH. Los datos de EEG se aplicaron a la memoria a corto plazo (LSTM), la máquina de vectores de soporte y los clasificadores de redes neuronales artificiales, pero la mayor precisión se obtuvo con LSTM.

Esta métrica se calculó como 88,88% en el canal "Fp1, F7" y 92,15% en el estado de reposo con los ojos cerrados demostrando ser eficaz en el diagnóstico del TDAH. En cuanto a Khullar, et al. [16], se utilizó el algoritmo de red neuronal convolucional bidimensional (CNN) y la red neuronal convolucional bidimensional híbrida: memoria a largo plazo a corto plazo (2D CNN–LSTM) para la clasificación del TDAH a partir de controles de desarrollo típico.

El método propuesto logró una mejora significativa en el análisis y detección de parámetros. Los resultados construyeron un modelo adecuado e inteligente para diagnosticar comparativamente el TDAH a partir de controles sanos.

Los métodos tuvieron una buena precisión con el modelo 2D híbrido donde mostró más del 98% de precisión. En el estudio, se analizaron los efectos de los estímulos fóticos a diferentes frecuencias y en diferentes canales en el diagnóstico de TDAH.

2.2. Enfoques basados en técnicas de aprendizaje automático para la identificación del TDAH

Anuradha, et al. [17] usó el algoritmo SVM; se implementó en una herramienta llamada Clementine, este software se utiliza para realizar minería de datos. Después de ejecutar una base de datos de acuerdo con el SVM, proporcionó su propia interpretación de los resultados. El algoritmo SVM demostró un porcentaje de 88,674 % de éxito.

Por otro lado, Bautista et al. [18] utilizó una extensión del método Deformación Dinámica del Tiempo (DTW) para medir la similitud entre dos secuencias temporales; dicha extensión se codificó con clasificadores de clase GMM y APE; junto a dispositivo Kinect© para obtener patrones de comportamiento. El modelo se aplicó a un conjunto de datos multimodal y obtuvo mejoras con respecto a las técnicas puras de DTW.

En Duda et al. [19] utilizaron seis algoritmos diferentes de aprendizaje automático en datos de 2,925 sujetos, implementando métodos de selección de características para adaptar cada algoritmo y reducir el conjunto original. Los algoritmos SVC, LDA, Categorical Lasso y Logistic Regression arrojaron resultados de entre el 0.962 y 0.965 de exactitud, siendo los mejores modelos para su clasificación.

Por su parte, Uluyagmur-Ozturk et al. [20] se enfocaron en la clasificación de los participantes con TDAH y los que contaban con el Trastorno del Espectro Autista (TEA) en función al reconocimiento de emociones; los datos se usaron en los clasificadores de Árboles de Decisión. Los resultados indicaron que el TDAH y el TEA se clasificaron con un 90 % de precisión mediante el uso del algoritmo AdaBoost.

En el trabajo de Itani et al. [21] desarrollaron modelos de soporte de diagnóstico interpretables con árboles de decisión, C4.5 y algunas reglas, capaces de implementar a una base de datos médica. Al aplicar este marco en el conjunto ADHD-200 junto con imágenes de fMRI, logró predecir sujetos con TDAH. Mohammadhasani et al. [22] desarrolló un módulo de Instrucción Asistida por Computadora (CAI) que incluye agentes pedagógicos virtuales, utilizados en el aprendizaje en línea, que guían a los usuarios en entornos multimedia y, brindar apoyo cognitivo y emocional a los estudiantes con TDAH.

El estudio demostró que los agentes generan beneficios educativos para los estudiantes con TDAH. Khanna & Das [23] desarrollaron un método para analizar la variación pupilo métrica utilizando el aprendizaje automático, con la hipótesis de que reflejaría con precisión los pacientes con TDAH. El resultado de este método superó las tasas de diagnóstico clínico en un 80%. Christiansen et al. [24] realizaron un estudio con el objetivo de validar si el algoritmo CAARS logra discriminar pacientes con TDAH. Al aplicar el algoritmo, logró diferenciar sujetos con TDAH, obesidad y ludopatía con una precisión global del 80%.

Por último, los autores Maniruzzaman M. et al. [25] presentaron una investigación sobre los factores de riesgo en niños con TDAH. El estudio ilustró que el clasificador Random Forest proporciona una excelente clasificación y predicción de niños con TDAH dando más del 85% de correcta clasificación.

En nuestro caso, para el desarrollo de este trabajo, se utilizaron técnicas de aprendizaje automático y técnicas de aprendizaje profundo, estas técnicas se aplicaron a varios conjuntos de datos, cuya naturaleza de los datos van desde datos numéricos, datos provenientes de Electroencefalogramas (EEG) hasta imágenes de resonancia magnética (MRI); estos conjuntos se describen en la siguiente sección.

Tabla 2. Comparación de resultados obtenidos en el conjunto de datos HYPERAKTIV.

Algoritmos AA	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Linear Regression	0.9047	0.9090	0.9166	0.9000	0.9166
CatBoost	0.9047	0.9047	0.9047	0.9047	0.9047
SVM	0.8571	0.8750	0.8750	0.8571	0.8750
Gradient Boosting	0.8571	0.8545	0.8611	0.8421	0.8611
AdaBoost	0.8571	0.8545	0.8611	0.8421	0.8611
Random Forest	0.8095	0.8136	0.8194	0.8000	0.8194
Decision Tree	0.7619	0.7777	0.7777	0.7619	0.7777
LightGBM	0.7619	0.7777	0.7777	0.7619	0.7777
XGBoost	0.7619	0.7777	0.7777	0.7619	0.7777
KNN	0.6666	0.6636	0.6666	0.6315	0.6666

3. Conjuntos de datos para la identificación del TDAH

En esta sección se describen los distintos conjuntos de datos utilizados para la identificación del TDAH. Se utilizaron 5 conjuntos de datos, cada uno con distintos enfoques y características.

- **HYPERAKTIV:** HYPERAKTIV [26] es un conjunto de datos público con datos relacionados con la frecuencia cardiaca y de datos de movimiento de 51 pacientes adultos sin TDAH y 52 pacientes clínicos con TDAH. HYPERAKTIV incluye datos salud, atributos como la edad y el sexo, datos de salida de una prueba neuropsicológica informatizada e información sobre el estado mental del paciente. El conjunto de datos HYPERAKTIV se utilizó en [27] para el diagnóstico del TDAH en adultos utilizando algoritmos de aprendizaje automático.
- **ADHD200:** Por su parte, ADHD200 [28] ofrece un conjunto de datos preprocesados procedentes del Concurso Mundial ADHD-200. ADHD200 consiste en 776 conjuntos de datos anatómicos y de RMf en estado de reposo, incluyendo 285 de niños y adolescentes con TDAH. ADHD200 incluye datos sobre edad, sexo, estado de diagnóstico, estado de medicación, medidas dimensionales de los síntomas del TDAH, puntuación de algunos de los cuestionarios más comunes para la detección del TDAH y cociente intelectual (CI).El conjunto de datos ADHD200 se utilizó en [29] para el desarrollo de métodos de clasificación del TDAH basados en CNN 3D y redes neuronales de aprendizaje profundo multicanal en [30]. Los resultados obtenidos en ambos estudios indicaron una exactitud del 69,15% al 95%, respectivamente.
- **Working Memory and Reward in Children with and without ADHD:** este conjunto de datos (disponible en OpenNeuro.org [31]) fue el resultado de datos registrados por resonancia magnética funcional (fMRI) y puntajes longitudinales de medidas estandarizadas para la capacidad cognitiva, síntomas de TDAH y habilidades de lectura.

Tabla 3. Comparación de los resultados obtenidos en el conjunto de datos ADHD-200.

Algoritmo AA	Accuracy	Precision	Recall	F1-Score	ROC-AUC
KNN	0.9750	0.9705	0.9791	0.9787	0.9791
CatBoost	0.9750	0.9736	0.9772	0.9767	0.9772
Decision Tree	0.9750	0.9800	0.9687	0.9795	0.9687
Gradient Boosting	0.9750	0.9800	0.9687	0.9795	0.9687
LightGBM	0.9750	0.9800	0.9687	0.9795	0.9687
XGBoost	0.9750	0.9800	0.9687	0.9795	0.9687
Random Forest	0.9750	0.9800	0.9687	0.9795	0.9687
AdaBoost	0.9750	0.9800	0.9687	0.9795	0.9687
Linear Regression	0.9500	0.9479	0.9479	0.9583	0.9479
SVM	0.9250	0.9194	0.9270	0.9361	0.9270

El objetivo de las tareas era explorar la memoria de trabajo y el procesamiento de la retroalimentación en niños con desarrollo típico y con diagnóstico de TDAH. Se recogieron datos de 79 niños con edades comprendidas entre los 8, 6 y 12 años; 35 niños tenían un diagnóstico formal de TDAH [32].

- **Working memory and reward in adults:** En un estudio posterior, se creó un conjunto de datos titulado "Working Memory and Reward in Adults" [33] (disponible en OpenNeuro.org) con las mismas características que su predecesor y que incluye datos de 24 individuos adultos que realizaron las mismas tareas. El conjunto de datos anteriormente mencionado tiene antecedentes en el estudio presentado por [34] y [35]. Booth et al. [34] analizaron datos de fMRI obtenidos de cuatro tareas de memoria de trabajo visoespacial (VSWM) para detectar casos de TDAH. Los resultados indicaron una exactitud del 92,5% en la clasificación del TDAH.

  Hammer et al. [35] probaron el efecto interactivo de la retroalimentación y la recompensa en la memoria de trabajo visoespacial en niños con TDAH. Las pruebas recogieron datos MRI de 17 niños con TDAH y 17 niños de control mientras realizaban un seguimiento espacial de letras en una pantalla. Los resultados indicaron que el rendimiento de los niños con TDAH es similar a los niños de sin TDAH sólo cuando se les dio retroalimentación mediante una recompensa.

- **EEG data for ADHD:** Por su parte, EEG DATA FOR ADHD [36] es un conjunto de datos que contiene información de 61 niños con TDAH y 60 pacientes sanos. El registro EEG se realizó durante tareas de atención visual, se pedía a los niños que contaran el número de personajes de cada imagen; cada imagen se mostraba inmediatamente después de la respuesta del niño.

El registro EEG se basó en registros EEG estándar de 19 canales (Fz, Cz, Pz, C3, T3, C4, T4, Fp1, Fp2, F3, F4, F7, F8, P3, P4, T5, T6, O1, O2) a una frecuencia de muestreo de 128 Hz, y se colocaron electrodos A1 y A2 en los lóbulos de las orejas.

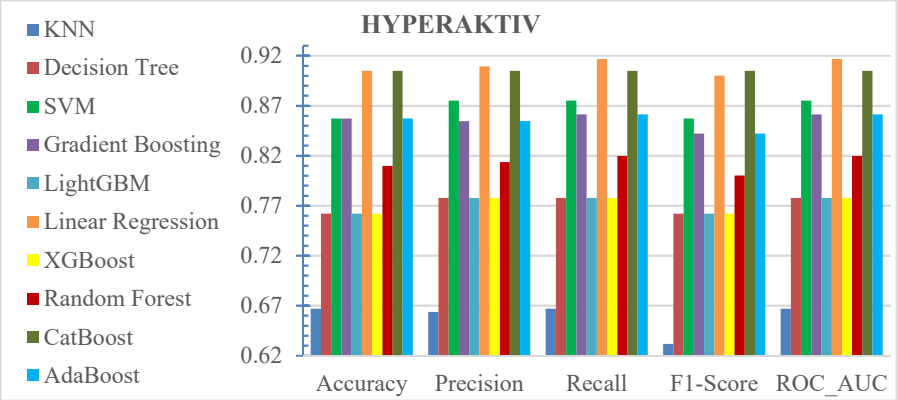


Fig. 1. Gráfica de los resultados de la tabla 2 para el conjunto de datos HYPERAKTIV.

El conjunto de datos EEG DATA FOR ADHD se utilizó en otros estudios para la detección del TDAH a partir de las características del EEG. En este sentido, Mohammadi et al. [37] clasificaron las características no lineales del EEG en niños con y sin TDAH como equivalentes a la atención. Los resultados confirmaron el defecto en el segmento del cerebro anterior de los niños con TDAH.

Barua et al. [38] utilizaron señales de EEG para plantear un nuevo modelo de clasificación manual para diferenciar a los individuos con TDAH. El modelo propuesto utilizó la Transformada Q Wavelet Sintonizable (TQWT) para generar sub-bandas Wavelet y un nuevo patrón de motivo ternario (TMP). Los resultados obtenidos mediante validaciones cruzadas arrojaron porcentajes de 95,57% y 77,93% de precisión en la clasificación.

La tabla 1 presenta una breve comparación de los conjuntos de datos utilizados, en esta tabla se muestran características como: cantidad de registros, número de atributos y breve descripción de los datos. Es importante mencionar que durante en análisis de los Datasets para su implementación, dos de ellos pertenecen al mismo grupo de datos, “Working Memory and Reward in Children with y without ADHD” y “Working Memory and Reward in Adults”, por lo que este último se omitió, quedando así sólo 4 conjuntos de datos para el análisis.

4. Análisis comparativo de técnicas de aprendizaje automático y aprendizaje profundo

En esta sección se detalla el análisis realizado de los conjuntos de datos revisados anteriormente con distintas técnicas de inteligencia artificial, las cuales aplican técnicas de aprendizaje automático y aprendizaje profundo.

Dentro de los algoritmos de aprendizaje automático se utilizaron: KNN (K-Nearest-Neighbor), Decision Tree, SVM (Support Vector Machine), LightGBM [39], Linear Regression [40], Gradient Boosting, XGBoost, CatBoost y AdaBoost [41].

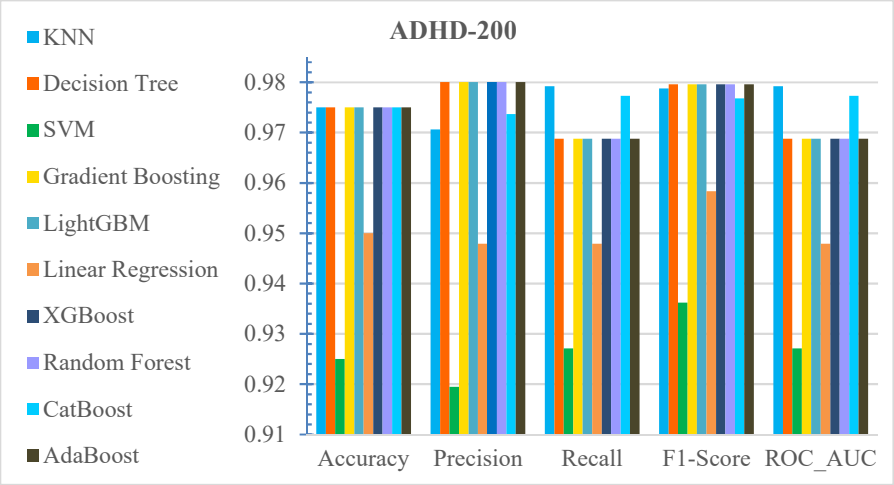


Fig. 2. Gráfica de resultados de la tabla 3 para el conjunto de datos ADHD-200.

Para la parte de aprendizaje profundo, se utilizaron los siguientes algoritmos: CNN (Convolutional Neural Network) y Multi-modality 3D CNN (es una variante de CNN, pero aplicada a imágenes en 3D) [42].

Estas técnicas de Aprendizaje Profundo se seleccionaron dado que la información de los conjuntos de datos a aplicar fue extraída con Matlab y que son compatibles con la herramienta Image Processing Toolbox de MATLAB.

Para los algoritmos de aprendizaje automático, se realizó una configuración experimental donde en primera instancia, se filtraron las características relevantes de cada conjunto de datos, evitando incluir aquellas características cuya presencia no aporta valor al resultado y omitiendo valores nulos.

Los conjuntos de datos se dividieron en un 80% de entramiento y un 20% de pruebas. Se agregaron más de un valor a cada parámetro utilizado, parámetros que varían según el algoritmo a utilizado, con el fin de utilizar la propiedad best_estimator_ al momento de realizar el entrenamiento para obtener los parámetros que mejor se ajustan y así obtener una clasificación óptima.

En cada algoritmo, se obtuvieron las siguientes métricas: exactitud (accuracy), precisión, exhaustividad (recall), F1-Score y el área bajo la curva (ROC-AUC). Se utilizó la matriz de confusión y otras métricas por ser ampliamente aceptadas en la literatura. Los algoritmos pertenecientes al aprendizaje automático fueron aplicados a los conjuntos de HYPERAKTIV y ADHD-200 en el lenguaje de programación Python junto con el apoyo de la biblioteca de funciones Scikit-Learn.

En la tabla 2 y en la tabla 3, se muestran los resultados obtenidos respectivamente. La Figura 1 y la Figura 2 muestra una gráfica comparativa de los resultados. Los conjuntos de datos evaluados con algoritmos de aprendizaje profundo se realizaron con Matlab. Se obtuvieron las cinco métricas ya mencionadas. En la Tabla 4 se muestra los resultados obtenidos para el resto de conjuntos de datos.

En la Figura 3a y la Figura 3b que se muestran a continuación, exponen de manera gráfica los resultados mostrados de la tabla 4 respectivamente.

Tabla 4. Resultados obtenidos con los algoritmos de aprendizaje profundo para los conjuntos de datos EEG data for ADHD (EEG ADHD) y Working Memory and Reward in Children with and without ADHD (WMRC ADHD).

Conjunto	Algoritmo AP	Accuracy	Precision	Recall	F1-Score	ROC-AUC
EEG ADHD	CNN	0.5833	0.5833	0.5533	0.5833	0.5533
WMRC ADHD	Multi-modality 3D CNN	0.6833	0.6833	0.6833	0.6538	0.6833

Los resultados obtenidos para los distintos conjuntos de datos analizados varían de significativamente según la procedencia de los datos, por lo que es un aspecto importante a considerar para la fiabilidad de los resultados.

5. Discusión

Los resultados obtenidos durante el análisis realizado arrojaron un rango de precisión del 57% al 91%, el rendimiento de cada modelo varía significativamente según el conjunto de datos utilizado. Este resultado podría deberse a diferentes factores, como la calidad o naturaleza de los datos, la distribución de las clases, la cantidad de datos de entrenamiento y la complejidad del modelo utilizado. Es posible que el modelo tenga un alto nivel de clasificación, pero aun así tenga problemas con la capacidad de generalización a nuevos datos. Por lo tanto, es importante tener en cuenta el resultado arrojado en otras métricas, como recall, F1-score o la curva ROC.

Ahora, profundizando más en los resultados obtenidos, se tiene que los algoritmos Regresión Lineal, CatBoost y SVM fueron los que arrojaron más del 85% de exactitud en el conjunto de datos HYPERAKTIV; mientras tanto que el conjunto de datos ADHD200 tuvo valores arriba del 97% de exactitud con los algoritmos de KNN, CatBoost y AdaBoost. Para el resto de los conjuntos de datos (EEG data for ADHD y Working Memory and Reward in Children with and without ADHD) fueron los menos precisos, arrojando resultados de entre el 58% y 68% respectivamente.

Es importante mencionar que debe considerarse la aplicación de un estándar para los conjuntos de datos relacionados al TDAH para garantizar la precisión y confiabilidad de los datos; además de hacer la colección de datos pública para futuras investigaciones y aún más si los datos fuesen multimodal, es decir, si la recolección de información de distintos estudios aplicados, como por ejemplo, imágenes fMRI, datos EEG y actividades que midan la concentración e hiperactividad de un mismo paciente se combinasen para favorecer aún más el desarrollo de herramientas con inteligencia artificial y de la exactitud resultante de las mismas para mejorar la detección, el diagnóstico y el tratamiento del TDAH.

Otro tema a discutir aplica a la selección de otros algoritmos a los aplicados aquí, considerando sus fortalezas y debilidades. La inclusión de otras técnicas de validación para evaluar su precisión y su comparación con los modelos tradicionales. Por último, considerando los resultados de este análisis, se determina que los biomarcadores más óptimos para la detección del TDAH son la combinación de: 1) datos de ondas cerebrales (obtenidos por EEG y que mide la concentración del individuo), 2) el registro de la actividad física y 3) ritmo cardiaco.

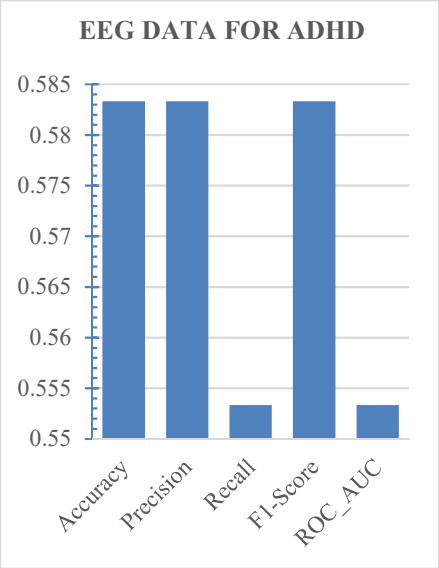


Fig. 3a. Gráfica de los resultados vistos en la tabla 4 (EEG DATA FOR ADHD).

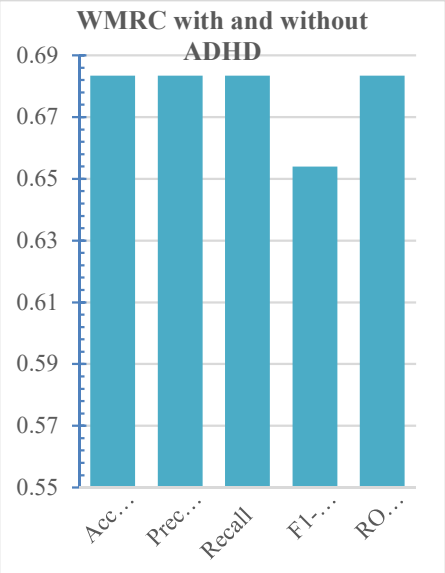


Fig. 3b. Gráfica de los resultados vistos en la tabla 4 (Working Memory and Reward in Children with and without ADHD)

No obstante, la variable relacionada con fMRI no se encuentra descartada, ya que el conjunto de datos utilizado contiene muy pocos registros para dar un diagnóstico preciso; pero en un futuro, con la obtención de un mayor número de registros, se podría obtener un mayor grado de precisión.

6. Conclusiones

La falta de precisión en los diagnósticos para la detección del TDAH evita que las personas con este trastorno no reciban el tratamiento adecuado a tiempo, generando problemas en su entorno social, escolar o laboral. En este artículo se analizaron diferentes conjuntos de datos con distintas técnicas de aprendizaje automático y aprendizaje profundo para identificar qué algoritmos ofrecen los mejores resultados, así como también los biomarcadores más relacionados para la identificación del TDAH.

Durante el desarrollo de este análisis se detectó que la exactitud en la detección puede variar significativamente según el conjunto de datos utilizado y las técnicas de inteligencia artificial utilizadas. Los algoritmos con mayor puntaje obtenido para esta métrica fueron SVM, Regresión Lineal, KNN, Decision Tree y CatBoost, aplicados sobre los conjuntos de datos HYPERAKTIV y ADHD200, donde los resultados arrojaron un rango entre un 85% y 97% de exactitud.

El desafío de la detección del TDAH con técnicas de aprendizaje automático y aprendizaje profundo requiere la colaboración de especialistas médicos y técnicos en aprendizaje automático para desarrollar algoritmos precisos y eficientes.

En resumen, este análisis arrojó resultados prometedores, demostrando que las técnicas de aprendizaje automático y aprendizaje profundo son útiles en la detección del TDAH, pero que necesitan de un conjunto de datos con suficiente información y características específicas para realizar una correcta clasificación.

Como trabajo a futuro debe considerarse el desarrollo de un test estadístico con el fin de garantizar validez y fiabilidad de los resultados, así como para tomar decisiones informadas basadas en datos objetivos.

También es importante considerar el uso de más conjuntos de datos similares o diferentes para complementar la información, también se considera el posible uso de otras técnicas de inteligencia artificial a las utilizadas en este trabajo, considerar también evaluar otro tipo de métricas como el tiempo de respuesta y realizar la evaluación en otros lenguajes de programación como R.

Agradecimientos. Este trabajo de investigación fue patrocinado por el Consejo Nacional de Ciencia y Tecnología de México (CONACYT) y la Secretaría de Educación Pública (SEP) de México a través del programa PRODEP. Adicional-mente, se agradece al Tecnológico Nacional de México (TecNM) por apoyar este proyecto.

Referencias

1. OMS: Salud mental del adolescente (2023) www.who.int/es/news-room/fact-sheets/detail/adolescent-mental-health
2. Centers for disease control and prevention: Trastorno por déficit de atención e hiperactividad (TDAH)
3. Rivera, F. B.: La elevada prevalencia del TDAH: posibles causas y repercusiones socioeducativas. Psicología educativa, vol. 22, no. 2, pp. 81–85 (2016) doi: 10.1016/ j.pse.2015.12.002
4. Jakhar, D., Kaur, I.: Artificial intelligence, machine learning and deep learning: definitions and differences. Clinical and experimental dermatology, vol. 45, no.1, pp. 131–132 (2020) doi: 10.1111/ced.14029
5. Gimenez, M., Peláez, D. B., Fisac, J. E. O., McVeigh, E. R., Carbayo, M. L.: Desarrollo de una herramienta para la detección automática del plano valvular mitral mediante algoritmos de deep learning. In: Libro de Actas del XXXVI Congreso Anual de la Sociedad Española de Ingeniería Biomédica, pp. 33–36 (2018)
6. Wang, W., Lee, J., Harrou, F., Sun, Y.: Early detection of parkinson's disease using deep learning and machine learning. IEEE Access, vol. 8, pp. 147635–147646 (2020) doi: 10.11 09/ACCESS.2020.3016062
7. Khullar, V., Salgotra, K., Singh, H. P., Sharma, D. P.: Deep learning-based binary classification of ADHD using resting state MR images. Augmented Human Research, vol. 6, no. 1, p. 5 (2021) doi: 10.1007/s41133-020-00042-y
8. Loh, H. W., Ooi, C. P., Barua, P. D., Palmer, E. E., Molinari, F., Acharya, U. R.: Automated detection of ADHD: Current trends and future perspective. Computers in Biology and Medicine, vol. 146, pp. 1–18 (2022) doi: 10.1016/j.compbiomed.2022.105525
9. Arthi, K., Tamilarasi, A.: A hybrid model in prediction of ADHD using artificial neural networks. International Journal of Information Technology and knowledge management, vol. 2, No. 1, pp. 209–215 (2009)
10. Kuang, D., He, L.: Classification on ADHD with deep learning. In: International Conference on Cloud Computing and Big Data, pp. 27–32 (2014) doi: 10.1109/CCBD.2014.42

11. Chu, K., C., Huang, H. J., Huang, Y. S.: Machine learning approach for distinction of ADHD and OSA. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1044–1049 (2016) doi: 10.1109/ASONAM.2016.7752370
12. Miao, B., Zhang, Y.: A feature selection method for classification of ADHD. In: 4th International Conference on Information, Cybernetics and Computational Social Systems, pp. 21–25 (2017) doi: 10.1109/ICCSS.2017.8091376
13. Shao, L., Zhang, D., Du, H., Fu, D.: Deep forest in ADHD data classification. In: IEEE Access, vol. 7, pp. 137913–137919 (2019) doi: 10.1109/ACCESS. 2019.2941515
14. Mao, Z., Su, Y., Xu, G., Wang, X., Huang, Y., Yue, W., Sun, L., Xiong, N.: Spatio-temporal deep learning method for ADHD fMRI classification. Information Sciences, vol. 499, pp. 1–11 (2019) doi: 10.1016/j.ins.2019.05.043
15. Tosun, M.: Effects of spectral features of EEG signals recorded with different channels and recording statuses on ADHD classification with deep learning. Physical and Engineering Sciences in Medicine, vol. 44, no. 3, pp. 693–702 (2021) doi: 10.1007/s13246-021-01018- x
16. Anuradha, J., Tisha, Ramachandran, V., Arulalan, K. V., Tripathy, B. K.: Diagnosis of ADHD using SVM algorithm. In: Proceedings of the Third Annual ACM Bangalore Conference, pp. 1–4 (2010) doi: 10.1145/1754288.1754317
17. Bautista, M. Á., Hernández-Vela, A., Escalera, S., Igual, L., Pujol, O., Moya, J., Violant, V., Anguera, M. T.: A gesture recognition system for detecting behavioral patterns of ADHD. In: IEEE Transactions on Cybernetics, vol. 46, no.1, pp. 136–147 (2014) doi: 10.11 09/TCYB.2015.2396635
18. Duda, M., Ma, R., Haber, N., Wall, D. P.: Use of machine learning for behavioral distinction of autism and ADHD. Translational Psychiatry, vol. 6, no. 2, pp. e732–e732 (2016) doi: 10.1038/tp.2015.221
19. Uluyagmur-Ozturk, M., Arman, A. R., Yilmaz, S. S., Findik, O. T. P., Genc, H. A., Carkaxhiu-Bulut, G., Yazgan, M. Y., Teker, U., Cataltepe, Z.: ADHD and ASD classification based on emotion recognition data. In: 15th IEEE International Conference on Machine Learning and Applications, pp. 810–813 (2017) doi: 10.1109/icmla.2016.0145
20. Itani, S., Lecron, F., Fortemps, P.: A multi-level classification framework for multi-site medical data: Application to the ADHD-200 collection. Expert Systems with Applications, vol. 91, pp. 36–45 (2018) doi: 10.1016/j.eswa.2017.08.044
21. Mohammadhasani, N., Fardanesh, H., Hatami, J., Mozayani, N., Fabio, R. A.: The pedagogical agent enhances mathematics learning in ADHD students. Education and Information Technologies, vol. 23, pp. 2299–2308 (2018) doi: 10.1007/s10639-018-9710-x
22. Khanna, S., Das, W.: A novel application for the efficient and accessible diagnosis of ADHD using machine learning. In: 2020 IEEE/ITU International Conference on Artificial Intelligence for Good, pp. 51–54 (2020) doi: 10.1109/AI4G50087.2020.9311012
23. Christiansen, H., Chavanon, M. L., Hirsch, O., Schmidt, M. H., Meyer, C., Müller, A., Rumpf, H. J., Grigorev, I., Hoffmann, A.: Use of machine learning to classify adult ADHD and other conditions based on the Conners' Adult ADHD Rating Scales. Scientific reports, vol. 10, no. 1, pp. 1–10 (2020) doi.org/10.1038/s41598-020-75868-y
24. Maniruzzaman, M., Shin, J., Hasan, M. A. M.: Predicting children with ADHD using behavioral activity: A machine learning analysis. Applied Sciences, vol. 12, no. 5, p. 2737 (2022) doi: 10.3390/app12052737
25. Hicks, S. A., Stautland, A., Fasmer, O. B., Førland, W., Hammer, H. L., Halvorsen, P., Mjeldheim, K., Oedegaard, K. J., Osnes, B., Giæver-Syrstad, V. E., Riegler, M. A., Jakobsen, P.: HYPERAKTIV: An activity dataset from patients with attention-deficit/hyperactivity disorder. In: Proceedings of the 12th ACM Multimedia Systems Conference, pp. 314–319 (2021) doi: 10.1145/3458305.3478454
26. Kaur, A., Kahlon, K. S.: Accurate identification of ADHD among adults using real-time activity data. Brain Sciences, vol. 12, no. 7, p. 831 (2022) doi: 10.3390/brainsci12070831
27. Nichols, N.: ADHD200 (2023) data.world/nicholsn/adhd-200

28. Zou, L., Zheng, J., Miao, C., Mckeown, M. J., Wang, Z. J.: 3D CNN based automatic diagnosis of attention deficit hyperactivity disorder using functional and structural MRI. IEEE Access, vol. 5, pp. 23626–23636 (2017) doi: 10.1109/ACCESS.2017.2762703
29. Chen, M., Li, H., Wang, J., Dillman, J. R., Parikh, N. A., He, L.: A multichannel deep neural network model analyzing multiscale functional brain connectome data for attention deficit hyperactivity disorder detection. Radiology: Artificial Intelligence, vol. 2, no. 1, pp. 1–9 (2019) doi: 10.1148/ryai.2019190012
30. Booth, J. R., Cooke, G. E., Gayda, J., Hammer, R., Lytle, M. N. Stein, M. A., Tennekoon, M. M.: Working memory and reward in children with and without attention deficit hyperactivity disorder (ADHD). OpenNeuro (2023) doi: 10.18112/openneuro.ds002424.v1.1.1
31. Lytle, M. N., Hammer, R., Booth, J. R.: A neuroimaging dataset on working memory and reward processing in children with and without ADHD. Data in Brief, vol. 31, p. 105801 (2020) doi: 10.1016/j.dib.2020.105801
32. Booth, J. R., Cooke, G. E., Gayda, J., Hammer, R., Lytle, M. N., Stein, M. A., Tennekoon, M.: Working memory and reward in adults. OpenNeuro (2023) doi: 0.18112/openneuro.ds002687.v1.1.1
33. Hammer, R., Cooke, G. E., Stein, M. A., Booth, J. R.: Functional neuroimaging of visuospatial working memory tasks enables accurate detection of attention deficit and hyperactivity disorder. NeuroImage: Clinical, vol. 9, pp. 244–252 (2015) doi: 10.1016/j.nicl.2015.08.015
34. Hammer, R., Tennekoon, M., Cooke, G. E., Gayda, J., Stein, M. A., Booth, J. R.: Feedback associated with expectation for larger-reward improves visuospatial working memory performances in children with ADHD. Developmental Cognitive Neuroscience, vol. 14, pp. 38–49 (2015) doi: 10.1016/j.dcn.2015.06.002
35. Nasrabadi, A. M., Allahverdy, A., Samavati, M., Reza-Mohammadi, M.: EEG data for ADHD/control children. IEEE Dataport (2020) doi: 10.21227/rzfh-zn36
36. Mohammadi, M. R., Khaleghi, A., Nasrabadi, A. M., Rafieivand, S., Begol, M., Zarafshan, H.: EEG classification of ADHD and normal children using non-linear features and neural network. Biomedical Engineering Letters, vol. 6, pp. 66–73 (2016) doi: 10.1007/s13534-016-0218-2
37. Barua, P. D., Dogan, S., Baygin, M., Tuncer, T., Palmer, E. E., Ciaccio, E. J., Acharya, U. R.: TMP19: A novel ternary motif pattern-based ADHD detection model using EEG signals. Diagnostics, vol. 12, no. 10, p. 2544 (2022) doi: 10.3390/diagnostics12102544
38. Mahesh, B.: Machine learning algorithms-a review. International Journal of Science and Research, vol. 9, pp. 381–386 (2020) doi: 10.21275/ART20203995
39. Su, X., Yan, X., Tsai, C. L.: Linear regression. Wiley interdisciplinary reviews: Computational Statistics, vol. 4, no. 3, pp. 275–294 (2012) doi: 10.1002/wics.1198
40. Bentéjac, C., Csörgő, A., Martínez-Muñoz, G.: A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, vol. 54, pp. 1937–1967 (2021) doi: 10.1007/s10462-020-09896-5
41. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., Farhan, L.: Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, vol. 8, pp. 1–74 (2021) doi: 10.1186/s40537-021-00444-8

ISSN 1870-4069

Seguimiento de objetos en video mediante el método de emparejamiento de bloques

Andrés Ely Pat-Chan[1], Francisco Javier Hernández-López[2], Mario Renán Moreno-Sabido[1]

[1] Instituto Tecnológico de Mérida, Tecnológico Nacional de México, México

[2] Consejo Nacional de Ciencia y Tecnología, Centro de Investigación en Matemáticas A. C., México

{mg13080859, mario.ms}@merida.tecnm.mx, fcoj23@cimat.mx

Resumen. El seguimiento de objetos en video (*video object tracking*) ha sido uno de los retos más relevantes en el campo de la visión computacional; actualmente, en la literatura existen cientos de métodos que realizan esta acción. El seguimiento de objetos en video contribuye a muchas aplicaciones que hoy en día son una necesidad en la vida del ser humano, como tratamiento de enfermedades crónicas, robótica, videovigilancia, la industria del entretenimiento, entre otras. En este artículo se presenta la implementación de un algoritmo para el seguimiento de objetos en video usando el método de emparejamiento de bloques. Se muestra el desarrollo de cinco versiones del algoritmo considerando estrategias para determinar si el objeto se está detectando de forma correcta, lo cual ofrece resultados interesantes ante los diferentes entornos y escenarios al que se pudiera enfrentar el algoritmo. Se utilizó un conjunto de datos de veinte secuencias de video para evaluar y comparar cada una de las versiones del algoritmo.

Palabras clave: Método de emparejamiento de bloques, seguimiento de objetos en video, umbral de aceptación.

Video Object Tracking Using Template Matching Method

Abstract. Video object tracking has been one of the most relevant challenges in computer vision; currently, hundreds of methods in the literature perform this action. Video object tracking contributes to many applications that are a necessity in human life today, such as the treatment of chronic diseases, robotics, video surveillance, the entertainment industry, and more. This article presents the implementation of an algorithm for video object tracking using template matching method. The development of five versions of the algorithm is shown considering strategies to determine if the object is being detected correctly, which offers interesting results in the different environments and scenarios that the

algorithm could face. A data set of twenty video sequences was used to evaluate and compare each version of the algorithm.

Keywords: Template matching method, video object tracking, acceptance threshold.

1. Introducción

El seguimiento de objetos en video ha sido uno de los retos más importantes en el campo de la visión computacional. A lo largo de los años, se ha trabajado para proponer diversos métodos que realizan esta acción. El seguimiento de objetos en video consiste en la acción de señalar por medio de diversos recursos visuales, tales como líneas, círculos o cuadros de colores, un objeto específico a lo largo de un video de manera automática [1].

Algunas aplicaciones del seguimiento de objetos en video que destacan hoy en día se encuentran en los sistemas inteligentes de videovigilancia, en la industria cinematográfica con las nuevas herramientas de edición de películas, en el campo de la medicina para detectar organismos celulares u otros patógenos, entre muchas más aplicaciones [2].

También en el área de la robótica se puede encontrar la implementación de seguimiento de objetos [3]. Al igual que en la medicina, con un sistema de seguimiento de movimiento para el tratamiento de pacientes con alguna enfermedad crónica [4].

Un problema en sistemas de videovigilancia ocurre cuando un operador debe revisar una cantidad enorme de videos para poder detectar objetos de interés, además debe seguir o rastrear a dichos objetos durante toda la secuencia de imágenes. Este trabajo es tedioso y conduce a que los operadores dejen pasar situaciones que pudieran ser importantes. Este trabajo se enfocó en la implementación de un algoritmo para contribuir en el desarrollo de sistemas de videovigilancia inteligente.

Actualmente, en la literatura existen un centenar de métodos de seguimiento de objetos, y cada uno realiza esta acción de manera distinta. Uno de los métodos de seguimiento de objetos en video elementales en la visión computacional es el de emparejamiento de bloques (*template matching*), del cual se hablará en este artículo.

El emparejamiento de bloques es un algoritmo, o método, elemental en el mundo del seguimiento de objetos en video; adopta el nombre de emparejamiento de bloques debido a que basta con tomar una plantilla del objeto a seguir para localizarlo en imágenes o fotogramas del video [5].

Algunos trabajos similares que se encuentran en la literatura utilizan el emparejamiento de bloques como fundamento para poner a prueba alguna nueva estrategia que consideren conveniente, con el fin de mejorar la acción de seguimiento de objetos en video [6].

Cabe mencionar que los retos en el seguimiento de objetos en video son interminables, y esa es la esencia y la inspiración para continuar con la investigación sobre el seguimiento de objetos en video. Por cada nuevo método propuesto para este fin, siempre surge alguna inquietud que propicie a seguir mejorando las técnicas y procesos que conllevan a encontrar un objeto a lo largo de un video.

De esta forma, se contribuye cada vez más a la información y, por ende, a las nuevas aplicaciones que estos, de alguna forma, puedan aportar.

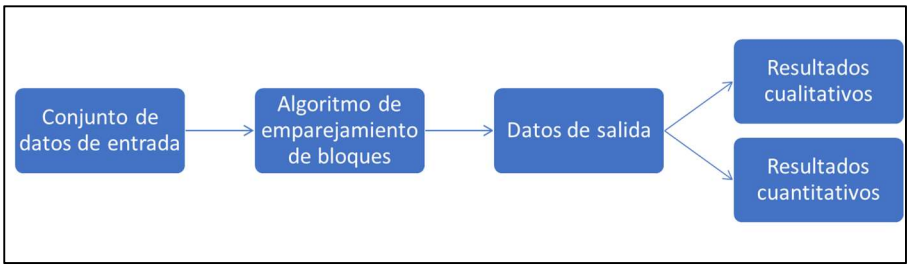


Fig. 1. Diagrama de la metodología propuesta.

El objetivo de este trabajo fue desarrollar un algoritmo para el seguimiento de objetos que automatice el trabajo humano de seguir el objeto a través de un video o una secuencia de imágenes.

En la sección 2 se describe la metodología que se utilizó para el desarrollo de este trabajo. En la sección 3 se muestran los resultados obtenidos en las pruebas. En la sección 4 se presentan las conclusiones a las que se llegaron, así como el trabajo a futuro.

2. Metodología

En la metodología propuesta están involucrados diversos elementos que en conjunto funcionan para lograr un objetivo en común, que en este caso es el de seguir un objeto a través de un video. En la Fig. 1 se observa un diagrama de bloques de la metodología propuesta. El primer paso es el establecimiento del conjunto de datos; actualmente, se cuentan con muchos recursos visuales que pueden servir para tales fines; algunos de ellos se pueden encontrar en internet, o incluso se pueden crear.

El segundo paso se refiere a aplicar un algoritmo de seguimiento de objetos; en el caso de este trabajo consistió en aplicar el método de emparejamiento de bloques. Posteriormente, se obtienen los datos de salida, los cuales se dividen en resultados cualitativos y resultados cuantitativos.

Los resultados cualitativos se obtuvieron al analizar de forma visual las cajas envolventes que devuelve el algoritmo sobre el objeto de interés a través de la secuencia de imágenes.

Los resultados cuantitativos se obtuvieron al comparar las cajas envolventes que devuelve el algoritmo con las cajas envolventes obtenidas de forma manual usando una métrica. A continuación, se describen de manera detallada los pasos de la metodología propuesta.

2.1. Conjunto de datos de entrada

En este trabajo, el recurso visual utilizado fue la secuencia de imágenes del conjunto de datos VOT [7]. Se decidió utilizar este conjunto de datos debido a que es muy conocido y recomendado para trabajos de seguimientos de objetos en video.

Tabla 1. Secuencia de imágenes del conjunto de datos de entrada.

Secuencia	Número de fotogramas	Breve descripción
Ball	603	Pelota en movimiento.
Board	698	Tarjeta de un circuito electrónico en movimiento.
Box	1161	Caja de cartón pequeña en movimiento.
Car	374	Auto saliendo de un estacionamiento.
Car_2	945	Camioneta blanca en calles.
Carchase	9928	Persecución policiaca.
Cup_on_table	1021	Una tasa sobre una mesa con la cámara moviéndose.
Dog1	1390	Peluche de perro en movimiento.
Gym	767	Gimnasta olímpica presentando su número.
Juice	404	Caja de jugo en una mesa con la cámara moviéndose.
Jumping	313	Persona saltando una cuerda.
Lemming	1336	Oso de peluche en movimiento.
Liquor	1741	Botellas en movimiento.
Mountain-bike	228	Motociclista saltando una rampa.
Person	948	Persona en movimiento.
Person_crossing	1018	Persona caminando por un parque.
Person_partially_occluded	306	Persona filmada desde diferentes ángulos.
Singer	351	Cantante presentando su número.
Sylvester	1345	Oso de peluche en movimiento.
Track_running	503	Corredora profesional presentando su número.

Otra ventaja es que cuenta con la información y etiquetas necesarias para trabajar con el desarrollo e implementación de algoritmos y métodos enfocados al seguimiento de objetos en video.

El conjunto de datos para este trabajo consistió en veinte secuencias de imágenes. En la Tabla 1 se presenta una breve descripción de las secuencias de imágenes que conforman el conjunto de datos de entrada.

En las diferentes secuencias de imágenes se pueden encontrar archivos de formatos JPEG y PNG que van desde los tamaños de 320x240 píxeles, hasta 640x480 píxeles. Cada secuencia posee información del objeto a seguir en un archivo de texto, lo cual facilita la evaluación de los algoritmos.

El número de fotogramas varía según la secuencia de imágenes; en este caso, van desde 228 hasta 9928 archivos de imágenes o fotogramas. Ahora bien, ya con los recursos visuales y la información del objeto a seguir, se procede a obtener la plantilla.

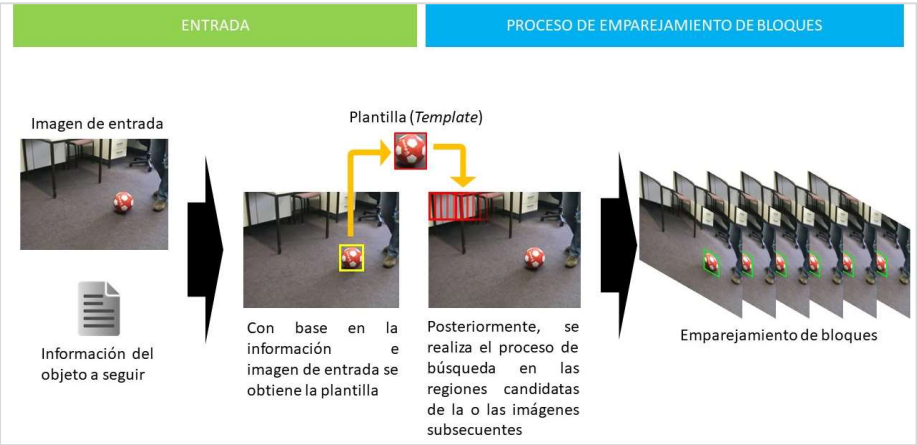


Fig. 2. Proceso general del emparejamiento de bloques.

Como primera instancia, se necesita una imagen de entrada que se obtiene de la secuencia de imágenes; por lo regular, siempre se utiliza el primer fotograma.

De igual forma, se necesita información sobre el objeto a seguir, la cual consiste en coordenadas (x, y) en la imagen, que indican donde se encuentra el objeto y las dimensiones que este abarca (ancho, alto). Posteriormente, con estos recursos se procede a ubicar y señalar en la imagen el objeto en cuestión. Con esa localización se obtiene la plantilla que es una imagen de dimensiones menores a la original, y que servirá como modelo para encontrarlo en los fotogramas subsecuentes.

2.2. Algoritmo de emparejamiento de bloques

Tomando el recurso visual, llámese video o secuencia de imágenes (*frames*), y la información sobre el objeto a seguir, la cual puede estar guardada en un archivo que proporciona el usuario, se puede obtener una plantilla con la cual se inicia el proceso de emparejamiento de bloques.

La plantilla es una imagen del objeto a seguir y está formada por todos los valores de color que tienen los píxeles en esa región seleccionada como plantilla. Con la plantilla establecida, lo que sigue es el proceso de barrido. Se le denomina de esta forma debido a que va recorriendo la imagen hasta encontrar la región que más se parezca a la plantilla [8]. El proceso de barrido lo hace dentro de una región específica de la imagen denominada región de búsqueda.

La región de búsqueda es un área que abarca los alrededores de donde probablemente se encuentre el objeto en la imagen subsecuente. Con esta estrategia se busca reducir el tiempo de procesamiento computacional y aumentar la velocidad de búsqueda. En la Fig. 2 se ejemplifica el método de emparejamiento de bloques de manera general.

Se realiza un cálculo matemático a la que se le denomina métrica, la cual sirve para determinar si la región analizada efectivamente es la más parecida a la plantilla.

Se ejemplificará lo anterior con lo siguiente: suponiendo que se quiere analizar una región de la imagen con la plantilla del objeto a seguir; para ello se necesita *"algo"* que indique que efectivamente ahí se encuentra dicho objeto [9].

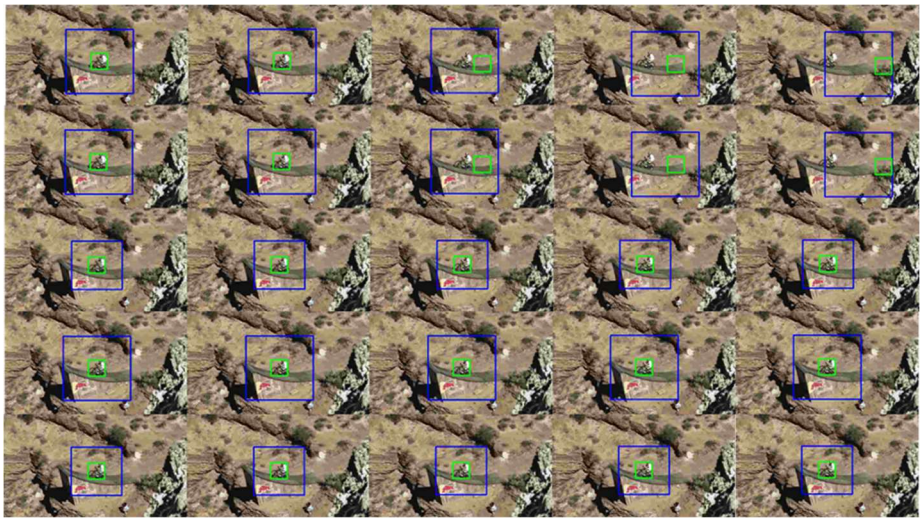


Fig. 3. Resultados de las estrategias procesando la secuencia de imágenes *Mountain-bike*. Cada fila de la figura representa los resultados de las estrategias V1, V2, V3, V4 y V5 respectivamente usando la métrica CCN.

Ese *"algo"* es un valor que es proporcionado por una de las métricas de las cuales se hablará más adelante en este artículo.

Lo antes descrito brinda una idea general de cómo se realiza el emparejamiento de bloques para encontrar el objeto en las imágenes subsecuentes.

Existen varias métricas para hacer el emparejamiento de bloques, sin embargo, para este trabajo se utilizó la suma de diferencias al cuadrado (SDC) y la correlación cruzada normalizada (CCN).

La SDC es un valor decimal que representa el error de analizar la región candidata contra la plantilla. Mientras más cercano a 0 sea este valor, significa que el error es mínimo y, por ende, en la región candidata analizada se encuentra el objeto.

Cabe mencionar, que de todas las regiones candidatas analizadas, siempre se debe considerar la que ofrezca el valor mínimo.

La SDC se denota de la siguiente forma:

$$SDC = \sum_{i=1}^{M} \left(I_{i,v} - T_i\right)^2, \tag{1}$$

donde M es el número de píxeles en la imagen, I es la imagen de entrada, T la plantilla o modelo del objeto a seguir, i denota un píxel en la imagen, y v denota una ventana del tamaño de T centrada en el píxel i.

La CCN, a diferencia de la SDC, tiene un rango acotado entre [-1, 1], en donde si el valor resultante es cercano a 1, entonces la región analizada es parecida a la plantilla. Esta métrica se denota de la siguiente forma:

$$CCN = \frac{\sum_{i=1}^{M}(T_i - \bar{T})(I_{i,v} - \bar{I_v})}{\sqrt{\sum_{i=1}^{M}(T_i - \bar{T})^2}\sqrt{\sum_{i=1}^{M}(I_{i,v} - \bar{I_v})^2}}, \tag{2}$$

Fig. 4. Comparación cualitativa de las métricas SDC (primera fila) y CCN (segunda fila) usando la estrategia V5 y procesando la secuencia de imágenes Singer.

donde $\overline{I_v}$ es el promedio de los valores de los píxeles de la imagen en la ventana v y $\overline{T}$ el promedio de los valores de los píxeles de la plantilla.

Con la ayuda de cualquiera de estas dos métricas es como se realiza el emparejamiento de bloques para encontrar la región donde se encuentra el objeto.

Técnicamente, con el paso del barrido se estaría concluyendo el método de emparejamiento de bloques, ya que una vez encontrada la región que más se le parece a la plantilla, el siguiente paso es marcar dicha región en la nueva imagen mediante un cuadro con un color que sobresalga.

Este mismo proceso se repite por cada nuevo fotograma. En este trabajo se realizaron algunas modificaciones que derivaron en cinco versiones (denotadas como V1, V2, V3, V4 y V5) de la implementación del algoritmo; estas versiones se denominaron estrategias y tienen como objetivo principal hacer más robusto el método de emparejamiento de bloques. A continuación, se explican estas estrategias.

2.3. Estrategias usando el método de emparejamiento de bloques

Una inquietud que surge con el método de emparejamiento de bloques (V1) es que siempre marcará una región en donde se considere que está el objeto. Existen casos en donde el objeto ya no está en la escena, o se encuentra ocluido por algún otro objeto; el método, por su naturaleza, siempre señalará en la imagen la región donde considere que está el objeto, aunque en realidad ya no exista en la imagen.

Tomando en cuenta estos casos, se realizaron algunas modificaciones al método que derivaron en versiones que mostraron mejoras significativas.

Una estrategia desarrollada fue la región de confianza (V2), que consiste en tomar los primeros n valores de la métrica que se esté utilizando, almacenarlos en un vector del tamaño de los primeros n valores, y calcular el promedio y desviación estándar para establecer umbrales de aceptación.

Entonces, si el próximo valor a analizar está dentro de estos umbrales, es aceptado y, por ende, se señala el objeto en la imagen en cuestión.

Posteriormente, se desarrolló la estrategia llamada plantilla dinámica (V3), que consiste en evaluar el valor de la métrica; si dicho valor está dentro de los umbrales de aceptación, entonces se señala al objeto en la nueva imagen, y con base en esta información se actualiza la plantilla. Con esto, se busca reducir el problema ocasionado en los casos en donde se tiene una plantilla fija y que no se actualiza a través del tiempo.

Otra estrategia (V4) consistió en combinar las estrategias V2 y V3 en busca de obtener mejores resultados. Por último, se desarrolló una estrategia que se denominó umbral de aceptación (V5), la cual consiste en utilizar las estadísticas de la estrategia

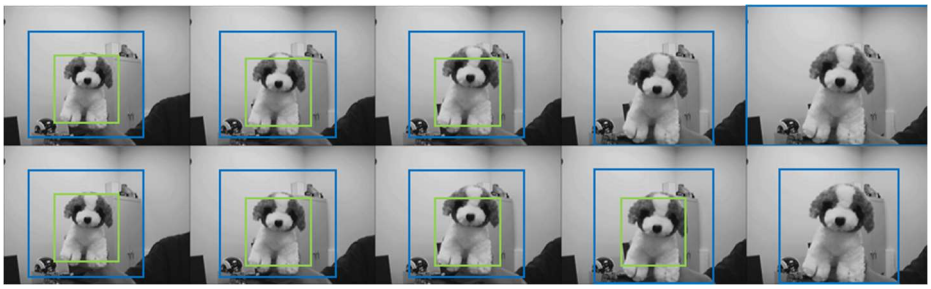


Fig. 5. Demostración cualitativa bajo cambios de escala del objeto a seguir. La primera fila corresponde a la métrica SDC y la segunda a la métrica CCN, usando la estrategia V5 y procesando la secuencia de imágenes Dog1.

V2, pero con la característica de que utiliza solamente valores positivos para crear dos umbrales de aceptación.

El primer umbral sirve para determinar si la región analizada es candidata para señalar la localización del objeto en la imagen. El segundo umbral determina si el objeto encontrado es adecuado para la actualización de la plantilla.

Los datos de salida de las diferentes estrategias, que son recursos visuales como videos con cuadros marcando el objeto, fueron sometidos a pruebas y evaluaciones que ofrecieron resultados interesantes, los cuales se describen a continuación.

3. Resultados

3.1 Resultados cualitativos

Al ejecutar cada una de las estrategias con las secuencias de imágenes se pudieron observar mejorías en cuanto al seguimiento del objeto. También se notó un buen ajuste en ciertas secuencias de video en la que la estrategia base (V1) no podía seguir el objeto.

En la Fig. 3. se observan los resultados de las estrategias procesando la secuencia de imágenes *Mountain-bike* del número de fotograma 15 hasta 19. La fila de la parte superior corresponde a los resultados de la estrategia base (V1), le sigue la fila de la estrategia V2, y así sucesivamente hasta llegar a la estrategia V5. Todos estos usando la métrica CCN. Es evidente como a partir de la estrategia V3, el objeto es seguido con precisión, mientras que en la estrategia V1 hay un desfase considerable.

Las métricas para el emparejamiento (SDC y CCN) jugaron un papel importante para encontrar una mejora a este algoritmo de objetos en video; La métrica CCN ofrece una cierta ventaja sobre la métrica SDC en cuanto a cambios de iluminación. En la Fig. 4 se muestran los resultados de la estrategia V5 procesando la secuencia de imágenes *Singer*, en donde se puede notar que ocurren cambios de iluminación en la escena.

Se observa que al usar la métrica SDC (primera fila de secuencia de imágenes) hay un desfase de la caja que envuelve al objeto, y posteriormente, la caja desaparece perdiendo al objeto. Por otro lado, al usar la métrica CCN (segunda fila de secuencia de imágenes) se muestra un seguimiento más preciso.

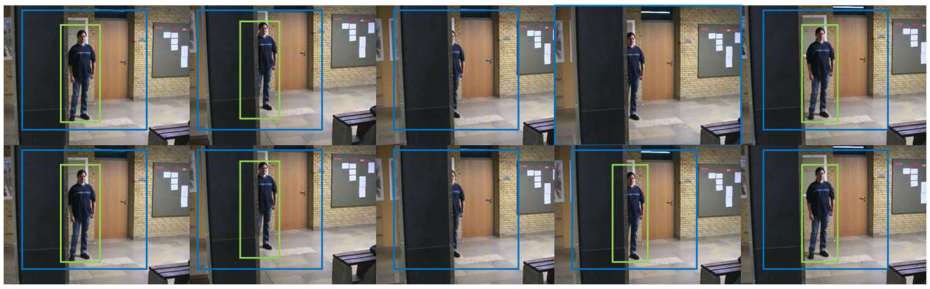


Fig. 6. Demostración cualitativa bajo oclusión del objeto a seguir. La primera fila corresponde a la métrica SDC y la segunda a la métrica CCN, usando la estrategia V5 y procesando la secuencia de imágenes Person_partially_occluded.

En la Fig. 5 se muestran los resultados de la estrategia V5 al procesar la secuencia de imágenes *Dog1* usando las métricas SDC (primera fila de secuencia de imágenes) y CCN (segunda fila de secuencia de imágenes).

Se observa que el algoritmo pierde al objeto cuando este presenta un cambio de escala, es decir, cuando el objeto está más cerca o más lejos de la cámara.

En la Fig. 6 se muestra una prueba de oclusión usando la estrategia V5, en donde se observa que las estrategias son capaces de encontrar el objeto después de tener una oclusión, siempre y cuando el objeto no cambie su tamaño a través de toda la secuencia de imágenes. La fila superior le corresponde a la métrica SDC, y la inferior a la métrica CCN, ambas de la estrategia V5, donde el algoritmo fue capaz de encontrar el objeto después de sufrir una oclusión.

3.2 Resultados cuantitativos

Para observar de manera cuantitativa la precisión y la robustez de un método de seguimiento de objetos en video ante diversos escenarios se emplean medidas o métricas existentes en la literatura [10].

En este trabajo se empleó la medida de exhaustividad (E) para obtener una ponderación de las diferentes versiones implementadas (derivadas del método original) usando los conjuntos de datos de entrada (secuencia de imágenes).

Para implementar la E se requiere de información verdadera e información estimada. La información verdadera se obtiene del conjunto de datos de entrada en donde se mencionó sobre la información y etiquetas del objeto a seguir; esta información viene incluida en el conjunto de datos de entrada, y son los cuadros o cajas envolventes reales del objeto.

Posteriormente, se necesitan los datos de salida de las diferentes versiones del algoritmo. Esta información es la del objeto que siguió el algoritmo a lo largo del video; normalmente son cuadros o cajas envolventes. Esta información que provee el algoritmo como datos de salida es la información estimada.

La E es la medida de rendimiento para una secuencia; esta medida (a veces denominada porcentaje de seguimiento correcto) indica cuántos cuadros de la salida del algoritmo o rastreador satisface el requisito en la superposición de los cuadros cuando el objeto fue visible; esta superposición se da cuando se comparan los cuadros de la

Tabla 2. Resultados de la E de cada estrategia considerando las métricas SDC|CCN.

Secuencia	V1	V2	V3	V4	V5
Ball	0.06|0.08	0.08|0.09	0.12|0.76	0.12|0.12	0.12|0.12
Board	0.01|0.15	0.17|0.05	0.18|0.16	0.18|0.18	0.18|0.18
Box	0.02|0.05	0.07|0.14	0.82|0.15	0.82|0.82	0.82|0.82
Car	0.17|0.10	0.10|0.18	0.18|0.55	0.18|0.18	0.18|0.18
Car_2	0.02|0.23	0.23|0.29	0.53|0.85	0.53|0.60	0.53|0.60
Carchase	0.04|0.01	0.01|0.04	0.04|0.16	0.04|0.03	0.04|0.03
Cup_on_table	0.02|0.08	0.08|0.13	1.00|0.88	1.00|1.00	1.00|1.00
Dog1	0.03|0.07	0.09|0.11	0.86|0.78	0.86|0.86	0.86|0.86
Gym	0.01|0.49	0.50|0.18	0.36|0.59	0.36|0.36	0.36|0.36
Juice	0.05|0.33	0.33|1.00	0.96|0.48	1.00|1.00	1.00|1.00
Jumping	0.29|0.80	0.80|0.99	0.99|1.00	1.00|0.99	1.00|1.00
Lemming	0.01|0.32	0.33|0.04	0.30|0.50	0.30|0.28	0.30|0.28
Liquor	0.02|0.04	0.04|0.17	0.25|0.53	0.25|0.25	0.35|0.25
Mountain-bike	0.07|0.06	0.06|0.26	0.88|0.07	0.98|0.99	0.99|1.00
Person	0.03|0.06	0.06|0.10	0.55|1.00	0.57|0.57	0.57|0.57
Person_crossing	0.01|0.07	0.07|0.06	0.37|0.82	0.37|0.41	0.37|0.41
Person_partially_occluded	0.07|0.18	0.18|0.53	0.94|1.00	1.00|0.98	1.00|1.00
Singer	0.20|0.20	0.20|0.17	0.46|0.54	0.46|0.47	0.46|0.47
Sylvester	0.01|0.03	0.03|0.05	0.92|0.35	0.94|0.98	0.94|0.98
Track runing	0.05|0.03	0.04|0.28	0.44|0.89	0.44|1.00	0.44|1.00
Promedio	0.06|0.17	0.18|0.24	0.56|0.60	0.57|0.60	0.58|0.61

información verdadera con los cuadros de información estimada. Se denota de la siguiente forma:

$$E = \frac{TP}{TP + FN}, \tag{3}$$

donde TP (*True Positive*) son los verdaderos positivos, y FN (*False Negative*) son los falsos negativos de la superposición.

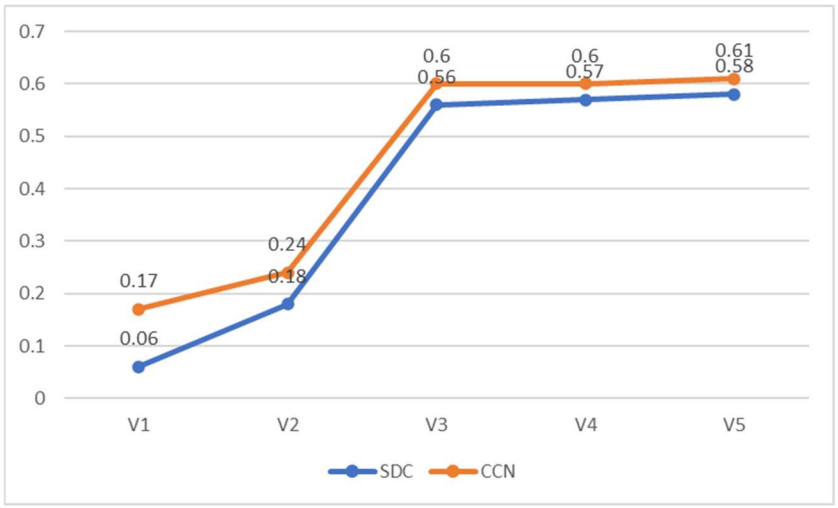


Fig. 7. Gráfica de resultados de la medida E.

Teniendo en cuenta lo anterior, y para tener una medida más objetiva, se decidió utilizar E como referente para mostrar los resultados. En la Tabla 2 se muestran los resultados de la E usando las diferentes estrategias.

En la Fig. 7 se presenta la gráfica que expresa los resultados de la E. En el eje horizontal se tienen las diferentes estrategias; para este trabajo se implementaron cinco.

En el eje vertical se tiene el resultado de la E promedio considerando todas las secuencias de video en cada una de las estrategias; entre más cercano sea este valor a uno, mejor es el método. La línea azul indica la métrica SDC, mientras que la línea naranja se usa para la métrica CCN.

4. Conclusiones y trabajo a futuro

Como se pudo observar en los resultados cuantitativos, se puede concluir que al aplicar la métrica CCN, ésta ofrece una mejora con respecto a la métrica SDC; esto se atribuye a que la primera es más robusta a los cambios de iluminación en comparación con la segunda.

También se concluye que las estrategias implementadas para el caso de la métrica SDC tienden a ser favorables para el seguimiento de objetos debido a que la línea azul apunta hacia arriba.

De manera cualitativa se observó que cuando se utilizó la métrica SDC en algunas secuencias de video con cambios de iluminación, en la escena se perdía el objeto cuando había cambio de iluminación, mientras que cuando se implementó la métrica CCN, éstos se reducían significativamente.

Las estrategias implementadas lograron una mejora en el método de emparejamiento de bloques para el seguimiento de objetos. Además, el uso de la métrica CCN mejoró los resultados.

Las limitantes que se encontraron en este trabajo son, en primera instancia, que se debe definir el objeto a seguir al inicio de la secuencia de video.

Otra limitante se presenta cuando los objetos cambian de escala; en este caso, el método ya no puede seguirlos.

Como trabajo a futuro se desarrollarán métodos de seguimiento de objetos basados en puntos característicos y descriptores de la imagen con el fin de acelerar el proceso de seguimiento y que el método sea robusto a oclusiones.

También se realizarán comparaciones de resultados con otros métodos de seguimiento de objetos, como lo son los basados en aprendizaje automático y puntos característicos.

Referencias

1. Pernici, F., Bimbo, A.: Object tracking by oversampling local features. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 12, pp. 2538–2551 (2014) doi: 10.1109/TPAMI.2013.250
2. Maggio, E., Cavallaro, A.: Video tracking: Theory and practice. Wiley (2010) https://www.wiley.com/en-sg/Video+Tracking:+Theory+and+Practice-p-9780470749647
3. Petrović, E., Leu, A., Ristić-Durrant, D., Nikolić, V.: Stereo vision-based human tracking for robotic follower. International Journal of Advanced Robotic Systems, vol. 10 (2013) doi: doi.org/10.5772/5612
4. Rolland, J. P., Fuchs, H.: Optical versus video see-through head-mounted displays in medical visualization. Presence: Teleoperators and Virtual Environments, vol. 9, no. 3, pp. 287–309 (2000) doi: 10.1162/105474600566808
5. Bovik, A.: The essential guide to video processing. Academic Press (2009) doi: 10.1016/B978-0-12-374456-2.X0001-1
6. Briechle, K., Hanebeck, U. D.: Template matching using fast normalized cross correlation. In: Proceedings of SPIE the International Society for Optical Engineering, vol. 4387, pp. 95–102 (2001) doi: 10.1117/12.421129
7. VOT Visual Object Tracking: VOT challenge (2013) https://www.votchallenge.net/
8. Corke, P.: Robotics, vision and control: Fundamental algorithms in MATLAB. Springer Cham (2017) doi: 10.1007/978-3-319-54413-7
9. Burger, W., Burge, M.: Principles of digital image processing: Core algorithms. Springer (2009) doi: 10.1007/978-1-84800-195-4
10. Padilla, R., Netto, S. L., Silva, E.: A survey on performance metrics for object-detection algorithms. In: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 237–242 (2020) doi: 10.1109/IWSSIP48289.2020.9145130

ISSN 1870-4069

Arquitectura de un módulo para la identificación de factores de riesgo para la detección de desórdenes hepáticos a partir del análisis de biomarcadores utilizando métodos de ensambles de aprendizaje automático

Luis Rodolfo Cabrera-Elías[1], José Luis Sánchez-Cervantes[2], Giner Alor-Hernández[1], Beatriz Alejandra Olivares-Zepahua[1], Luis Ángel Reyes-Hernández[1]

[1] Tecnológico Nacional de México
Instituto Tecnológico de Orizaba
México

[2] Consejo Nacional de Ciencia y Tecnología,
Tecnológico Nacional de México,
México

{m16011057, jose.sc, giner.ah, luis.rh}@orizaba.tecnm.mx,
bolivares@ito-depi.edu.mx

Resumen. La salud es importante para llevar una vida estable y duradera, pero la falta de cuidado aumenta el riesgo de enfermedades y dañar órganos, especialmente el hígado. En este trabajo se presenta la arquitectura de un módulo de aprendizaje automático basado en ensambles para identificar factores de riesgo en la detección de desórdenes hepáticos a través del análisis de biomarcadores. Además, se presenta el análisis de los trabajos relacionados con el tema y se examinan algoritmos de ensamble de aprendizaje automático, como bagging, boosting, para la detección de desórdenes hepáticos. Como trabajo a futuro se identificarán los principales biomarcadores para la detección de factores de riesgo, así como se diseñará un modelo de entrenamiento basado en el algoritmo de ensamble para la identificación de factores de riesgo. Finalmente, el módulo se entrenará y se integrará con interfaces Web y un repositorio de información.

Palabras clave: Biomarcadores, daño hepático, ensambles de aprendizaje automático.

Architecture of a module for identification of risk factors for the detection of liver disorders from biomarker analysis using ensemble machine learning method

Abstract. Health is important to lead a stable and long-lasting life, but lack of care increases the risk of diseases and damaging organs, especially the liver. This work presents the architecture of a machine learning module based on ensembles

to identify risk factors in the detection of liver disorders through biomarker analysis. In addition, an analysis of related works on the topic is presented and machine learning ensemble algorithms, such as bagging and boosting, are examined for the detection of liver disorders. As future work, the main biomarkers for detecting risk factors will be identified, and a training model based on the ensemble algorithm will be designed for the identification of risk factors. Finally, the module will be trained and integrated with web interfaces and an information repository.

Keywords: Biomarkers, liver damage, machine learning ensembles.

1. Introducción

El daño hepático causado por obesidad (hígado grado), o hepatotoxicidad generada por el consumo de medicamentos, drogas o incluso remedios caseros, por mencionar algunos casos; se está convirtiendo en un problema principal en la salud en México, tal como, la enfermedad por hígado graso no alcohólico (EHGNA) que afecta a un tercio de la población mundial, siendo México uno de los países cuya población reúne varios factores de riesgo para esta enfermedad y su prevalencia podría superar el 50% [1].

Los ensambles de aprendizaje automático son algoritmos integrados por algoritmos más simples (tradicionales), que permiten incrementar los porcentajes de sensibilidad y especificidad al momento de clasificar y detectar. Por lo cual, el uso de ensambles de aprendizaje automático permitirá identificar los factores de riesgo para la detección de desórdenes hepáticos a partir del análisis de biomarcadores.

En este artículo se propone un módulo de aprendizaje automático basado en ensambles para identificar factores de riesgo en la detección de desórdenes hepáticos y tomar medidas preventivas necesarias. El resto del trabajo incluye una revisión del estado del arte en la sección 2, la arquitectura del módulo en la sección 3, resultados de los datos en la sección 4, y las conclusiones y el trabajo futuro en la sección 5.

2. Trabajos relacionados

En esta sección se exponen los trabajos más destacables, los cuales tienen relación con las técnicas de ensamble automático. Doganer et al. [2] discutieron la importancia del uso de un ensamble de aprendizaje automático para la detección temprana de carcinomas de células renales, comparando métodos de stacking, bagging y boosting.

Los algoritmos de stacking obtuvieron los mejores resultados con una precisión del 86.7%. Verma et al. [3] compararon un nuevo método de ensamble de aprendizaje automático llamado "BBS method" (Bagging, Boosting, Stacking method, en español: Método de Bagging, Boosting y Stacking) con los algoritmos clasificadores apropiados.

El "BBS method" obtuvo una mayor precisión que sus contrapartes individuales. En [4], Buyrukoglu propuso mejorar los enfoques existentes para la detección temprana del Alzheimer mediante el uso de bagging, boosting y stacking, lo que resultó en una mejora en la clasificación del 3.2% al 7.2% respecto a los métodos anteriores.

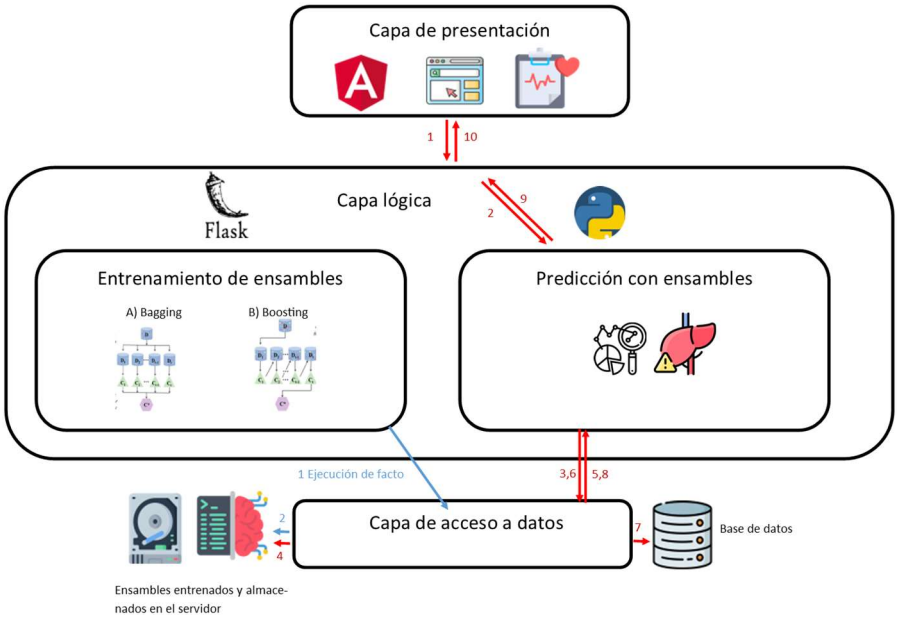


Fig. 1. Arquitectura del módulo de detección.

En [5], se utilizó un algoritmo de ensamble automático de bagging y boosting con 5 algoritmos de clasificación para predecir los síntomas de un infarto al miocardio en una etapa temprana. El algoritmo de bagging en conjunto con Random Forest obtuvo la mayor precisión con un 96.50%. Yadav et al. [6] compararon tres algoritmos basados en reglas utilizando ensambles de bagging y boosting para abordar la problemática de la diabetes mellitus.

El ensamble de bagging obtuvo una precisión del 98%. En [7], se propuso el uso de un ensamble de Bagging Weighted Voting (EBWvc) para abordar la problemática de la detección tardía del cáncer de mama. El EBWvc obtuvo una clasificación superior a las demás técnicas existentes.

Rahman F y Mahmood M [8] propusieron el uso de tres algoritmos clasificadores base, uno de ellos dirigido por bagging, para predecir enfermedades cardiacas mediante biomarcadores destacables. El modelo creado a partir del uso de Random Forest con Bagging obtuvo una precisión del 85.18%.

En [9], se mostró la importancia del uso de los registros electrónicos de salud (EHR) para obtener las variables de diagnóstico más relevantes para la predicción de la hepatitis C y la cirrosis mediante el uso de ensambles de aprendizaje automático.

Santos et al. [10] propusieron un enfoque de agrupamiento con sobre muestreo robusto para tratar con datos incompletos y predecir la supervivencia en un lapso de un año para pacientes con carcinoma hepatocelular.

En [11], se propuso la incorporación de la regresión lógica homogénea para apoyar la toma de decisiones y diagnósticos rápidos en el caso de un accidente cerebrovascular. La regresión logística homogénea obtuvo una mayor precisión que su contraparte simple. En la tabla 1 se presenta la comparación de los enfoques analizados en la sección 2. Trabajos Relacionados.

Tabla 1. Comparativa de los trabajos relacionados.

Artículo	Problema	Contribución	Tecnologías empleadas	Resultados
Doganer et al. [2]	El cáncer de células renales es normalmente asintomático, incluso en las etapas avanzadas y la posibilidad de dar con un diagnóstico temprano es baja.	Un modelo de ensamble automático de alto rendimiento para la detección temprana de carcinomas renales.	Ensambles de boosting Ensambles de bagging Ensambles de stacking, tales como: IB1, IBk, Kstar, LWL, REPTree, Random Forest y SMO.	El algoritmo de stacking REPTree obtuvo la más alta precisión del 86.7%.
Hakim et al. [5]	La carencia de detección temprana de los síntomas de un infarto al miocardio.	La comparación de dos enfoques de ensamble complementados por cinco algoritmos clasificadores base.	Bagging Boosting Support Vector Machine K-Nearest Neighbor Naive Bates Árbol de decisión Random Forest	El ensamble de bagging en conjunción con Random Forest obtuvo la clasificación de precisión más alta del 96.50%
Anisha y Saranya [11]	La falta de detección de los síntomas de un accidente cerebrovascular y las decisiones rápidas a tomar.	Un sistema que provee ayuda en la detección de un accidente cerebrovascular y las recomendaciones a seguir en dicho momento.	Regresión logística homogénea	El sistema propuesto obtuvo errores reducidos en la predicción y una precisión del 91% frente a la regresión logística simple para el diagnóstico de accidentes cerebrovasculares
Buyrukoglu[4]	La detección tardía del Alzheimer provoca pérdidas a la memoria del paciente.	Mejorar los enfoques ya existentes mediante la implementación de ensambles de aprendizaje automático.	Boosting Bagging Stacking	La clasificación incrementó en un porcentaje de entre 3.2% y el 7.2% respecto a los enfoques previamente diseñados.
Rahman y Mahmood [8]	La carencia de la capacidad de predicción para enfermedades coronarias.	Un modelo que permita predecir una enfermedad cardiaca, comparando tres clasificadores base con bagging.	Random Forest con ensamble de bagging K-Nearest Naive Bayes	El algoritmo de Random Forest con bagging alcanzó la puntuación más alta para predecir enfermedades coronarias.

Tabla 2. Comparativa de los atributos e instancias de los conjuntos de datos.

Conjunto de datos	Repositorio	Instancias	Atributos
Cirrhosis Prediction Dataset	Kaggle	424	19
Indian Liver Patient Dataset (ILPD):	UCI Machine Learning Repository	583	10
Liver Disorders Data Set	UCI Machine Learning Repository	345	6
Non-alcohol fatty liver disease (NAFLD)	Kaggle	17549	10

Tabla 3. Métricas obtenidas en el conjunto de datos Cirrhosis Prediction Dataset

Clasificador base	Exactitud	Precisión	Exahustividad	F1-Score	Area ROC
LR	71.43%	62.50%	45.45%	52.63%	65.41%
SVC	72.22%	62.50%	40.91%	50.70%	64.97%
KNN	68.25%	58.33%	31.82%	41.18%	59.81%
DT	73.02%	67.86%	43.18%	52.78%	66.10%
RF	69.05%	57.58%	43.18%	49.35%	63.05%

3. Arquitectura propuesta para el módulo

En esta sección se describe la propuesta y las tecnologías con las cuales se pretende desarrollar el módulo propuesto en este trabajo, además, se presenta los resultados obtenidos en el entrenamiento de los algoritmos de ensamble de aprendizaje automático seleccionados.

3.1. Diseño del módulo

En la Fig. 1 se muestra el diagrama de la arquitectura propuesta para este módulo de detección de factores de riesgo de desórdenes hepáticos. La arquitectura presentada en la Fig. 1 está separada en tres capas de las cuales se explicarán a detalle a continuación.

3.2. Capa de presentación

En esta capa se encuentra la interfaz gráfica en la que el médico o personal sanitario interactuará con la aplicación, se designó al framework Angular [12] el cual destaca por su flexibilidad y su capacidad de trabajar con aplicaciones Web grandes. En dicha interfaz, se le proporcionará al sistema los biomarcadores de un paciente en específico para que el módulo proporcione una predicción pertinente y mostrar en la pantalla dicho resultado obtenido.

Tabla 4. Métricas obtenidas en el conjunto de datos Indian Liver Patient Dataset.

Clasificador base	Exactitud	Precisión	Exahustividad	F1-Score	Area ROC
LR	72.00%	74.53%	93.75%	83.04%	53.26%
SVC	73.14%	73.14%	100.00%	84.49%	50.00%
KNN	68.57%	76.64%	82.03%	79.25%	56.97%
DT	73.14%	77.93%	88.28%	82.78%	60.10%
RF	73.71%	80.15%	85.16%	82.58%	63.85%

Tabla 5. Métricas obtenidas en el conjunto de datos Liver Disorders Data set.

Clasificador base	Exactitud	Precisión	Exahustividad	F1-Score	Area ROC
LR	75.00%	74.29%	86.67%	80.00%	72.88%
SVC	72.12%	75.41%	76.67%	76.03%	71.29%
KNN	67.31%	70.97%	73.33%	72.13%	66.21%
DT	70.19%	69.86%	85.00%	76.69%	67.50%
RF	73.08%	72.22%	86.67%	78.79%	70.61%

Cabe resaltar, que al usuario del módulo se le solicitarán únicamente los biomarcadores más importantes entre los que hasta el momento se han identificado los siguientes: tiempo de protrombina (en segundos), plaquetas por ml^3 y albúmina para cirrosis, y fosfatasa alcalina, aspartato aminotransferasa (AST) y alanina aminotransferasa (ALT) para desórdenes hepáticos y peso, género, altura e índice de masa corporal para hígado graso no alcohólico, los cuales se determinarán de manera definitiva en los trabajos a futuro de la presente investigación, por lo que no se le permitirá proseguir con la predicción en caso de no proporcionarlos.

3.3 Capa lógica

La capa lógica es la más importante de dicho módulo, ya que aquí se aloja el núcleo principal de este sistema, como backend se propone utilizar a Python [13] por ser un lenguaje de programación versátil y ampliamente utilizado en el campo de la inteligencia artificial en conjunción del framework Flask [14] que está diseñado para este lenguaje, además, permitirá darle funcionalidad de API REST al módulo propuesto.

Una vez introducidos los datos en la capa anterior son procesados en la sección de predicción de ensambles mostrada en la Fig. 1, donde se ejecutan una serie de modelos previamente entrenados mediante algoritmos de ensamble de aprendizaje automático.

El módulo estará integrado por dos de los mejores clasificadores base de boosting y de bagging. El sistema determinará de manera automática cual utilizar dependiendo de la carga de trabajo actual que sostenga. Consecutivamente almacenará dicha predicción

Tabla 6. Métricas obtenidas en el conjunto de datos Non-alcohol fatty liver disease.

Clasificador base	Exactitud	Precisión	Exahustividad	F1-Score	Area ROC
LR	92.27%	56.00%	10.10%	17.11%	54.71%
SVC	92.10%	0.00%	0.00%	0.00%	50.00%
KNN	92.00%	47.71%	12.50%	19.81%	55.66%
DT	92.27%	66.67%	4.33%	8.13%	52.07%
RF	92.33%	54.29%	18.27%	27.34%	58.47%

en la base de datos y el resultado obtenido se mostrará al médico para que proceda a prescribir un tratamiento oportuno.

La sección de entrenamiento de los algoritmos de ensamble que se muestra en la Fig. 1, se ejecutará únicamente por primera vez o al mejorar su entrenamiento. Si se realiza constantemente el entrenamiento, afectará en el rendimiento del módulo propuesto.

3.4 Capa de acceso a datos

La capa de acceso a datos provee las APIs necesarias para la comunicación de los componentes externos al sistema, tales como: la base de datos (donde se almacenarán los registros para llevar un historial) y el entrenamiento entrenado con anticipación. En la Fig. 1 se indica con flechas rojas y numeraciones el orden de ejecución de este módulo para evitar confusiones y se define de la siguiente manera:

- Punto 1: Lectura de biomarcadores en la interfaz gráfica.
- Punto 2: Recepción de la lectura en la capa lógica.
- Puntos 3, 4 y 5: Uso del ensamble entrenado con anterioridad y almacenado en la capa de acceso a datos para realizar una predicción con la información recibida y la obtención del resultado.
- Punto 6 y 7: Almacenaje del resultado en la base de datos de la aplicación para generar una historia del paciente.
- Punto 8 y 9: Devolver el resultado a la Capa de presentación.
- Punto 10: Observar el resultado en pantalla.

4. Conjuntos de datos para la identificación de desórdenes hepáticos

En esta sección se describirán 4 conjuntos de datos que se utilizarán para detección de desórdenes hepáticos mediante la ejecución de este módulo.

- **Cirrhosis prediction dataset:** [15] contiene información sobre pacientes con cirrosis hepática crónica y no crónica. Incluye 27 variables de entrada que incluyen edad, sexo, síntomas, signos de laboratorio, hallazgos histopatológicos y

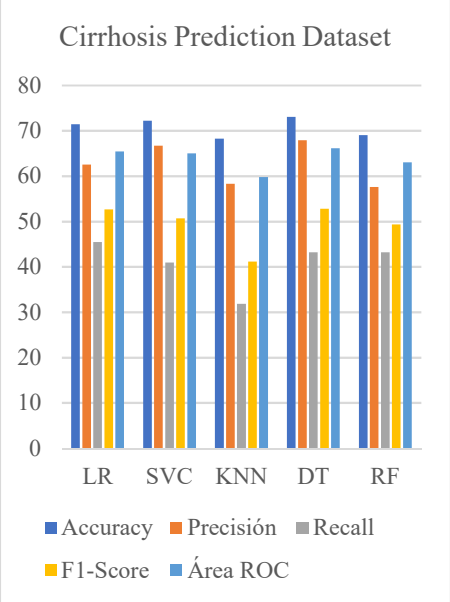


Fig. 2. Gráfica de las métricas obtenidas en el conjunto de datos cirrhosis prediction dataset.

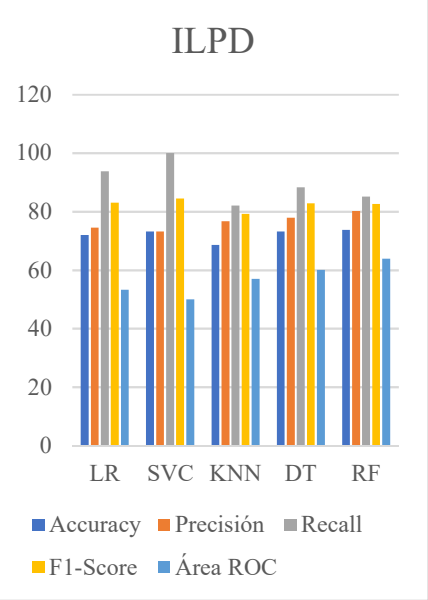


Fig. 3. Gráfica de las métricas obtenidas en el conjunto de datos indian liver patient dataset.

diagnósticos, así como una variable de salida que indica si el paciente tiene cirrosis crónica o no. El objetivo del dataset es predecir si un paciente tiene cirrosis crónica o no en función de las variables de entrada. El conjunto de datos contiene 386 casos, 186 de los cuales son casos positivos de cirrosis crónica. El dataset se encuentra en formato CSV y está disponible para descargar en Kaggle.

– **Indian liver patient dataset (ILPD):** ILPD [16] es un conjunto de datos médicos que contiene información sobre pacientes indios con y sin enfermedades hepáticas. El conjunto de datos consta de 583 instancias con 10 atributos, incluyendo información demográfica del paciente, resultados de pruebas de sangre y diagnósticos de enfermedades hepáticas. El objetivo del conjunto de datos es predecir si un paciente tiene una enfermedad hepática o no, lo que lo convierte en un conjunto de datos útil para tareas de clasificación binaria en el campo de la salud.
– **Liver disorders data set:** [17] este conjunto de datos proveniente del repositorio UCI Machine Learning contiene los resultados de pruebas de laboratorio de pacientes con trastornos hepáticos, como hepatitis viral, cirrosis, hemocromatosis y otros. El conjunto de datos consta de 345 instancias y 6 atributos, incluyendo el volumen corpuscular medio, la fosfatasa alcalina, la alanina aminotransferasa y el aspartato aminotransferasa. El objetivo del conjunto de datos es predecir si un paciente tiene o no un trastorno hepático basado en los valores de los atributos.

– **Non-alcohol fatty liver disease (NAFLD):** NAFLD [18] alojado en Kaggle contiene información sobre pacientes diagnosticados con hígado graso no alcohólico en un hospital universitario de Corea del Sur. Incluye datos demográficos y clínicos de

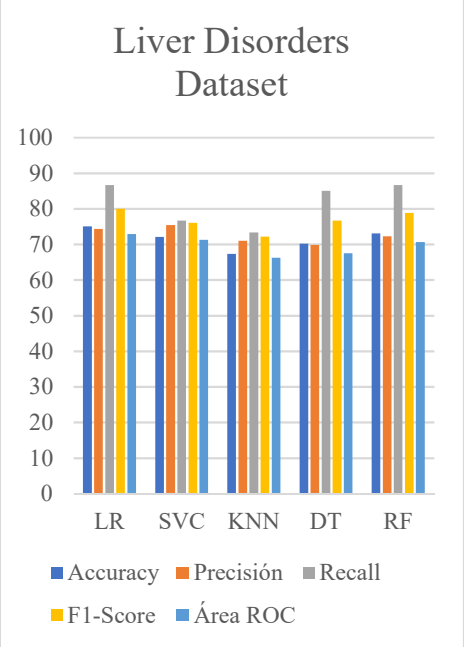


Fig. 4. Gráfica de las métricas obtenidas en el conjunto de datos liver disorders dataset.

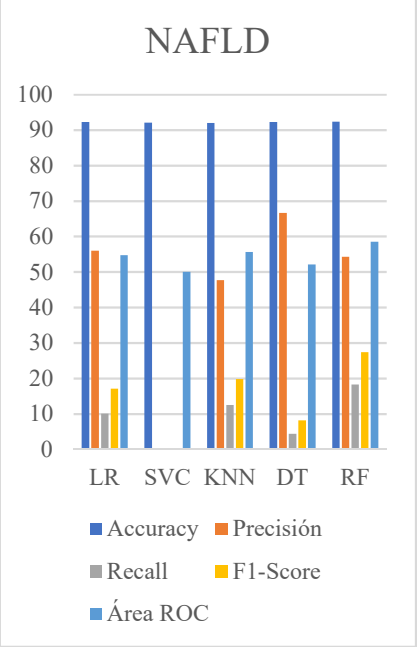


Fig. 5. Gráfica de las métricas obtenidas en el conjunto de datos non-alcohol fatty liver disease.

17,549 pacientes, como edad, género e índice de masa corporal. El objetivo del conjunto de datos es ayudar a predecir la progresión de la enfermedad y desarrollar un modelo de diagnóstico temprano.

La Tabla 2 muestra una comparación breve entre los cuatro conjuntos de datos mencionados anteriormente, haciendo hincapié en el número de instancias y los atributos.

5. Análisis comparativo de los conjuntos de datos mediante los algoritmos de ensambles con enfoque de Bagging

En esta sección se presentan la comparativa realizada a los conjuntos de datos mediante el uso de diferentes clasificadores base utilizando el enfoque de embolsado (bagging, en inglés) mediante el uso de Python y las bibliotecas de scikit-learn [19], las cuales están enfocadas en aprendizaje automático. Los algoritmos utilizados como clasificadores base para este análisis fueron los siguientes: LogisticRegression [20], SVM (Support Vector Machine) [21], KNN(K-Nearest-Neighbor) [22], DecisionTree [23] y RandomForest [24].

De los algoritmos se evaluaron y compararon métricas tales como: exactitud (accuracy), precisión (precision), exhaustividad (recall), F1-Score y el área ROC y se aplicaron a los 4 conjuntos de datos listados en la Tabla 2. Algunos conjuntos de datos

tienen mejores resultados para las predicciones que otros y esto es dependiente de los atributos de los cuales se encuentren conformados y mediante este análisis comparativo es posible determinar cuál clasificador base es más eficiente con los diferentes conjuntos que se le proporcionen.

6. Conclusiones y trabajo a futuro

La detección temprana de factores de riesgo para la detección de desórdenes hepáticos es esencial para prevenir daños graves y potencialmente mortales en el hígado. El desarrollo de este módulo tiene como objetivo ayudar a los médicos a identificar y tratar el problema antes de que se convierta en algo más grave y difícil de tratar y tomar las medidas pertinentes para dar un tratamiento que permita a los afectados por dichos desordenes mejorar su calidad de vida.

Como trabajo a futuro se considerarán los biomarcadores más importantes para la detección de los factores de riesgo para la detección de desórdenes hepáticos, tal cual, el análisis comparativo mediante el enfoque de algoritmos de ensamble de boosting, así como como los algoritmos de ensamble de aprendizaje automático a utilizar para este módulo, se tiene previsto utilizar las bibliotecas orientadas para Python enfocadas en aprendizaje automático conocidas como scikit-learn como parte del núcleo para realizar las predicciones pertinentes y proporcionar un resultado fidedigno, así como evaluar y comparar, el conjunto de datos y tipos de ensambles más precisos para esta problemática.

Para la validación del módulo propuesto en el futuro se hará uso de un caso de estudio como prueba de concepto que permita describir los resultados y conclusiones obtenidas. Finalmente se realizará una comparativa de los resultados obtenidos de los clasificadores mediante este módulo respecto a los trabajos mencionados anteriormente en la sección 2, estado del arte.

Agradecimientos. Los autores dan las gracias al Tecnológico Nacional de México por respaldar este trabajo. A sí mismo como el patrocinio por parte del Consejo Nacional de Ciencia y Tecnología (CONACYT).

Referencias

1. Bernal-Reyes, R., Castro-Narro, G., Malé-Velázquez, R., Carmona-Sánchez, R., González-Huezo, M., García-Juárez, I., Chávez-Tapia, N., Aguilar-Salinas, C., Aiza-Haddad, I., Ballesteros-Amozurrutia, M., Bosques-Padilla, F., Castillo-Barradas, M., Chávez-Barrera, J., Cisneros-Garza, L., Flores-Calderón, J., García-Compeán, D., Gutiérrez-Grobe, Y., Tijera, M. H., Kershenobich-Stalnikowitz, D., Guevara-Cetina, L. L., et. al.: Consenso mexicano de la enfermedad por hígado graso no alcohólico. Revista de Gastroenterología de México, vol. 84, no. 1, pp. 69–99 (2019) doi: 10.1016/j.rgmx.2018.11.007
2. Doğaner, A., Çolak, C., Küçükdurmaz, F., Ölmez, C.: Prediction of renal cell carcinoma based on ensemble learning methods. Middle Black Sea Journal of Health Science, vol. 7, no. 1, pp. 104–114 (2021) doi: 10.19127/mbsjohs.889492

3. Verma, A., Mehta, S.: A comparative study of ensemble learning methods for classification in bioinformatics. In: 7th International Conference on Cloud Computing, Data Science and Engineering Confluence, pp. 155–158 (2017) doi: 10.1109/confluence.2017.7943141
4. Buyrukoglu, S.: Improvement of machine learning models' performances based on ensemble learning for the detection of Alzheimer disease. In: 6th International Conference on Computer Science and Engineering pp. 102–106 (2021) doi: 10.1109/ubmk52708.20 21.9558994
5. Hakim, M. A., Jahan, N., Zerin, Z. A., Farha, A. B.: Performance evaluation and comparison of ensemble based bagging and boosting machine learning methods for auto mated early prediction of myocardial infarction. In: 12th International Conference on Computing Communication and Networking Technologies, pp. 1–6 (2021) doi: 10.1109/ICCCNT 51525.2021.9580063
6. Yadav, D. C., Pal, S.: An experimental study of diversity of diabetes disease features by bagging and boosting ensemble method with rule-based machine learning classifier algorithms. SN Computer Science, vol. 2, no. 1 (2021) doi: 10.1007/s42979-020-00446-y
7. Ponnaganti, N. D., Anitha, R.: A novel ensemble bagging classification method for breast cancer classification using machine learning techniques. Traitement du Signal, vol. 39, no. 1, pp. 229–237 (2022) doi: 10.18280/ts.390123
8. Rahman, F., Mahmood, M. A.: A dynamic approach to identify the most significant biomarkers for heart disease risk prediction utilizing machine learning techniques. Bangabandhu and Digital Bangladesh, pp. 12–22 (2022) doi: 10.1007/978-3-031-17181- 9_2
9. Chicco, D., Jurman, G.: An ensemble learning approach for enhanced classification of patients with hepatitis and cirrhosis. IEEE Access, vol. 9, pp. 24485–24498 (2021) doi: 10.1109/access.2021.3057196
10. Santos, M. S., Abreu, P. H., García-Laencina, P. J., Simao, A., Carvalho, A.: A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients. Journal of Biomedical Informatics, vol. 58, pp. 49–59 (2015) doi: 10.1016/j.jbi.20 15.09.012
11. Anisha, C., Saranya, K.: Early diagnosis of stroke disorder using homogenous logistic regression ensemble classifier. International Journal of Nonlinear Analysis and Applications, vol. 12, pp. 1649–1654 (2021) doi: 10.22075/ijnaa.2021.5851
12. Angular: What is angular? (2023) angular.io/guide/what-is-angular
13. Python: What is Python? Executive Summary (2022) www.python.org/doc/ essays/blurb
14. Python and REST APIs: Interacting with web services – real python (2022) real python.com/api-integration-in-python/#flask
15. Cirrhosis prediction dataset. Kaggle (2023) www.kaggle.com/datasets/fede soriano/cirrhosis-prediction-dataset
16. UCI machine learning repository: ILPD (indian liver patient dataset) data set (2022) archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+ Dataset%29
17. UCI machine learning repository: Liver disorders data set (2023) archive.ics.uci. edu/ml/datasets/liver+disorders
18. Non-alcohol fatty liver disease. Kaggle (2023) www.kaggle.com/datasets/ jamescorden/nonalcohol-fatty-liver-disease?select=nafld.csv

19. scikit-learn: Machine learning in Python. scikit-learn 1.2.2 documentation (2023) scikit-learn.org/stable/
20. sklearn.linear_model: LogisticRegression. scikit-learn 1.2.2 documentation, scikit-learn.org/stable/modules/generated/sklearn.linear_model.Logistic Regression.html
21. 1.4. Support vector machines. Scikit-learn 1.2.2 documentation (2023) scikit-learn.org/stable/modules/svm.html
22. sklearn.neighbors: KNeighborsclassifier. Scikit-learn 1.2.2 documentation (2023) scikit-learn.org/stable/modules/generated/sklearn.neighbors. KNeighborsClassifier.html
23. sklearn.tree.: DecisionTreeClassifier. Scikit-learn 1.2.2 documentation (2023) scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTree Classifier.html
24. sklearn.ensemble: RandomForestClassifier. Scikit-learn 1.2.2 documentation (2023) scikit-learn.org/stable/modules/generated/sklearn.ensemble.Random ForestClassifier.html

ISSN 1870-4069

Diseño de un sistema de reconocimiento de matrículas automotrices usando servidor Raspberry Pi e IA en Amazon Web Services

Ilse K. Leyva-Villanueva[1], Iván U. Aguilar-Pillado[1], Héctor R. Martínez-Anselmo[1], José J. Rodríguez-Senday[2]

[1] Universidad Tecnológica de Nogales, Sonora, México

[2] Instituto Tecnológico de Nogales, Sonora, México

karim.iklv@gmail.com

Resumen. El presente trabajo muestra cómo es posible que un sistema embebido constituido como un servidor apache, con pocos o casi nulos recursos de procesamiento para aplicaciones de inteligencia artificial pueda ser un medio para comunicarse y enviar información a los servicios en la nube de Amazon Web Services, AWS, y pueda utilizar esas aplicaciones para reconocimiento de matrículas automotrices. Los servicios en la nube que son proporcionados por AWS y Google son similares en cuanto a lo que ofrecen para el área de reconocimiento por visión artificial, pero se desconocen ciertos resultados cuando estos interactúan con sistemas embebidos como Raspberry Pi o cualquier otro. La finalidad es conocer la eficiencia de esos recursos en la nube para desarrollar aplicaciones en hardware que nos ayuden a simplificar algunas tareas de la vida diaria y posteriormente podamos iniciar con el desarrollo de aplicaciones diversas para solucionar algunas problemáticas de nuestro entorno social.

Palabras clave: AWS, Google, inteligencia artificial, sistemas embebidos.

Design of an Automotive License Plate Recognition System Using Raspberry Pi Server and AI in Amazon Web Services

Abstract. This paper shows how it is possible that an embedded system constituted as an Apache server, with little or no processing resources for artificial intelligence applications can be a means to communicate and send information to the Amazon Web Services cloud services, AWS, and can use those applications for license plate recognition. The cloud services provided by AWS and Google are similar in terms of what they offer in the area of machine vision recognition, but certain results are unknown when they interact with embedded systems such as raspberry pi or any other. The purpose is to know the efficiency of these resources in the cloud to develop applications in hardware that help us

to simplify some tasks of daily life and then we can start with the development of diverse applications to solve some of the problems of our social environment.

Keywords: AWS, Google, artificial inteligence, embedded system.

1. Introducción

Con los avances en electrónica y las comunicaciones, por internet han surgido los sistemas embebidos los cuales tienen un desempeño bastante aceptable para procesar información simple y son bastante buenos para comunicarse al internet y poder interactuar con páginas WEB y aplicaciones móviles.

Amazon Web Services, AWS, es un servicio en la nube que nos ofrece distintas características para el tratamiento de datos, siempre y cuando exista un medio que ayude a subirlos a la nube. Uno de los servicios que ofrece es el reconocimiento por visión artificial con algoritmos conceptuales de Machine y Deep learning.

El medio por el cual podemos subir esta información a la nube son los llamados sistemas embebidos, los cuales pueden ser constituidos como servidores apache mediante la instalación de ciertas librerías específicas. La desventaja de este tipo de hardware programable es su baja capacidad de procesamiento, aún y cuando se considera como una computadora personal, no tiene mucha memoria RAM, no tiene procesadores capaces de trabajar con algoritmos pesados en cuanto al cálculo necesario para procesar algoritmos de inteligencia artificial.

El presente trabajo muestra una manera de trabajar estos algoritmos de inteligencia artificial en la nube y los sistemas embebidos mediante el desarrollo de un proceso de investigación para reconocer matrículas automotrices.

2. Trabajos relacionados

Las organizaciones emplean cada vez más la IA para llevar a cabo complejas tareas de Inteligencia Artificial que antes se creía que solo los humanos eran capaces de realizar. En algunos ámbitos limitados del aprendizaje como servicio, la IA supera incluso el rendimiento de los humanos.

Para fomentar la difusión y aplicación de la IA, los proveedores de la nube, como Amazon, Google, IBM, Microsoft o Salesforce, han empezado a ofrecer aprendizaje automático, aprendizaje profundo, analítica e inferencia como servicio, llevando a la práctica el debate sobre el aprovisionamiento de capacidades de IA desde la nube [1]. [2] Explora el desarrollo de la Inteligencia Artificial y su impacto en los modelos de negocio, la organización y el trabajo.

En primer lugar, se ofrece una historia estilizada de la IA en la que se destacan los factores tecnológicos, organizativos y de mercado que fomentan su difusión y su potencial transformador. En segundo lugar, se analizan las consecuencias de la adopción de la IA para los modelos empresariales, la organización y el trabajo.

Este debate contribuye a mostrar cómo el desarrollo y la difusión de este ámbito tecnológico dan nueva fuerza al paradigma de lean production, tanto en el sector manufacturero como en el de servicios.

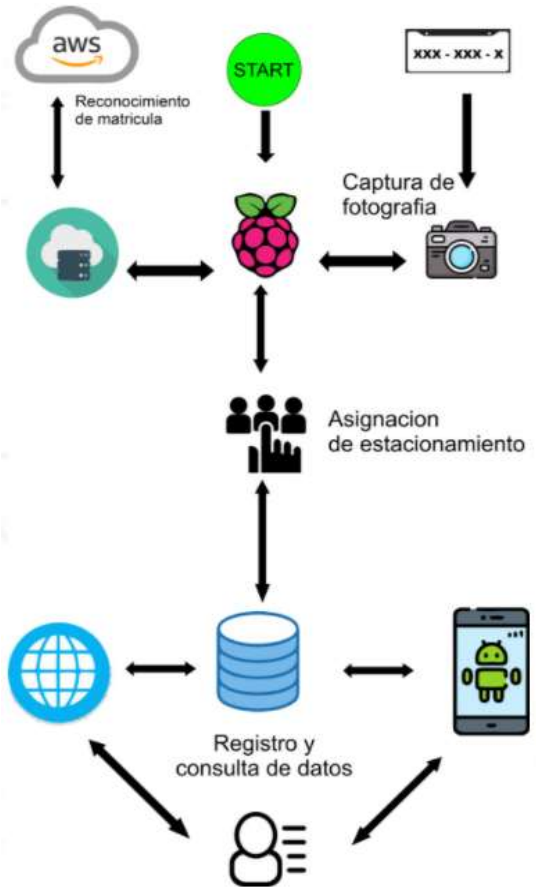


Fig. 1. Diagrama de bloques del proyecto propuesto.

Actualmente, existe un auge creciente sobre tecnologías basadas en microservicios y en el cloud computing, esto debido a su alta escalabilidad, mantenibilidad y facilidad para crear infraestructuras de forma segura. Amazon Web Services (AWS) ofrece diversos servicios que permiten convertir plataformas sencillas en aplicaciones robustas, usando diferentes tecnologías y bases de datos, así como también permite agregar seguridad tanto a las aplicaciones como a los datos que son la fuente principal de todo sistema [3].

En [4] se usó una unidad construida con Raspberry Pi, que se presenta como nodo central del sistema. Para recopilar datos de múltiples nodos sensores de forma rápida y enviar/recibir mensajes desde/hacia la aplicación que se ejecuta en la nube, se utiliza el protocolo ligero de transporte de mensajería publisher/subscriber, MQTT. [5] Presenta una solución que integra un asistente de voz en un escenario de oficina inteligente.

El interés particular era desarrollar el entorno para interactuar con dispositivos específicos de oficinas inteligentes, así como con herramientas de gestión de proyectos, ofreciendo nuevas perspectivas científicas que puedan guiar a los investigadores en futuros trabajos similares.

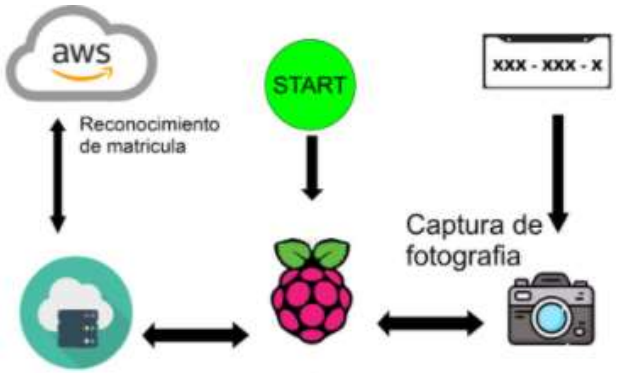


Fig. 2. Estructura de procesamiento con IA propuesta.

3. Sistema propuesto

Se propone un sistema de reconocimiento de matrículas automotrices que usa una Webcam para tomar estas fotografías, un sistema embebido raspberry pi configurado para trabajar como un servidor apache, el uso de los servicios de inteligencia artificial de AWS y todo esto trabajando en el concepto de internet de las cosas para probar la eficiencia que tienen estos algoritmos de inteligencia artificial. La figura 1 muestra el diagrama conceptual del proyecto.

La metodología propuesta para el término de este trabajo de investigación es dividir el trabajo en 2 casos.

La primera parte propuesta es constituir el servidor apache con la raspberri pi, configurada para la toma de fotografías de la matrícula del automóvil, esta puede ser guardada en cualquiera de los formatos conocidos, png, jpeg, etc.

Lo siguiente es a través del desarrollo de ciertos scripts de Python, que trabajan bien en el sistema embebido, subirlas a la plataforma de AWS, una vez estando estos datos en la nube de AWS, poder aplicarle los algoritmos de reconocimiento para determinar la cadena de caracteres de la matrícula, tal y como lo muestra la figura 2.

La segunda parte del proyecto, para proporcionarle un sentido de uso práctico es utilizar la información generada en la parte inteligente del mismo e interactuar mediante consultas a una base de datos construida en la nube de Google, en Firebase. Cuando el sistema inteligente terminó de procesar la información, se envía la cadena de caracteres de nueva cuenta al servidor con raspberry pi, no se sabe aún si es una cadena que corresponde a una matrícula.

Entonces, la raspberry establece una consulta a la base de datos para verificar que el usuario efectivamente se encuentre registrado y si eso resulta correcto, se enviará un aviso al dispositivo móvil del usuario avisándole que está activo en el sistema.

El usuario podrá interactuar también con una APP WEB para verificar su estado en el sistema. La figura 3 nos muestra cómo es esto posible.

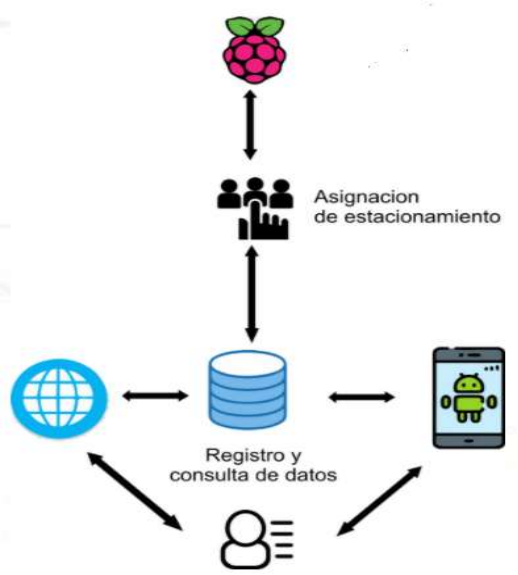


Fig. 3. Estructura del bloque de consulta para validar al usuario.

Fig. 4. Vista frontal de una matrícula.

4. Resultados

Se tomaron diversas fotografías de matrículas automotrices para probar el sistema de reconocimiento, en las cuales se cuidó que tuvieran ciertas características para hacer que el sistema tuviera retos en la identificación y que probablemente pudiera clasificar mal. La figura 4 muestra cómo sería una fotografía perfecta para el sistema, con una excelente nitidez y totalmente de frente.

La figura 5, muestra una fotografía con desviación hacia la izquierda y con menor grado de nitidez con respecto a la figura 4. Esta característica es con la finalidad de que el sistema de reconocimiento tenga un reto en la clasificación de la imagen.

La figura 6, muestra una matrícula escrita totalmente a mano y con un color distinto al que muestran de manera normal las matrículas automotrices, con el fin de agregarle otro reto más al sistema con el color y con los caracteres escritos de manera manual.

Fig. 5. Vista de lado de una matrícula.

Fig. 6. Matrícula escrita manualmente.

Se tomó un número de 30 fotografías que corresponden a la capacidad que tiene un estacionamiento, tratando de que cada una de las fotografías tuviera alguna de las características mostradas en las figuras 4, 5 y 6. Los resultados se muestran en las matrices de confusión indicadas en la tabla 2. Esta matriz de confusión propuesta es con la finalidad de mostrar las métricas de precisión y exhaustividad que nos servirán para determinar qué tan buena es la propuesta.

Los resultados se muestran la matriz de confusión de la tabla 2, tal como el ejemplo mostrado en la tabla 1.

- **TP:** Foto válida y el sistema la reconoció.
- **TN:** Foto no válida y el sistema no lo reconoció.
- **FP:** Foto no válida y el sistema la reconoce.
- **FN:** Foto válida y el sistema no la reconoce.

La matriz de confusión mostrada en la tabla 2, muestra como las 30 fotografías fueron clasificadas como TP, verdaderos positivos. Esto es, el renglón correspondiente tiene un valor booleano de 1 que corresponde a los que realmente es y la columna también tiene el mismo valor booleano de 1 que corresponde a lo que el sistema predice. Este tipo de matriz se utilizó para medir el rendimiento del sistema propuesto.

Tabla 1. Matriz de confusión propuesta.

		Predicción	
		0	1
Realidad	0	TN	FP
	1	FN	TP

Tabla 2. Matriz de confusión de resultados.

		Predicción	
		0	1
Realidad	0	0	0
	1	0	30

De los datos proporcionados por la matriz de confusión se calculó el parámetro de precisión, para determinar qué tan cercas estaban los resultados predichos de lo que realmente debería de ser:

$$Precisión = \frac{TP}{TP + TN} = 1. \quad (1)$$

Como podemos observar en el resultado, que fue de 1, lo podemos interpretar como excelente, ya que indica que cada una de las fotografías tomadas las clasificó de forma correcta:

$$Precisión = \frac{30}{30 + 0} = 1.$$

El siguiente parámetro calculado fue la exhaustividad para determinar que tan bien clasificaba el sistema:

$$Exhaustividad = \frac{TP}{TP + FN} = 1. \quad (2)$$

Como resultado de esto, el sistema indica que el hecho de identificar de manera correcta las 30 fotografías era lo indicado:

$$Exhaustividad = \frac{30}{30 + 0} = 1.$$

Adicionalmente, se obtuvieron Apps Web y Móviles que ayudan al usuario a utilizar de una manera más simple el sistema como es en el registro, en el acceso al sistema y sobre todo la App Web que sirve para que el administrador del sistema tenga datos confiables para la toma de decisiones, en las figuras 7, 8 y 9 se muestran estas interfaces.

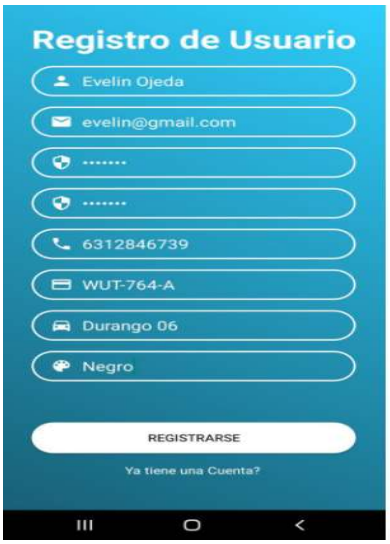


Fig. 7. App Móvil.

Fig. 8. Alta de usuarios en la App Web.

Fig. 9. Información para el administrado.

5. Conclusiones y trabajo a futuro

De acuerdo a lo que observamos en la matriz de confusión, las métricas indican que las herramientas de AWS en la nube con IA son excelentes inclusive, aún y cuando las fotografías de las matrículas tengan características que en su momento pueden hacer que el sistema identifique de manera errónea.

También probamos que es posible construir un servidor con el sistema embebido usado, raspberry pi, para preprocesar la fotografía de acuerdo a las instancias que solicita la plataforma de IA de AWS y llevar a cabo el trabajo de identificación.

El trabajo muestra también, como es posible que con las herramientas del internet de las cosas, IoT, se puede interconectar una App Web, App móvil, una base de datos de proveedores distintos a AWS, sistemas embebidos de tal manera que se construya un proyecto completo de gestión de información para el usuario.

Trabajos futuros implican darle aplicaciones al sistema desarrollado para ofertar diferentes servicios a los usuarios de las alternativas que ofrecen las empresas privadas como las instituciones públicas.

Referencias

1. Lins, S., Pandl., K. D., Teigeler, H., Thiebes, S., Bayer, C., Sunyaev, A.: Artificial intelligence as a service. Business and Information Systems Engineering, vol. 63, pp. 441–456 (2021) doi: 10.1007/s12599-021-00708-w
2. Fanti1, L., Guarascio, D., Moggi, M.: From Heron of Alexandria to Amazon's Alexa: A stylized history of AI and its impact on business models, organization and work. Journal of Industrial and Business Economics, vol. 49, pp. 409–440 (2022) doi: 10.1007/s40812-022-00222-4
3. Cárdenas-Sánchez, B. C., Olarte-Rojas, C. A.: Análisis de seguridad entre microservicios con Amazon web service. Revista Logos Ciencia and Tecnología, vol. 14, no. 2, pp. 42–52 (2022) doi: 10.22335/rlct.v14i2.1546
4. Khriji, S., Benbelgacem, Y., Chéour, R., Houssainil, D., Kanoun, O.: Design and implementation of a cloud-based event-driven architecture for real-time data processing in wireless sensor networks. The Journal of Supercomputing, vol. 78, pp. 3374–3401 (2022) doi: 10.1007/s11227-021-03955-6
5. Bogdan, R., Tatu, A., Crisan-Vida, M. M., Popa, M., Stoicu-Tivadar, L.: A practical experience on the Amazon Alexa integration in smart offices. Sensors, vol. 21, no. 734 (2021) doi: 10.3390/s21030734
6. Swaroop, T.: Create your own face recognition service with AWS Rekognition! (2022) https://youtu.be/oHSesteFK5c
7. Zaamout, E.: Laravel tutorial - Deploy any laravel app in AWS. AHT Cloud (2021) https://youtu.be/W2fQFbkEQo0
8. Aprendible: Aprende Laravel en 3 horas (2022) https://youtu.be/rQZmhqah0PQ
9. AWS: AWS SDK for PHP 3.x (2023) https://docs.aws.amazon.com/aws-sdk-php/v3/api/namespace-Aws.html
10. Microsoft: Overview of ASP.NET Core MVC (2022) https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-7.0

ISSN 1870-4069

Reconocimiento facial usando herramientas de IA de Amazon Web Services y sistemas embebidos

Eduardo Saavedra Quijada[2], Luis A Medina Muñoz[1], Felipe Morales Solís[1], Gabriel López Valencia[2]

[1] Instituto Tecnológico de Nogales, Sonora, México

[2] Universidad Tecnológica de Nogales, Sonora, México

eduardosaavedraq687@gmail.com

Resumen. Los diversos sistemas embebidos que actualmente están disponibles para los usuarios, aun y cuando tienen las características de una computadora personal, carecen de la capacidad de procesamiento necesaria para ejecutar trabajos de reconocimiento facial, a menos que, se le agreguen dispositivos coprocesadores como los neural computer stick de Intel o Coral USB accelerator de google. Amazon Web Services y recientemente google, tienen herramientas de inteligencia artificial que ayudan a acelerar la implementación de un proyecto de reconocimiento facial incorporadas en sus espacios virtuales llamados nube. El presente trabajo, muestra cómo es posible implementar acciones de reconocimiento facial en un sistema embebido, el cual es raspberry pi, cuya función es servir como un hardware auxiliar para subir fotografías grupales a la nube de Amazon Web Services y que con las herramientas de IA alojadas en ese medio se lleve a cabo el procesamiento pesado de reconocimiento facial y devuelva hacia al sistema embebido la información de cada uno de las personas presentes en la fotografía.

Palabras clave: Amazon web, google, reconocimiento facial, sistemas embebidos.

Facial Recognition Using AI Tools from Amazon Web Services and Embedded Systems

Abstract. The various embedded systems that are currently available to users, even though they have the features of a personal computer, lack the processing power needed to run facial recognition jobs, unless co-processor devices such as Intel's neural compute stick or Google's Coral USB accelerator are added. Amazon Web Services and recently Google have artificial intelligence tools that help accelerate the implementation of a facial recognition project embedded in their virtual spaces called cloud. This paper shows how it is possible to implement facial recognition actions in an embedded system, which is raspberry pi, whose

function is to serve as an auxiliary hardware to upload group photographs to the Amazon Web Services cloud and that with the AI tools hosted in that environment the heavy processing of facial recognition is carried out and returns to the embedded system the information of each of the people present in the photograph.

Keywords: Amazon web, google, facial recognition, embedded system.

1. Introducción

Los sistemas embebidos son herramientas de cómputo donde todos sus componentes están integrados en una sola placa, reduciendo con esto el tamaño, pero con características similares a una computadora personal, con una menor capacidad de procesamiento, sobre todo para ejecutar aplicaciones donde se utilicen herramientas de inteligencia artificial.

Estas debilidades del sistema embebido pueden ser cubiertas con el uso de herramientas de inteligencia artificial quizás, de los 2 proveedores más importantes como son Amazon Web Services y Google. Amazon Web Services provee de servicios en la nube a usuarios registrados para acelerar sus procesos de diseño basados en inteligencia artificial.

Estos procesos ayudan a los sistemas embebidos a formar parte de un proyecto de reconocimiento facial sin utilizar sus características de cómputo, únicamente convirtiéndose en un servidor de información hacia la nube, lo cual con sus propiedades de conectividad es relativamente sencillo.

Este trabajo propone utilizar Raspberri Pi como sistema embebido, el cual trabajará como medio para subir fotografías individuales o grupales a la nube de Amazon Web Services en la cual se llevará a cabo el procesamiento de la imagen con servicios de inteligencia artificial y regresará al servidor implementado con Raspberri Pi la información del o los individuos que están en esa fotografía.

2. Trabajos relacionados

En [1] se implementó en Python un prototipo de un sistema de reconocimiento facial usando machine learning, para lograrlo se usaron algunos temas relacionados con la inteligencia artificial y el impacto que ha tenido el desarrollo de estas tecnologías a nivel mundial, siendo implementadas en muchos campos como la medicina, la seguridad en los aeropuertos entre otras aplicaciones que han mejorado la calidad de vida de las personas.

La investigación “Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección” creo una aplicación web que permite monitorear el uso de mascarillas protectoras en ambientes públicos. Utilizando el framework Flask, en el lenguaje de Python, la aplicación cuenta con un panel de control que ayuda a visualizar los datos obtenidos. El proceso de detección utiliza el algoritmo Haar Cascade para clasificar rostros con y sin mascarillas protectoras.

Como resultado, la aplicación web es liviana y permite detectar y almacenar en la nube las imágenes capturadas y la posibilidad de un mayor análisis de datos.

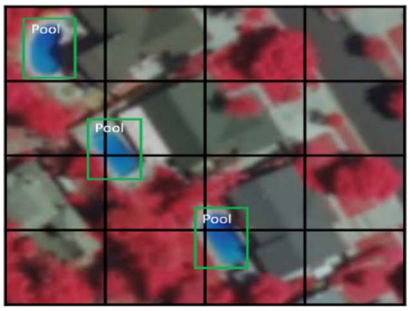


Fig. 1. Single shoot detector.

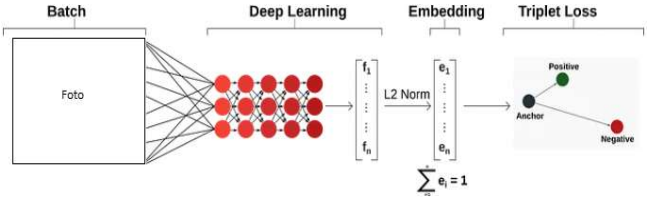


Fig. 2. Arquitectura de FaceNet.

El clasificador presenta precisión, revocación y f-score de 63%, 93% y 75%, respectivamente. Aunque la precisión fue satisfactoria, se realizarán nuevos experimentos para explorar nuevas técnicas de visión por computadora, como el uso de aprendizaje profundo [2].

En [4] muestran cómo implementar un servicio basado en la nube que proporciona los algoritmos faciales más avanzados de reconocimiento y detección de rostros con atributos bajo la plataforma Microsoft Azure.

Con lo que respecta a la implementación de estos servicios se llevaron a cabo experimentos diferenciados de cada una de las fases del desarrollo del proyecto, de modo que se puedan evaluar las fortalezas y debilidades del servicio en la nube. Los análisis de las imágenes procesadas se han centrado en observar el potencial de exactitud, la eficiencia y rapidez del servicio.

En [5] se presenta un método de reconocimiento de imágenes mediante el aprendizaje de diccionarios específicos de clase para separar las salidas de la red neuronal en características dependientes de la clase, potenciando así su capacidad de discriminación. Específicamente, se desarrolló una red de atención de clase (CANet) mediante la integración de un módulo de codificación de atención de clase específica (CAE) simple pero eficaz en la parte superior de las capas convolucionales.

En [6], se presenta una revisión de los avances recientes con sugerencias sobre las nuevas direcciones posibles para mejorar la eficiencia de los enfoques de reconocimiento visual relacionados con las DNN e inspirados en el cerebro, incluida la compresión eficiente de redes y las redes dinámicas inspiradas en el cerebro. Se Investiga no sólo desde el punto de vista del modelo, sino también de los datos (lo que no ocurre en los estudios existentes) y se centra en cuatro tipos de datos típicos (imágenes, vídeo, puntos y eventos).

Fig. 3.: Modo de prueba del sistema propuesto.

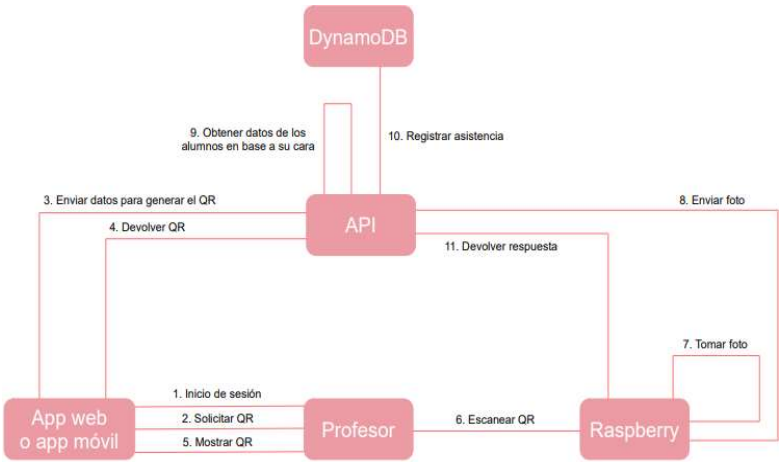


Fig. 4. Modelado gráfico de la aplicación.

Este estudio pretende ofrecer un resumen sistemático exhaustivo que pueda servir de valiosa referencia e inspirar tanto a investigadores como a profesionales que trabajen en problemas de reconocimiento visual.

3. Desarrollo

3.1. Single Shot Detector

El SSD tiene dos componentes: un modelo troncal y un principal. El modelo troncal suele ser una red de clasificación de imágenes preentrenadas como extractor de características. Normalmente se trata de una red como ResNet entrenada en ImageNet de la que se ha eliminado la última capa de clasificación totalmente conectada.

De este modo, obtenemos una red neuronal profunda capaz de extraer el significado semántico de la imagen de entrada, conservando al mismo tiempo la estructura espacial de la imagen, aunque a una resolución inferior.

Tabla 1. Porcentajes de similitud de Amazon Recognition.

No. de foto	% de similitud
Foto 1	99.999
Foto 2	99.997
Foto 3	99.999
Foto 4	99.983
Foto 5	99.981

Para ResNet34, la columna vertebral da como resultado 256 mapas de características de 7×7 para una imagen de entrada. El SSD principal no es más que una o varias capas convolucionales añadidas a esta columna vertebral y los resultados se interpretan como los cuadros delimitadores y las clases de objetos en la ubicación espacial de las activaciones de las capas finales.

En lugar de utilizar ventanas deslizantes, SSD divide la imagen utilizando una cuadrícula y hace que cada celda de la cuadrícula sea responsable de detectar objetos en esa región de la imagen. Detectar objetos significa simplemente predecir la clase y la ubicación de un objeto dentro de esa región.

Si no hay ningún objeto presente, lo consideramos como la clase de fondo y se ignora la ubicación. Por ejemplo, podríamos utilizar una cuadrícula de 4x4 como muestra la figura 1. Cada celda de la cuadrícula es capaz de emitir la posición y la forma del objeto que contiene [11]

3.2. FaceNet

FaceNet es un sistema de reconocimiento facial que aprende a asignar caras a una posición en un espacio multidimensional en el que la distancia entre puntos corresponde directamente a una medida de similitud facial.

1. Preprocesamiento: Un método utilizado para tomar un conjunto de imágenes y convertirlas todas a un formato uniforme, en este caso, una imagen cuadrada que contiene sólo la cara de una persona. Un conjunto de datos uniforme es útil para reducir la varianza durante el entrenamiento, ya que se disponen de recursos informáticos limitados al utilizar la TPU Edge.
2. Embedding: Un proceso fundamental para el funcionamiento de FaceNet, que aprende representaciones de caras en un espacio multidimensional donde la distancia corresponde a una medida de similitud de caras.
3. Clasificación: Paso final en el que se utiliza la información proporcionada por el proceso de incrustación para separar los distintos rostros [10].

3.3. Sistema Propuesto

El sistema propuesto se probó en un grupo de una institución educativa, identificando a los usuarios por clase asignada en diferentes horas y llevando un control

Fig. 5. Encuadre de caras para reconocimiento.

de lista de manera automatizada, en este caso solo se adjuntarán resultados de lo eficiente que es el sistema de reconocimiento facial.

El diagrama de bloque muestra que se puede usar una APP móvil o una APP web, el usuario es libre de elegir, de acuerdo a los recursos con los que cuenta, cuál de las 2 APPs usará. Iniciará sesión y solicitará generar un código QR que contenga la información que identifica a la materia que imparte, cuando este código es generado se le regresará al usuario a la APP para posteriormente ser usada en la siguiente etapa.

Un sistema embebido, raspberry pi, mediante el uso de una cámara WEB leerá el código QR y en un intervalo de tiempo de 5 segundos tomará una foto de manera grupal y subirá esta información a los servicios de Amazon para que inicie con el reconocimiento facial de las personas que se encuentren en dicha foto.

Las coincidencias de los rostros se basan en su geometría visual, incluida la relación entre los ojos, la nariz, las cejas, la boca y otras características faciales. Cuando Amazon Rekognition analiza imágenes, existe una línea alrededor del rostro, un recuadro, que determina la única parte de la imagen que Rekognition considera en su análisis.

A continuación, el análisis produce números de notación del objeto para la imagen que indican la "ubicación" de los elementos principales del rostro. Cuando los clientes ejecutan la búsqueda de un rostro, la tecnología compara los datos de la imagen fuente con cada una de las imágenes en las que busca. A partir de ahí, el servicio asigna a cada rostro de la imagen una puntuación de similitud. Este enfoque garantiza que Amazon Rekognition no tenga información sobre la identidad de una persona, sino solo de la probabilidad de que un rostro coincida con otro [9].

La tabla 1, indica la información del porcentaje de similitud de 5 fotos, tomadas aleatoriamente del conjunto de datos, para demostrar la forma como AWS indica el

Tabla 2. Matriz de confusión propuesta.

		Predicción	
		0	**1**
Realidad	**0**	TN	FP
	1	FN	TP

Tabla 3. Datos estadísticos tomando la fotografía de frente 126 fotos.

		Predicción	
		0	**1**
Realidad	**0**	0	0
	1	2	124

reconocimiento y como en esta información nos podemos dar cuenta que la identidad de la persona está reservada. Una vez finalizado el proceso de reconocimiento se hacen accesos a la base de datos para ver si es una persona existente o no y determinar la identificación de la misma con la información personal.

4. Resultados

Se construyó una matriz de confusión como la que se muestra en la tabla 3, la cual se llenó con la información obtenida del reconocimiento para calcular las métricas del sistema, que fueron 14 fotos grupales con 9 integrantes cada una, donde:

- **TP:** Está en la foto y el sistema lo reconoció.
- **TN:** No está en la foto y el sistema no lo reconoció.
- **FP:** No está en la foto y el sistema lo reconoce.
- **FN:** Está en la foto y el sistema no lo reconoce.

$$\text{Precisión} = \frac{124}{124+0} = 1, \tag{1}$$

$$\text{Recall} = \frac{124}{124+2} = 0.98, \tag{2}$$

$$\text{F}_1 = 2\left(\frac{(0.98)(1)}{0.98+1}\right) = 0.98, \tag{3}$$

$$\text{Precisión} = \frac{101}{101+0} = 1, \tag{4}$$

$$\text{Recall} = \frac{101}{101+25} = 0.80, \tag{5}$$

$$\text{F}_1 = 2\left(\frac{(0.80)(1)}{0.80+1}\right) = 0.88. \tag{6}$$

Tabla 4. Datos estadísticos tomando la fotografía agregando un cubrebocas a 2 personas.

		Predicción	
		0	**1**
Realidad	**0**	0	0
	1	25	101

5. Conclusiones y trabajo a futuro

De acuerdo a los resultados obtenidos, vemos primeramente que el posible implementar diversas aplicaciones con el concepto de internet de las cosas, es decir, comunicar APPs de servicios WEB y dispositivos móviles y los diferentes servicios en la nube como pueden ser de Google o AWS, en el caso de este trabajo fue con AWS. Todo esto no sería posible en el concepto en el cual se realizó el trabajo sin un sistema embebido como Raspberry Pi.

También, este trabajo muestra que es posible que los recursos de procesamiento para aplicaciones que ocupen inteligencia artificial, como lo es el reconocimiento de rostros, y de los cuales los sistemas embebidos no cuentan con la suficiente capacidad para ello, los recursos de procesamiento que se encuentran en el caso nuestro en AWS son suficientes y pueden ser utilizados en aplicaciones específicas.

Las métricas porcentuales indicadas por AWS en promedio, si las analizamos, nos dicen el sistema de reconocimiento es bastante bueno con alrededor de menos del 1% de error, datos que son comprobados al implementar las matrices de confusión y que arrojan una precisión en el reconocimiento facial correcto del 100%. Trabajos futuros implica darle aplicaciones educativas como toma de lista, tutorías y otros servicios relacionados con el alumno.

Referencias

1. Baresi, L., Colazzo, S., Mainetti, L., Morasca, S.: W2000: A modelling notation for complex web applications. In: Web Engineering, pp. 335–364 doi: 10.1007/3-540-28218-1_11
2. Koch, N., Knapp, A., Zhang, G., Baumeister, H.: UML-based web engineering. In: Web Engineering: Modelling and Implementing Web Applications, pp. 157–191 (2008) doi: 10.1007/978-1-84628-923-1_7
3. Sánchez-Santamaria, M., García-García, L. A.: La ingeniería web: Desarrollo de aplicaciones web de alta calidad. Revista de Divulgación Científico-Tecnológica del Gobierno del Estado de Morelos (2011)
4. Martín-Vera, P., Pons, C., Gonzáles, C., Giulianelli, D. A., Rodríguez, R. A.: Metodología de modelado de aplicaciones web móviles basada en componentes de interfaz de usuario. In: Argentine Symposiumon Software Engineering (2013)
5. Cheng, G., Lai, P., Gao, D., Han, J.: Class attention network for image recognition. Science China Information Sciences, vol. 66, no. 3 (2023) doi: 10.1007/s11432-021-3493-7
6. Wu, Y., Wang, D., Lu, X., Yang, F., Yao, M., Dong, W., Shi, J., Li, G.: Efficient visual recognition: A survey on recent advances and brain-inspired methodologies. Machine Intelligence Research, vol. 19, no. 5, pp. 366–411 (2022) doi: 10.1007/s11633-022-1340-5

7. Moroni, N., Señas, P.: Uso de grafos para el modelado de experiencias educativas colaborativas basadas en la web. In: VI Workshop de Tecnología Informática Aplicada en Educación, pp. 1134–1146 (2007)
8. Bouchrika, I., Ait-Oubelli, L., Rabir, A., Harrathi, N.: Mockup-based navigational diagram for the development of interactive web applications. In: Proceedings of the 2013 International Conference on Information Systems and Design of Communication, pp. 27–32 (2013) doi: 10.1145/2503859.2503864
9. AWS: Datos sobre el reconocimiento facial mediante inteligencia artificial. Amazon Web Services (2023) aws.amazon.com/es/rekognition/the-facts-on-facial-recognition-with-artificial-intelligence/
10. ArcGIS developers: How single-shot detector (SSD) works? ArcGIS API for Python (2023) evelopers.arcgis.com/python/guide/how-ssd-works/
11. Madio, P.: A facenet-style approach to facial recognition on the google coral development board. Towards Data Science (2019)
12. Lipschutz, S.: Theory and problems of set theory and related topics. McGraw-Hill, pp. 5–6 (1998)
13. Lipschutz, S., Lipson, M.: Discrete Mathematics. McGraw-Hill, pp. 1–22 (2007)
14. Swaroop, T.: Create your own face recognition service with AWS Rekognition! (2022)
15. AHT cloud: Laravel tutorial - Deploy any Laravel app in AWS (2021) youtu.be/W2fQFbkEQo0
16. Aprendible: Aprende Laravel en 3 horas (2022) youtu.be/rQZmhqah0PQ
17. AWS: AWS SDK for PHP 3.x (2023) docs.aws.amazon.com/aws-sdk-php/v3/api/namespace-Aws.html
18. Microsoft learn: Overview of ASP.NET Core MVC, ASP.NET (2023) learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-7.0

ISSN 1870-4069

Desarrollo de sistemas de apoyo al diagnóstico y aplicación de pruebas psicométricas mediante chatbots con inteligencia artificial para profesionales de la salud: AMEL-IA

Arturo Jair Soto-Bahena[1], Dania Nimbe Lima-Sánchez[2], Mahuina Campos-Castolo[2], Alejandro Alayola-Sansores[2], Germán Fajardo-Dolci[2], Jennifer Hincapié-Sánchez[3]

[1] Universidad Nacional Autónoma de México, Facultad de Psicología, México

[2] Universidad Nacional Autónoma de México, Facultad de Medicina, México

[3] Universidad Nacional Autónoma de México, Programa Universitario de Bioética, México

light_arn@outlook.es, infobiomedix@facmed.unam.mx, {dibfm, ale.alayola,german.fajardo, jhincapie }@unam.mx, ale.alayola@gmail.com

Resumen. A raíz del confinamiento social causado por la pandemia de COVID-19, los actos de violencia doméstica han ido en aumento, especialmente los suscitados dentro de las relaciones de pareja. Debido a esto, organismos tanto públicos como privados han comenzado a actuar a favor de las victimas lanzando programas de apoyo y atención a distancia. Es así que, gracias al avance tecnológico, este apoyo ha permitido la creación de Amelia, una healthbot con Inteligencia Artificial (IA) de IBM cuyo propósito es apoyar al profesional de la salud en cuanto a la detección, diagnóstico y prevención de signos de violencia de género.

Palabras clave: Género, Chatbot, IA.

Development of Diagnostic Support Systems and Application of Psychometric Tests Through AI-Powered Chatbots for Healthcare Professionals: AMEL-IA

Abstract. As a result of social confinement due to COVID-19, the cases of domestic violence had raised, especially those linked to couple relationships. Because of this, public and private organizations had launched support programs in favor of the victims. Technological advancement is also part of this, is that Amelia was created, a Healthbot made with Artificial Intelligence (AI) from IBM

whose purpose is to support the healthcare professional regarding to detection, diagnostic and prevention of gender violence.

Key words: Gender, Chatbot, AI.

1. Introducción

Desde el 20 de marzo de 2020 en México [1] comenzó, a recomendación del gobierno del país y de la Organización Mundial de la Salud un confinamiento social que obligó a las personas a permanecer en sus hogares mientras durara la pandemia del SARS-CoV-2, esto, desafortunadamente llevó a que la violencia doméstica y de género se agravara y dificultara a las victimas buscar ayuda profesional.

Es por ello que, para todo profesional de la salud, ya sea psicólogo, médico o incluso trabajador social, sea pertinente adaptarse a las nuevas tecnologías para mejorar y agilizar sus estrategias de detección e intervención. Sin embargo, en dichas profesiones, al ser disciplinas dominadas por la intervención práctica y la interacción persona a persona, pocas veces se toma en cuenta el papel que la tecnología puede jugar al momento de complementar procesos de sanación e intervención.

A nivel mundial existen 5.1 billones de usuarios con teléfonos inteligentes y más de 4 millones de usuarios de internet [2] por lo que es de vital importancia aprovechar esto a favor del avance y la atención médica.

En este caso, gracias a herramientas en línea y al avance tecnológico es que servicios como Watson Assistant de IBM [3] es que ha sido posible crear chatbots especializados en el cuidado de la salud, como Amelia, cuyo propósito y capacidad le permiten charlar con cualquier usuario que disponga de un dispositivo inteligente con acceso a internet, brindándole suficientes datos para determinar si necesita la intervención de un profesional o no.

2. Teoría

El 31 de diciembre de 2019, la Comisión Municipal de Salud de Wuhan, China, notifica un conglomerado de casos de neumonía en la ciudad, posteriormente se determina que estos casos fueron causados por un nuevo tipo de Coronavirus, es entonces cuando la Organización Mundial de la Salud (OMS) establece un estado de emergencia para abordar el brote.

Es así como el 11 de marzo de ese mismo año, la misma organización determina que la COVID-19 puede caracterizarse como una pandemia [4] lo que llevó a países de todo el mundo a entrar en un estado de cuarentena.

En México, fue a partir del mes de marzo de 2020 que se implementó la llamada “Jornada Nacional de Sana Distancia”, la cual consistió en que todos los habitantes del país debían permanecer en sus hogares y preferentemente evitar espacios abiertos.

Esto inicialmente no significó un cambio en los reportes de violencia de género, sin embargo, en meses posteriores la sigla de llamadas de emergencia al 9-1-1 relacionadas con la solicitud de ayuda por violencia contra las mujeres incrementó significativamente, pasando de 197,693 llamadas en el 2019 a 260,067 en el 2021, demostrando un aumento aproximado del 31.5% [5].

La Ley General de Acceso de las Mujeres a una Vida Libre de Violencia [6] define la violencia contra las mujeres como "cualquier acción u omisión basadas en su género que le cause daño o sufrimiento psicológico, físico, patrimonial, económico, sexual o la muerte a cualquier mujer, tanto en el ámbito privado como en el público".

Ante el mencionado incremento de la violencia en el país y la necesidad de atención psicológica, es pertinente para todo profesional buscar alternativas para mejorar y agilizar las estrategias de detección e intervención.

Normalmente se conocen más los tratamientos alternativos propios de cada cultura, como son el yoga, la acupuntura y hasta el taichí, que, aunque tienen cierto grado de acción terapéutica, su impacto puede resultar mínimo o dependiente de un acceso más especializado [7] es por ello que, para brindar una atención eficiente, es necesario un medio capaz de resolver estos problemas; el tiempo, la distancia y la accesibilidad.

Estudios como el de Epalza, en 2014 y Basantes, en 2017, afirman que la utilización de dispositivos móviles constituye un potencial para el desarrollo del aprendizaje de las personas más jóvenes, impulsando su motivación, satisfacción de interacción y además estimulan el pensamiento crítico y reflexivo [7].

Pero no solo eso, el hecho de que la mayor parte de la población ya disponga de dispositivos inteligentes, brinda la posibilidad de aprovechar esto a favor de los profesionales de la salud mediante un puente de comunicación entre ambos; aquí es cuando entran los robots de texto, o chatbots.

La importancia de la utilización de chatbots contra la intervención directa de persona a persona radica en diversos factores, el primero de ellos es la accesibilidad, siendo que gracias a que la mayor parte de personas poseen dispositivos inteligentes, resulta más rápido acceder a una página web que a un profesional inmediato.

De igual forma, en situaciones que requieran de distanciamiento social, o, incluso, ante la imposibilidad debida a la distancia física, disponer de un robot de texto continúa siendo más accesible. Otra de las razones es la velocidad de aplicación, ya que, en el caso del profesional le podría ser más eficiente utilizar a la inteligencia artificial como apoyo para recabar datos de diagnóstico o de contacto de manera simultánea en lugar de interactuar directamente con cada uno.

Fue en los 40 cuando el pionero de la computación y considerado padre de la Inteligencia Artificial, Alan Turing, rompió la línea entre ficción y realidad al escribir acerca de la inteligencia de las máquinas y la creación de un test capaz de atribuir el status de máquina pensante a una computadora; fue así como se creó el Test de Turing, una prueba que consiste en la interacción entre un chatbot generado por computadora, una persona real y un juez real, donde los primeros dos conversan y el último debe identificar quién es la persona, y si resulta que no lo logra, entonces la máquina habrá engañado a la percepción del ser humano.

Al principio fue muy complicado para muchas máquinas obtener un buen puntaje en dicho test, y ninguna lo logró. Sin embargo, con el tiempo, la tecnología fue mejorando, hasta que, en 2001 [8] un bot denominado Eugene Goostman, logró superar el Test de Turing con un 33% de éxito, lo cual es suficiente para considerársele inteligente.

El avance no terminó con Goostman, sino todo lo contrario, pues paralelamente IBM, o International Business Machines Corporation, se encontraba desarrollando computadoras cada vez más capaces de desafiar a la inteligencia natural humana, como Deep Blue, la cual se coronó con el campeonato mundial de ajedrez de 1997, o incluso Watson, el software con el que se trabaja en esta investigación, que fue capaz de

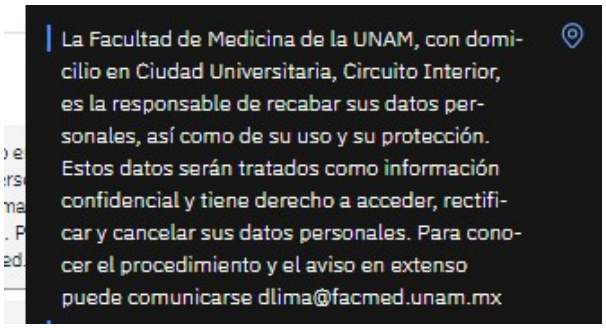


Fig. 1. Ejemplo del tratado de privacidad que se muestra al usuario después de contestar que sí permite que se recopilen sus datos.

finalizar el concurso de conocimientos estadounidense Jeopardy [9] para luego ser liberado al público como herramienta de libre acceso.

A mediados de la primera década de los 2000 comenzaron a desarrollarse los llamados robots sociales, los cuales son máquinas con aspectos humanoides programadas para interactuar con el ser humano, con un fin especializado.

Como el proyecto AURORA (AUtonomous RObotic platform as a Remedial tool for children with Autism), cuya finalidad fue la de probar la eficacia terapéutica de la interacción entre máquina y niños que se encuentran dentro del espectro autista, la cual resultó efectiva y dio origen al desarrollo de más máquinas con este propósito.

Otro ejemplo es NAO, un robot capaz de alentar a niños a realizar acciones y premiarlos por sus logros[10a], o Kaspar que es capaz de realizar movimientos y expresiones básicas que ayudan al niño a mejorar su capacidad de interacción [11]. Sin embargo, los costos de estos autómatas rondan cerca de los 9,000 euros (o 216,312 pesos mexicanos), en el caso de NAO, o 265 euros (6300 pesos mexicanos), en el caso de alternativas basadas en NAO, como otro autómata de origen español llamado Aisoy[10b].

En los años 2020 y 2021, en plena pandemia por SARS-CoV-2, es cuando los healthbots, que son chatbots destinados a la atención especializada para la salud, cobraron vital importancia, pues ante un contexto donde difícilmente las personas pueden salir de casa, su único recurso es la comunicación vía infraestructura, como es el internet. De igual manera, sumado a estas herramientas, el avance de la Inteligencia Artificial (IA) también se hizo más presente en la vida de todos.

Existen diferentes maneras de definir lo que es la Inteligencia Artificial (IA) dependiendo desde el enfoque desde el que se le observe. La primera de estas definiciones, que se basa en la Inteligencia Artificial como una disciplina de la informática, dicha en 2009 por el Dr. David Hanson Jr., director ejecutivo de Hanson Robotics, es “La inteligencia artificial es el campo de la informática que estudia cómo computar tareas tales como la percepción, el razonamiento, y el aprendizaje; y permitir así el desarrollo de sistemas que lleven a cabo estas capacidades” [12].

Otra definición, basada en el marco de la inteligencia humana y las ciencias cognitivas, es aquella dada por Marvin Minsky, citada en un obituario hecho por Javier Sampedro (2016) “Es la ciencia de hacer que las máquinas hagan cosas que requerirían inteligencia si las hubiera hecho un humano” [13].

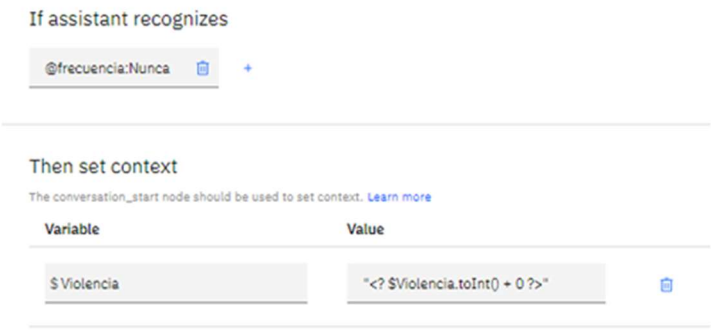


Fig. 2. Ejemplo de la variable $Violencia, para la cual se utiliza la línea de código [valor = "<? $valor.toInt() + 1 ?>"] para sumar un puntaje de 1 en caso de que la respuesta sea "Nunca".

3. Otros proyectos relacionados en México

Los healthbots, al ser principalmente código de programación, la mayoría están disponibles en la red, por lo que son capaces de ayudar en diversas tareas, ya sea de manera independiente, retroalimentando las respuestas de los usuarios, o de manera dependiente, funcionando como atención a clientes para conectar con agentes humanos [6].

Algunos de los healthbots más conocidos son Yana[14a] o Violetta[15b], que son de origen mexicano y ayudan al acompañamiento emocional y la psicoeducación de género, respectivamente.

En el caso de YANA, por sus siglas "You Are Not Alone" es un chatbot creado por Andrea Campos Guerra, es una aplicación que ha adquirido mucha popularidad en los últimos dos años, pues se basa en el acompañamiento emocional para descubrir si el usuario padece depresión o ansiedad.

Comenzó como un proyecto escolar en el año 2016, cuando la creadora, tras un episodio de depresión y haber tomado terapia cognitivo-conductual, tuvo la idea de automatizar lo que ella aprendió y llevarlo a una aplicación compatible con Android e IOS. Luego, en febrero de 2020, lanzó al mercado gratuitamente su aplicación, que desde entonces ha sido descargada más de 1.6 millones de veces[14b].

Luego está Violetta, una chatbot creada por alumnos del Instituto Tecnológico de Monterrey como parte de un proyecto para detectar casos de violencia doméstica. Este chatbot tiene un funcionamiento similar al de YANA, ya que es capaz de detectar palabras y frases que usarían víctimas potenciales de violencia doméstica, para luego, en caso de detectarla, brindar psicoeducación al usuario[15b].

Aunque estas dos últimas creaciones se han percibido de manera eficiente por la sociedad y los medios, realmente ninguna está vinculada a la comunidad de investigación científica, o, en este caso, a la Universidad Nacional Autónoma de México, la cual únicamente cuenta con un bot de renombre, el cual fue EMI, que en 2018 brindó información acerca de las elecciones presidenciales [16], pero que actualmente ya no se encuentra operativo.

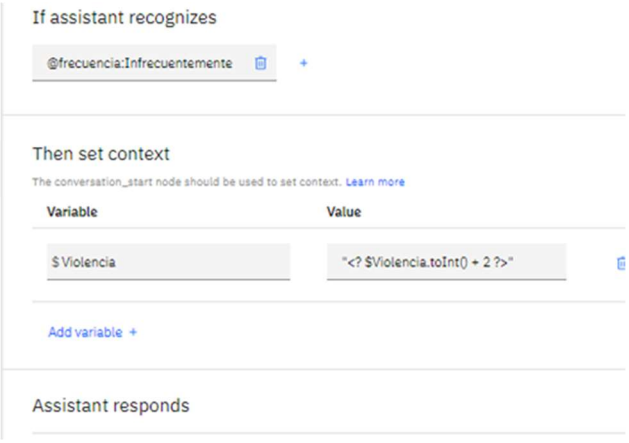


Fig. 3. Ejemplo de la variable $Violencia, para la cual se utiliza la línea de código [valor = "<? $valor.toInt() + 2 ?>"] para sumar un puntaje de 2 en caso de que la respuesta sea “Infrecuentemente”.

4. Aplicación y resultados de AMELIA

Se trabajó con una muestra de 55 participantes, todos pertenecientes al primer año del nivel licenciatura en la Facultad de Medicina de la UNAM. Como criterios de exclusión hubo que no fueran menores de 18 años o no pertenecieran a la carrera de Medicina.

Se trabajó con la Escala de Factores de Riesgo Asociados a Violencia de Género y la Escala de Autoestima de Rosenberg, ambos ya validados para su uso. El primero con el fin de encontrar factores de riesgo asociados a violencia de género en relaciones de pareja y el segundo para medir puntajes de autoestima. En el caso de ambas escalas, se convirtieron sus preguntas a un lenguaje informal que fuera más compatible con los fines del experimento.

Se utilizó el software de uso abierto Watson Assistant, de IBM, el cual brinda la posibilidad de programar de manera simple un chatbot. En el caso del experimento, se utilizó para crear a “Amelia”, a donde se implementarían los dos instrumentos anteriores.

En la programación de Amelia se incluyeron funcionalidades adicionales a las de generar puntajes correspondientes a las dos escalas. Entre estas funcionalidades se encuentra la posibilidad de reconocer el nombre del usuario y guardarlo en una variable cuando este se lo dice.

Adicionalmente, Amelia es capaz de guardar otras variables, tales como la edad, el consumo de alcohol, el permiso para recabar datos y cada uno de los puntajes que se obtienen con los 50 reactivos.

Es decir; si en una pregunta la respuesta es “Totalmente”, se genera una variable que vale 5 puntos, y si en la siguiente pregunta se recibe la misma respuesta, ambas variables se sumarán, creando un resultado de 10 puntos, que posteriormente servirán para determinar resultados.

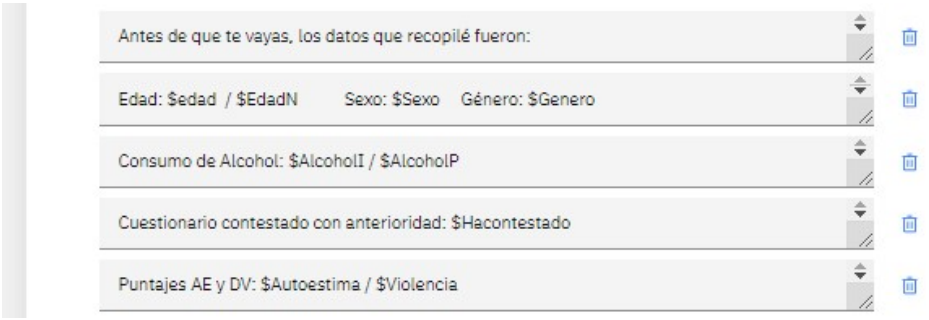


Fig. 4. Las variables que se muestran al usuario son: Edad, Sexo, Género, Consumo de Alcohol propio y de pareja, Si ha contestado antes el mismo cuestionario y finalmente los puntajes, uno de autoestima y otro de violencia.

Antes de comenzar el cuestionario, se le muestra un tratado de privacidad respecto al uso de sus datos personales. Como se muestra en la figura 1.

Luego, en la programación de las entidades del Healthbot, se utilizaron funciones capaces de agregar puntajes a las posibles respuestas que el usuario podía dar ante las preguntas. Siendo "nunca" el menor valor, "infrecuentemente" como el valor que le sigue, luego "frecuentemente", "muy frecuentemente" o "siempre", respectivamente. Como se puede observar ejemplos en las figuras 2 y 3.

El siguiente paso consistió en programar al algoritmo de Amelia para que reconociera más opciones de respuesta. Así, el usuario podría responder "Nunca" o "Jamás" o cualquier sinónimo, y el chatbot lo detectaría como la misma entrada, por lo que sumaría el mismo puntaje. Esto con el fin de hacer más amigable la interacción con el usuario.

5. Resultados

Al finalizar el cuestionario, al usuario se le muestra una recopilación de las variables guardadas durante su interacción con Amelia, como se muestra en la Figura 4.

Como punto final, respecto a los puntajes; si el usuario obtiene un puntaje de Violencia de Género equivalente o mayor a 17, que fue el que se manejó con la escala original, se le ofrece la posibilidad de contactar o ser contactado por un profesional.

En el caso de elegir "Contactar", se le muestran dos opciones distintas; una siendo el correo de contacto del departamento de Informática Biomédica y el otro número referente a la línea de atención psicológica a distancia de la Facultad de Psicología de la UNAM.

En cualquiera de ambos casos, se verifican los datos de contacto del paciente y se comienza a formular un expediente para poder realizar un seguimiento.

En el caso de elegir "Ser contactado/a", Amelia le solicita al usuario su correo electrónico y número de teléfono, los cuales se guardan como nuevas variables para que los encargados del proyecto recopilen y trabajen.

Una vez que se confirman estos datos, el equipo encargado del chatbot se encarga de canalizar a la persona con algún profesional pertinente; ya sea con psicólogos, trabajadores sociales o incluso con personal jurídico en caso de ser necesario.

Fig. 5. En la plataforma de IBM se muestra una gráfica de conversaciones y reconocimiento de palabras. Hasta el día de hoy, Amelia ha sido capaz de reconocer el 95% de los mensajes, que equivalen a 1778 mensajes de usuarios.

Afortunadamente, durante el piloteo no se presentaron situaciones en que algún participante requiriera de intervención psicológica, sin embargo, se sigue revisando constantemente las respuestas registradas en el proyecto para poder brindar un apoyo eficiente.

6. Análisis de datos

Watson Assistant permite la visualización de resultados en cuanto a reconocimiento de mensajes del usuario, es decir, cada que el chatbot reconoce lo que significa un mensaje de usuario y logra transformarlo a una variable cuantitativa. Como se puede observar en la figura 5.

7. Conclusiones

Aunque los bots con Inteligencia Artificial aún están lejos de poseer las capacidades comunicativas de una persona real, sí han demostrado ser más eficientes en cuanto a velocidad y accesibilidad que las aplicaciones clásicas de cuestionarios en aquellas disciplinas cuya tradición se basa principalmente en la interacción persona a persona, como la medicina o la psicología.

Gracias al uso de tecnología vinculada a las redes sociales y dispositivos inteligentes, se ha permitido a profesionales de la salud acceder a nuevas oportunidades que les permiten agilizar su proceso de intervención, pues la implementación de chatbots no significa que su labor se vea reemplazada, sino más bien, complementada.

De igual forma, para aquellos usuarios que por alguna razón no tenían la oportunidad de trasladarse a un consultorio para una evaluación, ya sea por tiempo o falta de recursos, basta con tener una computadora o un celular con acceso a internet para disponer del apoyo inmediato de un chatbot.

En este caso, Amelia ha demostrado ser de gran apoyo, gracias al alto porcentaje de reconocimiento de palabras y al algoritmo de la Inteligencia Artificial de IBM que le permiten, con ayuda del usuario programador, eventualmente aprender más palabras y

expresiones que permitirán una mejora de su funcionalidad, lo que permitiría no solamente utilizar al chatbot como aplicador de instrumentos, sino complemento profesional para el médico o el psicólogo.

Referencias

1. Cárdenas, K.: Se duplican llamadas de auxilio por violencia contra mujeres (2021) https://www.elsoldecuernavaca.com.mx/policiaca/se-duplican-llamadas-de-auxilio-por-violencia-contra-mujeres-7223502.html
2. Kemp, S.: Digital 2020: The Philippines (2020) https://datareportal.com/reports/digital-2020-philippines
3. International business machines corporation: Watson assistant: Build better virtual agents, powered by AI (2022) https://www.ibm.com/products/watson-assistant
4. Organización mundial de la salud, OMS: Cronología de la respuesta de la OMS a la COVID-19 (2021) https://www.who.int/es/news/item/29-06-2020-covidtimeline
5. Instituto nacional de las mujeres: Las mujeres y la violencia en tiempos de pandemia. Año 7, no. 3 (2021) http://cedoc.inmujeres.gob.mx/documentos_download/BA7N03%20Para%20Publicar%20con%20vo%20bo.pdf
6. Peña-Martínez, A. C.: Integración de las terapias alternativas y complementarias al sistema nacional de salud. Universidad Autónoma de Madrid (2018) https://repositorio.uam.es/bitstream/handle/10486/685022/pena_martinez_ana%20cristinatfg.pdf
7. Fresneda, C.: Un ordenador logra superar por primera vez el test de Turing (2014) https://www.elmundo.es/ciencia/2014/06/09/539589ee268e3e096c8b4584.html#:~:text=Por%20primera%20vez%20desde%20que,al%20nombre%20de%20Eugene%20Goostman
8. Rennie, J.: How IBM´s Watson computer excels at Jeopardy! PLOS Blogs (2013) http://www.cs.cornell.edu/courses/cs6700/2013sp/readings/01-a-Watson-Short.pdf
9. Rodríguez-Canfranc, P.: Robots sociales que ayudan al niño autista a abrirse al mundo. Fundación Telefónica (2021) https://telos.fundaciontelefonica.com/la-cofa/robots-sociales-que-ayudan-a-abrirse-al-mundo-al-nino-autista/
10. Corona, L.: Con una aplicación, esta desarrolladora buscaba tratar su depresión y terminó creando una plataforma de ayuda para 1.6 millones de personas. Business Insider México (2022) https://businessinsider.mx/aplicacion-depresion-premio-pandemia/
11. Bar-Cohen, Y., Hanson, D.: The coming robot revolution: Expectations and fears about emerging intelligent, humanlike machines. Editorial Springer (2009)
12. Sampedro, J.: Marvin Minsky, cerebro de la inteligencia artificial. El País (2016) https://elpais.com/elpais/2016/01/26/ciencia/1453809513_840043.html
13. Treviño, C.: ¡A la final del reto de Alibaba! Crean IA para detectar la violencia. Redacción CONECTA, El sitio de noticias del Tecnológico de Monterrey (2020). https://tec.mx/es/noticias/nacional/emprendedores/reto-global-inteligenciaartificial detecta-violencia
14. Webster, W.: Este robot ayuda a los niños autistas. RedBull (2018) https://www.redbull.com/mx-es/kaspar-el-robot-social-que-ayuda-a-ninos-autistas
15. Morris, R. R., Kouddous, K., Kshirsagar, R., Schueller, S. M.: Towards an artificially empathic conversational agent for mental health applications: System design and user perceptions. Journal of Medical Internet Research, vol. 20 no.6 (2018) doi: 10.2196/10148
16. Redacción Aristegui Noticias: EMI, el "chatbot" que brinda información sobre las elecciones (2018) https://aristeguinoticias.com/2105/mexico/emi-el-chatbot-que-brinda-informacion-sobre-las-elecciones/

ISSN 1870-4069

Aplicación de algoritmos de aprendizaje automático sobre un corpus depresivo digital

César-Jesús Núñez-Prado[2], Claudia Talavera Ortega[1], Liliana Chanona-Hernández[1], Grigori Sidorov[2]

[1] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, México

[2] Instituto Politécnico Nacional, Centro de Investigación en Computación, México

{cesar.jnprado, claudiatalaveraor, lchanona}@gmail.com, sidorov@cic.ipn.mx

Resumen. Uno de los principales objetivos que estimula al avance tecnológico; sin importar el campo de desarrollo, es el de hacer más sencilla y cómoda la vida de las personas. Estos avances en el área de la inteligencia artificial han impulsado a los algoritmos de aprendizaje automático para realizar la clasificación de ciertos objetos gracias al reconocimiento de patrones que descubren durante las fases del entrenamiento. Dichos algoritmos en conjunto con técnicas de procesamiento de lenguaje natural están capacitados para realizar la detección de emociones en el discurso escrito. Una de estas emociones es la depresión, la cual está catalogada como uno de los trastornos más comunes y además más letales que puede afectar a la población mundial y ello nos motiva a dirigir nuestra investigación a aplicar algunos modelos de aprendizaje automático a la detección del posible discurso de depresión sobre un corpus lingüístico digital en español construido a partir de mensajes publicados en la red social Twitter y evaluado por profesionales en el campo de la sicología.

Palabras clave: Inteligencia artificial, reconocimiento de patrones, aprendizaje automático, procesamiento de lenguaje natural, depresión.

Application of Machine Learning Algorithms on a Digital Depressive Corpus

Abstract. One of the main objectives that stimulates technological progress; regardless of the field of development, is to make people's lives simpler and more comfortable. These advances in the area of artificial intelligence have driven machine learning algorithms to perform the classification of certain objects thanks to the recognition of patterns that they discover during the training phases. These algorithms in conjunction with natural language processing techniques are capable of detecting emotions in written speech. One of these emotions is depression, which is ranked as one of the most common and also one of the most lethal disorders that can affect the world population and this motivates us to direct our research to apply some machine learning models to the detection of possible

depression discourse on a digital linguistic corpus in Spanish built from messages posted on the social network Twitter and evaluated by professionals in the field of psychology.

Keywords: Artificial intelligence, pattern recognition, machine learning, natural language processing, depression.

1. Introducción

De acuerdo con el Instituto Nacional de Estadística y Geografía en México (INEGI)[1], entre el 2016 y el 2019 se emitieron más de 140 millones de publicaciones en las redes sociales, por otra parte; la estadística[2] estima que entre las aplicaciones más populares en personas con un rango de edad entre los 16 y los 64 años, se encuentran: Facebook, WhatsApp, Instagram, TikTok, Twitter y Telegram.

Toda esta información respalda que el uso de las redes sociales es tan común que se ha convertido en un aspecto muy importante en la vida de muchos adolescentes y adultos jóvenes, los cuales utilizan este tipo de comunicación para compartir con otros usuarios aspectos relevantes y no relevantes de su vida diaria.

Principalmente, las redes sociales están enfocadas a compartir videos, imágenes o texto con cualquier usuario que tenga una cuenta activa dentro de la misma red social, (Aunque es posible que, en algunas redes sociales, las publicaciones solo estén disponibles para un grupo de personas en específico), a cualquier hora y en cualquier parte del mundo.

La mayoría de estas redes no poseen un filtro eficaz para validar si la información con la cual se aperturan las cuentas es verídica o falsa, por tal motivo; la identidad de muchos de los usuarios se puede conservar en un completo anonimato. Este tipo de acciones puede repercutir de manera negativa, ya que detrás del perfil creado se pueden esparcir noticias falsas, se pueden ocultar acosadores, etc.

En el anonimato no todo es negativo ya que cuando el usuario no se siente observado y enjuiciado todo el tiempo, se puede sentir con la libertad y confianza de expresar sus propios sentimientos reales y ello da la pauta para poder realizar el análisis de emociones en las publicaciones dentro de las redes sociales.

En los últimos años, el análisis de los sentimientos en las redes sociales ha ido en incremento y no sólo con fines de mercadotecnia sino con fines de índole social, ya que se analizan los sentimientos negativos, como la depresión, y la finalidad de ello es intentar reducir el índice de consecuencias letales que se pueden presentar en personas con este tipo de trastorno.

Esta investigación está enfocada a aplicar técnicas de procesamiento de lenguaje natural y algunos algoritmos de aprendizaje automático sobre un corpus digital en español desarrollado a partir de publicaciones en *Twitter* con la finalidad de analizar los mensajes y detectar de manera automática si el mensaje puede ser clasificado como un mensaje depresivo.

[1] www.inegi.org.mx/inegi/sociales.html

[2] es.statista.com/estadisticas/1035031/mexico-porcentaje-de-usuarios-por-red-social/

Vector de vocabulario							
hartar	morir	muerte	odiar	...	olvidar	querer	vida
Vector de presencia y ausencia							
1	0	1	1	...	0	1	1

Fig. 1. Ejemplo del vector de vocabulario y el vector de presencia y ausencia.

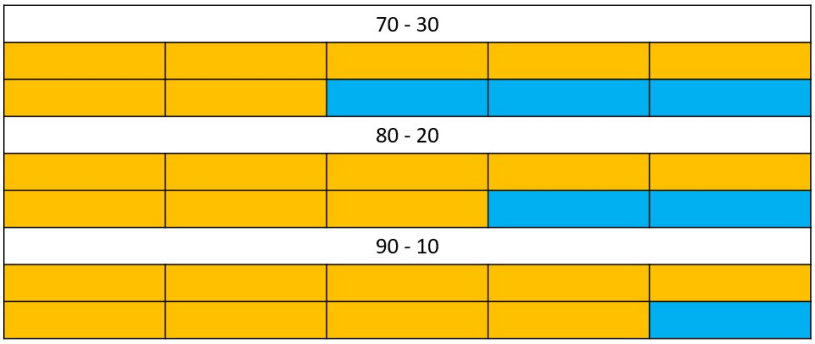


Fig. 2. Ejemplo de las métricas de validación.

La estructura de este artículo será la siguiente, se presentarán algunos trabajos relacionados, la aplicación de la metodología y finalmente se mostrarán los resultados obtenidos y las conclusiones finales de la investigación.

2. Trabajos relacionados

Los avances tecnológicos a nivel mundial forzaron la evolución de los medios de comunicación y con ello se modificó la manera tradicional de transmitir información. El alto uso de las redes sociales permite que toda la información publicada en ellas pueda ser analizada tanto con fines lucrativos como con fines de índole social.

En el análisis de texto extraído de Twitter encontramos a [1] en donde aplican técnicas de procesamiento de lenguaje natural para realizar un estudio social entre las publicaciones de los usuarios. Aplicaron una metodología que es capaz de rastrear[3] publicaciones semejantes, de acuerdo a ciertos dominios de información y los códigos fuentes fueron desarrollados en el lenguaje de programación python.

Entre las investigaciones de índole social se encuentra [2] en donde aplican minería de datos en el campo de la sicología para detectar depresión en usuarios de redes sociales de China. Su trabajo reporta una precisión aproximada del 80 % y tuvieron apoyo de la comunidad médica (sicólogos especializados en el tema).

El análisis de las emociones en las publicaciones en usuarios de redes sociales en China ha ido incrementando año con año debido al alza en las estadísticas de la tasa de suicidio, por ello en [3] aplican técnicas de análisis de sentimientos para detectar casos potenciales de depresión entre usuarios menores de edad.

[3] El término en inglés se conoce como «crawling»

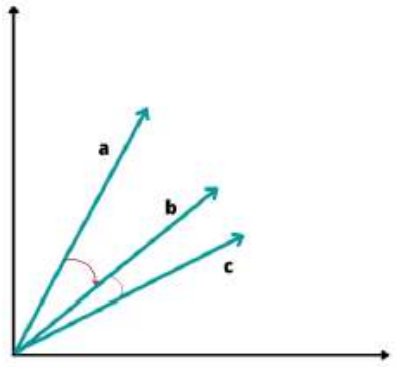


Fig. 3. Representación de 3 vectores.

Crearon una base de conocimiento de recursos encontrados en internet enfocados a la misma tarea y con el apoyo de sicólogos generaron una lista de vocabulario del sentimiento depresivo. Una vez que realizan el procesamiento de la información y obtienen la clasificación de los mensajes, se pondera si es necesario enviar una notificación de alarma a trabajadores sociales o a los padres de los usuarios.

Por último, en [4] indican que existe una correlación entre el alto uso de las redes sociales y el incremento de la depresión en las personas. En su investigación aplicaron los clasificadores: Máquinas de vectores de soporte (Support Vector Machines SVM), Bayes ingenuo (Naive Bayes) y árboles de decisión (Decision Trees). Compararon el historial de las publicaciones y generaron una métrica para calificar si era posible o no, que el usuario sufriera de depresión de acuerdo a los cambios de comportamiento en sus publicaciones en el último periodo de tiempo.

3. Aplicación de la metodología

En esta sección se explicará de manera detallada el proceso que se siguió para la aplicación de los algoritmos de aprendizaje automático sobre el corpus lingüístico digital en español enfocado a la depresión.

3.1. Preprocesamiento del corpus

El corpus lingüístico digital en español enfocado a la depresión cuenta con 1,623 mensajes y se encuentra etiquetado con sólo dos opciones: 1 (clase uno) si el mensaje es depresivo y 0 (clase cero) si el mensaje es no depresivo. Este corpus fue evaluado por personal profesional de la salud en el campo de la sicología con experiencia en la atención de pacientes en el área de la depresión. El procesamiento que se realizó sobre cada uno de los mensajes que componen el corpus se desarrolló en Python[4] y fue el siguiente:

[4] Se utilizó la version 3.9.12 de *Python*.

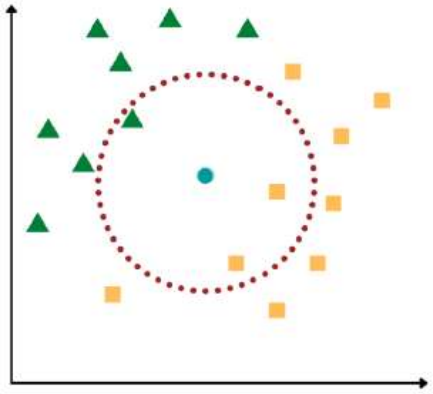


Fig. 4. Representación gráfica de k-vecinos más cercanos.

- Tokenización (es el proceso mediante el cual se obtienen las unidades mínimas de las oraciones, es decir; se separan las oraciones en palabras).
- Lematización (es la acción de encontrar la palabra raíz sin flexionar, por ejemplo: para el verbo «corriendo» el lema es «correr»).
- Etiquetado de las palabras (part of speech tagging) (un ejemplo de etiquetado: si aparece la palabra «correr» la etiqueta asociada es «verbo»).
- Eliminación de stop-words (es el conjunto de palabras que incluyen a los artículos definidos, indefinidos, preposiciones, etc).

Para la tokenización y la eliminación de stop-words se utilizó la biblioteca de NLTK (por sus siglas en inglés, Natural Language ToolKit)[5] y para el proceso de lematización y el etiquetado de las palabras se usó la biblioteca Stanza[6].

3.2. Vectorización

Debido a que no es posible realizar operaciones matemáticas sobre las palabras considerándolas sólo cadenas de caracteres, se aplicó la vectorización de los mensajes utilizando la técnica one hot encoding, la cual es capaz de transformar un mensaje en un vector numérico de presencia y ausencia.

El procedimiento inicial en esta técnica es formar un vector de vocabulario, este se consigue guardando en un arreglo cada una de las palabras del corpus sin repetir, es decir; contendrá sólo palabras únicas (por ello es importante el proceso de realizar la lematización de las entradas, ya que en diferentes mensajes se podrían encontrar las palabras «odié» y «odiado» y sólo se agregarían representaciones flexionadas de la misma raíz «odiar»). Generalmente, este vector ordena de manera alfabética todas las entradas.

Una vez que se cuenta con el vector de vocabulario, se procesa cada mensaje de manera individual y se genera un vector con sólo unos y ceros, en donde el uno

[5] Se utilizó la versión 3.7 de NLTK.

[6] Se utilizó la versión 1.4.2 de Stanza.

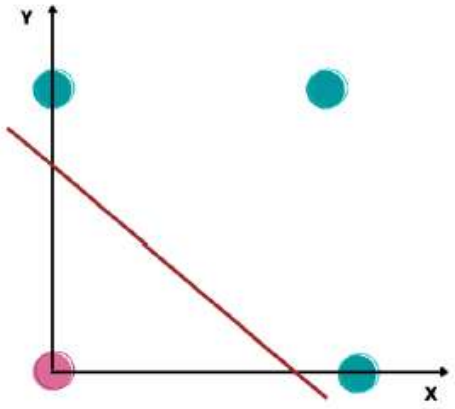


Fig. 5. Representación de clasificación con una red neuronal.

representa presencia y el cero ausencia. Este nuevo vector tiene exactamente la misma longitud que el vector de vocabulario. En la Figura 1 se visualiza un ejemplo de vectorización en ausencia y presencia.

3.3. División del corpus

Ya que se realizó la transformación de los mensajes a vectores numéricos, se realizó la segmentación del corpus con la finalidad de disponer de un rango de información para que los algoritmos escogidos puedan aprender y otra sección para examinar el funcionamiento de los mismos.

Se escogieron los métodos de validación 70 – 30, 80 – 20 y 90 -10, en donde la primera cifra corresponde al conjunto de entrenamiento y la segunda cifra al conjunto de prueba. De la figura anterior, los recuadros color naranja en cada métrica de validación representan al conjunto de entrenamiento y los recuadros azules son el conjunto sobre el cual se realizarán las pruebas.

3.4. Clasificación

Para realizar la etapa de la clasificación se escogieron 3 algoritmos de clasificación supervisada[7]: similitud coseno, k-vecinos más cercanos y perceptrón multicapa y a continuación se explicará de manera general el funcionamiento de cada uno de ellos.

3.4.1. Similitud coseno

Este algoritmo de clasificación compara dos vectores con las mismas dimensiones y obtiene un resultado que se encuentra entre cero y uno; cuando el cálculo obtenido tiende hacia cero[8] implica que los vectores casi no se parecen entre sí y por el contrario,

[7] En el paradigma supervisado se cuenta con las entradas etiquetadas, es decir; desde un inicio se conoce el resultado esperado y con ello se puede verificar de manera automática si los resultados obtenidos son correctos o incorrectos.

[8] Cuando los vectores tienen una similitud de cero se dice que los vectores son ortogonales entre sí.

Tabla 1. Precisión similitud coseno.

Algoritmo	Métrica	Precisión
Similitud coseno	90 – 10	71.62 %
	80 – 20	73.43 %
	70 – 30	75.42 %

cuando el resultado tiende hacia uno, se dice que los vectores se parecen demasiado entre sí.

Supongamos que la Figura 3 representa la gráfica en dos dimensiones de 3 vectores y se desea clasificar al vector «b» aplicando similitud coseno, se obtendrían dos resultados, el resultado entre «b y c» sería mayor que el resultado entre «b y a» por lo cual, al vector «b» se le asignaría la clase del vector «c».

Al aplicar este algoritmo sobre el corpus vectorizado, se debe calcular la similitud coseno del vector que se desea clasificar con todos los vectores del conjunto de entrenamiento, después buscar cual fue el resultado mayor, identificar su clase y asignarla al vector objetivo.

3.4.2. k-vecinos más cercanos

Este algoritmo de clasificación busca encontrar un número finito de elementos cercanos al elemento a clasificar y después realizar un conteo de clases entre dichos elementos, la clase con el número mayor de votos en este conteo es la clase que se asigna al vector objetivo.

Cabe mencionar, que cuando se trata de clasificación bi-clase (sólo 2 posibles clases a asignar), se escoge un número impar entre los elementos más cercanos a buscar con la finalidad de evitar empates entre los conteos.

De la Figura 4 supongamos que deseamos clasificar al círculo azul y que tenemos solo dos posibles clases «triángulo verde» y «cuadro amarillo», para evitar empates en el conteo elegimos k = 3, esto implica que el algoritmo deberá identificar los 3 elementos más cercanos[9] al elemento a clasificar y realizar el conteo.

En dicho conteo habría «triángulo verde = 1» y «cuadro amarillo = 2» por lo que la asignación de la clase al círculo azul sería cuadro amarillo. Los parámetros escogidos para este algoritmo con el corpus de depresión fue calcular la distancia euclidiana entre los vectores y el número de vecinos se varió con $k = 3$, $k = 5$, $k = 7$, $k = 9$ y $k = 11$.

3.4.3. Perceptrón multicapa

Este algoritmo pertenece a un conjunto de algoritmos conocidos como redes neuronales artificiales en donde se busca replicar (reproducir) el funcionamiento de las redes neuronales biológicas. De manera general, este tipo de redes están compuestas de 3 capas: la capa de entrada, la capa oculta y la capa de salida y cada capa puede contener «n» número de neuronas; cada neurona puede o no, estar enlazada con las demás capas por medio de aristas (pesos) que generalmente se inicializan con valores aleatorios.

[9] Generalmente se calcula la distancia euclidiana entre los elementos.

Tabla 2. Precisión k-vecinos más cercanos.

Algoritmo	Métrica	*k*	Precisión
k-vecinos más cercanos	90 – 10	3	76.68 %
	70 – 30	5	79.70 %
	90 – 10	7	81.75 %
	90 – 10	9	82.09 %
	70 – 30	11	82.29 %

Desde que se presenta información a clasificar en la capa de entrada hasta que se obtiene un valor en la capa de salida se le conoce como época y las neuronas de la capa oculta actúan a través de funciones de activación.

El resultado obtenido en la capa de salida se compara con el resultado esperado y de ser diferente se realiza un cambio en todos los pesos de la red neuronal. Este último proceso se suspende de acuerdo a dos condiciones: que se cumpla el número de épocas máximo establecido o que se alcance la precisión correcta esperada.

Consideremos que se desea clasificar de la Figura 5 al círculo color rosa a través de aplicar una red neuronal artificial, de manera gráfica la capa oculta de la red deberá dibujar una línea (llamado hiperplano en más de dos dimensiones) que sea capaz de dividir los elementos de las clases.

Cuando se compara la salida con el resultado esperado y se verifica que no se clasificó de manera correcta, los pesos de la red se modifican y con ello se modifica la ubicación de la línea.

Este proceso finaliza cuando la línea ha podido dividir correctamente ambas clases, es decir; los elementos de una clase se encuentran por encima de la línea y los elementos de la otra clase se encuentra por debajo[10]. Dentro de los parámetros que se eligieron para examinar el corpus fueron los siguientes:

- Número de épocas = 300.
- Taza de aprendizaje = 0.001.
- Funciones de activación:
  - Identity,
  - Logistic,
  - Relu,
  - Tanh.
- Optimizadores:
  - Adam,
  - Lbfgs,
  - Sdg.

[10] Lo cual sólo sucede cuando las clases son linealmente separables.

Tabla 3. Precisión perceptrón multicapa.

Algoritmo	Métrica	Función activación	Optimizador	Precisión
Perceptrón multicapa	90 – 10	Identity	Sgd	83.78 %
	80 – 20	Logistic	Sgd	84.12 %
	90 – 10	Relu	Sgd	84.22 %
	70 – 30	Tanh	Sgd	83.78 %

Se realizó una combinación completa entre todas las funciones de activación y los optimizadores conservando el número de épocas y la taza de aprendizaje para buscar los mejores resultados.

3.5. Resultados

En esta sección se muestran los mejores resultados obtenidos, de acuerdo con la métrica de validación aplicada en cada uno de los algoritmos de clasificación. En la Tabla 1 se muestran los resultados obtenidos con el algoritmo de similitud coseno, el mejor resultado se obtuvo con la métrica 70 – 30 con una precisión del 75.42 % y el promedio entre los resultados es de 73.15 %.

En la Tabla 2 se muestran los resultados aplicando el algoritmo de k-vecinos más cercanos considerando valores de 3, 5, 7, 9 y 11 para el valor de «k»; el mejor resultado se obtuvo con la métrica 70 – 30, con k = 11 con una precisión del 82.29 % y el promedio entre los resultados es de 80.5 %.

En la Tabla 3 se visualizan los mejores resultados obtenidos con el perceptrón multicapa, el mejor resultado se obtuvo con la combinación entre la métrica de validación 90 – 10, la función de activación «Relu» y el optimizador «Sgd» con una precisión del 84.22 % y el promedio entre los resultados mostrados en la Tabla 3 es de 83.97 %.

4. Conclusiones y trabajo a futuro

Se realizaron con éxito las pruebas sobre el corpus lingüístico digital en español con 3 algoritmos de aprendizaje automático con la finalidad de detectar el posible discurso depresivo. El promedio obtenido entre los mejores resultados fue de 73.15 % para similitud coseno, 80.5 % para k-vecinos más cercanos y del 83.97 % para el perceptrón multicapa.

El perceptrón multicapa fue el que dio los mejores resultados en las clasificaciones, pero se debe destacar que también fue el algoritmo que más tiempo y recursos computacionales requirió. Para el trabajo a futuro se plantea continuar con las descargas de los mensajes en Twitter para poder hacer más grande el corpus lingüístico digital y que los mensajes no sólo sean en español sino también incluir mensajes en el idioma inglés.

Respecto a las métricas de validación, buscaremos aplicar otras métricas tal como k-fold cross validation así como también nos gustaría poder aplicar otros algoritmos de aprendizaje automático y algoritmos de aprendizaje profundo. La depresión es un

perseguidor silencioso, por lo que consideramos de suma importancia enlazar este tipo de investigaciones con estudiantes de psicología, para ahondar en el estudio de trastornos mentales con apoyo de la tecnología y con ello estar atentos para evitar cualquier tipo de desgracia.

Referencias

1. Mendivelso, J. D., Baron, M. J. : Análisis social aplicando técnicas de lenguaje natural a información extraída de twitter. Scientia et Technica, vol. 24, no. 3, pp. 496–503 (2019) doi: 10.22517/23447214.21731
2. Wang, X., Zhang, C., Ji, Y., Sun, L., Wu, L., Bao, Z.: A depression detection model based on sentiment analysis in micro-blog social network, trends and applications in knowledge discovery and data mining. Lecture Notes in Computer Science, vol. 7867, pp. 201–213 (2013) doi: 10.1007/978-3-642-40319-4_18
3. Babu, N. V., Kanaga, E. G.: Sentiment analysis in social media data for depression detection using artificial intelligence: a review. SN Computer Science, vol. 3, no. 1 (2021) doi: 10.1007/s42979-021-00958-1
4. Alsagri, H. S., Ykhlef, M.: Machine learning-based approach for depression detection in twitter using content and activity features. IEICE Transactions on Information and Systems, vol. E103.D, no. 8, pp. 1825–1832 (2020) doi: 10.1587/transinf.2020edp7023
5. Cremades, S. Z., Gómez, J. M., Colorado, B. N.: Diseño, compilación y anotación de un corpus para la detección de mensajes suicidas en redes sociales. Procesamiento del Lenguaje Natural, no. 59, pp. 65–72 (2017) http://hdl.handle.net/10045/69092
6. Ameer, I., Arif, M., Sidorov, G., Gómez-Adorno, H., Gelbukh, A.: Mental illness classification on social media texts using deep learning and transfer learning (2022) doi: 10.48550/ARXIV.2207.01012
7. Bird, S., Klein, E., Loper, E.: Natural language processing with python: Analyzing text with the natural language toolkit. O'Reilly Media, 1st Edition (2009)
8. Sidorov, G.: Construcción no lineal de n-gramas en la lingüística computacional: N-gramas sintácticos, filtrados y generalizados. Sociedad mexicana de inteligencia artificial (2013)
9. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.: Stanza: A python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 101–108 (2020)
10. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases (2013) doi: 10.48550/ARXIV.1309.0238

ISSN 1870-4069

Propuesta de arquitectura de un sistema inteligente adaptativo para apoyo al aprendizaje de expresiones algebraicas

Brandon Azael Muciño-Santiesteban[1], María Antonieta Abud Figueroa[1], Ulises Juárez-Martínez[1], Lisbeth Rodríguez Mazahua[1], Mario Andrés Paredes-Valverde[2]

[1] Tecnológico Nacional de México,
Instituto Tecnológico de Orizaba,
División de Estudios de Posgrado e Investigación,
México

[2] Tecnológico Nacional de México,
Instituto Tecnológico Superior de Teziutlán,
México

{M22010147, maria.af, ulises.jm}@orizaba.tecnm.mx,
lrodriguezm@ito-depi.edu.mx, mario.pv@teziutlan.tecnm.mx

Resumen. En México el aprendizaje de la asignatura de matemáticas presenta varios problemas como el bajo interés de los alumnos en el tema aunado a la complejidad que representa. Una forma de abordar este problema es aprovechar los beneficios que el avance de la tecnología trae consigo y aplicar nuevos métodos de enseñanza con el objetivo de mejorar y agilizar los procesos de aprendizaje, siendo los sistemas tutores inteligentes adaptativos (STIA) una opción interesante. Un STIA ayuda a despertar el interés del estudiante optimizando su aprendizaje y permite al profesor identificar acciones que ayuden a mejorar el aprovechamiento de sus estudiantes. Por esta razón, en este artículo se propone una arquitectura base para el desarrollo de un sistema tutor inteligente adaptativo que apoye el aprendizaje de expresiones algebraicas para alumnos de primer año de secundaria.

Palabras Clave: Educación asistida por computadora, sistemas tutoriales inteligentes, sistema tutor adaptativo.

Proposed Architecture of an Adaptive Intelligent System to Support the Algebraic Expressions Learning

Abstract. In Mexico, the mathematics learning presents several problems such as the low interest of students in the subject combined with its complexity. One way to address this problem is to take advantage of the benefits that the advancement of technology brings and apply new teaching methods in order to improve and streamline the learning process, being the adaptive intelligent tutoring systems (AITS) an interesting option. An AITS helps to awaken the student's interest by optimizing their learning and allows the teacher to identify

actions that help to improve the achievement of their students. This paper proposes a base architecture for the development of an adaptive intelligent tutor system that supports the learning of algebraic expressions for first year of middle school students.

Keywords: Computer aided education, intelligent tutoring systems, adaptive tutoring system.

1. Introducción

En la actualidad en México el aprendizaje de las matemáticas se ve afectado por varios motivos, como lo son el bajo rendimiento académico y las aptitudes negativas que presentan los alumnos durante el aprendizaje de éstas.

Esto trae consigo dificultades para la comprensión de problemas matemáticos, así como inadecuadas estrategias para su solución.

Por otra parte, debido a los avances en las tecnologías de la información, las generaciones actuales de estudiantes presentan nuevas necesidades en su educación, por lo que se requiere de entornos de aprendizaje centrados en el usuario que garanticen que las aplicaciones les facilite una experiencia de aprendizaje positiva, siendo los sistemas de tutoría inteligente una opción interesante ya que imitan el estilo de enseñanza de un instructor brindando apoyo al estudiante durante su aprendizaje y evaluando la adquisición de conocimientos a lo largo del proceso educativo.

Estos sistemas de tutoría son importantes para apoyar al estudiante en su proceso de aprendizaje promoviendo su interés y motivación, ya que se adecúan a cada uno presentando contenidos de acuerdo a su grado de aprovechamiento, incrementando la motivación y el rendimiento académico.

En este trabajo se presenta la arquitectura para un sistema tutor inteligente adaptativo que sirva como una herramienta para el aprendizaje autónomo o como complemento a la escuela de estudiantes de primer año de secundaria en el aprendizaje básico de matemáticas, especialmente en el tema de expresiones algebraicas.

El sistema propuesto contempla los modelos básicos de los sistemas tutores inteligentes: modelo de dominio, de enseñanza y del estudiante, incorporando un método heurístico para la evaluación del aprendizaje.

El documento se organiza de la siguiente forma: la sección 2, estado del arte, presenta la revisión de trabajos relacionados con sistemas de tutoría inteligentes/educativos para fines educativos; en la sección 3 se presenta el diseño arquitectónico del software y sus componentes principales; en la sección 4 se establece la propuesta de solución; por último, se presentan las conclusiones y trabajo futuro.

2. Estado del arte

Diversos autores han realizado trabajos sobre el desarrollo de sistemas tutores inteligentes, en esta sección se presenta la revisión de los principales trabajos relacionados con sistemas de tutoría inteligente/adaptativos para fines educativos, con el objetivo de mostrar el impacto que estos lograron obtener con su implementación.

En [1], se presentó un sistema inteligente de tutoría para comprender la teoría de computación. Este sistema se basa en otros ITS (*Intelligent Tutoring System*) e implementa una arquitecta que se conforma de lo siguiente: "Modelo de estudiante", "Modelo de interfaz", "Modelo de dominio" y "Modelo pedagógico" el cual actúa como instructor virtual encargándose de evaluar el desempeño del alumno.

Con base en los resultados obtenidos por lección se estableció un porcentaje de aprendizaje (menor a 70%) que es utilizado para determinar si el alumno tiene que repasar todo lo visto durante la lección.

Rocha G. et al. [2] plantearon las posibilidades de mejora a la problemática del aprendizaje de las matemáticas y las negativas que se presentan a éstas con la ayuda del "Sistema Tutor Adaptativo" (STA) MyMathLab el cual tiene la posibilidad de configurar los temas y ejercicios que serán vistos por los alumnos.

Este sistema permite que los profesores visualicen el rendimiento de los estudiantes los cuales tienen la posibilidad de avanzar a su propio ritmo. Con base en las pruebas realizadas durante la investigación y el uso de la escala AMMEC (Aptitudes hacia las Matemáticas y hacia las Matemáticas Enseñadas por Computadora) se logró saber el cambio que tuvieron los alumnos con respecto a su aceptación hacia dicha área.

Shamara et al. [3] abordaron el problema de la dificultad de detectar la colaboración que presentan los alumnos al momento de trabajar en un entorno tecnológico, enfatizando que esto dificulta la evaluación de la colaboración de los estudiantes.

Para esto se propuso el uso de la causalidad de Granger, para analizar las relaciones causales, junto con el uso de un sistema tutor inteligente de fracciones, para el estudio de las relaciones entre los procesos cognitivos individuales y colaborativos que presentaban los estudiantes. Los resultados obtenidos indicaron que la correlación entre las distintas medidas como lo son mirada, individuales y colaborativas, pueden utilizarse en un sistema adaptativo.

En [4], se examinaron los elementos que conformaban a un Sistema Inteligente de Tutoría con el fin de que determinados artículos pudieran ser identificados en distintas bases de datos con la frase clave “Sistema Inteligente de Tutoría Adaptativa”. Para esto se utilizaron métodos de meta síntesis.

Los resultados obtenidos de esa meta síntesis indicaron que los STI fueron diseñados principalmente para campos como la tecnología, matemáticas, entre otras ramas de la educación, así como que estos comparten cuatro módulos básicos que son: conocimiento, estudiante, enseñanza e interfaz de usuario.

Ramírez-Noriega A. et al. [5] mencionaron que la principal tarea que realiza un Sistema Inteligente de Tutoría es proporcionar el conocimiento necesario a los estudiantes. Para cumplir con este objetivo, el sistema tiene que presentar una correcta evaluación de los conocimientos aprendidos por los estudiantes.

En este sentido, los autores propusieron un módulo de evaluación basado en redes bayesianas, para que, con distintas pruebas, se determinara el conocimiento del estudiante. Los resultados obtenidos mostraron que la implementación de redes bayesianas aumentó la precisión de la evaluación de los conocimientos obtenidos.

En [6], se abordó que los Sistemas Inteligentes de Tutoría son una tecnología que va en aumento y junto con eso su efectividad y eficiencia también ha mejorado. Por esto presentaron un STI para que los estudiantes aprendieran los conceptos básicos de matemáticas, principalmente para resaltar la importancia que tiene sumar y restar.

Los resultados obtenidos demostraron que los estudiantes presentaron una mayor facilidad al momento de estudiar en el sistema, ya que el material y los ejercicios, así como los niveles de dificultad eran eficientes.

En [7], se estudiaron los resultados que se obtenían de la interacción con un sistema híbrido de tutoría inteligente el cual era la combinación de un sistema tutor convencional y el sistema de evaluación y aprendizaje en espacios de conocimiento ALEKS por sus siglas en inglés.

El sistema resultante contó con una arquitectura basada en servicios, además de la utilización de diálogos de tutoría para que los problemas de álgebra mostrados tuvieran una auto explicación para los alumnos. Como resultado se concluyó que el uso de distintos sistemas de tutoría adaptativos, forman en conjunto una mejora potencial para el aprendizaje.

En [8], tomaron la posición de que el aprendizaje adaptativo inteligente (AAI) es como un aprendizaje digital el cual introduce a los estudiantes a un aprendizaje modular donde cada decisión tomada es registrada y utilizada para dar una mejor experiencia de aprendizaje para el alumno.

Para esto se tomaron en cuenta características como los intereses del alumno y el tiempo que utiliza para resolver un problema. Se llegó a la conclusión de que el Aprendizaje Adaptativo Inteligente es una herramienta de diagnóstico, así como un recurso de aprendizaje muy valioso para los maestros, estudiantes y padres.

Fouki M. et al. [9] abordaron que las plataformas de e-learning cada día adquieren más popularidad en las instituciones educativas, como las universidades abiertas y a distancia y los institutos de investigación, sin embargo, estas plataformas presentan problemas que no se han resuelto, por ejemplo, el hecho de que los profesores a distancia tienen dificultades para identificar de una manera correcta a sus alumnos, así como identificar los comportamientos de éstos.

Por lo cual desarrollaron una estrategia de aprendizaje inteligente y adaptada basada en el sistema de recomendación para ayudar a los profesores a tener un trabajo más eficiente.

En [10], se abordó que los sistemas de aprendizaje adaptativo se distinguen de los tradicionales al ofrecer una experiencia de aprendizaje personalizada a los estudiantes de acuerdo con sus diferentes estados de conocimiento.

Para esto evaluaron la efectividad del sistema de aprendizaje adaptativo "Yixue Squirrel AI" (o Yixue) en el aprendizaje de inglés y matemáticas en la escuela secundaria.

Los resultados sugieren que los estudiantes lograron un mejor rendimiento utilizando el sistema de aprendizaje adaptativo Yixue en relación con otras plataformas de aprendizaje adaptativo, así como las clases impartidas por los profesores.

La Tabla 1 presenta la comparación de la información de cada uno de los artículos descritos anteriormente con el objetivo de observar las similitudes y diferencias que estos presentan.

Después de analizar los artículos citados, se identificaron las características que presentan los sistemas tutores inteligentes a través de tres modelos principales: el modelo de dominio, el de tutor y el del alumno.

Se encontró también que en todos los casos estos sistemas facilitan el aprendizaje de los alumnos y ayudan al docente a identificar áreas de mejora.

Tabla 1. Análisis comparativo de los artículos relacionados.

Art.	Problema	Contribución	Resultados
Al-Nakhal M [1]	La necesidad de un sistema de apoyo a la enseñanza de la teoría de la computación.	Desarrollo de un Sistema Inteligente de Tutoría para la enseñanza de la teoría de la computación.	Un sistema con un diseño simple, el cual dio como resultado que los estudiantes comprendieran de manera fácil las lecciones.
Rocha G. et al. [2]	El bajo rendimiento y aptitudes negativas en el aprendizaje de las matemáticas en estudiantes de universidad en México.	Demostró la efectividad que se logra obtener en el aprendizaje gracias al uso de un STIA.	Mayor desarrollo de competencias matemáticas en los alumnos. Mejora en la aptitud hacia el aprendizaje por computadora de las matemáticas.
Shamara et al. [3]	La necesidad de encontrar la relación causal entre los procesos cognitivos individuales y colaborativos.	Implementación de seguimiento ocular para la mejora en la comprensión del aprendizaje colaborativo apoyado por computadora.	Se descubrió que los patrones de mirada colaborativa impulsan el enfoque individual.
Erümit A. et al. [4]	Necedad de definir los elementos de adaptación y los elementos del Sistema de Tutoría Inteligente (STI) utilizados en los Sistemas de Tutoría Inteligente Adaptativa (STIA).	Metasíntesis de los artículos identificados con la frase clave "sistema de tutoría inteligente adaptativa".	Se llegó a la conclusión de que los sistemas evaluados fueron diseñados para el uso efectivo de la tecnología en la provisión de entornos de aprendizaje atractivos y de calidad.
Ramírez-Noriega A. et al. [5]	Necesidad de medir el nivel de conocimiento de un estudiante al usar un STI.	Módulo de evaluación basado en red bayesiana.	El uso de red bayesiana proporciona mayor precisión de diagnóstico.
Abueloun N et al. [6]	Necesidad de herramientas que apoyen el aprendizaje de conceptos básicos de matemáticas.	Sistema Inteligente de Tutoría para ayudar a estudiantes en la comprensión de temas básicos de las matemáticas.	Mayor facilidad de estudio para alumnos. Mayor eficiencia en material y ejercicios para el aprendizaje.
Nye et al. [7]	Conocer el nivel de aprendizaje y las percepciones de los usuarios a partir del uso de un sistema hibrido de tutoría inteligente.	Se encontró que la asignación de condiciones experimentales y de control no muestran diferencias significativas en las ganancias de aprendizaje.	La integración de múltiples sistemas de tutoría adaptativa con estructuras complementarias muestra cierto potencial para mejorar el aprendizaje.
BreamBox Learning I [8]	La eficiencia del aprendizaje adaptativo inteligente, como tecnología para personalizar el aprendizaje de cada alumno.	Proporcionó un contexto en donde demostró el rol que desempeña el AAI en el aumento del rendimiento académico de los estudiantes.	Demostró el valor de los sistemas AAI deben ser tomados como una herramienta de diagnóstico, un recurso de aprendizaje y una fuente de datos valiosos para el maestro, el estudiante y los padres.
El Fouki M. et al. [9]	La dificultad que presentan los profesores a distancia para identificar las cualidades y aptitudes de sus alumnos al no verlos en persona.	Los resultados obtenidos servirán como medida para realizar correcciones en el proceso de aprendizaje de los estudiantes.	Muestra que el análisis profundo de los componentes principales y el aprendizaje por refuerzo podría aumentar el rendimiento de predicción de un algoritmo de red neuronal profunda.
Wei-Cui et al. [10]	Falta de una herramienta para evaluar la efectividad de un sistema de aprendizaje adaptativo.Evaluar la efectividad del sistema de aprendizaje adaptativo "Yixue Squirrel AI" en el aprendizaje de inglés y matemáticas en secundaria.	Demostraron la efectividad del programa en las materias de matemáticas e inglés.	Mayores ganancias de aprendizaje en las materias de inglés y matemáticas con Yixue en comparación con una clase en aula convencional.

Los trabajos revisados establecen la necesidad de utilizar un método de evaluación del aprendizaje, mismo que la arquitectura propuesta en este trabajo busca implementar a través de un método empírico de evaluación de aprendizaje basado en el seguimiento del conocimiento bayesiano (*BKT, Bayesian Knowledge Tracing),* el cual no ha sido

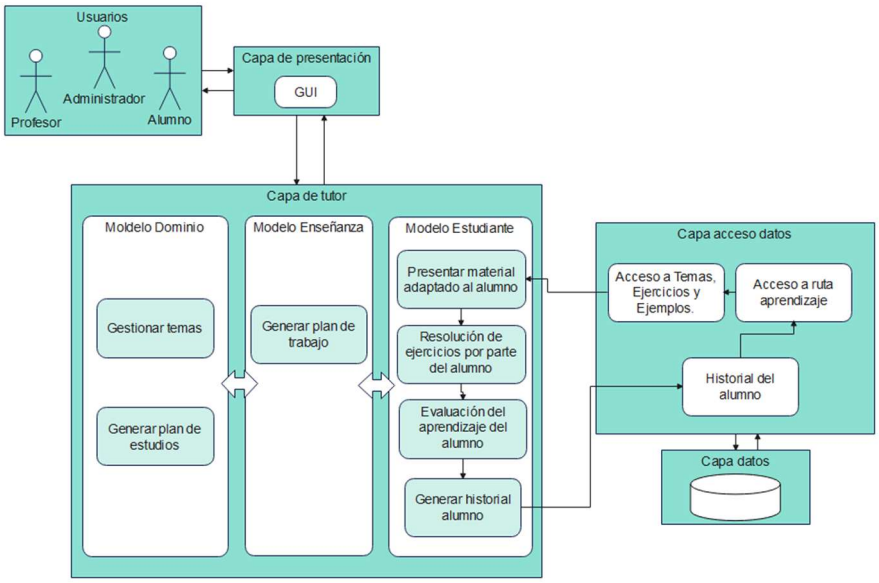


Fig. 1. Arquitectura del sistema inteligente adaptativo para el apoyo del aprendizaje de expresiones algebraicas.

reportado en los trabajos previos, lo que representa una de las principales contribuciones de este trabajo.

3. Arquitectura propuesta

En esta sección se presenta una propuesta de arquitectura para el desarrollo de un sistema inteligente adaptativo para el apoyo del aprendizaje de expresiones algebraicas. La arquitectura está diseñada en capas, las cuales contienen módulos que interactúan entre sí para el correcto funcionamiento del sistema.

La Figura 1 muestra un diagrama general de la arquitectura propuesta cuyos componentes son descritos a continuación.

Capa de presentación: la capa de presentación es la encargada de la interfaz gráfica de usuario donde se llevará a cabo la interacción con los diferentes tipos de usuario.

- El Administrador, que tiene la función de gestionar a profesores, alumnos y grupos.
- El Profesor, encargado de gestionar los temas, el plan de estudios y el plan de trabajo, así como la administración del historial de los alumnos.
- El Alumno, quien podrá visualizar los temas, los ejemplos y realizar los ejercicios para su posterior evaluación.

Capa de tutor: en esta capa se encuentran los modelos que conforman el sistema tutor adaptativo, los cuales controlan las distintas funciones de los usuarios. Estos modelos son:

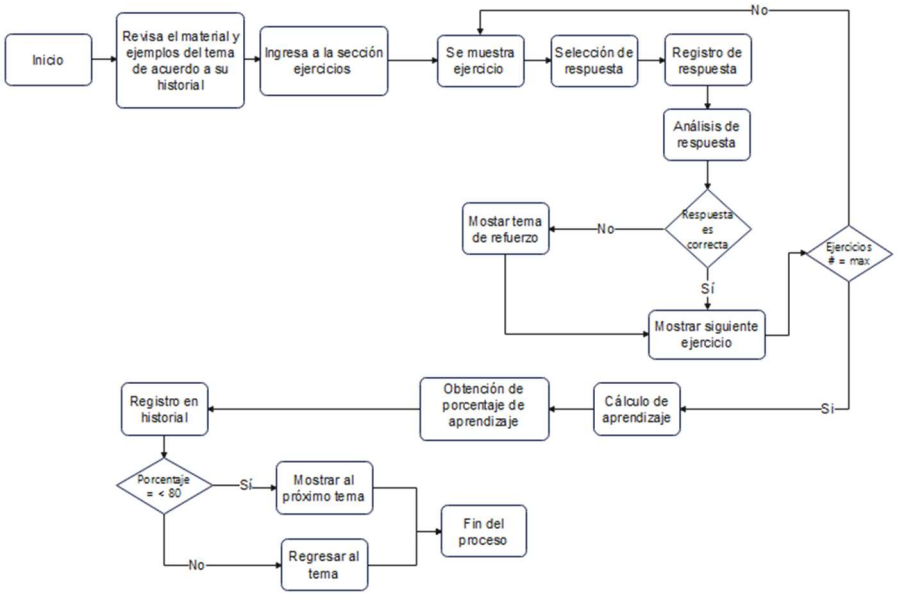


Fig. 2. Flujo de solución.

Modelo dominio: en este módulo se gestionará toda la información de los temas, en el cual el profesor ingresará el material didáctico que será presentado a los alumnos, y con este material generará el plan de trabajo programado para el aprendizaje de cada tema. Los temas considerados en aprendizaje de expresiones algebraicas de primer año de secundaria son: traducción de lenguaje común a lenguaje algebraico, identificación y representación de expresiones algebraicas, resolución de ecuaciones de la forma ax + b = c y resolución de ecuaciones de la forma ax + b = cx + d.

Modelo enseñanza: este modelo contempla los ejemplos y ejercicios que de una manera secuencial se presentarán al estudiante para apoyarle con aprendizaje de los temas. El profesor es el encargado de la selección y definición del plan de trabajo, con el cual establecerá el cómo y cuándo se mostrará el material seleccionado a los alumnos. Se considerarán ejercicios con los cuales se pueda identificar los principales errores que se presentan en el manejo de expresiones algebraicas como son: términos erróneos para referirse a una variable o acción algebraica, identificación errónea de coeficientes y literales, aplicación incorrecta de las propiedades opuestas a suma, resta, multiplicación y división.

Modelo estudiante: en este modelo se lleva el control del avance del alumno a través de su historial, el cual servirá para controlar los temas que se han estudiado y aprobado, así como los que faltan por lograr. Basado en este modelo el sistema presenta al estudiante los temas, ejemplos y ejercicios adaptados a la necesidad de acuerdo al historial de ejercicios realizados, considerando el proceso de evaluación del aprendizaje descrito en el punto 4.

Capa acceso datos: Esta capa da acceso a la información de las distintas fuentes de datos (Ruta de aprendizaje, temas, ejercicios, ejemplos e historial del alumno).

Capa de datos: Representa los datos que dan servicio a las capas superiores.

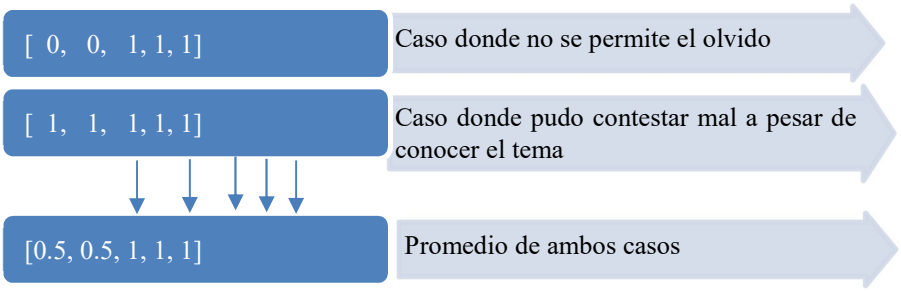


Fig. 3. Ejemplo de evaluación del aprendizaje.

4. Proceso de evaluación del aprendizaje

Una de las principales funciones del sistema propuesto es la adaptación del material que se presenta a los alumnos de acuerdo a los resultados obtenidos en su evaluación, por lo que en la Figura 2 se muestra el proceso propuesto para la evaluación del alumno y la adaptación de los materiales.

A continuación, se describe este proceso:

1) De acuerdo al nivel presentado en su historial del modelo del estudiante y al plan de trabajo del modelo de enseñanza, se mostrará al alumno el contenido del tema correspondiente y sus ejemplos.
2) Cuando el alumno considera que ya comprendió el tema, ingresará a la sección de ejercicios donde podrá realizar la actividad correspondiente a su avance de acuerdo a su historial.
3) El sistema seleccionará el ejercicio adecuado a su nivel de avance, descartando aquellos mostrados anteriormente.
4) De acuerdo a su respuesta el sistema verifica si ésta fue correcta o incorrecta.
5) Si la respuesta fue correcta pasa a seleccionar la siguiente pregunta (regresa al paso 3). Si fue incorrecta se mostrará material de refuerzo del tema y se presenta un nuevo ejercicio (regresa al paso 3).
6) Una vez alcanzado el número máximo de preguntas determinadas para el plan de trabajo del tema, se realiza el cálculo de aprendizaje utilizando el modelo de probabilidades empíricas basado en el seguimiento del conocimiento bayesiano (*BKT, Bayesian Knowledge Tracing)* propuesto en [11] el cual, de acuerdo a la secuencia de respuestas correctas e incorrectas, determina el nivel de conocimiento de un tema considerando que una respuesta correcta nunca puede ser seguida de una incorrecta (no se permite el olvido). En este modelo se considera que para un determinado tema $t \in T$, hay cuatro parámetros que representan probabilidades:
   - La probabilidad de que un estudiante domine el tema antes de intentar el primer problema asociado con t;
   - La probabilidad de que un estudiante: que actualmente no domina el tema, lo domine después de la próxima oportunidad de práctica;

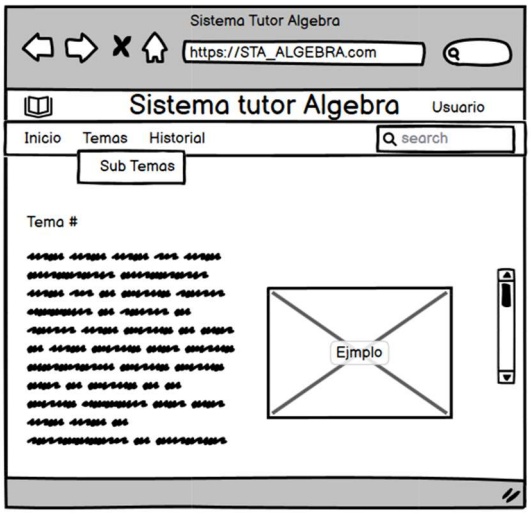


Fig. 4. Prototipo página temas/subtemas.

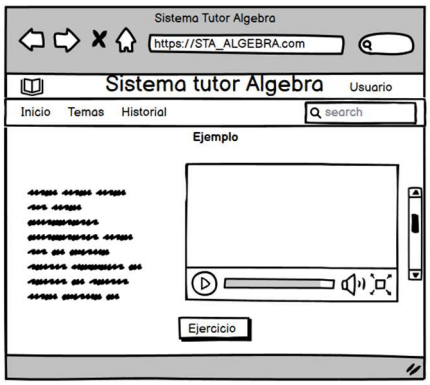


Fig. 5. Prototipo página ejercicios.

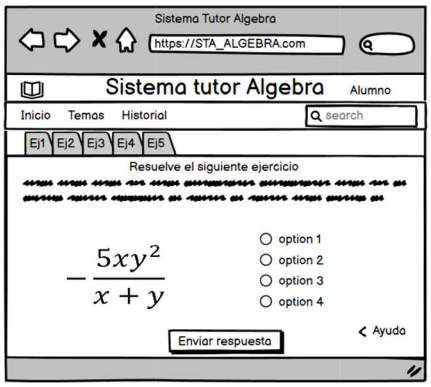


Fig. 6. Prototipo página ejercicios.

- La probabilidad de que un estudiante adivine la respuesta correcta a una pregunta a pesar de no conocer el tema (la adivina); y
- La probabilidad de que un estudiante responda incorrectamente a una pregunta a pesar de conocerlo (deslizamiento).

Por ejemplo, suponiendo que, de una secuencia de 5 preguntas, y tomando 1 como respuesta correcta y 0 como incorrecta, el estudiante tiene el siguiente resultado [1, 0, 1, 1, 1]. Considerando la heurística se tiene que los patrones que se ajustan son dos como se muestra en la Figura 3.

Tomando el promedio de ambos casos se tendrá un promedio de 80% de que el alumno tiene el conocimiento del tema.

La descripción completa de la heurística se encuentra en [11].

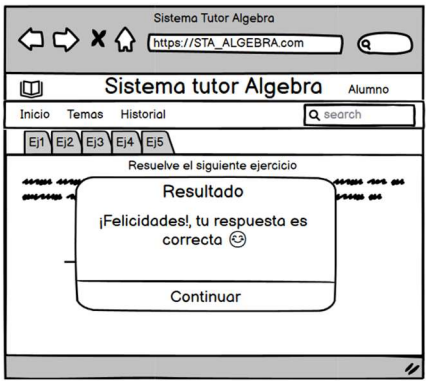


Fig. 7. Prototipo notificación resultado correcto.

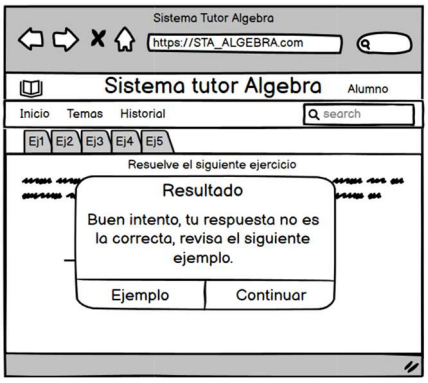


Fig. 8. Notificación resultado incorrecto.

7) Dependiendo del resultado, si el conocimiento resulta mayor o igual a 80 se considera que el alumno aprendió el tema y puede continuar con el siguiente, y en caso contrario se le redirige al inicio del tema actual para que revise nuevamente el material y ejemplos del mismo (regresa al paso 1).

A continuación, se presentan algunas de las interfaces gráficas de usuario más representativas del sistema. En primer lugar, la Figura 4 muestra la estructura de cómo se presentará la información a los alumnos, de la cual destaca el título del tema y una explicación de este.

En la Figura 5 se muestra la estructura que abarca la sección de ejemplos donde el estudiante visualizará un texto de explicación y una serie de pasos explicando cómo se resuelve.

En la Figura 6 se presenta la sección de ejercicios, la cual está conformada por el enunciado del ejercicio que realizará el estudiante, las opciones de respuesta del ejercicio (considerandos reactivos de opción múltiple, falso-verdadero y respuesta directa) y un enlace a la ayuda en caso de que la necesite.

En la Figura 7 se presenta la notificación correspondiente a respuestas correctas donde se mostrará un mensaje de felicitación y la opción de continuar. Por su parte la Figura 8 presenta una notificación para una respuesta incorrecta en la cual se presenta la opción de visualizar un ejemplo del ejercicio realizado.

Estos primeros prototipos se presentarán a profesores de nivel secundaria para que los evalúen y propongan mejoras y ya sean desarrollados en una versión ejecutable del sistema.

5. Conclusiones y trabajo a futuro

La estructura que caracteriza a los sistemas tutores inteligentes contempla los modelos de dominio, enseñanza y estudiante como elementos necesarios para el aprendizaje de los estudiantes haciendo el proceso de aprendizaje más atractivo y eficiente, consiguiendo con esto apoyar en la obtención de mejores resultados académicos.

En este trabajo se presenta la propuesta arquitectónica para un sistema tutor inteligente adaptativo que sirva como una herramienta para mejorar el aprendizaje de matemáticas en estudiantes de nivel secundaria, estableciendo una estrategia de selección de contenidos de acuerdo a los resultados previos de cada estudiante de manera que se adapte a su nivel de conocimiento y ritmo personal de aprendizaje, generando recomendaciones, brindando retroalimentación y permitiendo al profesor identificar situaciones de mejora y la toma de acciones que ayuden a incrementar el rendimiento de cada estudiante.

La arquitectura presentada se empleará en el desarrollo del sistema que se implemente con estudiantes de nivel secundaria y que sirva de apoyo en la obtención de mejores resultados de aprendizaje de las matemáticas.

Como trabajo futuro se contempla:

1. Evaluar los prototipos de pantallas con al menos un profesor de matemáticas de primero de secundaria y realizar los ajustes necesarios.
2. Diseñar y codificar el sistema.
3. Validar el sistema con una población muestra en un grupo primero de secundaria.

Agradecimientos. Se agradece al Tecnológico Nacional de México / Instituto Tecnológico de Orizaba por la oportunidad y apoyo otorgados para la realización de este trabajo. Este proyecto cuenta con el apoyo del Consejo Nacional de Ciencia y Tecnología (CONACyT).

Referencias

1. Al-Nakhal, M. A., Abu-Naser, S. S.: Adaptive intelligent tutoring system for learning computer theory. European Academic Research, vol. 4, no. 10 (2017)
2. Feregrino, G. R., López, J. A. J., Gómez, O. L. F., Méndez, G. R.: El rendimiento académico y las actitudes hacia las matemáticas con un sistema tutor adaptativo. PNA, Revista de Investigación en Didáctica de la Matemática, vol. 14, no. 4, pp. 271–294 (2020) doi: 10.30827/pna.v14i4.15202
3. Sharma, K., Olsen, J. K., Aleven, V., Rummel, N.: Measuring causality between collaborative and individual gaze metrics for collaborative problem-solving with intelligent tutoring systems. Journal of Computer Assisted Learning, vol. 37, no. 1, pp. 51–68 (2020) doi: 10.1111/jcal.12467
4. Erümit, A. K., Cetin, İ: Design framework of adaptive intelligent tutoring systems. Education and Information Technologies, vol. 25, no. 5, pp. 4477–4500 (2020) doi: 10.1007/s10639-020-10182-8
5. Ramírez-Noriega, A., Juárez-Ramírez, R., Martínez-Ramírez, Y.: Evaluation module based on bayesian networks to intelligent tutoring systems. International Journal of Information Management, vol. 37, no. 1, pp. 1488–1498 (2017) doi: 10.1016/j.ijinfomgt.2016.05.007
6. AbuEloun, N., Abu-Naser, S.: Mathematics intelligent tutoring system. International Journal of Advanced Scientific Research, vol. 2, pp. 11-16 (2017)
7. Nye, B. D., Pavlik, P. I., Windsor, A., Olney, A. M., Hajeer, M., Hu, X.: Skope-it (shareable knowledge objects as portable intelligent tutors): Overlaying natural language tutoring on an

adaptive learning system for mathematics. International Journal of STEM Education, vol. 5, no. 1 (2018) doi: 10.1186/s40594-018-0109-4
8. DreamBox learning: Intelligent adaptive learning: an essential element of 21st century teaching and learning (2014)
9. Fouki, M. E., Aknin, N., Kadiri, K. E. E.: Intelligent adapted e-learning system based on deep reinforcement learning. In: Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems (2017) doi: 10.1145/3167486.3167574
10. Cui, W., Xue, Z., Thai, K.: Performance comparison of an AI-based adaptivelearning system in china. In: Chinese Automation Congress, pp. 3170–3175 (2018) doi: 10.1109/cac.2018.8623327
11. Hawkins, W. J., Heffernan, N. T., Baker, R. S. J. D.: Learning bayesian knowledge tracing parameters with a knowledge heuristic and empirical probabilities. In: Intelligent Tutoring Systems, pp. 150–155 (2014) doi: 10.1007/978-3-319-07221-0_18

ISSN 1870-4069

Sistema para evaluar el grado de atención y aprovechamiento de estudiantes en correlación con la actividad ocular durante una actividad de comprensión lectora en computadora

Miguel Ángel Ramírez-Flores[1], María Antonieta Abud-Figueroa[1], Mario Andrés Paredes-Valverde[2], Ignacio López-Martínez[1], Ulises Juárez-Martínez[1]

[1] Tecnológico Nacional de México,
Instituto Tecnológico de Orizaba,
División de Investigación y Estudios de Posgrado,
México

[2] Tecnológico Nacional de México,
Instituto Tecnológico Superior de Teziutlán,
División de Estudios de Posgrado e Investigación,
México

{M21011181, maria.af, ulises.jm,
ignacio.lm}@orizaba.tecnm.mx,
mario.pv@teziutlan.tecnm.mx

Resumen. El presente trabajo describe el diseño y desarrollo de un sistema Web para dar soporte a la evaluación del grado de atención y aprovechamiento de estudiantes del primer año de secundaria durante una actividad de comprensión lectora en computadora. El software obtenido implementa técnicas de seguimiento ocular basadas en visión artificial, utilizando GazeCloudAPI y una cámara web convencional para registrar el seguimiento de la actividad ocular de los alumnos durante una lectura de comprensión. Para llevar a cabo la evaluación, se utilizó la prueba del Desarrollo de Movimiento Ocular (DEM), una lectura de comprensión ajustada a la cantidad de palabras de acuerdo al nivel de los estudiantes y una serie de preguntas y respuestas. Además, se llevó a cabo el análisis y categorización del mapa de calor generado durante el proceso de lectura utilizando las proporciones de color presentes en la imagen. La primera versión del sistema propuesto logró clasificar de manera similar a más del 65% de los evaluados con la clasificación obtenida en el aula. La principal ventaja de este sistema es que brinda una herramienta docente para obtener, analizar y visualizar la actividad ocular durante la lectura en computadora, lo que permite detectar problemas de atención y aprendizaje en los estudiantes y tomar medidas tempranas para mejorar su desempeño académico.

Palabras clave*:* Visión artificial, comprensión lectora, seguimiento ocular, desempeño académico.

System for Evaluation of the Degree of Attention and Achievement of Students in Connection with Ocular Activity During a Computer-Based Reading Comprehension Activity

Abstract. This paper describes the design and development of a web system to support the evaluation of the level of attention and performance of first-year middle school students during a computer-based reading comprehension activity. The obtained software implements eye-tracking techniques based on artificial vision, using GazeCloudAPI and a conventional webcam to record students' eye activity during a reading comprehension task. To carry out the evaluation, the Developmental Eye Movement (DEM) test was used, as well as a reading comprehension text adjusted to the students' level and a set of questions and answers. Additionally, the analysis and categorization of the heatmap generated during the reading process was carried out using the color proportions present in the image. The first version of the proposed system was able to classify more than 65% of the evaluated students similarly to the classification obtained in the classroom. The main advantage of this system is that it provides a teaching tool to obtain, analyze and visualize eye activity during computer-based reading, which allows for the detection of attention and learning problems in students and the early implementation of measures to improve their academic performance.

Keywords: Artificial vision, reading comprehension, eye tracking, academic performance.

1. Introducción

El bajo desempeño académico es un problema que afecta a muchos estudiantes en todo el mundo, independientemente de su nivel educativo. Si bien, este problema está relacionado con factores tales como falta de motivación, ansiedad, estrés y depresión, también está vinculado a la forma en que se enseña y se aprende.

Uno de los principales factores que se asocia con el bajo aprovechamiento académico es la falta de comprensión lectora, ya que limita la capacidad de aprendizaje de los estudiantes y contribuye a un bajo desempeño escolar. Muchos estudiantes tienen dificultades para entender lo que leen, lo que afecta significativamente su desempeño académico.

En México, muchos niños y jóvenes tienen dificultades para leer y comprender textos, lo que afecta su desempeño académico y los resultados generales del aprendizaje. Los resultados publicados por el Programa Internacional para la Evaluación de Estudiantes (PISA) en el 2018 indican que el 35% de los estudiantes mexicanos no alcanzó el nivel mínimo de competencia en tres áreas del conocimiento, destacando la comprensión lectora como uno de los mayores problemas [1].

Es esencial identificar y abordar los problemas de aprendizaje temprano para minimizar las consecuencias del bajo desempeño académico en los estudiantes. Con los avances actuales en tecnología, la educación aprovecha el uso de recursos tecnológicos para mejorar el desempeño académico y mitigar los problemas de comprensión lectora.

En este sentido, el análisis de la mirada mientras se realiza una lectura es una de las áreas de interés tecnológico para la educación. El seguimiento ocular es una herramienta cada vez más valiosa en la investigación sobre el comportamiento visual y la interacción de los estudiantes durante la lectura. Analizando la actividad ocular de los estudiantes, es posible identificar problemas de atención y aprendizaje que podrían estar afectando su desempeño académico [2].

Con base en lo antes mencionado, este trabajo describe el desarrollo de un componente de software que utiliza técnicas de seguimiento ocular basadas en visión artificial para obtener la actividad ocular de estudiantes de nivel secundaria, mediante una cámara web convencional, lo que facilita su implementación en un entorno educativo.

El objetivo principal de este componente es analizar la correlación entre la comprensión lectora y la actividad ocular de un estudiante durante el proceso de aprendizaje. La contribución de este trabajo es proporcionar una herramienta docente que permita la obtención, análisis y visualización de la actividad ocular de los estudiantes, así como una evaluación y aproximación de la comprensión de los estudiantes durante la lectura en computadora.

El resto de este manuscrito se estructura de la siguiente manera. El capítulo 2 presenta los trabajos más relacionados con el presente proyecto, mientras que el diseño y desarrollo del sistema propuesto son descritos en el capítulo 3. El capítulo 4 muestra los resultados obtenidos de la aplicación del software desarrollado a un grupo de nivel secundaria. Finalmente, las conclusiones y trabajo a futuro se discuten en el capítulo 5.

2. Trabajos relacionados

En esta sección se describen y discuten algunos de los trabajos más relevantes en el contexto de detección del nivel de comprensión en la lectura mediante el análisis de movimientos oculares. En el estudio presentado en [3], se propuso un enfoque para mejorar la comprensión lectora a través del análisis de datos de seguimiento ocular y características del texto. Se utilizó una detección automática para generar anotaciones en forma de traducción de palabras o resumen de oraciones complejas.

En el estudio piloto, se logró una precisión promedio del 80,6% ± 6,3% en la anotación generada automáticamente, lo que mejoró la eficiencia de lectura en inglés. Finalmente, este trabajo obtuvo una herramienta útil para mejorar la comprensión lectora en diferentes contextos.

En [4], se utilizó el rastreador ocular GazePoint® GP3 HD para investigar los problemas de los estudiantes de programación y sus estrategias de aprendizaje. Participaron 36 universitarios y se encontró que los estudiantes adoptaban diferentes estrategias de aprendizaje para diferentes tipos de sentencias de programación.

Se observó un aumento en las fijaciones oculares en los apuntes de clase para los alumnos que no comprendían del todo la estructura del código, lo que se utilizó como un indicador temprano para determinar el nivel de aprendizaje de los alumnos.

En [5], se investigaron los patrones de comportamiento durante la lectura. Para ello, se registraron los movimientos oculares de 32 participantes mediante el rastreador Tobii® X2-30 y se propuso un modelo en dos etapas, donde los usuarios primero

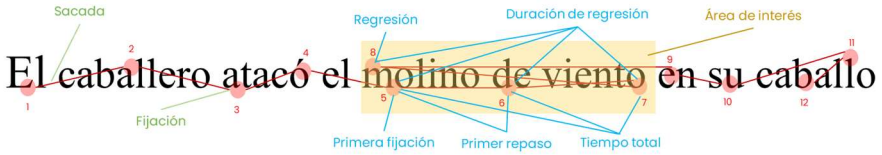


Fig. 1. Actividad ocular de un usuario durante una lectura.

buscan respuestas en el documento y luego realizan la verificación entre las respuestas candidatas.

Los resultados sugieren que el modelo obtenido mejora el desempeño y ayuda a comprender cómo los humanos leen y buscan respuestas. En [6] se analizó el uso de tecnologías de seguimiento ocular para medir el procesamiento cognitivo en un entorno de aprendizaje en línea.

Se recopilaron datos en tiempo real sobre el estado cognitivo de los usuarios, como la atención, la concentración, el cansancio, la relajación, el estrés y el éxito en la resolución de tareas. Los datos obtenidos permitieron el desarrollo de un sistema llamado Protus®, que utiliza tecnologías de reconocimiento ocular para identificar automáticamente el estado de aprendizaje y el estado mental/emocional del usuario.

Por otro lado, en [7], se exploraron métodos para clasificar el nivel de comprensión de un lector utilizando el EyeLink® 1000 como rastreador ocular. Se utilizaron modelos de red profunda para predecir la comprensión de lectura general con un nivel de precisión del 65%.

Los modelos que predijeron las otras variables consideradas no tuvieron un mejor desempeño que la precisión nula, excepto para el primer pasaje, donde la red neuronal convencional (CNN) es ligeramente mejor. Estos hallazgos tienen implicaciones importantes para las comunidades de investigación sobre educación, aprendizaje y capacitación.

En [8], se presentó un sistema de lectura interactivo controlado por los ojos utilizando el Gazepoint® GP3 (ECIRS) en lugar del ratón para controlar el texto digital. Se realizó un experimento comparativo entre un grupo de estudiantes que utilizaron ECIRS y otro grupo que utilizó un sistema de lectura interactivo controlado por ratón (MCIRS) para leer dos tipos de textos en inglés.

Los resultados mostraron que el grupo de ECIRS superó significativamente al grupo de MCIRS en la comprensión lectora del artículo y mejoró la comprensión lectora de los estudiantes independientemente del campo más que la de los estudiantes dependientes del campo. En [9], se propone una relación entre las medidas de movimiento ocular y la predicción de la comprensión lectora.

Se desarrolló una red neuronal para predecir puntajes de comprensión subjetiva y puntajes de cuestionarios. La intensidad lectora se relacionó directamente con la comprensión y, la información obtenida permitió crear entornos de aprendizaje dinámicos que usen el movimiento de los ojos para predecir la dificultad de las preguntas y proporcionar retroalimentación a los instructores sobre el comportamiento de los estudiantes.

El trabajo presentado en [10] describe un enfoque automático que utiliza el método de agrupamiento K-means para analizar el movimiento ocular durante la lectura de un texto de divulgación científica utilizando el rastreador ocular Tobii® EyeX. El análisis

Tabla 1. Precio al público de Eye Trackers.

Compañía	Producto	Precio de venta (aprox)
Tobii-pro®	Tobbi Eye Tracker 5	$307.33 USD
	Tobbi X2 – 30 Hz	$15,000 USD
Eye-link®	EyeLink 1000 plus	$22,950 USD
Smart-eye®	Smart Eye AI-X	$2,345 USD
Gaze-point®	GazePoint GP3 HD	$2,250 USD

identificó tres patrones de comportamiento de lectura. El estudio contribuyó al campo de la enseñanza al proporcionar evidencia y recomendaciones de la importancia del posicionamiento de elementos visuales durante el aprendizaje.

En el estudio descrito en [11] se desarrollaron algoritmos para combinar los datos obtenidos del rastreador ocular Gazepoint GP3® con los patrones de lectura que se producen durante la comprensión de un texto.

Esto permitió la creación de diversas funcionalidades, tales como la evaluación de la comprensión, la detección del nivel de interés y el desplazamiento automático. Como resultado de esta combinación, se logró una precisión del 27 % superior en la detección en comparación con el seguimiento realizado solamente por el rastreador ocular.

En [12], se investigó la posibilidad de predecir el nivel de comprensión lectora mediante el análisis del movimiento ocular. Se utilizó el rastreador ocular Tobii® X1 Light para analizar el patrón de movimiento ocular de 10 estudiantes de posgrado no hablantes de inglés mientras leían artículos en inglés de diferentes niveles de dificultad.

Se aplicaron técnicas de aprendizaje automático para identificar características en el reconocimiento de la comprensión de los lectores, logrando un modelo que identifica diferentes niveles de comprensión con una mejora del rendimiento del 30% por encima de la referencia.

El resumen, el campo de la investigación de los movimientos oculares y su relación con la comprensión lectora está en constante expansión, incluyendo el uso de técnicas de seguimiento ocular en la lectura de texto impreso y la detección de patrones de movimiento ocular en la lectura en línea.

Se desarrollaron diferentes enfoques para modelar y predecir el nivel de comprensión de la lectura a partir de los datos de seguimiento ocular mediante herramientas de alta precisión.

En este sentido, se examinaron estudios relacionados con la aplicación de rastreadores oculares en la educación y se concluye que existe una buena oportunidad para desarrollar un sistema basado en visión por computadora de costo accesible para su implementación en instituciones educativas.

Esto permitiría obtener datos de los registros oculares utilizando cámaras web convencionales y métodos de evaluación durante una lectura de comprensión, contribuyendo con una herramienta para los docentes que les permita clasificar el aprovechamiento de los alumnos durante una lectura de comprensión y realizar acciones prematuras que eviten problemas relacionados con el bajo desempeño escolar.

Tabla 2. Herramientas de seguimiento ocular.

Característica	Trackin.js	GazeRecorder	WebGazer.js	RealEye
Tiempo de rastreo	1 min	1 min	--	45 s
Visualización de actividad ocular	×	Mapa de calor/Video	×	✓
Personalización	✓	✓	×	×
Calibración	✓	✓	✓	✓
Uso de Webcam	✓	✓	✓	✓
Código abierto	✓	✓	✓	×
Versión premium	×	✓	×	✓

3. Desarrollo propuesto

Esta sección describe las características y funcionalidades más representativas del sistema propuesto, además, se discuten diferentes aspectos tecnológicos que han sentado las bases para el desarrollo de este.

3.1. Análisis del entorno de los rastreadores oculares actuales

El seguimiento ocular es un conjunto de dispositivos, técnicas y mecanismos tecnológicos que permiten detectar la presencia de una persona y seguir en tiempo real lo que están mirando. La tecnología convierte los movimientos oculares en datos que contienen información como la posición de la pupila, el vector de la mirada para cada ojo y el punto de enfoque, lo que logra utilizarse como una modalidad de entrada adicional en diversas aplicaciones.

Esta tecnología ofrece una amplia gama de aplicaciones y beneficios en el análisis y seguimiento de los movimientos oculares de las personas [13]. Los dispositivos que se utilizan para rastrear el movimiento de los ojos se denominan eye trackers y aunque la tecnología de eye-tracking aparentemente parece un término reciente, los primeros intentos de rastrear los movimientos del ojo comenzaron a fines del siglo XIX y no del todo agradables para los participantes del estudio.

En aquel entonces, los participantes en los estudios se veían obligados a llevar un molde de yeso cubriendo el ojo con varillas que sobresalían hacia fuera para indicar la posición del ojo en relación con el objeto observado. A medida que avanzó la tecnología, se desarrollaron rastreadores oculares que aún requerían cubiertas para los ojos, pero utilizaban dispositivos similares a los lentes de contacto modernos [14].

Estos dispositivos tecnológicos se idearon para visualizar la atención visual mediante la recolección de movimientos oculares cuando se exponen a diferentes estímulos (tal como se ilustra en la Figura 1). La tecnología de seguimiento ocular permite capturar los datos de la mirada para analizarlos e identificar problemas específicos o predecir su comportamiento.

Los movimientos oculares revelan información sobre cómo el usuario se comporta. Analizando la mirada es posible identificar lo que se ve, en qué orden y durante cuánto tiempo, información que es valiosa para identificar qué hace el alumno mientras realiza una lectura y utilizar esta información para realizar mejoras en los procesos y materiales educativos.

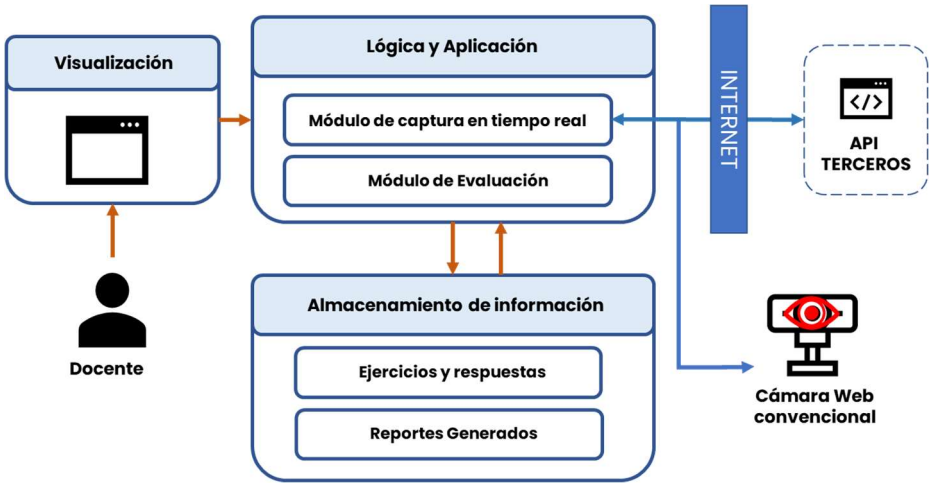


Fig. 2. Arquitectura del sistema propuesto.

Actualmente, para realizar este análisis de seguimiento ocular existen en el mercado diferentes dispositivos para el seguimiento de ojos tales como son Tobii-pro®, Eye-link®, Smart-eye®, Gaze-point®, por mencionar algunos (ver Tabla 1). Sin embargo, la adquisición de uno de estos instrumentos de seguimiento ocular profesional representa una inversión económica muy elevada para instituciones educativas y de investigación que deseen utilizar esta tecnología en sus estudios y proyectos, como se muestra en la Tabla 1.

Por lo tanto, existe la necesidad de buscar alternativas más económicas y accesibles para llevar a cabo este tipo de análisis. Una opción que surgió recientemente para el seguimiento ocular es el uso de cámaras web convencionales y software de seguimiento de ojos basado en visión artificial. Este enfoque utiliza algoritmos para procesar las imágenes capturadas por la cámara y determinar la posición de los ojos y los movimientos oculares del usuario.

La ventaja de esta alternativa es que la mayoría de las computadoras modernas ya están equipadas con cámaras web integradas, lo que significa que no se requiere una inversión adicional en hardware. Además, muchos de los programas de seguimiento de ojos basados en visión artificial están disponibles como software libre y de código abierto, lo que permite a los investigadores y desarrolladores personalizar y ajustar el software para satisfacer sus necesidades específicas.

3.2. Comparativa de herramientas de seguimiento ocular

En el campo del seguimiento ocular, existen diversas herramientas disponibles para llevar a cabo estudios y análisis de la actividad visual. Sin embargo, muchas de estas herramientas resultan ser costosas, lo que dificulta su acceso para aquellos interesados en realizar investigaciones o proyectos de manera independiente.

En este sentido, la Tabla 2 presenta una comparativa de herramientas de seguimiento ocular que utilizan una cámara web convencional en su versión gratuita. Estas

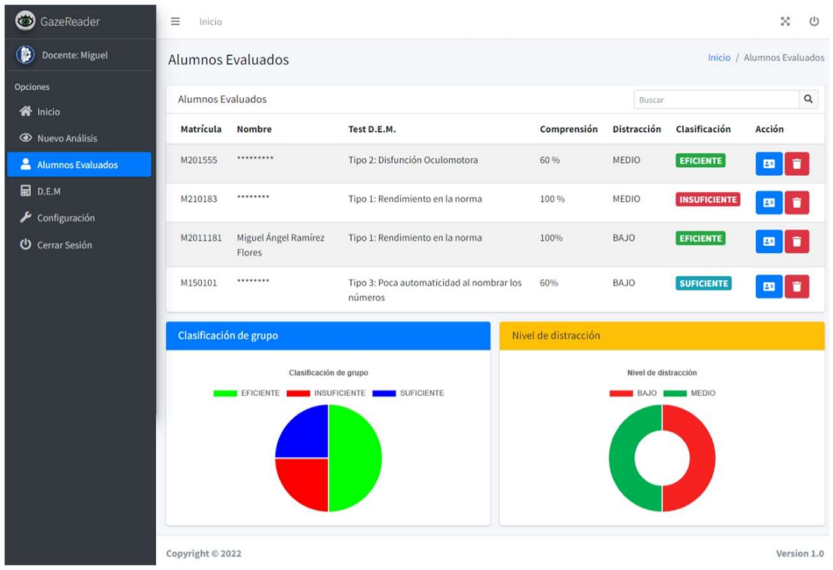


Fig. 3. Interface de usuario.

herramientas permiten a los usuarios llevar a cabo análisis de seguimiento ocular sin la necesidad de adquirir dispositivos costosos.

La comparación se basa en características importantes como tiempo de rastreo, facilidad de uso, compatibilidad con diferentes plataformas, entre otras. Aunque la precisión de estos sistemas de seguimiento de ojos basados en visión artificial aún no alcanza la de los dispositivos de seguimiento ocular profesionales, la tecnología está avanzando rápidamente y se espera que mejore con el tiempo.

Además, la accesibilidad y la facilidad de uso de estas soluciones de seguimiento de ojos las hacen ideales para estudios y proyectos a pequeña escala, como en el ámbito educativo y de investigación en ciencias sociales.

3.3. Elementos del sistema Web propuesto

El software propuesto se desarrolló con la finalidad de capturar, analizar y evaluar la actividad ocular de un usuario mientras realiza una lectura de comprensión en una computadora. Como se aprecia en la Figura 2, la arquitectura del sistema aquí propuesto se divide en diversos módulos que permiten la aplicación de técnicas de seguimiento de ojos basadas en visión artificial y obtener una estructura preliminar para evaluar el comportamiento y aprovechamiento del usuario a través de su mirada.

Luego de un análisis previo de tecnologías disponibles de desarrollo de software, se decidió utilizar el lenguaje de programación PHP y JavaScript para implementar las funcionalidades del sistema y la API GazeCloudAPI® para el módulo de captura en tiempo real mediante una cámara Web convencional.

Se decidió utilizar esta API en particular debido a su alta precisión en la localización de la posición ocular, el sistema de calibración integrado, así como su capacidad para

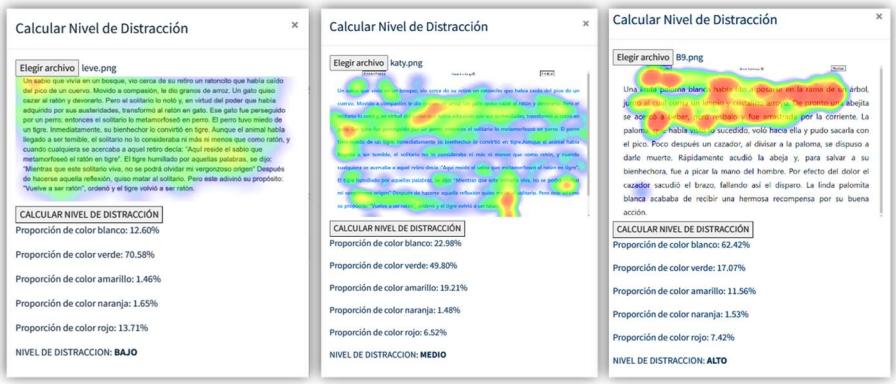


Fig. 4. Clasificador de mapas de calor por proporción de color.

ser implementada de manera rápida y eficiente en el sistema con respecto a las demás herramientas analizadas, como se muestra en la Tabla 2.

3.4. Prototipo del sistema

El sistema Web propuesto tiene como objetivo evaluar y clasificar el nivel de comprensión lectora de los estudiantes de secundaria en una escala de **EFICIENTE, SUFICIENTE** e **INSUFICIENTE** mediante el nivel de desarrollo oculomotor y la estimación del aprovechamiento en una lectura en computadora.

El sistema se basa en un panel de control para visualizar la información registrada, un conjunto de textos preparados acorde al nivel, un quiz de preguntas y respuestas, así como un sistema de calificación de resultados, todo ello implementado con lenguajes de programación como JavaScript y PHP, utilizando la base de datos MariaDB para almacenar los resultados de los estudiantes (Figura 3), y adoptando el modelo arquitectónico MVC (Modelo-Vista-Controlador).

El sistema utiliza técnicas de seguimiento ocular basadas en visión artificial mediante una cámara Web convencional y la implementación de una API para la obtención de datos en tiempo real mediante un mapa de calor sobre el comportamiento de lectura del estudiante.

El sistema de clasificación de imágenes se basa en la proporción de colores en la imagen, lo que permite determinar el nivel de distracción en una lectura de comprensión. Utilizando el mapa de calor generado por la API, se identifican las áreas de mayor actividad ocular y se analiza la proporción de colores presentes en esas áreas. Se establecen tres categorías de clasificación: **ALTO, MEDIO Y BAJO**, en función del nivel de distracción que presenta la imagen.

Para la clasificación de imágenes se utiliza un algoritmo de procesamiento de imágenes que analiza la distribución de los colores en cada píxel de la imagen. Se establecieron proporciones y combinaciones de colores personalizados en un rango que incluye blanco, verde, amarillo, naranja y rojo para determinar los límites entre las diferentes categorías de clasificación, como se observa en la Figura 4. A cada imagen

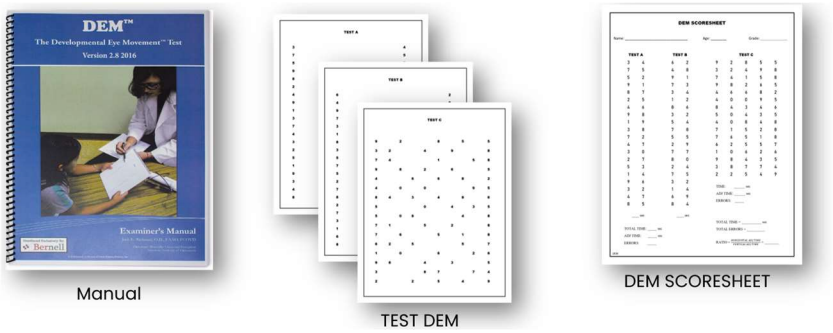


Fig. 5. The Developmental Eye Movement Test (Test DEM).

se le asigna una etiqueta que indica el nivel de distracción que presenta, según el color predominante en las áreas donde se enfoca la vista durante la lectura.

Una vez que las imágenes se clasificaron, se utilizan para analizar los patrones de distracción que se presentan durante la lectura, con el propósito de identificar los puntos críticos y ajustar los datos para lograr una evaluación automática precisa y mejorar las estrategias de enseñanza.

Asimismo, los colores en la imagen más intensos de rojo y naranja se relacionan con una distracción y el aumento en el tiempo de fijación en una zona determinada, lo que sugiere una baja velocidad lectora o desconocimiento de palabras en el texto. Por otro lado, los colores más claros, como el verde y el amarillo, indican una posición ocular correcta y una velocidad de lectura adecuada.

4. Presentación y análisis de datos

En esta sección se describe el caso de estudio desarrollado para evaluar el sistema propuesto. En primer lugar, se describe el lugar donde se llevó a cabo la prueba y se presentan las métricas utilizadas para evaluar el desempeño del sistema. Los resultados obtenidos se analizan y se comparan con las expectativas establecidas previamente. Estos resultados son fundamentales para determinar la eficacia del sistema propuesto y su capacidad para cumplir con los objetivos establecidos.

4.1. Caso de estudio

El caso de estudio se desarrolló en la Escuela Telesecundaria 2 de enero, ubicada en la comunidad de Huiloapan de Cuauhtémoc, Veracruz, en colaboración con dos maestras y una psicóloga. Cabe destacar que dicha escuela forma parte del programa de Unidades de Servicios de Apoyo a la Educación Regular (USAER), que busca proporcionar apoyo a estudiantes con necesidades educativas especiales.

En este caso de estudio, se contó con la colaboración de dos maestras responsables de los grupos analizados y una psicóloga, quienes desempeñaron un papel fundamental en la interpretación y validación de los resultados obtenidos.

Hubo un total de 38 alumnos de primer año de secundaria en la muestra, los cuales se distribuyeron en dos grupos de participantes con el propósito de evaluar su

desempeño en una actividad de comprensión lectora mediante el uso de un sistema de seguimiento ocular basado en visión por computadora.

Para llevar a cabo la evaluación de los alumnos se empleó el sistema desarrollado que implementa tecnología del seguimiento de la actividad ocular a través de una cámara web convencional. Y con el fin de medir su motricidad ocular, se utilizó el test de desarrollo ocular conocido como: The developmental eye movement test (DEM) (Figura 5).

Dicha prueba permite clasificar a los evaluados en cuatro tipos, según sus resultados y edad: Tipo 1, correspondiente al rendimiento en la norma; Tipo 2, relacionado con disfunción oculomotora; Tipo 3, caracterizado por poca automaticidad al nombrar los números; y Tipo 4, definido por deficiencias tanto en la automaticidad como en la función oculomotora.

Previamente, al inicio de las pruebas, se realizó una reunión de capacitación con los maestros y los estudiantes para la explicación detallada del uso del sistema y las instrucciones de la actividad. Además, se habilitó un espacio temporal en la biblioteca escolar para la realización de las pruebas, con el fin de evitar interrupciones y garantizar un ambiente adecuado para la realización de la actividad.

Posteriormente, con los datos obtenidos en las pruebas realizadas, se clasificó a los estudiantes según su estimación de aprovechamiento y desarrollo oculomotor en una escala de Eficiente, Suficiente e Insuficiente. Las maestras y la psicóloga participaron activa y colaborativamente en el desarrollo del proyecto, lo que resultó crucial para obtener datos relevantes para el sistema, así como de resultados precisos, la interpretación y validación adecuada de los mismos.

Estas profesionales aportaron su amplia experiencia y conocimiento en el ámbito educativo, lo que permitió una visión integral y contextualizada de los resultados obtenidos a través del seguimiento de la actividad ocular de los estudiantes en relación del aprovechamiento durante la actividad de comprensión lectora en computadora.

La labor de las maestras y la psicóloga incluyó no solo la interpretación de los datos obtenidos a través del seguimiento ocular, sino también la identificación de patrones, tendencias en el comportamiento y la motricidad ocular de los estudiantes.

Además, se encargaron de generar una clasificación de sus estudiantes acorde a su desempeño visto en clases y comparar los resultados obtenidos posteriormente a la implementación del sistema, así como de proponer recomendaciones y estrategias para mejorar el desempeño académico de los estudiantes en el área de comprensión lectora y desarrollo oculomotor.

4.2. Evaluación de métricas

Luego de la implementación de las pruebas mediante el sistema propuesto, se presenta en esta sección el análisis de resultados de los 38 alumnos de primer año de secundaria evaluados. Aquí se presentan los resultados obtenidos y se incluye la Tabla 3 que compara la clasificación otorgada por la profesora con la otorgada por la plataforma.

El análisis de estos resultados permite evaluar la efectividad del sistema en la clasificación del nivel de aprovechamiento de los estudiantes y su correlación con la actividad ocular durante la lectura.

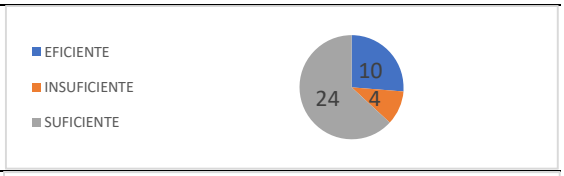
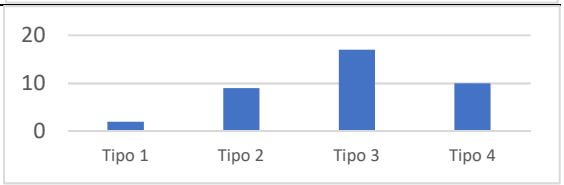
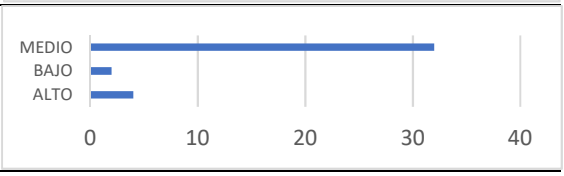
Tabla 3. Comparativa de clasificación.

Alumno	Clasificación otorgado por la profesora	Clasificación asignada por la plataforma
Alumno 1	Suficiente	Eficiente
Alumno 2	Suficiente	Suficiente
Alumno 3	Suficiente	Suficiente
Alumno 4	Suficiente	Suficiente
Alumno 5	Suficiente	Suficiente
Alumno 6	Suficiente	Suficiente
Alumno 7	Insuficiente	Insuficiente
Alumno 8	Eficiente	Eficiente
Alumno 9	Eficiente	Suficiente
Alumno 10	Eficiente	Eficiente
Alumno 11	Suficiente	Suficiente
Alumno 12	Eficiente	Eficiente
Alumno 13	Suficiente	Eficiente
Alumno 14	Suficiente	Suficiente
Alumno 15	Suficiente	Eficiente
Alumno 16	Suficiente	Eficiente
Alumno 17	Suficiente	Eficiente
Alumno 18	Suficiente	Eficiente
Alumno 19	Suficiente	Eficiente
Alumno 20	Suficiente	Eficiente
Alumno 21	Eficiente	Eficiente
Alumno 22	Suficiente	Eficiente
Alumno 23	Suficiente	Insuficiente
Alumno 24	Suficiente	Suficiente
Alumno 25	Insuficiente	Insuficiente
Alumno 26	Suficiente	Suficiente
Alumno 27	Eficiente	Eficiente
Alumno 28	Suficiente	Suficiente
Alumno 29	Eficiente	Eficiente
Alumno 30	Eficiente	Eficiente
Alumno 31	Eficiente	Suficiente
Alumno 32	Eficiente	Suficiente
Alumno 33	Insuficiente	Insuficiente
Alumno 34	Suficiente	Suficiente
Alumno 35	Suficiente	Suficiente
Alumno 36	Suficiente	Suficiente
Alumno 37	Suficiente	Suficiente
Alumno 38	Insuficiente	Insuficiente

Con los resultados obtenidos de la evaluación y clasificación, se realizó una comparación entre la clasificación hecha por las docentes y la generada por el sistema, y se encontró que la plataforma clasificó de manera similar a más del 65% de los estudiantes.

Esto sugiere que la plataforma puede clasificar a los estudiantes con un nivel significativo de similitud en comparación con la clasificación que obtienen en el aula, lo que demuestra que la implementación del sistema propuesto tiene un alto grado de

Tabla 4. Resultados obtenidos de la muestra.

Característica	Resultado
Clasificación del grupo	
Tipo de desarrollo ocular (DEM) en la muestra	
Nivel de distracción durante la lectura	

precisión en la clasificación de los estudiantes y es viable para su implementación en entornos educativos a un costo accesible.

Los resultados estadísticos de la muestra, como la clasificación del grupo, nivel de distracciones, entre otras, se muestra en la Tabla 4. Además, se observó que el sistema propuesto demostró una mayor efectividad en la clasificación de los estudiantes en los niveles de **Suficiente** e **Insuficiente**.

Posteriormente a la evaluación, se identificaron algunos aspectos que podrían mejorar la precisión de la clasificación de los estudiantes, como el porcentaje de evaluación en los test DEM y el nivel de distracción. Para determinar las posibles causas de estos hallazgos, se realizó un análisis exhaustivo de los resultados, y se está trabajando en ajustes para el sistema de clasificación con el fin de mejorar la precisión en la clasificación de los estudiantes en los tres niveles.

5. Conclusiones y trabajo a futuro

El incremento constante de estudiantes en México con un bajo aprovechamiento es una problemática alarmante en el ámbito educativo. Uno de los principales factores relacionados con este bajo rendimiento escolar es la falta de comprensión lectora. En los últimos tiempos, los avances tecnológicos permiten invertir en investigación, desarrollo de sistemas y dispositivos para detectar patrones que afecten el aprendizaje de los estudiantes.

En este sentido, el uso de dispositivos de seguimiento ocular se considera una herramienta prometedora para ayudar a analizar y comprender la relación entre la actividad ocular y el aprendizaje de los alumnos y aunque los dispositivos de seguimiento ocular profesionales siguen siendo la opción más precisa y confiable, los

sistemas de seguimiento de ojos basados en visión artificial ofrecen una alternativa más accesible y económica para aquellos con un presupuesto limitado.

En este trabajo, se presentó un sistema Web como herramienta docente que ayuda a facilitar la identificación y obtención del comportamiento visual de los estudiantes a través de una cámara Web convencional, así como una aproximación de su comprensión durante la lectura en computadora, para clasificarlos en una escala de eficiente, suficiente e insuficiente de acuerdo a los resultados obtenidos en las pruebas de aprovechamiento lector y motricidad ocular.

La investigación contó con la participación de 38 alumnos del primer año de secundaria bajo el consentimiento de las autoridades escolares y la colaboración de 2 maestras y una psicóloga.

La principal ventaja de esta herramienta es su capacidad para permitir a los docentes evaluar, analizar y visualizar la actividad ocular, así como el nivel de aprovechamiento de los estudiantes durante la lectura en computadora, lo que permite detectar problemas de atención y aprendizaje en los estudiantes de igual manera tomar medidas tempranas para mejorar su desempeño académico y la posibilidad de integrar el sistema en el plan de estudios, para que los docentes puedan utilizarlo como una herramienta de diagnóstico y seguimiento en el aula.

Como investigaciones futuras, se tiene la intención de profundizar en el análisis y detección de trastornos visuales que afecten el aprendizaje de los estudiantes en todos los niveles educativos mediante la implementación de dispositivos de seguimiento ocular accesibles y rentables.

También se pretende incorporar diversos algoritmos de aprendizaje automático con el fin de mejorar aún más la eficiencia y la validez de los resultados obtenidos a través del sistema. Esta tarea será esencial para poder identificar problemas visuales tempranos en los estudiantes y proporcionar intervenciones adecuadas para garantizar un desempeño académico óptimo.

Agradecimientos. Este trabajo de investigación fue financiado por el Consejo Nacional de Ciencia y Tecnología de México (CONACYT) y la Secretaria de Educación Pública (SEP) de México. De igual manera, se agradece al Tecnológico Nacional de México (TecNM) por el apoyo para la realización de este trabajo.

Referencias

1. Salinas, D., Moraes, C., Schwabe, M.: Programa para la evaluación internación de alumnos (PISA) PISA 2018- Resultados. OECD 2019, vol. I-III, pp. 1–12 (2019) https://www.oecd.org/pisa/publications/PISA2018_CN_MEX_Spanish.pdf
2. The Padula institution of vision rehabilitation: Attention deficit disorder (ADD) and vision problems (2019) padulainstitute.com/attention-deficit-disorder-vision-problems/
3. Guo, W., Cheng, S.: An approach to reading assistance with eye tracking data and text features. In: Adjunct of the 2019 International Conference on Multimodal Interaction, Association for Computing Machinery, no. 7, pp. 1–7 (2019) doi: 10.1145/33515 29.3360659
4. Cheng, G., Poon, L., Lau, W., Zhou, R.: Applying eye tracking to identify students' use of learning strategies in understanding program code. In: Proceedings of the 3rd International Conference on Education and Multimedia Technology, Association for Computing Machinery, pp. 140–144 (2019) doi: 10.1145/3345120.3345144

5. Zheng, Y., Mao, J., Liu, Y., Ye, Z., Zhang, M., Ma, S.: Human behavior inspired machine reading comprehension. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, pp. 425–434 (2019) doi: 10.1145/3331184.3331231
6. Ivanović, M., Klašnja-Milićević, A., Ivković, J., Porta, M.: Integration of eye tracking technologies and methods in an E-learning system. In: Proceedings of the 8th Balkan Conference in Informatics, Association for Computing Machinery, no. 29, pp. 1–4 (2017) doi: 10.1145/3136273.3136278
7. Ahn, S., Kelton, C., Balasubramanian, A., Zelinsky G.: Towards predicting reading comprehension from gaze behavior. In: ACM Symposium on Eye Tracking Research and Applications, Association for Computing Machinery, no. 32, pp. 1–5 (2020) doi: 10.1145/3379156.3391335
8. Chang, C., Chen, C., Lin, Y.: A visual interactive reading system based on eye tracking technology to improve digital reading performance. In: 2018 7th International Congress on Advanced Applied Informatics, pp. 182–187 (2018) doi: 10.1109/IIAI-AAI.2018.00043
9. Copeland, L., Gedeon, T.: Measuring reading comprehension using eye movements. In: IEEE 4th International Conference on Cognitive Infocommunications, pp. 791–796 (2013) doi: 10.1109/CogInfoCom.2013.6719207
10. Lin, W., Kotakehara, Y., Hirota, Y., Murakami, M., Kakusho K., Yueh, H.: Modeling reading behaviors: An automatic approach to eye movement analytics. IEEE Access, vol. 9, pp. 63580–63590 (2021) doi: 10.1109/ACCESS.2021.3074913
11. Bottos, S., Balasingam, B.: Tracking the progression of reading using eye-gaze point measurements and hidden Markov models. IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 10, pp. 7857–7868 (2020) doi: 10.1109/tim.2020.2983525
12. Li, J., Ngai, G., Leong, H. V., Chan S.: Your eye tells how well you comprehend. In: IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), pp. 503–508 (2016) doi: 10.1109/COMPSAC.2016.220
13. Aware biometrics: Reconocimiento del iris. Aware Trusted since 1993 (2021) https://www.aware.com/es/reconocimiento-del-iris/
14. Schall, A., Romano-Bergstrom, J.: Introduction to eye tracking. Eye Tracking in User Experience Design, pp. 3–26 (2014) doi: 10.1016/b978-0-12-408138-3.00001-7

ISSN 1870-4069

Uso de metaheurísticas para entrenamiento de redes neuronales artificiales

Yoqsan Angeles García[1], Hiram Calvo[1],
Álvaro Anzueto Ríos[2]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

[2] Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas,
México

{yangelesg2020,hcalvo}@cic.ipn.mx, aanzuetor@ipn.mx

Resumen. Los algoritmos basados en métodos metaheurísticos son aplicados a problemas de optimización y han demostrado un gran sentido de convergencia en sus soluciones, es por eso que en este trabajo se ha considerado su aplicación para determinar los valores de los pesos sinápticos en una arquitectura neuronal que ha sido configurada como aproximador de funciones. Se han elegido tres funciones típicas (benchmark) como pruebas de rendimiento, además, de considerar el error cuadrático medio (ECM), como función objetivo o de ajuste para los algoritmos metaheurísticos. El análisis de los resultados indica que los algoritmos de abejas y evolución diferencial funcionan bien para problemas de alta dimensionalidad y recocido simulado presenta la mejor relación entre tiempo y valor óptimo. En general, este estudio demuestra la eficiencia de las metaheurísticas para resolver problemas de entrenamiento de redes neuronales artificiales.

Palabras clave: Colonia artificial de abejas, evolución diferencial, recocido simulado, optimización por enjambre de partículas, redes neuraonales artificiales.

Use of Metaheuristics for Artificial Neural Network Training

Abstract. The algorithms based on metaheuristic methods are applied to optimization problems and have demonstrated a high degree of convergence in their solutions. Therefore, this study considers their application to determine the values of synaptic weights in a neural architecture configured as a function approximator. Three typical functions (benchmarks) were chosen as performance tests, in addition to using mean squared error (MSE) as the objective or fitting function for the metaheuristic algorithms. The analysis of the results indicates that artificial bee colony and differential evolution algorithms perform well for

high-dimensional problems, while simulated annealing has the best trade-off between time and optimal value. Overall, our findings demonstrate the efficacy of metaheuristic methods in solving training problems in artificial neural networks.

Keywords: Artificial bee colony, differential evolution, simulated annealing, particle swarm optimization, artificial neural networks.

1. Introducción

Las redes neuronales tienen diversas aplicaciones en diferentes áreas, tales como clasificación, control, procesamiento de imágenes, procesamiento de lenguaje natural, etc. La literatura clásica sobre redes neuronales muestra el método de descenso de gradiente como la forma de optimización de los pesos de las redes neuronales [1, 2]. El descenso de gradiente es un algoritmo basado en derivadas para encontrar la dirección de máximo decremento de una función [3]. Sin embargo, existen métodos de optimización que involucran procesos estocásticos.

Estos métodos son las metaheurísticas. Estos métodos sirven como una alternativa al método de descenso de gradiente y sus variantes [4]; pues, a pesar de que ha demostrado un desempeño satisfactorio en términos de los resultados obtenidos, existen trabajos en los que se muestra que existen circunstancias en las que puede no converger a un valor óptimo [5, 6]. Aunque existen diferentes algoritmos metaheurísticos [7], para este trabajo se han considerado desarrollar 5 metodologías de tipo metaehurísticas, las cuales son:

- Recocido Simulado.
- Optimización por Enjambre de Partículas.
- Algoritmos Genéticos.
- Colonia Artificial de Abejas.
- Evolución Diferencial.

La elección de metaheurísticas adecuadas para el entrenamiento de redes neuronales es fundamental para lograr una optimización efectiva. En este sentido, los algoritmos genéticos, la colonia artificial de abejas, la optimización por enjambre de partículas, el recocido simulado y la evolución diferencial son cinco metaheurísticas que pueden justificarse por su eficacia en problemas de optimización no lineales y no convexas.

Estas técnicas también tienen la capacidad de manejar múltiples objetivos y soluciones no factibles, lo que las hace ideales para el entrenamiento de redes neuronales complejas.

El objetivo de esta investigación es comparar las características y el desempeño de las metaheurísticas seleccionadas en el entrenamiento de redes neuronales, con el fin de evaluar su eficacia y determinar cuál de ellas es más adecuada para este fin.

Asimismo, la exploración de otras metaheurísticas puede ser de gran interés para futuras investigaciones en el campo de la optimización de redes neuronales. Otras metaheurísticas que también podrían ser consideradas son la búsqueda tabú, la

Tabla 1. Algunos trabajos relacionados.

Título	Parámetros de optimización	Metaheurística	Año
Particle Swarm Optimization of Neural Network Architectures and Weigths	Optimización de pesos y arquitecturas en Perceptrón Multicapa en problemas en el área médica	PSO	2007
Optimizing connection weights in neural netwokrs using the whale optimization algorithm	Optimización de pesos en Perceptrón Multicapa para 20 problemas de clasificación	Algoritmo de la Ballena (algoritmo poblacional)	2016
Artificial Neural Network training using metaheuristics for medical data classification: An experimental study	Optimización de pesos en Perceptrón Multicapa para clasificación de imágenes médicas	Comparación de 14 metaheurísticas diferentes	2022

búsqueda armónica, colonia de hormigas, entre otras. A continuación se da un breve resumen de cada una.

1.1. Recocido simulado (SA)

El recocido simulado (SA por sus siglas en inglés de Simulated Annealing) es un algoritmo que simula el comportamiento del enfriamiento lento de sistemas físicos simples, por ejemplo en el proceso de producción de acero. La idea de la que parte es que, al calentar los metales, las moléculas que lo componen están en movimiento.

Sin embargo, tras un lento proceso de enfriamiento, tales partículas alcanzan un estado de mínima energía, a pesar del movimiento aleatorio de las moléculas. Este método de recocido simulado permite explotar el conocimiento físico de tales procesos para tratar de encontrar el mínimo absoluto de cualquier función matemática [7].

1.2. Optimización por enjambre de partículas (PSO)

El algoritmo de Optimización de Enjambre de Partículas (PSO por sus siglas en inglés de Particle Swarm Optimization), fue desarrollado por Kennedy y Eberhart [8] y se basa en el comportamiento social que se ha observado en distintos grupos de individuos como pueden ser parvadas de aves, enjambres de insectos o bancos de peces.

El comportamiento grupal es el resultado de la interacción de dos factores mutuamente dependientes: las conductas individuales exhibidas por cada miembro y la conducta emergente que se manifiesta en el comportamiento colectivo del grupo. Cada individuo transmite información al resto del grupo, lo cual resulta en un proceso que permite a los miembros encontrar un valor adecuado.

Básicamente, PSO consiste en un algoritmo iterativo basado en una población de individuos, en la que cada partícula, o miembro, explora una parte del espacio de búsqueda en busca de soluciones óptimas y comparte la información con el resto del enjambre.

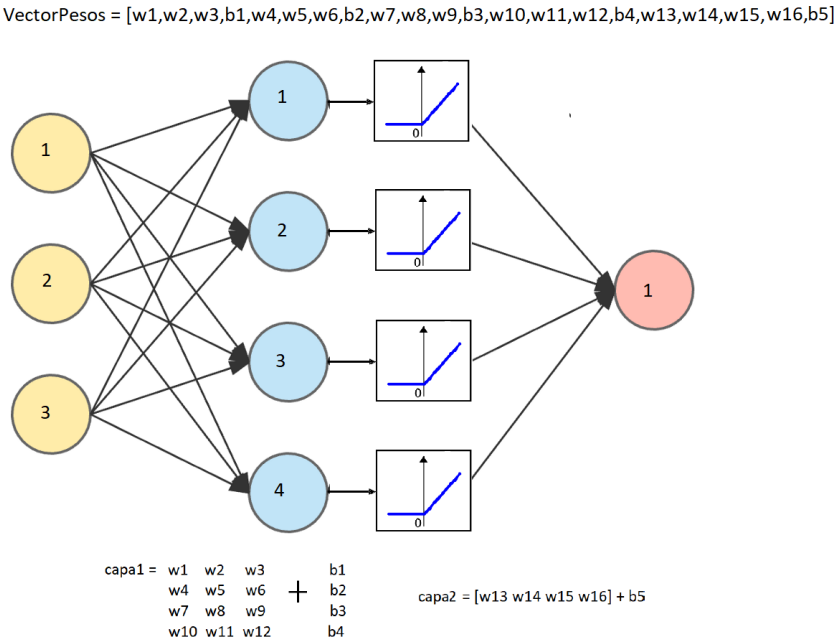


Fig. 1. Representación gráfica del acomodo de un vector de pesos en las capas de una red neuronal.

1.3. Algoritmos genéticos (GA)

Los Algoritmos Genéticos (GA por sus siglas en inglés de Genetic Agorithm) parten de la idea de la evolución darwiniana. Se utilizan para problemas de optimización con una población de individuos, los cuales se componen de un cromosoma de la forma $x = [x_1, x_2, ..., x_n]$.

De forma clásica, el cromosoma es un número binario, donde cada gen o bit x_i, únicamente puede tener valores de 1 y 0; sin embargo, también puede adaptarse para números reales en cada gen. El algoritmo genético considera una población inicial de M cromosomas de largo b, generados aleatoriamente.

Para evaluar el desempeño de estos individuos se requiere decodificar cada componente en caso de trabajar con números binarios o evaluar la función en caso de números reales.

Durante el proceso evolutivo, el algoritmo genético selecciona a los miembros más aptos de la población para realizar el proceso de cruza o reproducción, en el cual se tienen nuevos individuos, idealmente mejores que los padres y que formarán la nueva población.

También existe un proceso de mutación, en el cual se muta un gen del cromosoma de acuerdo a una probabilidad dada. La mutación sirve para evitar la pérdida de diversidad, producto de genes que han convergido a un cierto valor para toda la población, y que por tanto, no pueden ser recuperados por el operador de recombinación [9].

Tabla 2. Funciones, dimensiones y tamaño de red.

Función	Dimensión	Dominio	Datos de entrenamiento totales	Parámetros de Red Neuronal
Sphere	2	-5,5	1x250	31
Rastrigin	2	-5,5	1x250	31
Griewank	2	-50,50	1x250	31
Sphere	11	-5,5	10x2500	1201
Rastrigin	11	-5,5	10x2500	1201
Griewank	11	-50,50	10x2500	1201

Tabla 3. Datos estadísticos de ejecución para la función Spehre en 2D.

Metaheurística	Mejor	Peor	Promedio	Mediana	STD	Promedio de épocas hasta convergencia
SA	0.2615	2.2798	1.0512	0.9178	0.7456	1834
GA	0.3195	1.5513	0.9079	0.9240	0.4875	950
ABC	0.1873	0.5080	0.3271	0.33	0.1365	1092
PSO	0.5846	8.5009	3.1326	1.2450	3.4476	685
DE	0.1351	0.5797	0.2608	0.2065	0.1817	1133

1.4. Colonia artificial de abejas (ABC)

Colonia de Abejas Artificiales (ABC por sus siglas en inglés de Artificial Bee Colony) es una técnica propuesta por Karaboga y Basturk [10] que se basa en el comportamiento de las abejas melíferas para encontrar las mejores fuentes de alimento como individuo y también para que el resto de la colmena aproveche ese alimento al comunicar esta información a otras abejas.

ABC es un algoritmo diseñado para resolver problemas de optimización combinatoria, que se basa en poblaciones donde las soluciones, llamadas fuentes de alimento, son exploradas por las abejas. El objetivo de estas abejas es descubrir nuevas fuentes de alimento que tengan cada vez mayores cantidades de néctar.

Para el caso de optimización, entre más cercano al óptimo sea el valor de una función, significa una mayor cantidad de néctar. Sin embargo, si después de explorar una fuente de alimento no encuentran una mejor en la vecindad, otras abejas exploradoras se moverán aleatoriamente por el espacio de búsqueda para encontrar una nueva fuente de alimento. De esta forma se obtienen soluciones óptimas locales e, idealmente, el óptimo global.

En un sistema ABC, las abejas artificiales se mueven en un espacio de búsqueda multidimensional eligiendo fuentes de néctar dependiendo de su experiencia pasada y de sus compañeras de colmena. ABC trata de balancear los métodos de exploración y explotación.

1.5. Evolución diferencial (DE)

La evolución diferencial (DE por sus siglas en inglés de Differential Evolution) es una rama de la computación evolutiva desarrollada por Rainer Storn y Kenneth Price [11] para optimización en espacios continuos.

Tabla 4. Datos estadísticos de ejecución para la función Rastrigin en 2D.

Metaheurística	Mejor	Peor	Promedio	Mediana	STD	Promedio de épocas hasta convergencia
SA	49.75	55.41	52.95	54.62	2.9123	2124
GA	46.02	49.51	47.52	47.46	1.3475	842
ABC	44.28	46.24	45.53	45.56	0.8037	1514
PSO	45.90	75.02	54.67	51.03	11.5921	696
DE	47.44	49.32	48.38	48.62	0.7484	1221

Tabla 5. Datos estadísticos de ejecución para la función Griewank en 2D.

Metaheurística	Mejor	Peor	Promedio	Mediana	STD	Promedio de épocas hasta convergencia
SA	48.71	50.89	49.93	50.25	1.04	1343
GA	45.37	47.65	46.81	47.16	0.9372	1072
ABC	47.04	47.65	47.12	47.12	0.25	1013
PSO	47.02	50.81	48.61	47.94	1.55	739
DE	47.22	49.78	48.24	47.89	1.05	1970

En este método, las variables se representan mediante números reales. Parte de una población inicial generada aleatoriamente y, a diferencia de los algoritmos genéticos, se seleccionan tres individuos como padres. Uno de estos individuos es el padre principal y éste será el que se modifica con la información de los otros dos padres.

Si el valor resultante es mejor que el del padre principal, entonces se reemplaza, de lo contrario, se descarta y se conserva al padre principal. En el algoritmo de ED, para la generación de nuevos vectores, se suma la diferencia de pesos entre dos vectores miembros de la población a un tercer vector miembro o padre principal.

2. Estado del arte

El trabajo sobre metaheurísticas para el uso en redes neuronales es amplio, existen trabajos en los que se recopilan diversos trabajos sobre el tema. En el trabajo Advances of metaheuristic algorithms in training neural networks for industrial applications [12] se realiza un resumen los algoritmos metaheurísticos de los últimos 20 años y se indica la eficiencia que han tenido al utilizarse para entrenamiento de redes neuronales artificiales en aplicaciones en la industria.

Se llega a la conclusión de que las metaheurísticas tienen una buena relación entre exploración y explotación, lo que les da cierta ventaja en redes neuronales de propagación hacia adelante. Sin embargo, tal como dice el teorema de No Free Lunch, ningún algoritmo de estos ha servido mejor para resolver todos los problemas, por lo que aún queda la tarea de elegir el mejor método y adaptarlo para la aplicación deseada.

A pesar de la diversidad de trabajos antes mencionada, el enfoque de las metaheuristicas aplicadas en redes neuronales no se utliza para el cálculo de los pesos, se utiliza para optimización de hiperparámetros.

Tabla 6. Datos estadísticos de ejecución para la función Sphere en 11D.

Metaheurística	Mejor	Peor	Promedio	Mediana	STD	Promedio de épocas hasta convergencia
SA	24.61	45.76	26.87	35.84	7.53	37020
GA	144.42	202.28	164.42	171.34	30.82	10488
ABC	34.65	45.86	42.59	41.84	3.97	20124
PSO	911.38	1553	951.94	1322	244	2369
DE	16.38	189.2	26.45	27.47	78.14	5621

Tabla 7. Datos estadísticos de ejecución para la función Rastrigin en 11D.

Metaheurística	Mejor	Peor	Promedio	Mediana	STD	Promedio de épocas hasta convergencia
SA	2287	2416	2383.6	2337	50.36	2880
GA	2359	2590	2499.0	2549	38.87	9039
ABC	2458	2708	2511.8	2514	95.69	9733
PSO	3773	4365	3944.4	4093	222	1400
DE	2199	2364	2287.1	2349	31.98	11072

En la Tabla 1 se presenta una recopilación de trabajos previos que han abordado el problema, tanto, de la obtención de los pesos sinápticos de arquitecturas neuronales con perceptrones multicapa, como los hiperparámetros en redes neuronales convolucionales para la tarea de clasificación de imágenes, demostrando el hecho que este es un tópico de interés actual para los investigadores [13, 14, 15].

3. Desarrollo de la solución

En esta sección se describen los pasos que se llevaron a cabo para dar solución al problema planteado.

3.1. Descripción de la solución al problema planteado

Este trabajo se realizó en MATLABr2022B bajo la licencia de estudiante. Sin embargo, no se hizo uso de la DeepLearningToolbox incluida. La forma en la que se minimiza el error en todos los algoritmos aquí presentados es la modificación de un número n de pesos de la red neuronal por cada época, cambiando entre cada algoritmo la forma en la que se modifican estos pesos.

La forma en la que se codifican las entradas para realizar el proceso de optimización es el siguiente: Se inicializa un vector con n datos, siendo n el número de variables que puede modificarse en la red neuronal, para este caso, es el conjunto de pesos W y bias b de la red neuronal.

Teniendo el vector de pesos, es posible realizar modificaciones dentro del vector para posteriormente ejecutar la red y calcular el error cuadrático medio entre la salida deseada y la salida obtenida de la red neuronal. La Figura 1 muestra un ejemplo de red neuronal de tres capas, en la parte superior, el vector que se utiliza para ejecutar la red neuronal.

Tabla 8. Datos estadísticos de ejecución para la función Griewank en 11D.

Metaheurística	Mejor	Peor	Promedio	Mediana	STD	Promedio de épocas hasta convergencia
SA	130.19	174.19	170.08	150.26	19.08	6440
GA	51.02	113.02	66.99	97.86	36.18	6446
ABC	116.96	230.75	121.05	127.25	6.69	1226
PSO	45.95	170.1	49.27	67.69	50.73	4071
DE	206.24	230.75	216.96	227.25	6.69	1226

Tabla 9. Datos estadísticos de ejecución para la función Sphere, Rastrigin y Griewank en 11D con datos desconocidos.

Función	Metaheurística	Error de entrenamiento	Error de prueba	Tiempo por 100 épocas en segundos
Sphere	SA	26.87	15.82	0.1297
Sphere	GA	162.42	107.86	5.3941
Sphere	ABC	42.59	23.33	12.0833
Sphere	PSO	951.94	848.56	5.42
Sphere	DE	26.45	13.80	8.6071
Rastrigin	SA	2383.6	3008	0.1297
Rastrigin	GA	2499.0	2846	5.3941
Rastrigin	ABC	2511.8	2851	12.0833
Rastrigin	PSO	3944.4	4087	5.42
Rastrigin	DE	2287.1	2721	8.6071
Griewank	SA	170.08	94.06	0.1297
Griewank	GA	66.99	41.97	5.3941
Griewank	ABC	121.05	90.32	12.0833
Griewank	PSO	49.27	32.27	5.42
Griewank	DE	216.96	162.39	8.6071

En la parte inferior se muestra cómo se acomodan los elementos del vector en forma de matrices para realizar las operaciones necesarias en la ejecución de la red. A continuación se describen los parámetros para cada uno de los algoritmos metaheurísticos:

– **ABC**: Este algoritmo requiere conocer el tamaño de la población inicial, llamado Np con valor de 50, también requiere el límite N para mejorar una solución.

– **GA**: Este algoritmo requiere conocer el tamaño de la población inicial, llamado Np con valor de 50, la probabilidad de cruza, llamada Mc con valor de 1, el tipo de selección de padres que es por torneo con una población de 2 miembros por torneo, $M = 0.01$, el tipo de cruza y el tipo de mutación.

Para este trabajo el tipo de cruza es el método conocido como cruza de dos puntos y la mutación se realiza cambiando el gen x por un número aleatorio dentro del espacio de búsqueda. La selección de la nueva población se hace de forma elitista eligiendo los Np miembros con mejor aptitud, incluyendo padres e hijos.

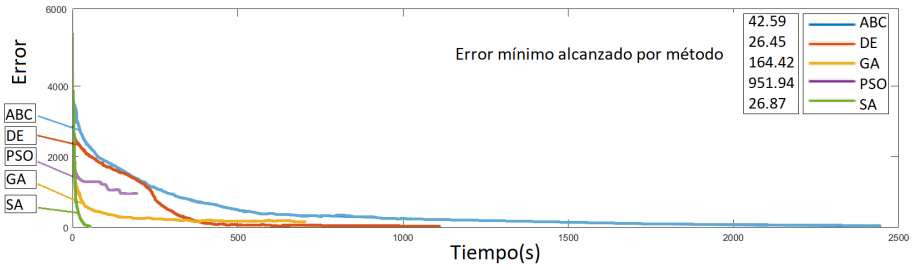


Fig. 2. Comparación de error en el entrenamiento para la función Sphere en 11 dimensiones.

– **SA**: Este algoritmo requiere conocer el tipo de enfriamiento, el tamaño de vecindad para elección de nuevas soluciones. El tipo de enfriamiento usado en este trabajo es un enfriamiento lineal, el cual requiere conocer la temperatura inicial, de 10 y DeltaT de 0.5.

– **DE**: Este algoritmo requiere conocer el tamaño de la población, llamado Np con valor de 50, un factor de escala, llamado alpha con un valor de 0.3 y la probabilidad de cruza, llamada Pcr con un valor de 0.4.

– **PSO**: Este algoritmo requiere conocer el tamaño de la población, llamado Np con valor de 50, un coeficiente de aceleración personal con un valor de 1, un coeficiente de aceleración social con un valor de 2 y un factor de escala de desaceleración, llamado w con un valor de 0.75.

En cada algoritmo usado en este trabajo, el número de pesos a modificar por cada época varía de acuerdo a la diferencia del error nuevo y el anterior. En un principio se modifica 2 % de los pesos totales. Al alcanzar un número de épocas con una diferencia de error menor a un δ dado, el número de pesos a modificar se reduce a 1 %, posteriormente a 0.1 % y por último a un peso por época. Para los algoritmos se tienen dos criterios de paro: uno es el número máximo de épocas, que es de 45,000 y un máximo de iteraciones en las que la diferencia entre el error actual y el error previo es menor a un ϵ dado, en este caso $\epsilon = 0.1$.

3.2. Evaluación

Para la evaluación de las redes neuronales, se eligió que éstas resuelvan el problema de aproximación de funciones. Se toman tres funciones matemáticas a aproximar: Sphere [1], Rastrigin [2] y Griewank [3].

Estas funciones se toman de [16], donde se hace la comparación similar a la propuesta aquí planteada; pero comparando tres tipos de redes neuronales en lugar de métodos de entrenamiento. Se prueban las funciones en dos y once dimensiones:

$$f(x) = \sum_{i=1}^{D} x_i^2, \tag{1}$$

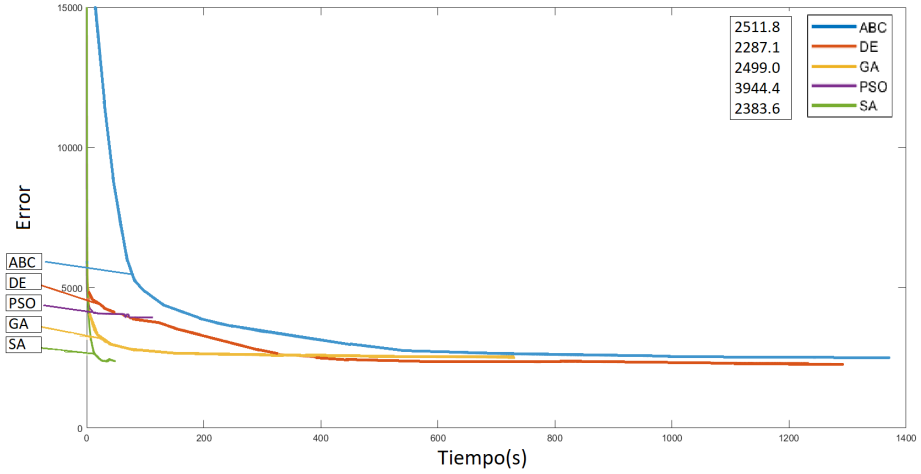


Fig. 3. Comparación de error en el entrenamiento para la función Rastrigin en 11 dimensiones.

$$f(x) = \sum_{i=1}^{D} x_i^2 - 10cos(2\pi x_i) + 10, \tag{2}$$

$$f(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} cos(\frac{x_i}{\sqrt{i}}) + 1. \tag{3}$$

La Tabla 2 muestra características de las funciones. El número de datos en la capa de entrada es el número de dimensión menos uno. Por ejemplo, para la función de dos dimensiones, sólo hay una neurona en la capa de entrada. Se utilizan dos arquitecturas neuronales dependiendo del número de dimensión de la función.

Para el caso de dos dimensiones, la red neuronal consta de tres capas con las siguientes características: la capa uno consta de una sola neurona y una función de activación sigmoide, la capa dos consta de 10 neuronas y una función de activación sigmoide, la capa tres consta de una sola neurona y una función de activación lineal.

Para el caso de once dimensiones: la capa uno consta de 10 neuronas y una función de activación sigmoide, la capa dos consta de 100 neuronas y una función de activación sigmoide, la capa tres consta de una sola neurona y una función de activación lineal.

Para evaluar el desempeño de las redes neuronales, se calcula el error cuadrático medio (MSE por sus siglas en inglés), el cual compara la diferencia entre la salida deseada o target y la salida obtenida por la red. La fórmula para el cálculo del error cuadrático medio se muestra en la ecuación 4:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (x_i - y_i)^2. \tag{4}$$

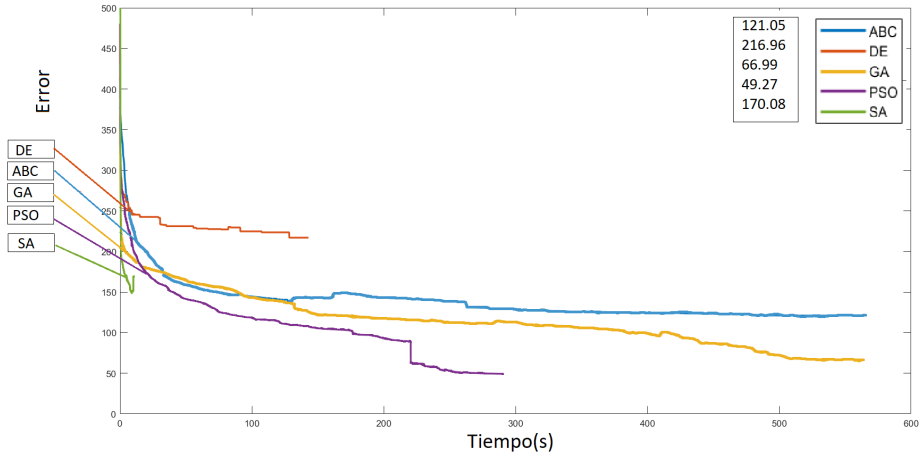


Fig. 4. Comparación de error en el entrenamiento para la función Griewank en 11 dimensiones.

4. Experimentos y resultados

En esta sección se presenta los resultados obtenidos al evaluar cada uno de los algoritmos metaheurísticos con las tres funciones de prueba. Cada función fue ejecutada en 20 ocasiones y se registraron de ellas los valores de error cuadrático medio, considerando el valor mínimo, máximo y promedio de las 20 ejecuciones. Se han recopilado 60 registros por cada sistema metaheurístico al evaluar las tres funciones.

Los resultados para las funciones Sphere, Rastrigin y Griewank en dos dimensiones se pueden ver en las Tablas 2, 3 y 4 respectivamente. Para el caso de once dimensiones se pueden ver en las Tablas 5, 6 y 7. Es importante mencionar que los resultados mostrados son únicamente del entrenamiento, los resultados con datos de prueba se muestran más adelante en las Tablas 8, 9 y 10.

Como se mencionó anteriormente, las tablas [3, 8] muestran los datos tomados del error cuadrático medio para el entrenamiento. Para la fase de prueba, se generaron nuevos datos aleatoriamente. Para este caso, se prueba únicamente con las funciones en 11 dimensiones pues fueron las más complejas para aproximar.

El vector de entrada de prueba es de 10x2500 datos; y se generaron aleatoriamente en el mismo rango del de los datos de entrenamiento. Para esta fase de prueba se tomaron las mejores configuraciones de pesos de las pruebas de entrenamiento con cada algoritmo.

Con el propósito de facilitar una comparación visual, las Figuras 2, 3 y 4 muestran una representación del comportamiento del error promedio de entrenamiento para cada método. Es preciso destacar que la variación en el tamaño de las gráficas mostradas es consecuencia de la divergencia en el número de épocas necesarias para alcanzar la convergencia en cada método. Sin embargo, el tiempo de ejecución de los métodos no es dependiente de este factor, pues cada método tarda una cantidad diferente de tiempo por época.

5. Conclusiones y trabajo futuro

Este estudio demostró que las metaheurísticas pueden ser una alternativa al descenso de gradiente para entrenar redes neuronales, aunque cada metaheurística tiene ventajas y desventajas. Se recopilaron y almacenaron los resultados para crear una base de conocimiento que ayude a seleccionar el tipo de red neuronal y el algoritmo metaheurístico más apropiado para una tarea específica.

En la Tabla 9 se muestra que en varios casos el error en los datos de prueba fue menor que en los de entrenamiento, lo que indica una buena generalización de las redes neuronales. Además, la gran diferencia en el error cuadrático medio en la función Rastrigin en comparación con las funciones Sphere y Griewank se debe a las diferencias en el dominio de las funciones, lo que afecta el desempeño de los algoritmos de optimización y, a su vez, el error de la red neuronal. Es crucial considerar este factor al evaluar el desempeño de las metaheurísticas en diferentes problemas de optimización.

Referencias

1. Hagan, M. T., Demuth, H. B., Beale, M.: Neural network design. PWS Publishing Co (1997)
2. Rafiq, M. Y., Bugmann, G., Easterbrook, D. J.: Neural network design for engineering applications. Structures, vol. 79, no. 17, pp. 1541–1552 (2001) doi: 10.1016/s0045-7949 (01)00039-6
3. Ruder, S.: An overview of gradient descent optimization algorithms (2016) doi: 10.48550/arXiv.1609.04747
4. Zhang, Z.: Improved adam optimizer for deep neural networks. In: IEEE/ACM 26th International Symposium on Quality of Service, pp. 1–2 (2018) doi: 10.1109/iwqos. 2018.8624183
5. Sankararaman, K. A., De, S., Xu, Z., Huang, W. R., Goldstein, T.: The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In: International conference on machine learning, pp. 8469–8479 (2020) doi: 10.48550/ARXIV.1904.06963
6. Dogo, E. M., Afolabi, O. J., Nwulu, N. I., Twala, B., Aigbavboa, C. O.: A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In: International Conference on Computational Techniques, Electronics and Mechanical Systems, pp. 92–99 (2018) doi: 10.1109/CTEMS.2018.8769211
7. Talbi, G.: Metaheuristics: From design to implementation. John Wiley and Sons (2009)
8. Kennedy, J., Eberhart, R. C.: A discrete binary version of the particle swarm algorithm. In: IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, vol.5, pp. 4104–4108 (1997) doi: 10.1109/ICSMC.1997.637339
9. Mitchell, M.: Genetic algorithms: An overview. Complex, vol. 1, no. 1, pp. 31–39 (1995)
10. Karaboga, D., Basturk, B.: Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. Lecture Notes in Computer Science, pp. 789–798 (2007) doi: 10.1007/978-3-540-72950-177
11. Storn, R., Kenneth, P.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, vol. 11, no. 4, pp. 341–359 (1997) doi: 10.1023/a:1008202821328
12. Chong, H. Y., Yap, H. J., Tan, S. C., Yap, K. S., Wong, S. Y.: Advances of metaheuristic algorithms in training neural networks for industrial applications. Soft Computing, vol. 25, no. 16, pp. 11209–11233 (2021)doi: 10.1007/s00500-021-05886-z

13. Carvalho, M., Ludermir, T. B.: Particle swarm optimization of neural network architectures and weights. In: 7th International Conference on Hybrid Intelligent Systems, pp. 336–339 (2007) doi: 10.1109/his.2007.45
14. Aljarah, I., Faris, H., Mirjalili, S.: Optimizing connection weights in neural networks using the whale optimization algorithm. Soft Computing, vol. 22, no. 1, pp. 1–15 (2016) doi: 10.1007/s00500-016-2442-1
15. Si, T., Bagchi, J., Miranda, P. B. C.: Artificial neural network training using metaheuristics for medical data classification: An experimental study. Expert Systems with Applications, vol. 193, pp. 116423 (2022) doi: 10.1016/j.eswa.2021.116423
16. Yang, S., Ting, T. O., Man, K. L., Guan, S. U.: Investigation of neural networks for function approximation. Procedia Computer Science, vol. 17, pp. 586–594 (2013) doi: 10.1016/j.procs.2013.05.076

ISSN 1870-4069

Configuración de hiperparámetros mediante algoritmos de optimización: Aplicación en la predicción de enfermedades cardiovasculares

Eduardo Sánchez-Jiménez, Yasmín Hernández,
Javier Ortiz-Hernández, Alicia Martínez-Rebollar,
Hugo Estrada-Esquivel

Centro Nacional de Investigación y Desarrollo Tecnológico,
Morelos,
México

{m22ce005, yasmin.hp, javier.oh, alicia.mr,
hugo.ee}@cenidet.tecnm.mx

Resumen. El desarrollo de modelos predictivos involucra la selección de un algoritmo de aprendizaje automático, los cuales tienen un conjunto de hiperparámetros que controlan su entrenamiento y desempeño. Los valores adecuados para los hiperparámetros dependen de la naturaleza de los datos, por lo que es necesario definirlos para cada modelo de aprendizaje que se construya. Se plantean las técnicas de optimización como mecanismos de configuración de hiperparámetros para buscar una estructura eficiente para el entrenamiento de los algoritmos de aprendizaje. Este artículo presenta la aplicación de las técnicas *Grid Search, Random Search* y *Bayesian Optimization* para configurar los hiperparámetros de los algoritmos *Random Forest, Support Vector Machine* y *XGBoost.* Se busca construir modelos de predicción de enfermedades cardiovasculares, por lo que se utilizó el conjunto de datos *Cleveland* de la Universidad de California en Irvine. Se identificó que el modelo *Random Forest* logró el mejor rendimiento al obtener un 90% en la métrica *Precision*, utilizando el optimizador bayesiano.

Palabras clave: Ajuste de hiperparámetros, enfermedades cardiovasculares, técnicas de optimización.

Hyperparameter Configuration by Optimization Algorithms: Application in the Prediction of Cardiovascular Disease

Abstract. The development of predictive models involves the selection of a machine learning algorithm, which has a set of hyperparameters that controls its training and performance. The appropriate values for the hyperparameters depend on the nature of the data, so it is necessary to define them for each learning model to be built. Optimization techniques are proposed as a hyperparameter configuration mechanism to find an efficient structure for the training of learning algorithms. This paper presents the application of Grid Search, Random Search and Bayesian Optimization techniques to configure the hyperparameters of

Random Forest, Support Vector Machine and XGBoost algorithms. The aim is to build efficient prediction models for cardiovascular disease, so we used the Cleveland dataset from the University of California, Irvine. It was identified that the *Random Forest* model achieved the best performance by obtaining 90% in the *Precision* metric, using the Bayesian optimizer.

Keywords: Hyperparameter tuning, cardiovascular disease, optimization techniques.

1. Introducción

La predicción de enfermedades cardiovasculares (EC) es un área de interés en la medicina debido a su alta prevalencia y mortalidad en todo el mundo. Los modelos de aprendizaje automático se han implementado con éxito en la predicción de EC, ya que pueden considerar múltiples factores de riesgo y proporcionar una evaluación precisa del riesgo cardiovascular [1].

El diseño de modelos predictivos [2, 5] se considera un proceso largo y complejo que implica seleccionar el mejor algoritmo y determinar la configuración de los hiperparámetros [6]. Los hiperparámetros son importantes porque controlan el comportamiento del algoritmo en la fase de entrenamiento lo que genera un impacto significativo en el rendimiento de los modelos.

Por lo general, en la construcción de modelos predictivos se omite el ajuste de valores para los hiperparámetros, lo que conlleva al uso de los algoritmos bajo una configuración predeterminada, es decir, utilizar los valores establecidos por los desarrolladores de las bibliotecas de aprendizaje automático [1].En algunos casos, porque no se tiene conocimiento del impacto que tienen los hiperparámetros en el desempeño de los modelos, o porque no se cuentan con mecanismos de configuración para entrenar los algoritmos.

Entre las técnicas de optimización más comunes se encuentran la *Grid Search* (GS), *Random Search* (RS) y *Bayesian Optimization* (BO). En GS se realiza la exploración sistemática y se evalúan todas las combinaciones posibles de valores de hiperparámetros [2]. Por otro lado, RS, selecciona combinaciones de hiperparámetros al azar dentro de un rango predefinido [3]. BO utiliza una función de probabilidad para encontrar la mejor combinación de hiperparámetros mediante la exploración guiada de la superficie de la función objetivo [4].

En este artículo se construyen modelos para predecir EC utilizando datos de fuentes públicas, en particular el conjunto de datos *Cleveland* del repositorio de la UCI [5]. Se busca mejorar el desempeño de los modelos predictivos a partir de la configuración de los hiperparámetros mediante un enfoque de optimización.

Sin embargo, primero es importante identificar la variedad de hiperparámetros con los que cuentan los algoritmos de aprendizaje automático. Posteriormente, entender los fundamentos de la optimización en el aprendizaje automático para tener conocimiento del comportamiento general de las técnicas de optimización y poder plantear un espacio de búsqueda de acuerdo con los algoritmos que se desean modelar.

El resto del artículo se organiza de la siguiente manera: en la sección 2 se describen los fundamentos de la optimización y los elementos que conforman dicho proceso, se presentan los hiperparámetros de los algoritmos *Random Forest* (RF) [6], *Support*

Vector Machine (SVM) [7] y *Extreme Gradient Boosting* (XGBoost) [8] y algunas investigaciones relacionadas.

En la sección 3 se muestran los resultados del análisis exploratorio y preprocesamiento del conjunto de datos *Cleveland*.

En la sección 4 se presentan los resultados de los experimentos de optimización y se brinda un análisis de los resultados para identificar los modelos que presentan mayor porcentaje de precisión en la predicción de personas con EC. Finalmente, la sección 5 se exponen las conclusiones y se propone algunas guías para el trabajo futuro.

2. Antecedentes y trabajo relacionado

2.1. Optimización matemática

Un problema de optimización consiste en la búsqueda de valores para determinadas variables, comúnmente llamadas variables de decisión, de forma que, cumpliendo un conjunto de restricciones proporcionan el mejor valor posible para una función objetivo, que utiliza para medir el rendimiento del sistema que se estudia. Un problema de optimización con restricciones puede expresarse como [9]:

$$\min/\max_{\mathrm{x}} f(x), \tag{1}$$

con respecto a:

$$g_i(x) \leq 0, i = 1,2,\dots,m, \tag{2}$$

$$h_i(x) = 0, j = 1,2,\dots,p. \;\; x \in X, \tag{3}$$

donde $g_i(x) \leq 0$ es la función de restricción de desigualdad; $h_i(x) = 0$ es la función de restricción de igualdad y X es el dominio de la variable de decisión x. El objetivo de las restricciones es limitar los posibles valores de la solución óptima a ciertas áreas del espacio de búsqueda, denominadas región factible. Así, la región factible R_f de x puede ser representada como:

$$R_f = \{x \in X \mid g_i(x) \leq 0, h_i(x) = 0\}. \tag{4}$$

2.2. Algoritmos de aprendizaje automático supervisado

El objetivo del aprendizaje supervisado es obtener una función óptima en el modelo predictivo f^* para minimizar la función objetivo $\mathcal{L}(f(x), y)$ que calcula el error entre las etiquetas de las predicciones estimadas $f(x)$ y las etiquetas verdaderas y. El modelo predictivo óptimo f^* puede obtenerse mediante [10]:

$$f^* = \arg\min \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y), \tag{5}$$

donde n es el número de datos de entrenamiento, x_i es el vector de características de la i-ésima instancia, y_i es la salida real correspondiente, y $\mathcal{L}$ es el valor de la función objetivo de cada muestra.

A continuación, se describen los algoritmos que fueron considerados para el desarrollo de esta investigación. En [3], 11] se mencionan los hiperparámetros que tienen un mayor impacto en la fase de entrenamiento. En la Tabla 1 se describen los principales hiperparámetros, etiquetados con base en la biblioteca de aprendizaje automático s*cikit-learn* [12].

2.2.1.Random forest

RF [6] es un algoritmo que ensambla una colección de árboles entrenados con un subconjunto de características y un subconjunto de datos tomados aleatoriamente. Cada árbol puede hacer un trabajo de predicción relativamente bueno, pero es probable que exista un sobreajuste en parte de los datos. Al construir muchos árboles, todos los cuales funcionan bien y se ajustan de diferentes maneras, podemos reducir la cantidad de sobreajuste promediando sus resultados [13].

2.2.2. Support vector machine

El objetivo de SVM es encontrar un hiperplano de separación entre las clases cuando estas no pueden separarse linealmente. El hiperplano de separación es un plano que divide el espacio de características en dos partes, una para cada clase. La distancia entre el hiperplano y los puntos a la frontera con cada clase (vectores de soporte) se llama margen, y el mejor hiperplano es aquel que presenta mayor margen [7].

2.2.3. Extreme gradient boosting

XGBoost [8] genera modelos secuenciales y utiliza la técnica de *boosting* para combinar modelos simples y menos precisos en modelos que mejoren la precisión de los casos que no se han predicho correctamente hasta ese momento. El proceso de ajuste de cada árbol se realiza utilizando la técnica de gradiente descendente estocástico (GDE), que es un método de optimización utilizado en el aprendizaje profundo.

2.3. Proceso de optimización de hiperparámetros

Los mecanismos de optimización de hiperparámetros que cuentan con cuatro elementos principales, según lo establece [3]: i) un regresor o un clasificador, ii) un espacio de búsqueda o configuración de los hiperparámetros, iii) el método de optimización para buscador el mejor modelo y iv) función objetivo (métrica de evaluación) para medir y comparar el rendimiento de las diferentes configuraciones de hiperparámetros. Para problemas de OH, el objetivo de este trabajo es obtener:

$$x^* = \arg\max_{x \in X} f(x). \tag{6}$$

x^* es la configuración de hiperparámetros que produce el valor óptimo de $f(x)$; y x es un hiperparámetro que puede tomar cualquier valor en el espacio de búsqueda *X*. La función objetivo $f(x)$ determina lo bien que se comporta un modelo de aprendizaje.

Esta función compara las predicciones del modelo con la clase real para determinar el nivel de error del algoritmo.

Tabla 1 Hiperparámetros de los algoritmos RF, SVM, XGBoost.

Algoritmo	Hiperparámetro	Descripción
RF	*n_estimators*	Número de árboles considerados para el modelado
	criterion	Mecanismo para medir la calidad de una división
	max_features	Número de características que se utilizan para construir cada árbol de decisión en el bosque
	max_depth	Profundidad máxima del árbol
	min_samples_split	Número mínimo de muestras que se requieren para dividir un nodo interno en dos subnodos
	min_samples_leaf	Número mínimo de muestras para formar una hoja
SVM	*c*	Margen entre los vectores de soporte y el hiperplano. Un valor de *C* bajo permitirá errores en la clasificación, por otro lado, un valor de *C* más alto buscará un margen de separación más robusto
	kernel	Función matemática que mapea los datos de entrada a un espacio de características de mayor dimensión (*linear, rfb, poly, sigmoid*)
	degree	Se utiliza con el *kernel poly*. El grado determina la complejidad de la función de decisión y, por lo tanto, la capacidad del modelo para ajustarse a los datos
XGBoost	*n_estimators*	Número de árboles a utilizar
	learning_rate	Controla la velocidad de aprendizaje del modelo
	gamma	Controla la complejidad del modelo y evitar el sobreajuste. El valor óptimo de gamma depende de los datos
	subsample	Fracción de observaciones que deben ser muestras aleatorias para cada árbol
	max_depth	Profundidad máxima de los árboles
	colsample_bytree	Fracción de características que se seleccionan al azar en cada árbol

2.3.1. Algoritmos de optimización de hiperparámetros

Los métodos de optimización evalúan de forma eficientes las combinaciones de valores de los hiperparámetros asociados a un algoritmo de aprendizaje automático. Esto permite a los usuarios no expertos desarrollar modelos bien estructurados [14].

La técnica GS [2] funciona evaluando el producto cartesiano de un conjunto finito de valores del espacio de búsqueda definido por el usuario. Es fácil ejecutar GS en paralelo porque cada ensayo se ejecuta individualmente y el resultado es independiente de los de otros ensayos.

Sin embargo, GS sufre la maldición de la dimensionalidad porque el consumo de recursos informáticos aumenta exponencialmente cuando hay hiperparámetros.

Además, un rango de muestreo limitado es aceptable para GS porque no es deseable que haya demasiadas configuraciones.

Por otra parte, RS [3] en lugar de probar todos los valores en el espacio de búsqueda, como es el caso de GS, RS toma aleatoriamente un número predefinido de valores de los límites superior e inferior del espacio de búsqueda como valores de hiperparámetros candidatos, y luego entrena el algoritmo hasta agotar el presupuesto definido. La base

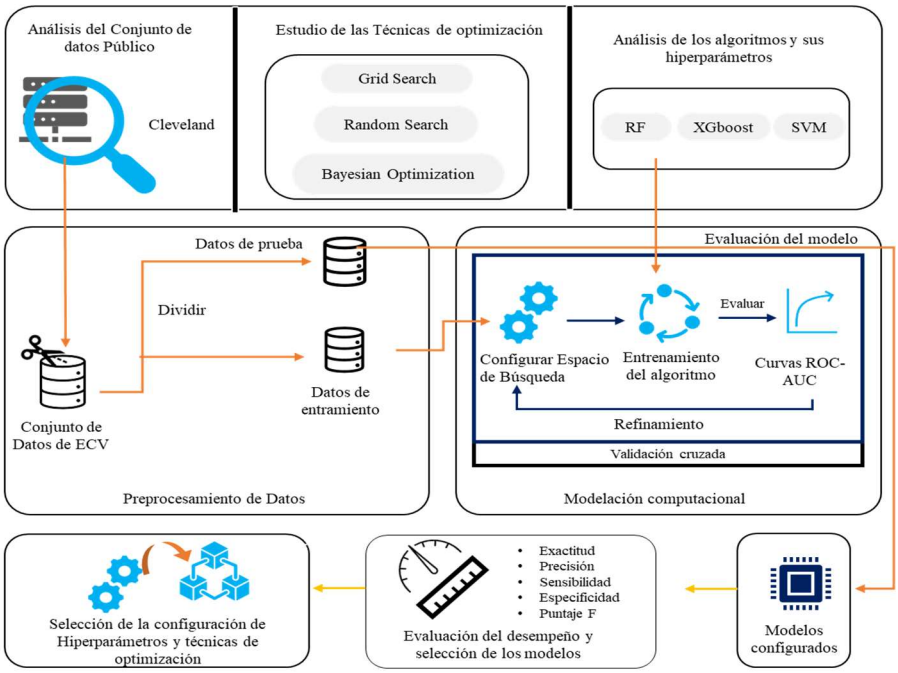


Fig. 1. Metodología de ajuste de Opti.

teórica de RS es que, si el espacio de configuración es lo suficientemente grande, posiblemente se detecte la combinación de hiperparámetros con un óptimo global.

BO [15] utiliza la teoría de la probabilidad bayesiana para encontrar los mejores hiperparámetros de manera más eficiente. BO comienza con la definición de una función objetivo que se desea optimizar, en este caso la función de probabilidad del modelo. A medida que se ejecutan los experimentos, se actualiza la distribución de probabilidad posterior para reflejar los resultados observados. El objetivo es encontrar el conjunto de hiperparámetros que maximice la función objetivo dada la distribución de probabilidad posterior [16].

2.4. Trabajo relacionado

En [17] mostraron que la configuración de los hiperparámetros es a menudo más importante que la elección del algoritmo de aprendizaje automático. Además, en [18] estudiaron la importancia de ajustar determinados hiperparámetros, más que la importancia de afinarlos en su totalidad.

Trabajos como: [3, 19, 21] formularon el proceso de configuración de los hiperparámetros como un problema de optimización, para encontrar los valores de los hiperparámetros que maximicen la tasa de predicciones positivas con respecto a las métricas que evalúan el rendimiento de algoritmos.

La selección de la función objetivo es el primer paso en los métodos de aprendizaje automático. Con la función objetivo determinada, se suelen utilizar métodos exactos y heurísticos para resolver el problema [22].

Tabla 2. Descripción del conjunto de datos *Cleveland.*

Variables	Descripción	Rango
Age	Edad del paciente en años	29-77
Sex	Sexo del paciente	0: hombre, 1: mujer
Cp	Dolor torácico	1: AT, 2: AA, 3: DNA, 4: A
Trestbps	Prensión arterial en reposo	94-200
Chol	Colesterol sérico	126-564
Fbs	Azúcar en sangre en ayunas > 120 mg/dl	0: No, 1: Si
Restecg	Resultados del electrocardiograma	0: normal, 1: anomalía-onda ST-T, 2: hipertrofia del LV
Thalach	Frecuencia cardíaca alcanzada	71-202
Exang	Angina inducida por el ejercicio	0: No, 1: Si
Oldpeak	Depresión del segmento ST	0.0-62.0
Slope	Pendiente del segmento ST de ejercicio máximo	1: pendiente-ascendente, 2: grasa, 3: pendiente-descendente
Ca	Número de vasos mayores coloreados por fuoroscopia	0-3
Thalassemia	Trastorno sanguíneo hereditario que hace que tu cuerpo tenga menos hemoglobina de lo normal	3: normal, 6: defecto-fijo, 7: defecto irreversible
HeartDisease	Variable objetivo	0: ausencia d, 1: presencia

En [23] se optimizaron dos algoritmos SVM para la predicción de EC. El primer modelo es lineal y fue regularizado con la técnica *L1*, mientras que el segundo fue regularizado con la técnica *L2* y utilizaron diferentes *kernels*, incluyendo el lineal y el de la función de base radial (rbf).

Por otro lado, en [24] se estableció un sistema optimizado basado en el algoritmo *XGBoost* para la predicción de EC. Para obtener la solución óptima de forma sistemática y eficaz se aplicó la técnica BO para configurar los hiperparámetros al algoritmo. En [25] se propuso un marco de trabajo basado principalmente en los algoritmos k-Nearest Neighbors (kNN), SVM y RF entrenados con el conjunto de datos de EC *Cleveland*.

3. Predicción de enfermedades cardiovasculares

En esta sección se muestra la metodología de OH aplicada (ver la Fig. 1). Se analiza el conjunto de datos *Cleveland*, las técnicas de optimización GS, RS y BO, y se evalúan los algoritmos de aprendizaje automático RF, SVM y XGBoost para identificar los hiperparámetros más relevantes (ver Tabla 1) con el objeto de construir un modelo de aprendizaje para apoyo en la detección de EC. Posteriormente, se lleva a cabo el

Tabla 3. Estadísticos de las variables cuantitativas del conjunto de datos *Cleveland.*

Variables	Min	Max	Media	Mediana	Moda	Desv. Est.	Varianza
Age	29	77	54.43	56	58	9.039	81.69
Trestbps	94	200	131.68	130	120	17.60	309.75
Chol	126	564	246.69	241	197	51.77	2680.84
Thalach	71	202	149.60	153	162	22.87	523.26
Oldpeak	0	6.2	1.039	0.80	0	1.161	1.34

preprocesamiento del conjunto de datos, que incluye el dividirlo en el conjunto de entrenamiento y prueba. La intención de contar con registros de prueba es para de evaluar a los modelos con registros que desconoce y medir el rendimiento a partir de las distintas métricas.

La fase de entrenamiento de cada algoritmo se realizó con un proceso de validación cruzada de 10 particiones. El proceso de modelado computacional se presenta como un proceso iterativo debido a que la búsqueda de los valores óptimos para los hiperparámetros de un algoritmo de aprendizaje puede requerir múltiples iteraciones.

Durante este proceso, se entrenan los algoritmos con diferentes combinaciones de valores de hiperparámetros, y para evaluar el rendimiento del modelo se analiza su curva ROC para determinar su capacidad para distinguir entre las diferentes clases de datos.

Este proceso se repite hasta encontrar una combinación de hiperparámetros que produzca el mejor rendimiento posible en el conjunto de datos dado. Finalmente, se aplican las métricas *Accuracy, Precision, Recall, Specificity y F-Score* para seleccionar los modelos con las combinaciones de hiperparámetros que proporcionan mayor rendimiento.

3.1 Descripción del conjunto de datos

El conjunto de datos fue recopilado por el *Cleveland Heart Institute*. Contiene información (demográfica, antecedentes médicos, signos vitales) de 303 personas a los que se le aplicaron una serie de pruebas de diagnóstico; de ellos, 164 presentan EC, el resto (139) tienen ausencia de EC.

En relación con el sexo de los pacientes, el 32% son pacientes mujeres y 68% pacientes hombres. En la Tabla 2 se describen los 14 atributos que contiene el conjunto de datos *Cleveland* y el intervalo de valores que presenta cada variable.

3.2 Análisis de correlación

La correlación se define como una medida estadística que indica la fuerza y la dirección de la relación lineal entre dos variables [26]. Conocer la correlación entre variables es importante porque puede proporcionar información valiosa sobre la relación entre dos o más variables, Para el conjunto de datos *Cleveland* se aplicó la

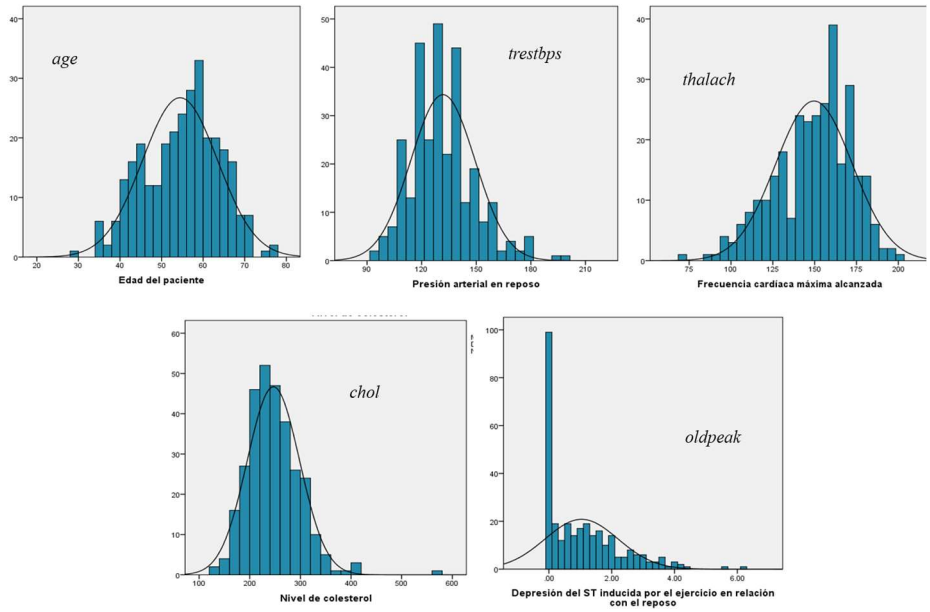


Fig. 2. Histogramas y curva de normalidad de las variables cuantitativas.

correlación de *Spearman* ya que esta técnica es menos sensible a los valores atípicos presentes y es más adecuada para variables que no presentan normalidad en su distribución.

Los pares de variables que presentan mayor relación son: *oldpeak* - *thalach*, *thalach* - *age* y *trestbp* - *age*. Las dos primeras correlaciones son negativas y débiles, lo que indica que a medida que una variable disminuye, la otra variable también. Por otro lado, la última relación es positiva débil, lo que nos hace inferir que a medida que aumenta la edad en las personas también su tensión arterial incrementa.

3.3 Preprocesamiento de datos

El análisis exploratorio de los datos incluye la visualización de los datos a través de histogramas y diagramas de caja y bigotes. El análisis exploratorio de datos es una parte importante del proceso de preprocesamiento de datos porque ayuda a garantizar que los datos estén limpios, sean coherentes y estén listos para su uso en el análisis posterior [27]. En la Tabla 3 se describen los estadísticos descriptivos de las variables cuantitativas del conjunto de datos *Cleveland*.

AT: Angina Típica, AA: Angina Atípica, DNA: Dolor no Anginoso, A: Asintomático

En la Fig. 2 se presentan los histogramas de frecuencias de las variables cuantitativas. Todas las variables presentan un pequeño sesgo en su distribución, por ejemplo, la variable *chol, oldpeak y trestbps* tienen un sesgo a la izquierda, es decir, tienen una cola asimétrica extendida hacia los valores positivos. Además, conforme el

Tabla 4. Matriz de confusión de los algoritmos RF, SVM y XGBoost.

Técnica	Iter.	Algoritmo	VP	FP	FN	VN	VP+VN
Configuración predeterminada	1	RF	35	7	8	41	76
	1	SVM	37	7	6	41	78
	1	XGBoost	37	9	6	39	76
GS	Todas	RF	36	8	7	40	76
	Todas	SVM	37	7	6	41	78
	Todas	XGBoost	37	9	6	39	76
RS	100	RF	36	5	7	43	79
	100	SVM	34	4	9	44	78
	100	XGBoost	36	7	7	41	77
BO	30	RF	36	4	7	44	80
	30	SVM	37	5	6	43	80
	30	XGBoost	37	6	7	41	78

polígono de frecuencia que se muestra en cada histograma podemos inferir que las distribuciones de cada variable no provienen totalmente a una distribución normal.

Otro tema muy importante en el análisis exploratorio de los datos es poder identificar los datos que se encuentran fuera del rango normal o esperado de los datos, llamados datos atípicos (*outliers*). Las variables con *outliers* son: *Trestbps, Chol, Thalach* y *Oldpeak*.

Para tratar este tipo de datos se consideró un mecanismo que los reemplaza por los valores más cercanos dentro de un rango definido. Para este caso, en cada atributo se consideró como *outliers* el 1% como extremo inferior y el 3% extremo superior de la distribución.

Existen numerosos algoritmos de aprendizaje automático que necesitan que tanto las variables de entrada como de salida se expresen en términos numéricos, ya que no pueden procesar directamente los datos de las etiquetas en su forma original [28]. El conjunto de datos *Cleveland* presenta variables categóricas las cuales fueron transformadas a partir del método *OneHot Encoding* [24].

El método consiste en crear una columna para cada valor único en la variable categórica original y asignar un valor binario de 0 o 1 a cada columna, según la pertenencia del registro a esa categoría. Las variables que se procesaron mediante la técnica fueron: *Cp, Restecg, Slope, Ca* y *Thalassemia*.

Además, cuando las variables tienen distintas escalas, es importante tener en cuenta que esto puede afectar la interpretación de los análisis y los modelos estadísticos. Es posible que las variables con mayores escalas tengan una influencia desproporcionada en los resultados en comparación con las variables con escalas más pequeñas.

Se ha identificado que los datos de dominios médicos son en su mayoría discretos, por lo que estandarizarlos es esencial para hacer converger las características de estos. En este trabajo se consideró un mecanismo de estandarización robusta para generar una nueva escala. Es uno de los métodos más populares para definir una única escala para todas las características utilizando los percentiles 25 y 75 de la distribución de datos para cada característica.

Tabla 5. Métricas de desempeño de los algoritmos RF, SVM y XGBoost.

Técnica.	**Iter.**	**Algoritmo**	***Acc. Train***	***Acc. Test***	***Precision***	***Recall***	***Specifity***	***F-score***
Configuración predeterminada	1	RF	81.14	83.52	83.33	81.40	85.42	82.35
	1	SVM	84.50	85.71	84.09	86.05	85.42	85.06
	1	XGBoost	80.62	83.52	80.43	86.05	81.25	83.15
GS	Todas	RF	84.91	83.52	81.82	83.72	83.33	82.76
	Todas	SVM	84.91	85.71	84.09	86.05	85.42	85.06
	Todas	XGBoost	84.43	83.52	80.43	86.05	81.25	83.15
RS	100	RF	87.99	86.81	87.80	83.72	89.58	85.71
	100	SVM	88.39	85.71	89.47	79.07	91.67	83.95
	100	XGBoost	88.78	84.62	83.72	83.72	85.42	83.72
BO	30	RF	83.93	**87.91**	90.00	83.72	91.67	86.75
	30	SVM	83.09	**87.91**	88.10	86.05	89.58	87.06
	30	XGBoost	84.02	**85.71**	86.05	84.09	87.23	85.06

4. Resultados y discusión

En esta sección se exponen los resultados tras aplicar las técnicas de optimización. Los experimentos se llevaron a cabo mediante el uso de diversas bibliotecas de aprendizaje automático, tales como *scikit-learn* y *optunity*, junto con el lenguaje de programación *Python*. Asimismo, cabe destacar que la experimentación fue realizada en una computadora HP-Victus con un procesador Core™ i5-12450H de 2.00 GHz, 8 GB de memoria RAM y con sistema operativo Windows 11.

Como primer paso, se procedió a dividir el conjunto de datos *Cleveland* en dos subconjuntos. El 70% de las muestras (aproximadamente 212 registros) se utilizaron para la fase de entrenamiento de los tres algoritmos, utilizando una validación cruzada de 10 *folds*. Además, se definió un número máximo de iteraciones o combinaciones que se evaluarán por parte de las técnicas RS y BO.

Finalmente, se empleó el 30% de las muestras (cerca de 91 registros) para la etapa de prueba de los modelos previamente entrenados. En la Tabla 4 se muestra el total de registros clasificados como: verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN) de los modelos que mostraron mayor rendimiento en relación con las técnicas de tres técnicas de OH. Además, se muestra el total de registros clasificados correctamente.

En la Tabla 5 se presentan los resultados de los modelos al evaluar su desempeño mediante las métricas *Accuracy, Precision, Recall, Specificity* y *F-Score*, evaluados con el conjunto de prueba.

Se observa que la técnica de optimización BO genera los modelos RF, SVM y XGBoost que presentan mayor desempeño, en relación con la métrica *Accuracy*. En términos generales estos modelos han logrado predecir correctamente la mayoría de las muestras del conjunto de datos de prueba en relación con el conjunto total de muestras de prueba.

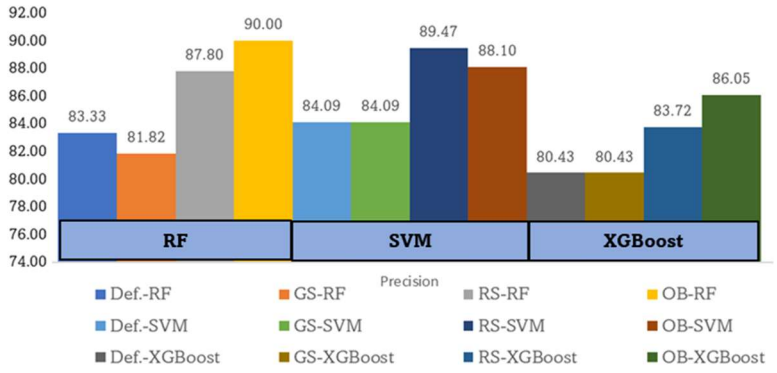


Fig. 3. Comparación de la métrica *precision* de modelos resultantes.

Para identificar el modelo con mayor capacidad para predecir a las personas que padecen EC, es importante enfocarnos en la métrica de *precision*. En la Fig. 3 se puede apreciar que en términos generales los tres modelos (RF, SVM, XGBoost) presentaron mayor rendimiento en diversos procedimientos de ajuste de hiperparámetros con respecto a la configuración de los hiperparámetros por defecto. Utilizando la técnica de optimización BO el modelo RF alcanzó el mayor rendimiento (90%).

4.1 Configuraciones de hiperparámetros identificadas con mejor desempeño

La Tabla 6 muestra para cada algoritmo las configuraciones de hiperparámetros que se obtuvieron a partir de las técnicas de OH. Estas configuraciones dan al modelo el mayor desempeño en relación con el espacio de búsqueda propuesto. Se observa que algunos valores de los hiperparámetros varían en cada técnica de OH.

Se ha realizado un comparativo entre los resultados de los modelos propuesto en este trabajo con algunos modelos reportados en la literatura. El desempeño de los modelos configurados mediante la técnica BO supera a los modelos propuestos en [27]. Además, el rendimiento obtenido por los modelos RF-RS y RF-BO es superior al de [28], donde el modelo RF obtiene un 83% en la métrica *Accuracy* en la fase de entrenamiento.

5. Conclusiones y trabajo a futuro

En este trabajo se construyeron modelos de predicción de EC a partir de un enfoque experimental que optimiza los hiperparámetros y mejora el desempeño de los modelos. Como parte del proceso se describieron los hiperparámetros más importantes de los algoritmos RF, SVM y XGBoost que tienen un impacto directo en la fase de entrenamiento. Así mismo planteamos el enfoque de OH y describimos las características de las técnicas de optimización GS, RS y BO.

El principal reto de este trabajo fue definir los mecanismos de entrenamiento para los tres algoritmos de aprendizaje automático mediante las técnicas de OH. Además, se consideró un mecanismo adicional que presenta a los modelos con la configuración predeterminada de sus hiperparámetros. Este mecanismo sirvió de referencia para el hacer el contraste de la precisión de cada modelo cuando se entrenan a partir de técnicas

Tabla 6. Configuraciones de hiperparámetros con mejor desempeño.

Algoritmo	Hiperparámetro	GS	RS	BO
RF	*n_estimators*	50	52	30
	max_features	0.1	1.0	0.2
	max_depth	5	5	9
	min_samples_split	2	7	7
	min_samples_leaf	6	5	4
	bootstrap	False	True	True
SVM	*c*	2	40	14
	gamma	0.1	0.01	0.001
	kernel	rfb	poly	sigmoid
	degree	1	2	4
XGBoost	*n_estimators*	45	55	65
	learning_rate	0.1	0.06	0.15
	gamma	1	6.4	0.4
	subsample	0.8	0.5	0.8
	max_depth	5	5	7

de optimización. Como resultado de los experimentos se identificó que los modelos RF y SVM optimizados con las técnicas BO y RS, respectivamente, presentan un mayor desempeño. La búsqueda de los valores de cada hiperparámetro estuvo restringido al espacio de búsqueda que se planteó en cada mecanismo experimental.

En el estado del arte se han planteado métodos de optimización heurísticos, incluyendo técnicas basadas en inteligencia de enjambre y algoritmos evolutivos. Como trabajo a futuro se plantearán el desarrollo de mecanismos de configuración de hiperparámetros considerando estas técnicas.

Referencias

1. Probst, P., Bischl, B., Boulesteix, A.: Tunability: importance of hyperparameters of machine learning algorithms. The Journal of Machine Learning Research, vol. 20, no. 1, pp 1934–1965 (2018) doi: 10.48550/ARXIV.1802.09596
2. Yu, T., Zhu, H.: Hyper-parameter optimization: a review of algorithms and applications (2020) doi: 10.48550/ARXIV.2003.05689
3. Yang, L., Shami, A.: On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, vol. 415, pp. 295–316 (2020) doi: 10.1016/j.neucom.2020.07.061
4. Thornton, C., Hutter, F., Hoos, H. H., Leyton-Brown, K.: Auto-weka. In: Proceedings of the 19th ACM, International Conference on Knowledge Discovery and Data Mining, pp. 847–855 (2013) doi: 10.1145/2487575.2487629
5. Detrano, R.: Enfermedades cardiovasculares. Organización Panamericana de la Salud (2023) www.paho.org/es/temas/enfermedades-cardiovasculares
6. Breiman, L.: Random forests. Machine Learning, vol. 45, no. 1, pp. 5–32 (2001) doi: 10.1023/a:1010933404324
7. Breiman, L.: Support-vector networks. Machine Learning, vol. 45, no. 1, pp. 5–32 (2001) doi: 10.1023/a:1010933404324

8. Chen, T., Guestrin, C.: Xgboost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016) doi: 10.1145/2939672.2939785
9. Paredes, G. E., Rodríguez, A. V.: Aplicaciones de programación no lineal. OmniaScience Scholar (2016) doi: 10.3926/oss.21
10. Gambella, C., Ghaddar, B., Naoum-Sawaya, J.: Optimization problems for machine learning: A survey. European Journal of Operational Research, vol. 290, no. 3, pp. 807–828 (2021) doi: 10.1016/j.ejor.2020.08.045
11. Elgeldawi, E., Sayed, A., Galal, A. R., Zaki, A. M.: Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. Informatics, vol. 8, no. 4, pp. 79 (2021) doi: 10.3390/informatics8040079
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research, vol. 12, no. 85, pp. 2825−2830 (2011)
13. Müller, A. C., Guido, S.: Introduction to machine learning with python: A guide for data scientists. O'Reilly Media (2016)
14. Schmidt, M., Safarani, S., Gastinger, J., Jacobs, T., Nicolas, S., Schulke, A.: On the performance of differential evolution for hyperparameter tuning. In: International Joint Conference on Neural Networks, pp. 1–8 (2019) doi: 10.1109/ijcnn.2019.8851978
15. Injadat, M., Salo, F., Nassif, A. B., Essex, A., Shami, A.: Bayesian optimization with machine learning algorithms towards anomaly detection. In: IEEE Global Communications Conference, pp. 1–6 (2018) doi: 10.1109/glocom.2018.8647714
16. Hazan, E., Klivans, A., Yuan, Y.: Hyperparameter optimization: A spectral approach. In: 6th International Conference on Learning Representations (2017) doi: 10.48550/ARXIV.1706.00764
17. Lavesson, N., Davidsson, P.: Quantifying the impact of learning algorithm parameter tuning. In: Proceedings of the 21st National Conference on Artificial Intelligence, vol. 1, pp. 395–400 (2006)
18. Weerts, H. J. P., Mueller, A. C., Vanschoren, J.: Importance of tuning hyperparameters of machine learning algorithms. Journal of Machine Learning Research, vol. 20, pp. 1–32 (2019) doi: 10.48550/ARXIV.2007.07588
19. Pannakkong, W., Thiwa-Anont, K., Singthong, K., Parthanadee, P., Buddhakulsomsiri, J.: Hyperparameter tuning of machine learning algorithms using response surface methodology: A case study of ANN, SVM and DBN. Mathematical Problems in Engineering, vol. 2022, pp. 1–17 (2022) doi: 10.1155/2022/8513719
20. Khourdifi, Y., Bahaj, M.: Heart disease prediction and classification using machine learning algorithms optimized by particle swarm optimization and ant colony optimization. International Journal of Intelligent Engineering and Systems, vol. 12, no. 1, pp. 242–252 (2019) doi: 10.22266/ijies2019.0228.24
21. Andonie, R.: Hyperparameter optimization in learning systems. Journal of Membrane Computing, vol. 1, no. 4, pp. 279–291 (2019) doi: 10.1007/s41965-019-00023-0
22. Ali, L., Niamat, A., Khan, J. A., Golilarz, N. A., Xingzhong, X., Noor, A., Nour, R., Bukhari, S. A. C.: An optimized stacked support vector machines based expert system for the effective prediction of heart failure. IEEE Access, vol. 7, pp. 54007–54014 (2019) doi: 10.1109/access.2019.2909969
23. Budholiya, K., Shrivastava, S. K., Sharma, V.: An optimized XGBoost based diagnostic system for effective prediction of heart disease. Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 7, pp. 4514–4523 (2022) doi: 10.1016/j.jksuci.2020.10.013

24. Gupta, A., Kumar, R., Arora, H. S., Raman, B.: MIFH: A machine intelligence framework for heart disease diagnosis. IEEE Access, vol. 8, pp. 14659–14674 (2020) doi: 10.1109/access.2019.2962755
25. El-Hashash, E. F., Shiekh, R. H. A.: A comparison of the Pearson, Spearman rank and Kendall tau correlation coefficients using quantitative variables. Asian Journal of Probability and Statistics, pp. 36–48 (2022) doi: 10.9734/ajpas/2022/v20i3425
26. Chang, V., Bhavani, V. R., Xu, A. Q., Hossain, M.: An artificial intelligence model for heart disease detection using machine learning algorithms. Healthcare Analytics, vol. 2, pp. 100016 (2022) doi: 10.1016/j.health.2022.100016
27. Rajeswari, D., Thangavel, K.: The performance of data normalization techniques on heart disease datasets. International Journal of Advanced Research in Engineering and Technology, vol. 11, no. 12, pp. 2350–2357 (2020) doi: 10.34218/IJARET.11.12.2020.222

ISSN 1870-4069

Identificación de procesos de eventos discretos cíclicos temporizados

Marina Montes-Partida, Ernesto López-Mellado

Centro de Investigación y de Estudios Avanzados,
Unidad Guadalajara,
México

{marina.montes, e.lopez}@cinvestav.mx

Resumen. Esta investigación se inscribe en el contexto de la minería de procesos; en particular, en el descubrimiento de procesos de eventos discretos, donde los modelos obtenidos, expresados mediante redes de Petri temporizadas (RPT), son obtenidos a partir de una secuencia S de eventos fechados. El presente artículo propone un método de descubrimiento de procesos temporizados, donde los modelos obtenidos son RPT. La temporización de las transiciones es determinística, la cual se expresa mediante dos parámetros: duración media e intervalo de los tiempos de disparo. La propuesta extiende un método de descubrimiento básico mediante de una técnica de refinamiento estructural y temporal a la RPT *N* descubierta para mejorar su precisión. Se analiza la distribución de las duraciones calculadas en cada transición para determinar si hay más de un grupo (clúster) de duraciones alrededor de un valor; en tal caso se refina la transición en dos o más transiciones según el número de clústeres. Posteriormente, las nuevas transiciones T' se reemplazan en S de acuerdo a su clúster de duraciones y se determina su patrón de ocurrencia; éste se expresa como una red de Petri N' con nuevos lugares, la cual se fusiona con **N** usando T' para obtener la RPT refinada.

Palabras clave: Minería de procesos, descubrimiento de procesos temporizados, refinamiento estructural y temporal, redes de Petri temporizadas.

Discovering Timed Cyclic Discrete Event Processes

Abstract. This research is in the scope of process mining, focusing on discrete event process discovery, where the discovered models are expressed by timed Petri nets (TPN) from a dated event sequence S. This paper presents a timed process discovery method that builds transition-TPN, where the timing is expressed in two deterministic parameters: average and interval firing durations. This proposal extends a former discovery method through a structural and temporal refining technique applied to a discovered TPN *N* to improve its precision. For each transition, the distribution of computed firing delays is analyzed to determine if there is more than one durations cluster; in such a case, the transition is refined into two or more new transitions according to the number of clusters. Afterwards, the set of new transitions T' are replaced in S; then, their occurrence pattern is synthesized as a PN *N'* using new places; the *N'* is merged with N through *T'* to obtain the refined TPN.

Keywords: Process mining, timed process discovery, structural and temporal refining, timed Petri nets.

1. Introducción

La identificación o descubrimiento de procesos es una forma de extracción de conocimiento sobre un proceso, donde el comportamiento observado, en la forma de secuencias de ejecución de eventos, tareas, o actividades, es representado por modelos formales.

Procesos de eventos discretos. En el caso de procesos de eventos discretos, los formalismos describen relaciones estado-eventos; los más usados son los autómatas finitos (AF) y las redes de Petri (RP). En el ámbito de los procesos de flujo de trabajo, las secuencias que describen el comportamiento son capturadas desde el inicio hasta el fin de una ejecución (cases); éstas son en gran cantidad.

En cambio, en el ámbito de los procesos industriales, las secuencias son pocas y muy largas; la ejecución de los procesos (jobs) se registra de manera continua y no se conoce la delimitación de dichos procesos. En ambos contextos los eventos, representados por símbolos, pueden tener información adicional relativa a la actividad, los recursos asignados y el instante de ejecución.

Un tipo de procesos de eventos discretos, es el de los sistemas de manufactura automatizados. Éstos, en general son estructurados como un sistema compuesto de un controlador y una planta interactuando en un ciclo cerrado, los cuales intercambian señales para llevar a cabo la ejecución de las tareas; el controlador envía comandos y la planta informa su estado a través de las señales de sensores.

Identificación/descubrimiento. El comportamiento del proceso puede ser registrado por la observación de las señales intercambiadas cada vez que ocurra un cambio en ellas. A partir de estas observaciones se genera una secuencia de vectores de entrada/salida, las cuales son la entrada a un sistema de descubrimiento o identificación; el sistema convierte la secuencia de señales a una secuencia de eventos la cual se procesa para construir un modelo que expresa el comportamiento del proceso desde el punto de vista del controlador. Este enfoque, ilustrado en la Figura 1, ha sido abordado en [1, 2, 3, 4]. El método de identificación se presenta en dos pasos.

En el primer paso se descubre el comportamiento observable generando los eventos de entrada asociados a las transiciones que producen cambios en los lugares asociados a las salidas (marcado de la red de Petri). Posteriormente, estas transiciones forman una secuencia de eventos S que replican el comportamiento funcionamiento del sistema.

El segundo paso, presentado por [3, 5] descubre el comportamiento no observable a partir de la secuencia S. Primero se determinan las relaciones causales y concurrentes observadas en los eventos, es decir, en el disparo de cada transición en S.

Finalmente, los modelos observable y no observable son fusionados para conseguir el modelo final. En estos métodos el modelo construido no contiene información relativa a las duraciones de las tareas o eventos.

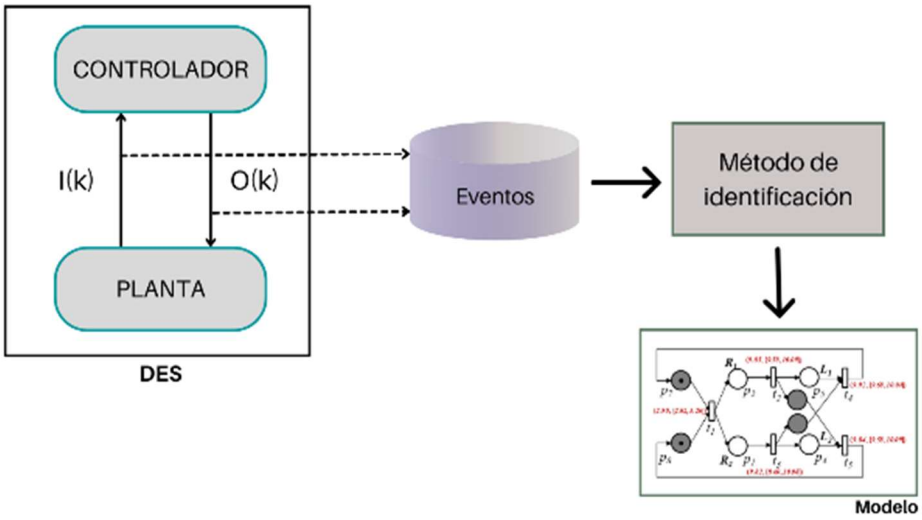


Fig. 1. Proceso de identificación para un DES.

Modelos temporizados. De acuerdo con [6], el tiempo en una red de Petri puede ser asociado con la duración de una operación o con el tiempo esperado previo a la ocurrencia de un evento. Trabajos relevantes sobre este tema se presentan en [7] donde se aborda la identificación del sistema basado en programación lineal entera, y en [8] donde se propone un algoritmo de aprendizaje que calcula RP ordinarias de acuerdo con la medición de flujos de salida cíclicos.

Propuesta. Este artículo se enfoca en el aspecto del tiempo; los modelos a descubrir capturan esa información con RP temporizadas (RPT) en las transiciones. Se presenta un método de identificación/descubrimiento de procesos temporizados donde el modelo obtenido es una RPT. La propuesta extiende un método previo [4] mediante de una técnica de refinamiento a una primera RPT identificada, la cual puede agregar nuevas transiciones con temporización más precisa.

Organización. El artículo está organizado como sigue. La sección 2 presenta los conceptos básicos de RP y el planteamiento del problema. La sección 3 describe la primera parte del método. La sección 4 presenta la fase de refinamiento. La Sección 5 ilustra la propuesta mediante un caso de estudio de tipo académico.

2. Preliminares y planteamiento del problema

2.1. Redes de Petri

Definición 1. (Red de Petri). Una estructura de red de Petri ordinaria G es un grafo bipartita representado por la tupla $G = (P, T, F)$ donde: $P = \{p_1, p_2, \dots, p_{|P|}\}$ y $T = \{t_1, t_2, \dots, t_{|T|}\}$ son conjuntos finitos de vértices denominados lugares y transiciones respectivamente; $F \subseteq P \times T \cup T \times P$ es una relación que representa los arcos que van de lugares a transiciones y viceversa. [9].

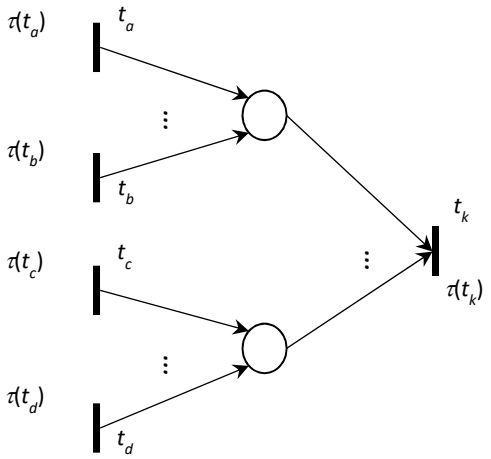


Fig. 2. Posibles transiciones precedentes a t_k.

Definición 2. (Red de Petri temporizada). Una transición temporizada es la tupla $N_\delta = (G, M_0, Tim)$, donde G es una PN ordinaria, M_0 es el marcado inicial; Tim es una función que asocia el tiempo a las transiciones; ésta puede ser de dos tipos:

- $\delta: T \rightarrow \Re^{\geq 0}$: una función que asigna un valor no negativo a cada $t_j \in T$; tal valor representa el tiempo de disparo de t_j contado a partir de su habilitación [10].
- $\iota: T \rightarrow \Re^{\geq 0} \times \Re^{\geq 0}$: una función de intervalo (ventana) de tiempo $[l_j, u_j]$ de disparo de la transición $t_j \in T$; t_j debe ser disparada después de l_j o antes de u_j unidades de tiempo contadas a partir del instante en que se habilita t_j [11].

Cuando $\delta(tj) = 0$ ó $\iota(tj) = [0, 0]$, t_j se dispara en cuanto está habilitada; a t_j se le llama transición inmediata. Las transiciones temporizadas e inmediatas suelen representarse por rectángulos vacíos y llenos, respectivamente.

Definición 3. (Registro de eventos temporizado). Un Registro de eventos temporizado (fechado) $\lambda_t = \{\sigma_{ti}\}$ es un conjunto de trazas $\lambda_t = \{\sigma_{ti} | \sigma_{ti} = (A_1, d_1) \ldots (A_i, d_i) \ldots (A_K, d_K)\}$, donde $A_j \in \boldsymbol{\Sigma}$ representa la tarea en la posición j y $d_j \in \Re^{\geq 0}$ es el instante en que se registra A_j. Cada traza se registra desde el tiempo $\tau = 0$.

2.2. Planteamiento del problema

El problema de identificación de procesos de eventos discretos temporizados se puede enunciar como sigue.

Definición 4. Sea λ_t un registro de eventos temporizado generado por un proceso de eventos discretos funcionando normalmente. El problema de identificación consiste en obtener una RPT la cual describa el comportamiento del proceso registrado en λ_t.

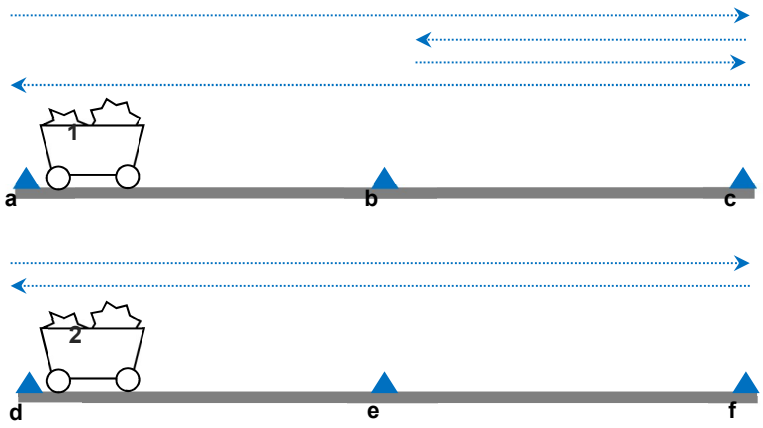


Fig. 3. Representación de proceso a ser modelado.

En [4] se presentó un método de identificación que aborda el problema anterior, el cual propone una estrategia en dos etapas. En la primera etapa, se procesa λ_t sin tomar en cuenta las fechas para obtener una RP cíclica no temporizada *N*.

En la segunda etapa, se determina la temporización de las transiciones; para ello se ejecuta λ_t en *N* determinando las duraciones de cada t_j cada vez que se dispara en la secuencia. Al final se tiene un conjunto de duraciones calculadas para cada transición; estos datos se procesan para obtener $\delta(t_j)$ y $\iota(t_j)$.

Los valores de los conjuntos de duraciones pueden ser dispersos, por lo que los parámetros $\delta(t_j)$ y $\iota(t_j)$ no representarían cercanamente el comportamiento temporal del proceso. Por esta característica, en el presente artículo se propone una extensión al método referido para mejorar la exactitud (*accuracy*) de la RPT descubierta.

A partir del análisis de los datos sobre las duraciones, se determina si existe más de una agrupación (clúster) de datos; en tal caso la transición es remplazada por un número de transiciones igual al número de clústeres. Posteriormente, se determina una sub-red con dichas transiciones, la cual ordena el disparo en las secuencias $\sigma_{t\,i}$. A este problema se le llama Refinamiento, sobre el cual se enfoca el presente artículo.

Definición 5. Sea N una RPT obtenida a partir de λ_t. El problema de refinamiento de una RPT consiste en extender N agregando un conjunto de subredes N_r tales que la composición $N||N_r$ ejecute λ_t y las temporizaciones de las transiciones en se aproximen al comportamiento observado.

3. Identificación de procesos temporizados

En el contexto de procesos temporizados, en [4] se presenta una extensión a los métodos en [12, 13] donde a partir de una secuencia S_τ de eventos con sus tiempos de ejecución (eventos fechados) se analiza cada ocurrencia de las distintas transiciones

Algoritmo 1. Refinamiento de TPN.

Entrada: $N=(P,T,F)$, S, $\gamma(T)$, Tim
Salida: N_R, Tim

1. $R(t_j) \leftarrow \varnothing$
2. $\forall t_j \in T$:
  If EMgetComponents ($\gamma(t_j)$) > 1
  then
    lab ← GetLabel($\gamma(t_j)$); // obtiene el conj nombres de las tr replicadas
    $R(t_j)$ ← Rename (t_j, lab); // obtiene las tr replicadas
    $\forall t_r \in R(t_j)$
    $\gamma(t_r) \leftarrow$ GetTime($R(t_j)$); //obtiene los tiempos para cada t_r de $R(t_j)$
3. $S_P \leftarrow Pr(S', R(t_j))$ // proyección
  N' ← IdentPN(S_P)
  $N_R \leftarrow$ Merge(N, N') // composición síncrona
4. $\forall\ t_k \in$ lab
  $\delta(t_k) \leftarrow$ average($\gamma(t_k)$);
  $\iota(t_k) \leftarrow$ [min($\gamma(t_k)$), max($\gamma(t_k)$)];
  $Tim_R(t_k) \leftarrow (\delta(t_k), \iota(t_k))$;
  Tim ← Tim ∪ Tim_R
5. Return N_R, Tim

para determinar el tiempo ocurrido entre ellas y finalmente asignar valores de tiempo a cada transición.

La obtención de los valores de tiempo para una PN en [4] se determina una función Tim, a partir de una secuencia de eventos temporizados S_τ y la estructura de la red de Petri.

La estrategia consiste en analizar S_τ comparando cada transición t_k en S_τ(k) con una o varias transiciones previas en la secuencia y determinando el tiempo que ocurre entre $\tau(t_k)$ de S_τ(k) y la $\tau(t_k)$ correspondiente en la secuencia S_τ(k-i).

Estas comparaciones se realizan en base a la semántica de las redes de Petri en que el tiempo transcurrido asociado a una transición t_k representa la duración máxima de la permanencia de una marca que habilita t_k

De esta manera, cada vez que aparece t_k en la secuencia S_τ, se analiza la sub-secuencia anterior a S_τ(k), hasta una aparición previa de t_k o hasta que se haya llegado al inicio de la secuencia, es decir, S_τ(1). En esta sub-secuencia, se analiza la ocurrencia de las transiciones inmediatamente anteriores a t_k en el modelo; la Figura 2 ilustra los elementos a considerar para la obtención de tiempos referentes a t_k.

Para cada lugar previo a t_k deberá detectarse una de sus transiciones de entrada en la sub-secuencia previa a t_k en S_τ. Entre esas transiciones, siendo t_r la transición cuyo instante de registro es el último, el tiempo máximo de residencia de las marcas en el marcado que habilita t_k es $\delta = \tau(t_k) - \tau(t_r)$, lo que es el tiempo transcurrido asociado a t_k para esta sub-secuencia. El algoritmo descrito obtiene para cada transición t_j, el promedio de duración del marcado que habilita t_j, y sus valores mínimo y máximo.

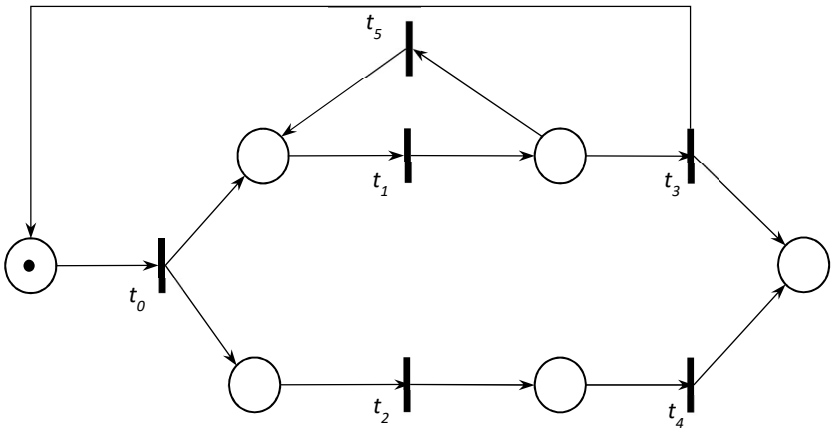


Fig. 4. Modelo que representa proceso de Fig. 3.

Tabla 1. Cálculo de tiempos asociados a un proceso.

Transición	**Tim(t_j)**	**$\gamma(t_j)$**
t_0	(4.017, [3.90, 4.15])	3.93, 3.95, 4.03, 4.04, 3.97, 3.93, 4.06, 3.97, 4.14, 4.05, 3.93, 3.90, 4.02, 3.91, 4.02, 4.14, 4.14, 4.01, 4.15, 4.06
t_1	(7.488, [4.90,10.09])	9.92, 5.01, 10.04, 4.92, 9.94, 4.91, 10.08, 4.94, 9.92, 5.17, 10.01, 4.90, 9.92, 4.93, 10.01, 4.91, 10.02, 5.03, 10.05, 5.01, 10.09, 4.91, 9.94, 5.06, 10.03, 5.11, 9.94, 4.95
t_2	(9.884, [9.67,10.06])	10.02, 9.96, 10.06, 9.88, 9.94, 9.99, 9.93, 9.80, 9.99, 9.76, 10, 9.73, 9.88, 9.80, 9.83, 9.93, 9.67, 9.71, 9.88, 9.93
t_3	(9.901, [9.80,10.06])	9.87, 9.91, 9.94, 9.92, 9.80, 9.80, 9.80, 9.85, 9.83, 10.05, 9.91, 9.93, 9.81, 9.98, 9.98, 9.84, 10.06, 10.01, 9.83, 9.91
t_4	(9.954, [9.82,10.10])	9.82, 10, 10.02, 9.90, 10.04, 10.10, 9.91, 10.07, 9.87, 10.05, 9.94, 9.86, 9.91, 10.06, 9.98, 9.86, 9.93, 9.92, 10.01, 9.84
t_5	(4.809,[4.47,5.10])	4.73, 5.06, 4.7, 4.86, 4.69, 4.66, 4.49, 4.74, 4.91, 5.1, 4.96, 5, 5.09, 4.69, 4.47, 4.91, 4.60, 4.91, 5.08, 4.53

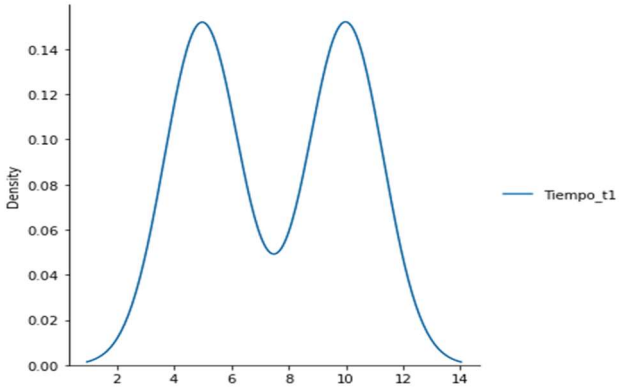


Fig. 5. Clústeres obtenidos para una transición.

4. Refinamiento de redes de Petri temporizadas

4.1. Enfoque

El método propuesto aborda el caso en que se cuenta con un modelo obtenido mediante algún método de identificación, así como una secuencia de eventos con sus tiempos de ejecución. Siguiendo la propuesta de [4], se generan conjuntos de valores de tiempo de disparo para cada transición; estos tiempos son analizados para identificar si los valores de duración pueden agruparse en dos o más conjuntos de valores (clúster).

Si se detecta más de una agrupación, es necesario modificar localmente el modelo agregando transiciones para hacerlo más exacto. En este trabajo se utiliza el algoritmo esperanza-maximización (EM) [14], dado que permite obtener el número de clústeres alrededor de diversos valores medios sin necesidad de especificarlos previamente, a diferencia del algoritmo k-means [15, 16].

Al aplicar el algoritmo EM para obtener los clústeres y valores medios del conjunto de datos, se obtienen etiquetas asignadas a cada elemento para identificar a que clúster pertenece cada valor en la secuencia.

Con esta información, una transición se sustituirá por tantas transiciones como clústeres se encuentren a fin de ajustar el modelo inicial. En particular para este trabajo se ha utilizado la biblioteca de Python scikit-learn [18], específicamente el objeto GaussianMixture que permite la implementación del algoritmo EM.

4.2. Método de refinamiento

El método propuesto se puede dividir en cinco etapas:

1. Aplicación del algoritmo EM al conjunto de duraciones $\gamma(t_j)$ calculadas para cada transición t_j, para obtener el número de componentes con sus valores medios respectivos.

Tabla 2. Tiempos de t_1 agrupados por clúster.

Transición	$\gamma(t_j)$
t_1	9.92, 10.04, 9.94, 10.08, 9.92, 10.01, 9.92, 10.01, 10.02, 10.05, 10.09, 9.94, 10.03, 9.94
t_1',	5.01, 4.92, 4.91, 4.94, 5.17, 4.90, 4.93, 4.91, 5.03, 5.01, 4.91, 5.06, 5.11, 4.95

2. Obtener las etiquetas correspondientes a cada una de las medias. Al aplicar el algoritmo EM se obtiene, además del número de clústeres existente en los datos, las etiquetas que permiten identificar a que clúster pertenece cada uno de los elementos analizados. Esta información es útil ya que ayudará a agrupar los valores pertenecientes a cada una de las transiciones replicadas.
3. A partir de las etiquetas obtenidas se replican las transiciones t_j que tiene más de un clúster. En el modelo *N*, t_j y sus réplicas tendrán los mismos $^\bullet t_j$ y $t_j^\bullet$. Las nuevas transiciones (T') se renombran y se actualizan en la secuencia original S; se obtiene una secuencia S'.
4. Hacer una proyección de S' sobre las nuevas T' renombradas Pr (S', T'). Sobre esa proyección, se aplica el método de identificación a S' para obtener una subred N' con T' y nuevos lugares P'.
5. Combinar la subred N' recién creada con la red de Petri original N para generar el modelo refinado (N||N').

El procedimiento se repite para cada transición t_j que tenga más de un clúster en el análisis de $\gamma(t_j)$ del Paso 1. Cada paso del método se describe con más detalle a continuación.

4.2.1 Análisis de las duraciones calculadas

Dado que se pueden encontrar variaciones de los tiempos en $\gamma(t_j)$, estos conjuntos se analizan con el algoritmo EM. En una primera instancia, se le especifica un rango definido de componentes para probar.

El algoritmo entrega el valor 1 (número de componentes) cuando los valores de tiempo están alrededor de una sola media; si este es el caso, esa transición no requerirá ajustes y la siguiente transición podrá ser procesada. Cuando el algoritmo devuelva un valor de 2 o más componentes para una transición, se aplicarán los pasos consecuentes.

4.2.2 Replicar transiciones con más de un clúster

Si un $\gamma(t_j)$ tiene dos o más componentes se deben crear nuevas transiciones. Entonces se utiliza un método que proporciona las etiquetas para las muestras de datos; cada dato de entrada tendrá asociado un número de etiqueta y habrá tantas etiquetas diferentes como clústeres en la transición. En el modelo N todas las transiciones replicadas de t_j se agregan usando los mismos lugares de entrada y de salida de t_j. Las transiciones creadas son renombradas para ser incluidas en S.

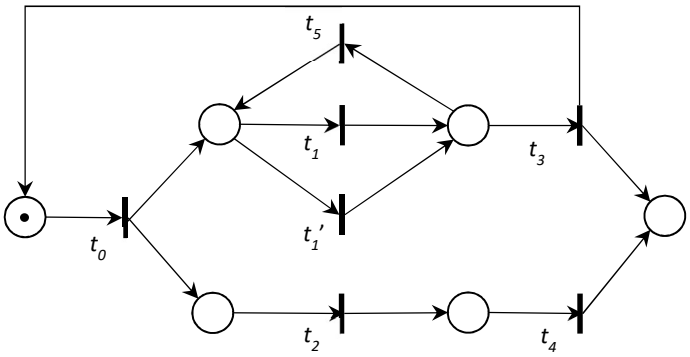


Fig. 6. Subred creada a partir de proyección en S.

4.2.3 Renombrar y sustituir las transiciones en S

Analizando las etiquetas, la transición se replica y se deben renombrar. Así, por ejemplo, t_j se puede convertir en t_j, y t_j' dependiendo de las etiquetas correspondientes. Las transiciones replicadas se incluyen en un conjunto T_j. Con la información proporcionada por las etiquetas, se relaciona cada valor de tiempo de la transición que se está tratando con el clúster al que pertenece.

A partir de esta relación se determina el número de transiciones nuevas representan ahora a t_j; para cada una se generan nuevos nombres y se determina la inclusión de las transiciones replicadas en la secuencia original. La secuencia actualizada se denomina S'.

4.2.4 Obtención de una subred con las transiciones replicadas

Luego de la sustitución, deberá modificarse la red original para incluir las nuevas transiciones. Para ello cada transición replicada tendrá los mismos lugares de entrada t_j y de salida t_j que t_j.

Adicionalmente, se realiza una proyección de S' sobre las transiciones replicadas T' $(Pr(S', T'))$; esto genera una secuencia S_P, formada por transiciones de T', a la cual se le aplicará un método de identificación [5] para obtener una RP N' con las transiciones de T' y nuevos lugares P'.

4.2.5 Fusionar la subred creada con el modelo original

Finalmente, se fusiona la subred creada N' con el modelo N de la TPN con las transiciones replicadas de acuerdo al método propuesto en [17]. La nueva red resultado de la composición síncrona N||N' y las temporizaciones correspondientes a las transiciones en T', mejora su exactitud. Los pasos anteriores se resumen en el Algoritmo 1.

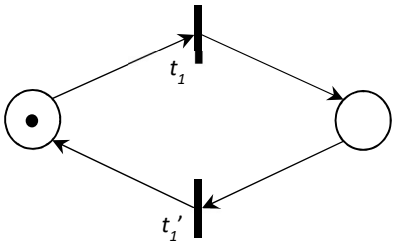


Fig. 7. Subred creada a partir de proyección en S.

5. Caso de estudio

A continuación, se presenta un ejemplo de aplicación del algoritmo propuesto. En la Figura 3 se muestra un proceso que puede ser modelado por una TPN. Se trata de un proceso con dos dispositivos móviles (carros) que se desplazan en vías separadas. Cada carro tiene dos tipos de movimiento, hacia la derecha y hacia la izquierda; en la figura se indica la secuencia de desplazamientos que tienen que hacer cada carro.

El carro 1 realiza una secuencia que toma más tiempo; es claro ver que los desplazamientos del carro 1, tanto a la derecha como a la izquierda, tienen duración diferente. Si en el modelo el movimiento hacia la derecha se representa mediante una transición, las duraciones calculadas tendrán variaciones.

Al aplicar el método propuesto, se deberá encontrar dos transiciones distintas que representen el movimiento a la derecha, pero con diferentes tiempos asignados muy similares.

Considere que en el proceso previo se definieron las siguientes actividades $T = \{t_0, t_1, t_2, t_3, t_4, t_5\}$, donde t_1 y t_2 representan el desplazamiento a la derecha de los carros 1 y 2 respectivamente; t_3 y t_5 representan los desplazamientos largo y corto, respectivamente, a la izquierda del carro 1; t_0 es el evento de inicio. Las secuencias registradas de la ejecución del proceso son:

S = *t0, t1, t2, t5, t1, t4, t3, t0, t2, t1, t5, t4, t1, t3, t0, t1, t2, t5, t1, t4, t3, t0, t2, t1, t5, t4, t1, t3, t0, t1, t2, t5, t1, t4, t3, t0, t2, t1, t5, t1, t4, t3, t0, t1, t2, t5, t1, t4, t3, t0, t2, t1, t5, t1, t4, t3, t0, t1, t5, t2, t4, t1, t3, t0, t1, t5, t2, t1, t4, t3, t0, t2, t1, t4, t5, t1, t3, t0, t1, t2, t4, t5, t1, t3, t0, t1, t2, t5, t4, t1, t3, …*

S_τ = *(t0, 3.93), (t1, 13.85), (t2, 13.95), (t5, 18.58), (t1, 23.59), (t4, 23.77), (t3, 33.64), (t0, 37.59), (t2, 47.55), (t1, 47.63), (t5, 52.69), (t4, 57.55), (t1, 57.61), (t3, 67.52), (t0, 71.55), (t1, 81.49), (t2, 81.61), (t5, 86.19), (t1, 91.1), (t4, 91.63), (t3, 101.57), (t0, 105.61), (t2, 115.49), (t1, 115.69), (t5, 120.55), (t4, 125.39), (t1, 125.49), (t3, 135.41), (t0, 139.38), (t1, 149.3), (t2, 149.32), (t5, 153.99), (t1, 159.16), (t4, 159.36), (t3, 169.16), (t0, 173.09), (t2, 183.08), (t1, 183.1), (t5, 187.76), (t1, 192.66), (t4, 193.18), (t3, 202.98), (t0, 207.04), (t1, 216.96), (t2, 216.97), (t5, 221.45), (t1, 222.38), (t4, 226.88), (t3, 236.68), (t0, 240.65), (t2, 250.45), (t1, 250.66), (t5, 255.4), (t1, 260.31), (t4, 260.52), (t3, 270.37), …*

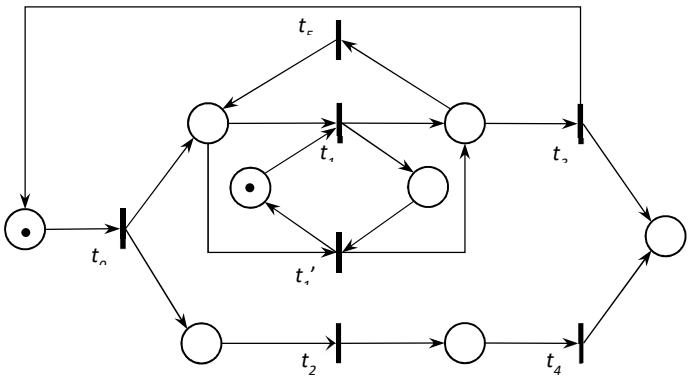


Fig. 8. RPT obtenida después de refinamiento.

A partir de la secuencia no fechada S, un método de identificación obtendrá la RP mostrada en la Figura 4. En cuanto a la determinación de los tiempos de ejecución, en la Tabla 1 se muestra un listado de tiempos que pueden ser obtenidos mediante el método descrito en [4] en base a S_τ y la RP de la Figura 4.

Analizando la tabla, se puede observar que el tiempo promedio para la t_1 es de 7.488, con un intervalo de tiempo entre 4.90 y 10.09. Esto se debe a que la transición se ejecuta algunas veces con valores de tiempo alrededor de 5, y otras veces alrededor de 10, lo que genera un intervalo amplio y una media no muy exacta.

Siguiendo los pasos descritos del algoritmo propuesto, en primer lugar, se realiza una búsqueda de clústeres en los tiempos de cada transición. Se obtiene un clúster para t_0, t_2, t_3, t_4, t_5 y dos clústeres para t_1 como se muestra en la figura 5. Luego, el resto del algoritmo se aplica a esta transición. Al aplicar el algoritmo EM, además del número de clústeres, se obtienen los valores de las medias y una etiqueta para cada elemento, que identifica a que clúster pertenece dicho elemento.

De esta manera los valores de las medias son 4.98 y 9.99. Por su parte, los elementos de tiempo que anteriormente pertenecían a t_1 hora se dividen como se muestra en la tabla 2. Una vez teniendo los clústeres, las transiciones son renombradas como t_1 y t_1' (T'). La RP *N* se refina de acuerdo a manera descrita anteriormente quedando como se muestra en la Figura 6. Por otro lado, T' = {t_1 y t_1'} son incluidas en la secuencia original S para obtener S' de la siguiente manera:

S' = $t_0, t_2, t_1, t_5, t_1', t_4, t_3, t_0, t_1, t_2, t_5, t_1', t_4, t_3, t_0, t_1, t_5, t_2, t_4, t_1', t_3, t_0, t_1, t_5, t_2, t_1', t_4, t_3, t_0, t_2, t_1, t_4, t_5, t_1', t_3, t_0, t_1, t_2, t_4, t_5, t_1', t_3, t_0, t_2, t_1, t_5, t_4, t_1', t_3, t_0, t_1, t_2, t_5, t_4, t_1', t_3, \ldots$

Haciendo la proyección de S'en t_1 y t_1' se obtiene:

$$S_P = Pr(S', T') = t_1, t_1', t_1, t_1', t_1, t_1', t_1, t_1', t_1, t_1', t_1, t_1', t_1, t_1', t_1, t_1'$$

Aplicando el método de identificación a S_P se obtiene la RP mostrada en la Figura 7. Las nuevas transiciones obtenidas t_1 y t_1' serán agregadas a la red de Petri inicial respetando el Pre y el Post de la transición original t_1. Esto se ilustra en la figura 6. En el ejemplo, se realiza la proyección de S en las transiciones recién etiquetadas y se obtiene una PN como la que se muestra en la Figura 7.

Finalmente, la subred creada N' es fusionada con el modelo N de la TPN (N_R=N||N') siguiendo los pasos indicados en [17]; ésta red es mostrada en la Figura 8. Los parámetros de tiempo correspondientes a las transiciones en T' son Tim(t_1) = (9.99, [9.92, 10.09]) y Tim(t_1') = (4.982, [4.90, 5.17]).

6. Conclusión

Se ha presentado un método para identificar procesos temporizados donde los modelos usados son redes de Petri temporizadas. La propuesta extiende un algoritmo básico existente que determina parámetros de tiempo para una red de Petri.

El nuevo método permite abordar casos en que los valores de tiempo calculados para algunas transiciones son muy variables. A partir de un primer modelo N obtenido y el cálculo de las temporizaciones, el algoritmo propuesto determina un refinamiento estructural de N y un aumento de la exactitud de las temporizaciones. Cuando se presentan estos casos de dispersión, el método aumenta la exactitud del modelo.

Esta propuesta permite obtener modelos más exactos en la temporización; además, su desempeño es bastante eficiente respecto a los trabajos publicados sobre identificación de procesos temporizados.

Actualmente se estudian situaciones en las que los valores de tiempo para una transición no sigan una distribución normal o en las que la media no represente fielmente el comportamiento del tiempo del sistema. Se analizan medidas de dispersión para cubrir más escenarios y describir de manera adecuada el comportamiento de los parámetros temporales.

También se trabaja en la definición de medidas para evaluar la exactitud de los modelos con respecto a los registros de eventos temporizados, tomando en consideración las desviaciones del instante de ocurrencia de los eventos con respecto al disparo esperado de las transiciones en el modelo.

Referencias

1. Meda-Campaña, M., Ramirez-Treviño, A., López-Mellado, E.: Asymptotic identification of discrete event systems. In: Proceedings of the 39th IEEE Conference on Decision and Control, pp. 2266–2271 (2000) doi: 10.1109/CDC.2000.914135
2. Meda-Campaña, M., López-Mellado, E.: Identification of concurrent discrete event systems using Petri nets. In: Proceedings of the 17th IMACS World Congress on Computational and Applied Mathematics, pp. 11–15 (2005)
3. Estrada-Vargas, A. P., López-Mellado, E., Lesage, J. J.: A black-box identification method for automated discrete-event systems. IEEE Transactions on Automation Science and Engineering, vol. 14, no. 3, pp. 1321–1336 (2017) doi: 10.1109/TASE.2015.2445332
4. Rodríguez-Pérez, E., Tapia-Flores, T., López-Mellado, E.: Identification of timed discrete event processes. Building input-output petri net models, vol. 16, pp. 153–167 (2016)
5. Tapia-Flores, T., López-Mellado, E., Estrada-Vargas, A. P., Lesage, J. J.: Discovering Petri net models of discrete-event processes by computing T-invariants. IEEE Transactions on Automation Science and Engineering, vol. 15, no. 3, pp. 992–1003 (2017) doi: 10.1109/ TASE.2017.2682060

6. David, R., Alla, H.: Discrete, continuous, and hybrid Petri nets. Springer (2010) doi: 10.1007/978-3-642-10669-9
7. Basile, F., Chiacchio, P., Coppola, J.: Identification of time Petri net models. IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 9, pp. 2586–2600 (2016) doi: 10.1109/TSMC.2016.2523929
8. Meda-Campaña, M. E., Medina-Vazquez, S.: Synthesis of timed Petri net models for on-line identification of discrete event systems. In: 9th IEEE International Conference on Control and Automation, pp. 1201–1206 (2011) doi: 10.1109/ICCA.2011.6137968
9. Murata, T.: Petri nets: Properties, analysis and applications. In: Proceedings of the IEEE, vol. 77, no. 4, pp. 541–580 (1989) doi: 10.1109/5.24143
10. Ramchandani, C.: Analysis of asynchronous concurrent systems by timed Petri nets. Ph. D. Thesis, Massachusetts Institute of Technology (1973) http://hdl.handle.net/1721.1/13739
11. Merlin, P. M.: A study of the recoverability of computing systems. University of California, Irvine (1974)
12. Estrada-Vargas, A. P., López-Mellado, E., Lesage, J. J.: Input–output identification of controlled discrete manufacturing systems. International Journal of Systems Science, vol. 45, no. 3, pp. 456–471 (2014) doi: 10.1080/00207721.2012.724098
13. Estrada-Vargas, A. P., Lesage, J. J., López-Mellado, E.: A stepwise method for identification of controlled discrete manufacturing systems. International Journal of Computer Integrated Manufacturing, vol. 28, no. 2, pp. 187–199 (2015) doi: 10.1080/0951192X.2013.874591
14. Moon, T. K.: The expectation-maximization algorithm. IEEE Signal Processing Magazine, vol. 13, no. 6, pp. 47–60 (1996) doi: 10.1109/79.543975
15. Alldrin, N., Smith, A., Turnbull, D.: Clustering with EM and K-means. University of San Diego, California, Tech Report, pp. 261–295 (2003)
16. Li, M. J., Ng, M. K., Cheung, Y. M., Huang, J. Z.: Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters. IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 11, pp. 1519–1534 (2008) doi: 10.1109/TKDE.2008.88
17. López-Mellado, E., Flores-Tapia, T.: Refining discovered Petri nets by sequencing repetitive components. In: International Workshop on Algorithms and Theories for the Analysis of Event Data Conference (2017)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot M., Duchesnay, E.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research, vol. 12, pp. 2825–2830 (2011)

ISSN 1870-4069

Diseño empírico de una arquitectura de perceptrón multicapa binario residual

Agustín Solís Winkler, José Luis Tapia Fabela,
Santiago Osnaya Baltierra

Universidad Autónoma del Estado de México,
México

{asolisw, jtapiaf, sosnayab}@uaemex.mx

Resumen. Desde su introducción en 2015 por Courbariaux, las redes neuronales binarias han surgido como una alternativa prometedora para reducir los grandes requisitos de cómputo, memoria y almacenamiento en modelos de aprendizaje profundo, permitiendo su implementación en dispositivos con recursos limitados, como los utilizados en el cómputo de frontera. No obstante, estos modelos presentan una notable degradación en su precisión en comparación con sus equivalentes de punto flotante. En este estudio, se emplean las últimas recomendaciones de diseño de arquitecturas binarias para desarrollar empíricamente una arquitectura de perceptrón multicapa binario residual que reduce los efectos adversos del proceso de binarización. La intención es proporcionar un modelo sencillo de implementar que no requiera un nivel experto en diseño de redes neuronales para su uso en dispositivos de baja potencia.

Palabras clave: Red neuronal binaria, red neuronal, perceptrón multicapa, compresión de redes, cuantización, binarización.

Empirical Design of a Residual Binary Multilayer Perceptron Architecture

Abstract. Since its publication in 2015 by Courbariaux, binary neural networks have proven to be a good alternative for reducing computing, memory, and storage requirements in deep learning models, which allows them to be implemented on devices with limited resources, such as those used in edge computing. However, binary models present a notable degradation in accuracy compared to their floating-point counterparts. Considering the multilayer perceptron as a flexible model of machine learning, we use the design recommendations of binary architectures of the state of the art in this work to empirically develop a binary residual multilayer perceptron architecture, which reduces the degradation caused by the binarization process. The aim is to have a model that is simple to implement and that does not require an expert level in the design of neural networks on the part of its users to deploy models that work on hardware-constrained devices.

Keywords: Binary neural network, neural network, multilayer perceptron, model compression, quantization, binarization.

1. Introducción

Como han explicado [1], los métodos de aprendizaje profundo basados en el uso de redes neuronales prealimentadas de múltiples capas o perceptrones multicapa, los cuales son una extensión del modelo original del perceptrón propuesto por [2], han tenido gran éxito académico y comercial en muchas áreas de interés como el reconocimiento y procesamiento de imágenes, reconocimiento del habla, procesamiento del lenguaje natural, traducción automática, desempeño superior al humano en juegos y hasta conducción autónoma de automóviles, por lo que han sido fundamentales en el éxito y consolidación de la inteligencia artificial en nuestros días como hacen notar [3, 4].

Aunque los mejores modelos de aprendizaje profundo obtienen resultados espectaculares en el laboratorio, su tamaño los vuelve poco útiles en aplicaciones del mundo real [5]. Desde los trabajos de [6, 7] se ha planteado la necesidad de reducir los requerimientos de los modelos de aprendizaje profundo para permitir su uso en dispositivos de bajos recursos [8].

Dado que las aplicaciones basadas en el uso de redes profundas tienen gran auge en la actualidad, esta necesidad ha conducido al desarrollo del campo de la compresión de redes neuronales [9]. Dentro de este campo, la cuantización [10] y la binarización [11,12] presentan gran interés debido a la sencillez de sus conceptos e implementación.

La binarización consiste en reducir la precisión de los parámetros de la red neuronal hasta un solo bit, lo que representa la forma más extrema de la cuantización [12]. Esta técnica permite ahorrar espacio de almacenamiento y memoria, además de reducir los requerimientos de procesamiento y tiempo de ejecución al reemplazar las operaciones de punto flotante por operaciones lógicas y de conteo de bits como lo proponen [13] y confirman otros autores como [14, 15].

A pesar de su atractivo, las redes neuronales binarias presentan problemas de notable degradación y pérdida de exactitud en comparación con sus equivalentes de punto flotante según se ha expresado en estudios previos [15, 16].

Sin embargo, se han desarrollado técnicas y distintos enfoques para abordar estos problemas, los cuales ha sido resumidos por [12], entre los que podemos contar la reducción de errores de cálculo durante la cuantización, la mejora de la función de pérdida y la aproximación del gradiente; la aplicación de diferentes estrategias de entrenamiento y el diseño de arquitecturas específicas para redes binarias.

Este último enfoque resulta particularmente interesante ya que busca crear modelos más eficientes y compactos al mismo tiempo que trata de optimizar la red y mejorar su exactitud. Aunque aún no se cuenta con una explicación matemática formal del funcionamiento de las redes neuronales binarias como explican [17], se sabe que las redes neuronales son aproximadores universales de funciones continuas, según describen [18].

Adicionalmente, [19] han demostrado la aproximación universal de funciones mediante redes neuronales binarias de tres capas. Estos hallazgos sugieren que es posible diseñar un perceptrón multicapa binario capaz de aproximar la solución de problemas que puedan ser modelados. Por lo tanto, el presente trabajo propone desarrollar, de manera empírica, una arquitectura de perceptrón multicapa binario residual utilizando ampliación e inserción capas adicionales, así como el uso atajos.

El objetivo es determinar si la exactitud obtenida con estos cambios de arquitectura puede compensar la pérdida de información derivada de la binarización y si la precisión obtenida es comparable con la del perceptrón multicapa de punto flotante, con un menor costo computacional.

2. Trabajos relacionados

Se han desarrollado diversos trabajos para mejorar la precisión y exactitud de las redes neuronales binarias, y se han realizado importantes avances en la teoría desde su introducción en 2015. El modelo original de red neuronal binaria, BinaryConnect de [20] binariza los pesos, pero no las activaciones. Al año siguiente el mismo equipo presentó BinaryNet que ya utiliza activaciones binarias [11].

Posteriormente, el modelo XNOR-NET de [13], incluyó todos los métodos propuestos en la BinaryConnect original, pero agregando un valor de ganancia para compensar la pérdida de información durante la binarización y cambiando el orden de algunas capas para mejorar el entrenamiento. Sin embargo, aunque los términos de ganancia mejoran la precisión de la red, su cálculo resulta costoso.

En los últimos años, se han presentado varios enfoques para mejorar la precisión y eficiencia de las redes neuronales binarias. Uno de estos enfoques es el propuesto por [21], denominado Binarized Neural Networks (BNN), que utiliza una aproximación probabilística para la binarización de los pesos y activaciones, lo que ha demostrado ser más eficiente en términos de memoria y cómputo que los modelos anteriores.

Sin embargo, esta aproximación muestra una disminución en la precisión, lo que ha sido abordado mediante el uso de técnicas como la cuantización de pesos y activaciones y la adición de capas con precisión de punto flotante.

Otra propuesta de 2016 es la DoReFa-Net por [22], que trata de mejorar las ineficiencias de XNOR-NET, utilizando diferente precisión y longitud para los pesos, las activaciones y los cálculos inversos durante el entrenamiento. Sin embargo, su método es complejo y no ha demostrado una mejora significativa sobre las operaciones de bits. Junto con esto, en los últimos años se han presentado avances en el diseño de arquitecturas de redes binarias, como Binary DenseNet por [23], que utiliza conexiones de atajo para aumentar el flujo de información entre capas.

Además, aconsejan utilizar el uso de capas de reducción con precisión de punto flotante para preservar la información y mejorar la exactitud de las redes binarias. Otro enfoque de los mismos autores es la MeliusNet publicada por [24] y mencionada por [25] que utiliza un Bloque Denso para incrementar la capacidad de las características. Estos avances en el diseño de arquitecturas muestran que se están realizando esfuerzos para mejorar la precisión de las redes binarias y superar sus limitaciones iniciales.

3. Conceptos

Esta sección describe los principios de la binarización de redes neuronales, y algunos de los conceptos principales que se utilizan en la implementación del modelo de perceptrón propuesto.

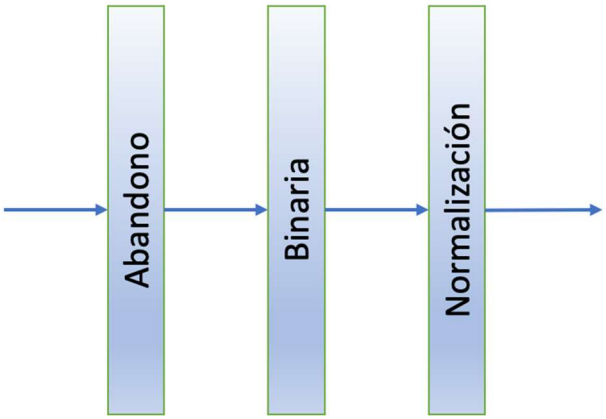


Fig. 1. Bloque de construcción de un perceptrón multicapa binario.

3.1. Red neuronal binaria

Una red neuronal binaria es un tipo de modelo de red neuronal propuesto por primera vez en 2015 por [20] en el cual solo las capas de entrada y salida se representan con valores de punto flotante y tanto los pesos como activaciones en las capas ocultas se representan con valores binarios.

La idea detrás de la binarización es restringir los valores de los pesos y las activaciones a +1 y -1 durante el entrenamiento, lo que permite reemplazar las operaciones de multiplicación y acumulación de punto flotante por operaciones XNOR y POPCOUNT de 1 bit que se ejecutan a mayor velocidad que las operaciones de 32 bits (Simons & Lee, 2019).

3.2. Implementación de la binarización

La función signo se utiliza tanto para binarizar las entradas, como activación, transformando en binarios los valores de punto flotante [11]:

$$signo(x) = \begin{cases} +1 \; si \; x \geq 0, \\ -1 \; si \; x < 0. \end{cases} \tag{1}$$

3.3. Retropropagación

Para entrenar una red utilizando los pesos binarizados, (Simons & Lee, 2019) aplican la propuesta de [11], para el uso de la técnica del estimador directo (STE) propuesta por [26] para poder entrenar una red binarizada con la función signo [12]:

$$\boldsymbol{b_w} = signo(\boldsymbol{w}). \tag{2}$$

El método del estimador directo aproxima el gradiente pasando por alto el gradiente de la capa en cuestión, y convirtiendo el gradiente problemático en una función de identidad [14]:

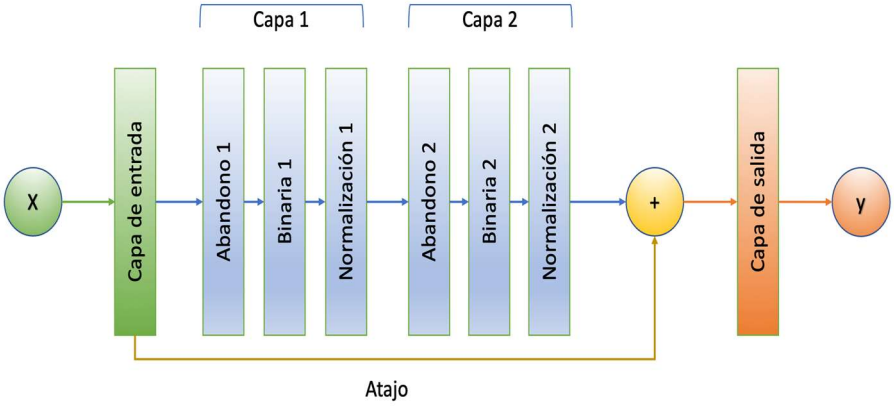


Fig. 2. Perceptrón multicapa binario residual de dos capas ocultas.

$$\frac{\partial L}{\partial \boldsymbol{w}} = \frac{\partial L}{\partial \boldsymbol{b_w}}. \tag{3}$$

Para las binarizar activaciones, se les aplica también la función signo y se utiliza el estimador directo en el recorrido hacia atrás, de la misma forma en que se binarizan los pesos. La función signo se utiliza también como función de activación en la red binaria. Si la entrada a la función de activaciones muy grande, se cancela el gradiente en el recorrido hacia atrás utilizando [14]:

$$\frac{\partial L}{\partial \boldsymbol{a}} = \frac{\partial L}{\partial \boldsymbol{b_a}} \times 1_{|a|\leq 1}, \tag{4}$$

donde $\boldsymbol{a}$ es la entrada de valor real de la función de activación y $\boldsymbol{b_a}$ es la salida binarizada de la función de activación.

3.4. Atajos

Un atajo es una conexión que lleva la información de la entrada de un bloque a la salida del mismo mediante una conexión de identidad [27]. Los atajos ayudan a aumentar el flujo de información entre dos bloques de la red, que se denominan residuales, evitando el problema de desvanecimiento del gradiente y sobreajuste en redes muy profundas. Los atajos son mencionados en el trabajo de [28].

3.5. Cuellos de botella

En una red neuronal se dice que en un diseño existe un cuello de botella, cuando el ancho de una capa es menor que el ancho de la capa anterior. Aunque la eliminación de los cuellos de botella es aconsejada por [29, 23], su efectividad es mayor en redes convolucionales que en perceptrones.

Tabla 1. Parámetros de los conjuntos de datos.

Conjunto	Objetos	Entrada	Clases	Instancias de Entrenamiento	Instancias de Prueba
MNIST	Imágenes 28x28 grises 8 bits	784	10	60000	10000
IMDB	Reseñas de películas	10000	2	25000	25000
Reuters	Noticias de 1986	5000	46	8982	2246
CIFAR10	Imágenes de 32x32 en color de 24 bits	3072	10	50000	10000

3.6. Capas de normalización por lotes

La normalización por lotes se aplica a los resultados de la capa anterior antes de la siguiente función de activación. Calcula la media y la varianza de un lote durante el entrenamiento, normalizando los datos con estos valores, ayudando a estabilizar el gradiente y reduciendo la dependencia del tamaño del lote durante el entrenamiento.

Cuando se aplica a las capas de las redes binarias, la normalización por lotes ayuda a mejorar la exactitud. [30] sugieren el abandono de las capas de normalización; desafortunadamente, el método propuesto en su lugar solo es aplicable para redes convolucionales. Sin embargo, reconocen que la normalización es una técnica bien conocida para estabilizar y acelerar el entrenamiento de los modelos.

Para el caso de redes binarias, nuestras pruebas iniciales confirmaron que el uso de capas de normalización por lotes ayuda al incremento de la exactitud. Una parte significativa de las pruebas consistió en estudiar el efecto de las capas de normalización en los modelos binarios.

3.7. Capas de abandono

Adicionalmente a la técnica de capas de normalización por lotes, la técnica de capas de abandono es muy utilizada para retardar el sobreajuste y mejorar la capacidad de generalización de los modelos. La idea es cambiar a cero algunas de las entradas durante el entrenamiento con una frecuencia definida. Las entradas que no son cero se incrementan en uno de forma que la suma de todas las entradas no cambia [31]. La ventaja de este método es que no agrega costo computacional.

4. Implementación y pruebas

El modelo multicapa binario que se propone, está basado en un bloque de construcción que reemplaza las capas totalmente conectadas en un perceptrón tradicional. Esta sección describe la arquitectura del modelo, la forma en que se

Tabla 2. Valores de los hiperparámetros.

Parámetro	**Valor**
Eras	10
Tamaño de lote	32
Ritmo de entrenamiento	0.001
Ritmo de abandono	0.05
Cuantización	8 bits
Tamaño de grupo (capas de agrupamiento)	8

implementan los atajos, y los resultados preliminares de las pruebas realizadas con los cuatro conjuntos de datos.

4.1. Bloque de construcción

Para el desarrollo del perceptrón multicapa binario residual, se consideraron las recomendaciones presentadas en especial por [23, 32], así como los componentes descritos en la sección anterior, en línea con la definición de red neuronal binaria.

Para construir el modelo, las capas de entrada y de salida se mantienen en punto flotante mientras que se utiliza un bloque de construcción que consiste en una secuencia ordenada de las siguientes capas: capa de abandono, capa totalmente conectada binaria y una capa de normalización por lotes como se muestra en la figura 1.

Este bloque se repite según el número de capas requeridas por el modelo. Si el perceptrón tiene más de una capa oculta se agrega una conexión de atajo, pero ajustando su anchura al de la última capa oculta.

4.2. Implementación de atajos

La función del atajo es pasar la información de la capa de entrada al final de la red, donde se combina con la salida de la última capa oculta para que contribuya al valor determinado en la capa de salida. Por lo general las conexiones de atajo se implementan utilizando capas convolucionales con filtros de 1x1, pero para el caso de este modelo se toma un enfoque diferente.

Se realizaron diferentes pruebas para la implementación de atajos siendo las más adecuadas las siguientes:

- Implementación mediante capa totalmente conectada sin función de activación.
- Implementación mediante capa cuantizada totalmente conectada sin función de activación. Para los experimentos, se utilizó cuantización de 8 bits, para reducir la memoria requerida por el atajo a la cuarta parte.
- Implementación de punto flotante utilizando una capa de agrupamiento máximo de una dimensión. Esta implementación requiere otra capa adicional totalmente conectada para ajustar con el tamaño de la siguiente capa cuando el ancho de la capa siguiente no se ajusta con exactitud a la salida de la capa de agrupamiento, lo

Tabla 3. Resultados con MNIST.

Modelo	Exactitud	Tamaño	Comentario
F-512	0.9811	1590.04	Punto flotante
F-128,88	0.9797	440.32	Punto flotante
F-64	0.9729	198.79	Punto flotante
B-128N, 88N, Q	0.9543	87.35	Binaria, normalizada, atajo
B-128N, 128N,128N	0.9538	25.79	Binaria normalizada
B-128N, 128N, Q	0.9502	120.79	Binaria, normalizada, atajo
B-128N, 128N	0.9490	22.29	Binaria, normalizada
B-128N, 88N	0.9488	19.63	Binaria, normalizada
B-128, 128N	0.9479	21.29	Binaria, normalizada
B-128, 88N	0.9478	16.63	Binaria, normalizada
B-128, 88	0.9361	17.95	Binaria equivalente
B-64	0.9093	9.91	Binaria equivalente
B-512	0.8969	71.04	Binaria equivalente

cual puede ocurrir cuando el cociente de la entrada y el tamaño de grupo no corresponden con el ancho de la capa destino.

4.3. Modelo propuesto

Utilizando el bloque de construcción y la conexión de atajo, la cual se suma con la última capa oculta, se obtiene un perceptrón binario multicapa residual se representa como se puede ver en la figura 2. Como puede observarse, la arquitectura propuesta reemplaza cada capa oculta de un perceptrón multicapa estándar por un bloque compuesto de una capa de abandono, una capa densa totalmente conectada binaria y una capa de normalización por lotes.

Para las pruebas realizadas, el ancho de la capa totalmente conectada binaria se mantiene igual al del perceptrón equivalente, para poder comprobar que las mejoras de precisión se deben a la adición de capas y no al incremento del ancho estas, aunque en aplicaciones reales se puede utilizar un factor de escala para mejorar la representatividad.

4.4. Pruebas

Conjuntos de datos e hiperparámetros

Para las pruebas se utilizaron cuatro conjuntos de datos: MNIST, IMDB, Reuters y CIFAR10 con la idea de mostrar la flexibilidad del perceptrón multicapa en tareas de clasificación de imágenes y textos, tanto binarias como categóricas. De cada uno de estos conjuntos se separan 10000 instancias de entrenamiento para el conjunto de validación, excepto del conjunto Reuters, del cual, por su tamaño, solo se separan 1000 ejemplos. La tabla 1 muestra las características de estos conjuntos.

Para enfocarnos en el cambio de la exactitud al modificar la arquitectura de la red, para todos los casos se utilizó un conjunto de valores similar para los hiperparámetros, en particular [17] sugiere utilizar tamaños de lote pequeños, por lo que en nuestro caso de utiliza 32. Los valores de los hiperparámetros se resumen en la Tabla 2.

Tabla 4. Resultados con IMDB.

Modelo	Exactitud	Tamaño	Comentario
F-250	0.8684	9767.58	Punto flotante
F-32,32	0.8569	1254.38	Punto flotante
F-64,32,16	0.8637	2510.5	Punto flotante
F 16,16	0.8510	626.19	Punto flotante
B-16, 16, Q	0.8596	176.07	Binaria, atajo
B-32, 32, Q	0.8443	352.19	Binaria normalizada, atajo
B-64,32N, 16, Q	0.8457	235.58	Binaria, normalizada, atajo
B-32,32N,32	0.8455	40.07	Binaria, normalizada
B-32, 32N, 32, Q	0.8583	352.69	Binaria, normalizada, atajo
B-250	0.8579	306.79	Binaria equivalente
B-32,32	0.8431	39.57	Binario equivalente
B-64,32,16	0.8442	78.94	Binaria equivalente
B-16.16	0.8450	19.75	Binaria equivalente

Identificación de modelos

La arquitectura del modelo se puede obtener de su descripción, para la que se utiliza se utiliza una notación de la forma (B -DxN, DxN, DxN, N, A) en donde:

B: designa a un modelo binario.

D: indica que se utiliza una capa de abandono antes de una capa totalmente conectada.

x: es un valor numérico que denota el ancho de la capa.

N: indica que se utiliza una capa de normalización en esa posición.

A: esta posición indica el uso de una conexión de atajo. Los valores posibles pueden ser:

F: Denota un atajo de puto flotante implementado mediante una capa totalmente conectada.

P: Denota un atajo de punto flotante implementado mediante una capa de agrupamiento.

Q: Denotan atajos implementados mediante una capa totalmente conectada cuantizadas a 8 bits.

4.5. Resultados preliminares con MNIST.

Para MNIST, se utilizaron como base modelos de punto flotante de una a tres capas ocultas. Los modelos binarios incluyeron de la misma manera de una a tres capas ocultas, más normalización. Para este conjunto de datos aún no se hacen pruebas exhaustivas de todas las combinaciones, y tampoco se utilizaron capas de abandono que se comenzaron a utilizar en un punto más tardío.

Están pendientes por realizar pruebas exhaustivas de todas las combinaciones por lo que podrían encontrase cambios en las arquitecturas de mejor desempeño. Sin embargo, podemos observar en la tabla 3, que los modelos de punto flotante obtienen la mayor

Tabla 5. Resultados con el conjunto Reuters.

Modelo	Exactitud	Tamaño	Comentario
F-64, 64	0.7845	3778.18	Punto Flotante
B-64N, 64N	0.7850	365.38	Binaria normalizada
B-64N, 64N, QN	0.7796	381.49	Binaria, normalizada, atajo
B-128N, 64N, 64N, Q	0.7787	407.05	Binaria normalizada, atajo
B-64N, 64N, 64N, Q	0.7685	N/D	Binaria normalizada, atajo
B-64N, 64N, P	0.7631	52.74	Binaria normalizada, atajo de agrupamiento
B-64, 64	0.7311	128.87	Binaria equivalente

exactitud entre todos los modelos, y los tres modelos binarios equivalentes la menor exactitud.

De los modelos modificados, el modelo de dos capas con atajo cuantizado (B-128N, 88N, Q) es el que tiene el mayor desempeño seguido del modelo de tres capas con normalización (B-128N, 128N, 128N), y en tercer lugar el modelo (B-128N, 128N, Q).

Estos resultados sugieren que, para la tarea de clasificación de imágenes, los atajos cuantizados representan una mejora que, aunque más costosa que una red sin atajos, da mejores resultados. También podemos notar que tres capas con normalización y sin cuellos de botella obtienen buenos resultados.

4.6. Resultados preliminares con IMDB

El desempeño con el conjunto de datos de IMDB resulta muy diferente al de MNIST. En la tabla 4 podemos notar un comportamiento bastante diferente, en donde dos de los perceptrones binarios equivalentes logran un buen desempeño, seguidos de un perceptrón de dos capas con atajo cuantizado y otro de 3 capas con atajo cuantizado también.

4.7. Resultados preliminares con el conjunto Reuters

Para el conjunto de datos de Reuters, se prueban modelos de dos y tres capas, cuyos resultados se muestran en la tabla 5.

4.8. Resultados preliminares con el conjunto CIFAR10

El conjunto de datos CIFAR10, supera las posibilidades de clasificación de los perceptrones, con los cuales se puede logran una exactitud máxima de 0.59 siempre y cuando se haga un preprocesamiento más extenso de la entrada, incluyendo aumento de las características y recorte de la imagen [33]. Sin embargo, en la tabla 6 se presentan las pruebas realizadas sin preprocesamiento para mostrar las bondades del modelo propuesto.

Se puede observar que los modelos B256N, 256N, P y el B-256N, 256N, 256N, P logran una exactitud de 0.4507 y 0.4580 respectivamente. Se puede notar además que los modelos sin modificaciones obtienen los promedios más bajos.

Tabla 5. Resultados con CIFAR10.

Modelo	Exactitud	Tamaño	Comentario
F-256	0.4224	3083.04	Punto flotante
F-256, 256	0.4576	3340.04	Punto flotante
F-256, 256, 256	0.4350	3597.04	Punto flotante 3 capas
B-256N, 256N, P	0.4507	464.04	Binaria, normalizada
B-256N, 256N, QN	0.4301	891.04	Binaria, normalizada, atajo
B-256N, P	0.4061	451.04	Binaria normalizada, atajo de agrupamiento
B-256N, QN	0.3755	880.04	Binaria, normalizada, atajo
B- 256N, 256N, 256N, PN	0.4580	N/D	Binaria normalizada, atajo de agrupamiento
B- 256N, 256N, 256N, Q	0.4201	N/D	Binaria normalizada, atajo
B-256	0.2612	107.04	Binaria equivalente
B-256, 256	0.1000	116.04	Binaria equivalente
B-256, 256, 256	0.1000	125.04	Binaria equivalente

5. Conclusiones y trabajo futuro

En conclusión, se realizaron pruebas sobre variantes del modelo de perceptrón multicapa binario residual con cuatro diferentes conjuntos de datos.

De los experimentos realizados se puede notar que efectivamente los perceptrones multicapa binarios equivalentes en general tienen menor desempeño que los de punto flotante.

También podemos observar que al agregar capas de normalización la exactitud del perceptrón mejora, y si adicionalmente se utilizan atajos siempre mejora la exactitud del perceptrón, lo cual confirma que estos mejoran el flujo de información. Un atajo cuantizado supera a una capa adicional en el desempeño, aunque podemos notar que es más costoso en el tamaño del modelo.

Aunque falta realizar más experimentos, podemos adelantar que el modelo de perceptrón multicapa binario residual cumple su función de ser una alternativa sencilla de implementar y siempre proporciona mejores resultados que el perceptrón binario simple equivalente al de punto flotante.

Hasta donde tenemos información, este tipo de atajos no se utilizado en modelos de perceptrón multicapa, son más comunes en modelos convolucionales, en los que se utilizan para regenerar la información de entrada al concluir un bloque de convolución.

Una vez que se terminen de realizar todas las series de experimentos, podremos contar con información suficiente para intentar un análisis de correlación y realizar un análisis de sensibilidad sobre los datos recolectados, que se espera arrojen más información sobre la exactitud del modelo.

A futuro, se continuará trabajando en redes neuronales binarias, pero ampliando al campo a redes convolucionales. Aunque el perceptrón multicapa es un modelo flexible, existen aplicaciones que se benefician más con modelos más sofisticados.

Referencias

1. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. The MIT Press (2016)
2. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, vol. 65, no. 6, pp. 386–408 (1958) doi: 10.1037/h0042519
3. Chollet, F.: Deep learning with python. Manning Publications Co (2017)
4. Aggarwal, C. C.: Artificial intelligence: A textbook. Springer Nature Switzerland AG (2021) doi: 10.1007/978-3-030-72357-6
5. Pokhrel, S.: 4 Popular model compression techniques explained. Xailient (2022) xai lient.com/blog/4-popular-model-compression-techniques-explained/
6. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. Advances in Neural Information Processing Systems, vol. 2, pp. 598–605 (1989)
7. Han, S., Pool, J., Tran, J., Dally, W. J.: Learning both weights and connections for efficient neural networks (2015) doi: 10.48550/ARXIV.1506.02626
8. Neill, J. O.: An overview of neural network compression (2020) doi: 10.48550/ ARXIV.2006.0366
9. Cheng, Y., Wang, D., Zhou, P., Zhang, T.: Model compression and acceleration for deep neural networks: the principles, progress, and challenges. IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 126–136 (2018) doi: 10.1109/msp.2017.2765695
10. Novac, P., Hacene, G. B., Pegatoquet, A., Miramond, B., Gripon, V.: Quantization and deployment of deep neural networks on microcontrollers. Sensors, vol. 21, no. 9, pp. 2984 (2021) doi: 10.3390/s21092984
11. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or –1 (2016) doi: 10.48550/ARXIV.1602.02830
12. Yuan, C., Agaian, S. S.: A comprehensive review of binary neural network. Artificial Intelligence Review (2023) doi: 10.1007/s10462-023-10464-w
13. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: XNOR-Net: ImageNet classification using binary convolutional neural networks (2016) doi: 10.48550/ARXIV.1603.05279
14. Simons, T., Lee, D.: A review of binarized neural networks. Electronics, vol. 8, no. 6, pp. 661 (2019) doi: 10.3390/electronics8060661
15. Qin, H., Gong, R., Liu, X., Bai, X., Song, J., Sebe, N.: Binary neural networks: A survey. Pattern Recognition, vol. 105, pp. 107281 (2020) doi: 10.1016/j.patcog.2020.107281
16. Bethge, J., Yang, H., Bornstein, M., Meinel, C.: Back to simplicity: how to train accurate BNNs from scratch? (2019) doi: 10.48550/ARXIV.1906.08637
17. Alizadeh, M., Fernández-Marqués, J., Lane, N. D., Gal, Y.: An empirical study of binary neural networks' optimisation. In: International Conference on Learning Representations, Conference Blind Submission (2018)
18. Hagan, M. T., Demuth, H. B., Beale, M. H., de-Jesús, O.: Neural network design. Martin Hagan, 2nd Edition (2014)
19. Redfern, A. J., Zhu, L., Newquist, M. K.: BCNN: A binary CNN with all matrix ops quantized to 1-bit precision. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2021) doi: 10.1109/cvprw53098.2021.00518
20. Courbariaux, M., Bengio, Y., David, J.: BinaryConnect: Training deep neural networks with binary weights during propagations. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 2, pp. 3123–3131 (2015) doi: 10.48550/ ARXIV.1511.00363
21. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: Training neural networks with low precision weights and activations. Journal of Machine Learning Research, vol. 18, no. 1, pp. 6869–6898 (2017)

22. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients (2016) doi: 10.48550/ARXIV.1606.06160
23. Bethge, J., Yang, H., Bornstein, M., Meinel, C.: BinaryDenseNet: Developing an architecture for binary neural networks. In: IEEE/CVF International Conference on Computer Vision Workshop, pp. 1951–1960 (2019) doi: 10.1109/iccvw.2019.00244
24. Bethge, J., Bartz, C., Yang, H., Chen, Y., Meinel, C.: MeliusNet: Can binary neural networks achieve MobileNet-level accuracy? (2020) doi: 10.48550/ARXIV.2001.05936
25. Yuan, C., Agaian, S. S.: A comprehensive review of binary neural network. Artificial Intelligence Review (2023) doi: 10.1007/s10462-023-10464-w
26. Hinton, G., Teleman, T.: Divide the gradient by a running average of its recent magnitude (2012) www.aminer.org/pub/5b076eb4da5629516ce741dc/lecture-rmsprop-divide-the-gradient-by-a-running-average-of-its-recent
27. Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., Cheng, K.: Bi-Real Net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm. In: Computer Vision - European Conference on Computer Vision, pp. 747–763 (2018) doi: 10.1007/978-3-030-01267-0_44
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
29. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015) doi: 10.1109/cvpr.2015.7298594
30. Chen, T., Zhang, Z., Ouyang, X., Liu, Z., Shen, Z., Wang, Z.: "BBB - BN = ?": Training binary neural networks without batch normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4619–4629 (2021)
31. Google: Keras: The python deep learning API (2023) keras.io
32. Li, H., De, S., Xu, Z., Studer, C., Samet, H., Goldstein, T.: Training quantized nets: a deeper understanding. In: 31st Conference on Neural Information Processing Systems (2017) doi: 10.48550/ARXIV.1706.02379
33. Parvathi, C.: Classification of cifar-10 dataset using a multi-layer perceptron model and a convolutional neural network model (2023) github.com/ parvathic/ml-cifar10

ISSN 1870-4069

Algoritmo de estimación de distribuciones para la segmentación de imágenes por umbralización multinivel

Jorge Armando Ramos Frutos, Israel Miguel Andrés, Diego Oliva

Centro de Innovación Aplicada en Tecnologías Competitivas, Centro de Pogrados, México

{jramos.estudiantepicyt,imiguel}@ciatec.mx, diego.oliva@academicos.udg.mx

Resumen. La segmentación de imágenes es una operación en el procesamiento digital de imágenes que divide una imagen en regiones que son completamente distintas. Existen distintos métodos para segmentar una imagen y uno de los métodos se denomina umbralización multinivel. La segmentación de imágenes por umbralización multinivel consiste en optimizar una función objetivo (Otsu) que evalúa el grado de diferencia entre las regiones. La función objetivo Otsu se utiliza para separar las imágenes en diferentes regiones maximizando la varianza entre las clases. En el presente artículo el objetivo es desarrollar un Algoritmo de Estimación de Distribuciones (EDA) que separe las imágenes maximizando la función objetivo Otsu. La experimentación se lleva a cabo utilizando un conjunto de imágenes de la Universidad de Berkeley. Se utiliza el EDA para segmentar las imágenes y se realiza una evaluación de la función objetivo (Otsu), la razón señal-ruido ($PSNR$) y, el índice de similitud de estructuras ($SSIM$). En la experimentación se observa que el valor de Otsu, $PSNR$ y $SSIM$ son más grandes al dividir la imagen en una mayor cantidad de regiones. Los resultados que se obtienen con el EDA hasta este punto de la investigación no se han comparado; bajo las mismas condiciones, con otros algoritmos, pero se observa un desempeño adecuado en la segmentación de las imágenes propuestas porque se alcanzan a distinguir las diferentes regiones de las imágenes segmentadas. La mayoría de los algoritmos metaheurísticos utilizan hiper-parámetros para su funcionamiento y el EDA no requiere de esa información para trabajar, esto es lo novodoso en propuesta del EDA para resolver el problema de segmentación de imágenes por umbralización multinivel.

Palabras clave: Algoritmo de estimación de distribuciones, segmentación, umbralización multinivel.

Estimation of Distribution Algorithm for Image Segmentation by Multilevel Thresholding

Abstract. Image segmentation is an operation in digital image processing that divides an image into regions that are completely distinct. There are different methods to segment an image and one of the methods is called

multilevel thresholding. Image segmentation by multilevel thresholding consists of optimizing an objective function (Otsu) that evaluates the degree of difference between regions. The Otsu objective function is used to separate the images into different regions by maximizing the variance between the classes. In this article the objective is to develop an Estimation of Distribution Algorithm (EDA) that separates the images by maximizing the Otsu objective function. The experimentation is carried out using a set of images from the University of Berkeley. The EDA is used to segment the images and an evaluation of the objective function (Otsu), the signal-to-noise ratio ($PSNR$) and the structure similarity index ($SSIM$) is performed. In experimentation it is observed that the value of Otsu, $PSNR$ and $SSIM$ are larger when dividing the image into a larger number of regions. The results obtained with the EDA up to this point of the investigation have not been compared; under the same conditions, with other algorithms, but an adequate performance is observed in the segmentation of the proposed images because it is possible to distinguish the different regions of the segmented images. Most of the metaheuristic algorithms use hyper-parameters for their operation and the EDA does not require this information to work, this is the novelty in the EDA proposal to solve the image segmentation problem by multilevel thresholding.

Keywords: Estimation of distribution algorithm, segmentation, multilevel thresholding.

1. Introducción

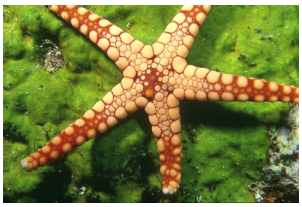
El procesamiento digital de imágenes es un proceso de aplicación de varias operaciones en una imagen para obtener información útil o generar una imagen con mejores características [1, 18]. Una de las operaciones consiste en segmentar la imagen en regiones [7]. Se pueden mencionar cinco técnicas para segmentar una imagen [15]: umbralización [3, 8, 12], métodos basados en bordes [5], métodos basados en regiones [19], redes neuronales artificiales [10, 13], entre otros.

La segmentación por umbralización se lleva a cabo con un histograma que grafica la cantidad de píxeles en cada nivel de iluminación de una imagen en escala de grises. La información del histograma se utiliza para el cálculo de la función objetivo que se utiliza como criterio de separación de las regiones que comprenden la imagen. Otsu [20] es una función objetivo que se obtiene calculando la varianza entre clases de las regiones en que se quiere dividir la imagen.

Al maximizar la función objetivo Otsu se obtienen las regiones de la imagen con la mayor varianza entre clases y son semejantes en la intensidad de sus píxeles. En la segmentación de imágenes por umbralización el objetivo principal es particionar la imagen en clases homogéneas, donde los elementos de cada clase comparten propiedades en común.

Las metaheurísticas son utilizadas para encontrar las posiciones de los umbrales que maximizan la función objetivo para que las regiones de las imágenes sean completamente distintas entre sí. Existe gran cantidad de metaheurísticas que se utilizan para resolver el problema de segmentación de imágenes por umbralización.

a) Mandril b) Pimientos c) Aeroplano



d) Calzado e) Estrella f) Avión

Fig. 1. Imágenes utilizadas para obtener resultados con el EDA.

Algunas metaheurísticas son: algoritmos genéticos (AG) [11], algoritmo diferencial evolutivo (DE) [6], colonia de abejas (ABC) [9], entre otras. El Algoritmo de Estimación de Distribuciones (EDA) es una metaheurística basada en el AG que cambia las operaciones de cruza y mutación por una operación de muestreo.

El EDA genera una población inicial con la codificación necesaria para cada tipo de problema, selecciona algunos individuos utilizando como criterio el valor que otorga al ser evaluado y genera un muestreo, al final realiza una operación de reemplazo para generar una población nueva. Wang *et al.* utiliza un EDA para resolver el problema de segmentación, pero ellos sólo resuelven el problema con dos umbrales.

Existen algunas métricas adicionales que ayudan con la evaluación de las metaheurísticas en el problema de segmentación de imágenes por múltiples umbrales como lo son la razón señal-ruido ($PSNR$, por sus siglas en inglés) y el índice de similitud de estructuras ($SSIM$, por sus siglas en inglés) [2]. Con estas métricas es posible tomar decisiones y ver cómo se comportan los algoritmos con diferentes imágenes y en diferentes niveles de umbralización. No se ha reportado en la literatura un EDA para la umbralización multinivel y que se analice utilizando las métricas.

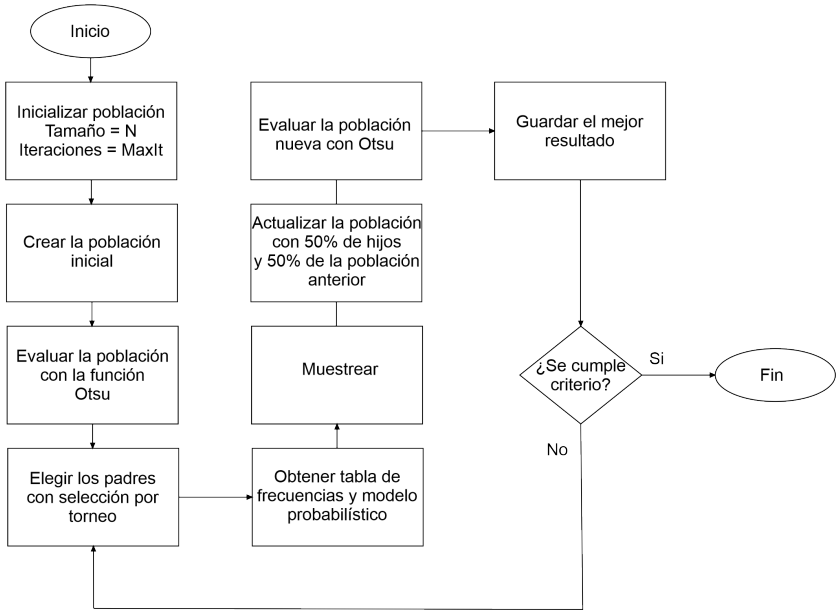


Fig. 2. Diagrama de flujo del EDA.

Por lo tanto, se debe establecer un algoritmo que maximice la función objetivo Otsu y que otorgue buenos resultados en las métricas de similitud de estructuras. El contenido de este documento se divide en 6 secciones.

La Sección 2 habla del Marco Teórico de la investigación el cual define el problema de segmentación por umbralización y se definen las operaciones realizadas en el EDA. En la Sección 3, se colocan las condiciones experimentales con una descripción de la computadora que se utilizó para correr el EDA y el conjunto de imágenes utilizado para obtener resultados.

La Sección 4 se establecen las métricas ya mencionadas y se presenta un diagrama de flujo del funcionamiento del EDA. Por último en la Sección 5, se muestran los resultados obtenidos con el EDA propuesto para la función Otsu y las métricas.

2. Marco teórico

En esta sección se hablará de la segmentación por Otsu y del EDA. Se explicará el problema de segmentación por umbralización y la función objetivo que se propone para evaluar el EDA. También se describen las operaciones que realiza el EDA para llevar a cabo la optimización.

2.1. Segmentación multinivel por el método de Otsu

En la segmentación multinivel por umbralización se trata de dividir una imagen en regiones lo más distintas posibles. Existen algunos métodos como Otsu [16] que utilizan un criterio de separabilidad de las regiones.

Tabla 1. Resultados obtenidos con el EDA.

Imagen	Th	Otsu	PSNR	SSIM	Mejor
Mandril	2	1222.27886	37.5637616	0.73919049	[99 146]
	3	1326.55088	45.9666508	0.84477138	[90 127 162]
	4	1369.14673	56.0732861	0.92796017	[75 103 131 163]
	5	1386.89423	60.9947407	0.95013415	[68 101 131 152 175]
Pimientos	2	1912.52694	48.2122084	0.84821121	[83 146]
	3	2038.36763	56.2551189	0.9100436	[77 124 169]
	4	2122.08977	60.0245722	0.93436348	[61 104 136 170]
	5	2141.54018	67.1628198	0.95907517	[62 100 132 156 182]
Aeroplano	2	1706.30453	39.5384322	0.86342072	[118 176]
	3	1777.11955	49.4709847	0.92512015	[100 146 187]
	4	1818.97398	68.3978748	0.98467007	[73 111 162 202]
	5	1847.10466	73.9106507	0.99257013	[71 113 145 176 202]
Huarache	2	1994.1147	38.5790877	0.81679751	[131 190]
	3	2059.91767	46.6695781	0.87818936	[96 145 193]
	4	2081.87112	52.9217747	0.90974377	[90 133 171 202]
	5	2100.42416	56.4903036	0.92509441	[80 109 147 185 204]
Estrella	2	2562.82841	42.1057814	0.65284185	[86 159]
	3	2796.15621	49.505938	0.76500284	[68 119 179]
	4	2882.21268	54.6438462	0.81816347	[55 92 135 183]
	5	2927.87502	58.9060441	0.85393707	[50 83 118 150 196]
Avión	2	721.289127	51.2531776	0.79990287	[71 139]
	3	772.025719	58.5738219	0.86842808	[53 88 142]
	4	795.310477	61.3557121	0.88538171	[51 79 103 151]
	5	809.106223	68.0878796	0.91964491	[36 60 84 106 148]

Otsu maximiza la variación entre las regiones utilizando el histograma de frecuencias para la obtención de las varianzas de clases. La Ec. 1 muestra cómo se establecen las clases para la umbralización multinivel:

$$\begin{aligned} C_0 &= \{I_{ij} \in I(x,y) | 0 \leq I_{ij} \leq th_1 - 1\}, \\ C_1 &= \{I_{ij} \in I(x,y) | th_1 \leq I_{ij} \leq th_2 - 1\}, \\ &\vdots \\ C_k &= \{I_{ij} \in I(x,y) | th_k \leq I_{ij} \leq th_{L-1}\}, \end{aligned} \tag{1}$$

donde C_k es la clase k-ésima que corresponde a la región k-ésima, $I(x,y)$ corresponde a los píxeles de la imagen original, th_k es el umbral k-ésimo que se coloca en el histograma para que la imagen original sea segmentada.

Para aplicar el método de Otsu [16], es necesario obtener medias y varianzas esperadas de las distintas regiones, estó ayudará a obtener el criterio de separabilidad de las regiones en la imagen. La Ec. 2 muestra como es posible calcular la media de las clases [17]:

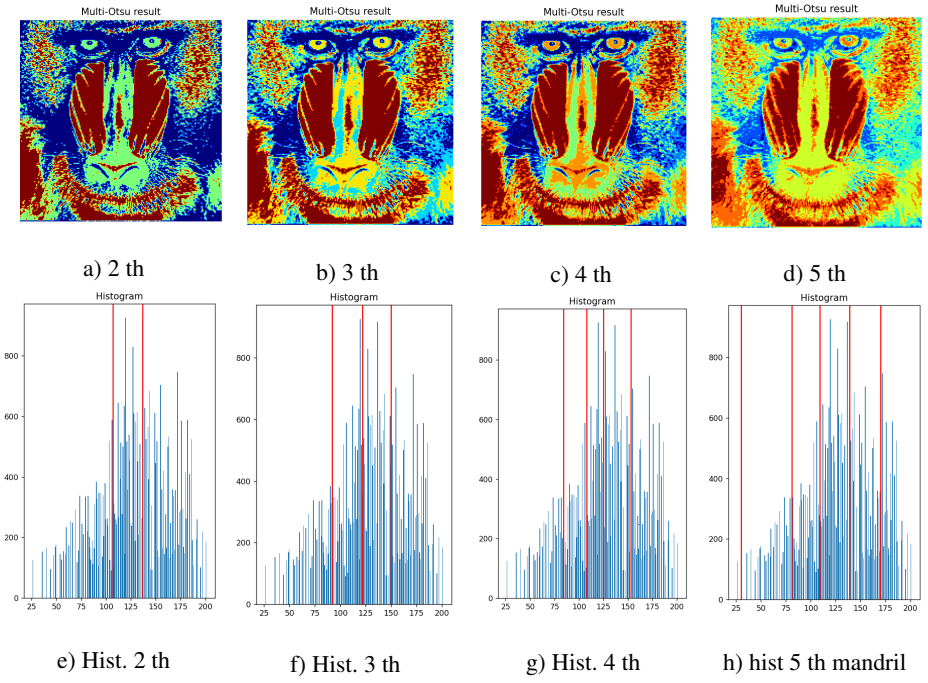


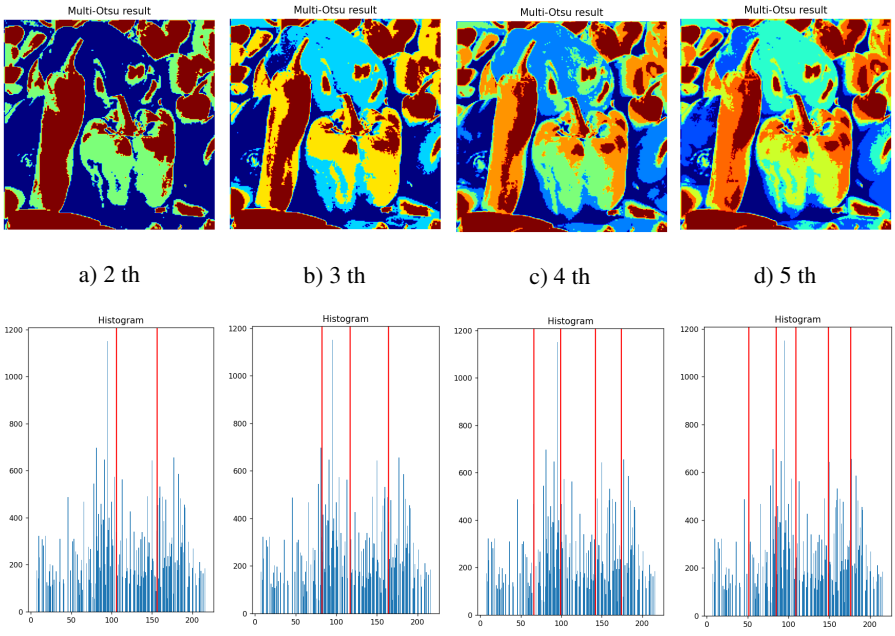
Fig. 3. Imágenes umbralizadas mandril.

$$\begin{aligned}
\mu_0 &= \sum_{i=0}^{t_1-1} iP(i|C_0) = \frac{1}{\omega_0} \sum_{i=1}^{t_1-1} ip_i, \\
\mu_1 &= \sum_{i=t_1}^{t_2-1} iP(i|C_1) = \frac{1}{\omega_1} \sum_{i=t_1}^{t_2-1} ip_i, \\
&\vdots \\
\mu_k &= \sum_{i=k}^{L-1} iP(i|C_k) = \frac{1}{\omega_k} \sum_{i=k}^{L-1} ip_i,
\end{aligned} \tag{2}$$

donde i es el nivel de intensidad que va de $[0, 255]$, p_i es la probabilidad de encontrar el nivel de intensidad i-ésimo dada la Ec. 3:

$$p_i = \frac{n_i}{N}, \tag{3}$$

donde n_i es la cantidad de píxeles en el nivel i y N es la cantidad de píxeles en la imagen. Mientras que las varianzas de las clases se obtienen con la Ec. 4. Se puede ver que con el criterio de Otsu se utiliza la estadística para ver qué tan diferentes son las regiones entre sí:



a) 2 th b) 3 th c) 4 th d) 5 th

e) Hist. 2 th f) Hist. 3 th g) Hist. 4 th h) Hist. 5 th

Fig. 4. Imágenes umbralizadas pimientos.

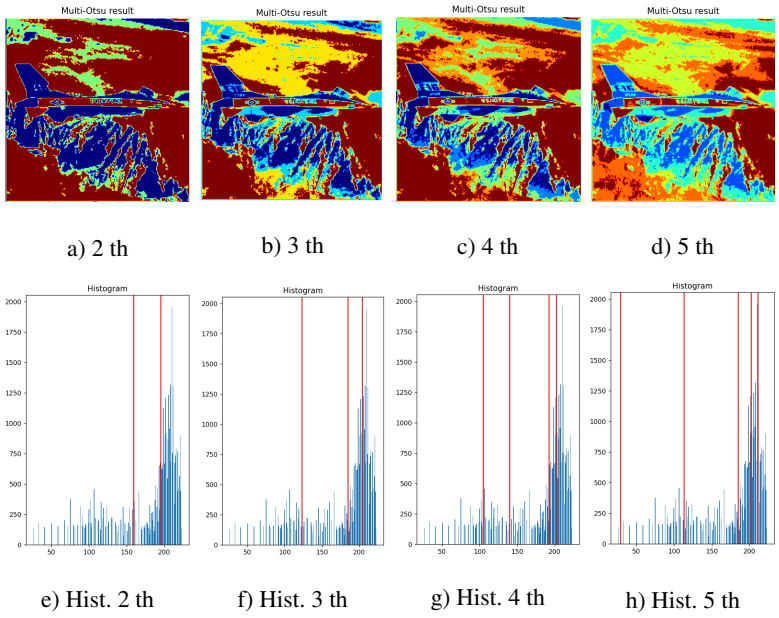
$$\begin{aligned}
\sigma_0^2 &= \sum_{i=0}^{t_1-1}(i-\mu_0)^2 P(i|C_0) = \frac{1}{\omega_0}\sum_{i=1}^{t_1-1}(i-\mu_0)^2 p_i,\\
\sigma_1^2 &= \sum_{i=t_1}^{t_2-1}(i-\mu_0)^2 P(i|C_1) = \frac{1}{\omega_1}\sum_{i=t_1}^{t_2-1}(i-\mu_1)^2 p_i,\\
&\vdots\\
\sigma_k^2 &= \sum_{i=k}^{L-1}(i-\mu_0)^2 P(i|C_k) = \frac{1}{\omega_k}\sum_{i=k}^{L-1}(i-\mu_k)^2 p_i.
\end{aligned} \tag{4}$$

Por lo tanto, se debe maximizar el criterio de Otsu para múltiples regiones como se muestra en la Ec. 5:

$$th_1^*, th_2^*, \cdots, th_K^* = \underset{th_1^*, th_2^*, \cdots, th_K^*}{\max} F(th_1^*, th_2^*, \cdots, th_K^*). \tag{5}$$

2.2. Algoritmo de estimación de distribuciones

El EDA que se propone realiza tres operaciones: selección, muestreo y reemplazo, después de generar una población inicial y evaluarla [14]. Cada una de las operaciones mencionadas se describe a continuación. La Ec. 6 se propone para generar la población inicial de umbrales. La primer población se genera de manera aleatoria [20] utilizando números pseudoaleatorios con una distribución uniforme:



a) 2 th b) 3 th c) 4 th d) 5 th

e) Hist. 2 th f) Hist. 3 th g) Hist. 4 th h) Hist. 5 th

Fig. 5. Imágenes umbralizadas aeroplano.

$$th_{ij} = L_j + \text{rand}_j \times (U_j - L_j), \tag{6}$$

donde th_{ij} es el j-ésimo umbral del poblador i-ésimo, $rand_j$ es un número pseudoaleatorio entre cero y uno, L_j y U_j son los límites inferior y superior, respectivamente, estos equivalen van de $[0, 255]$. Cuando se tiene la población inicial, se evalúan los pobladores con la función objetivo Otsu, que se describe en la Sección 2.1.

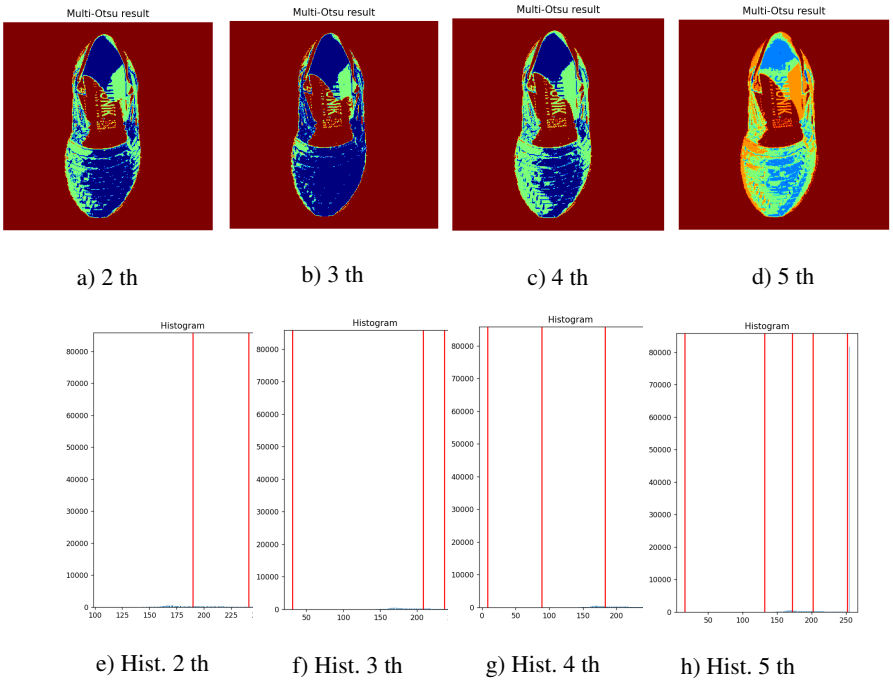
Teniendo las evaluaciones de los pobladores se realiza la operación de selección por torneo. En la selección por torneo se toman dos pobladores al azar y se elige el que tenga una evaluación de la función objetivo más grande, en este caso porque es un problema de maximización. De la operación de selección se genera un conjunto de padres que será utilizado en el muestreo.

Para el muestreo se genera una tabla de frecuencias de los umbrales que aparecen en cada posición de los padres que se seleccionan para obtener las frecuencias relativas con la Ec. 7. Estas frecuencias relativas se usan para muestrear, aquel que tiene una mayor frecuencia relativa tiene una probabilidad más alta de ser elegido para formar parte de los hijos:

$$p(th_{kj}) = \frac{n(th_{kj})}{N}. \tag{7}$$

La probabilidad de obtener el umbral j de la clase k es $p(th_{kj})$, $n(th_{kj})$ es la cantidad de veces que aparece el umbral j en la clase k y, N es el número de pobladores que se definen y se generan en la población inicial.

Por último, se realiza un reemplazo con el $50\,\%$ de la población con la que se trabajo y el $50\,\%$ de los hijos que se obtienen por muestreo.



a) 2 th b) 3 th c) 4 th d) 5 th

e) Hist. 2 th f) Hist. 3 th g) Hist. 4 th h) Hist. 5 th

Fig. 6. Imágenes umbralizadas calzado.

3. Métodos y materiales

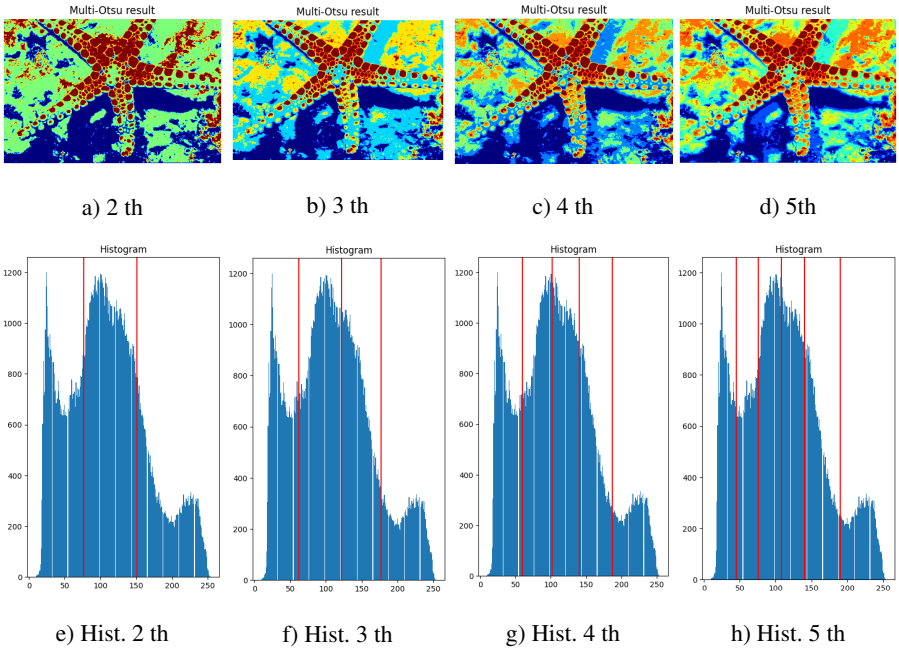
En esta sección, se muestran las condiciones experimentales y las imágenes con que se trabajan para evaluar el EDA.

3.1. Condiciones experimentales

Para llevar a cabo las corridas del EDA se utilizó una computadora con sistema operativo Windows 11, con procesador Core i7 de décima primera generación, 12 GB de memoria RAM, el lenguaje de programación Python versión 3.10.

3.2. Conjunto de imágenes

Se utilizaron cuatro imágenes para obtener resultados con el EDA propuesto (Fig. 1). Tres de las imágenes con un tamaño de 200 por 200 píxeles (Mandril, Pimientos y, Aeroplano), la imagen del Huarache con un tamaño de 300 por 400 píxeles y, las últimas dos imágenes (estrella de mar y avión) de 481 por 321. Este conjunto de imágenes se trabajó en escala de grises para realizar la operación de segmentación.



a) 2 th b) 3 th c) 4 th d) 5th

e) Hist. 2 th f) Hist. 3 th g) Hist. 4 th h) Hist. 5 th

Fig. 7. Imágenes umbralizadas estrella.

4. Implementación

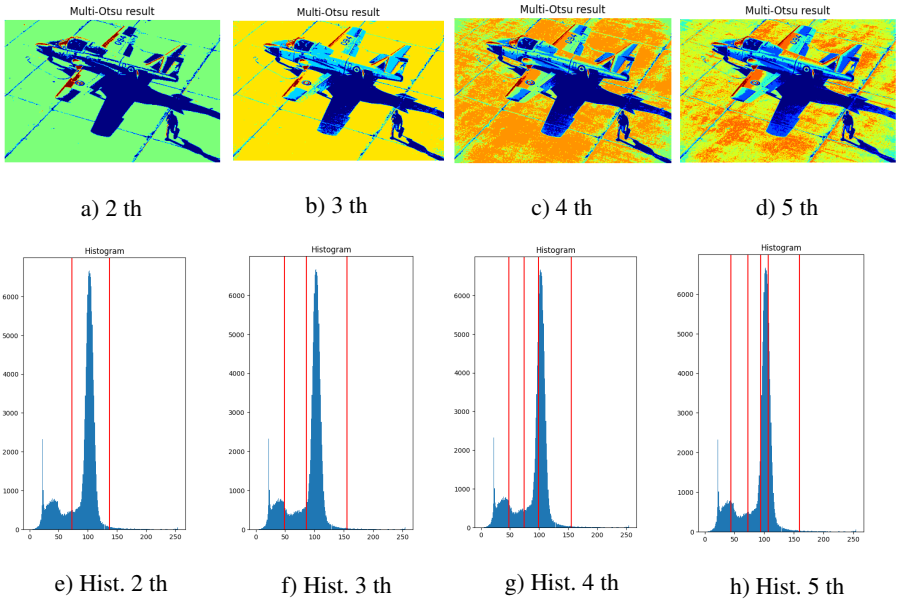
En esta sección se muestra el diagrama de flujo del algoritmo propuesto y algunas métricas que se utilizan para medir la calidad de la segmentación como: tiempo de procesamiento, razón señal-ruido (PSNR, por sus siglas en inglés) [4] y, el índice de similitud de estructuras (SSIM, por sus siglas en inglés) [21].

4.1. Algoritmo propuesto

El algoritmo propuesto inicia con definir la cantidad de iteraciones y la cantidad de pobladores generados en la población inicial. Después de elegir estos parámetros, se genera la población inicial. De la población inicial que se genera, se realiza la selección por torneo. Con los individuos electos se realiza un muestreo para obtener algunos hijos.

Dados estos hijos se realiza el reemplazo con el $50\,\%$ de hijos y $50\,\%$ de pobladores correspondientes a la población anterior para la generación de una nueva población. Si se cumple el criterio de paro (cantidad de iteraciones) el algoritmo deja de trabajar. Lo ya descrito se muestra en Fig. 2. El EDA, en comparación con otras metaheurísticas, no requiere de hiper-parámetros para su operación.

Los hiper-parámetros que se utilizan en otras metaheurísticas se deben configurar para obtener óptimos resultados. En el problema de umbralización multinivel no se ha reportado el EDA como método de solución. Por lo que se propone un EDA que genera su modelo probabilístico basado en la frecuencia de ocurrencia de los números entre 0 y 255 en cada nivel.



a) 2 th b) 3 th c) 4 th d) 5 th

e) Hist. 2 th f) Hist. 3 th g) Hist. 4 th h) Hist. 5 th

Fig. 8. Imágenes umbralizadas avión.

4.2. Métricas

El PSNR mostrado en la Ec. 8 es la razón señal-ruido que mide la diferencias que existen entre la imagen original y la imagen segmentada en promedio:

$$\text{PSNR} = 20 \log \left(\frac{255}{\text{RMSE}} \right), \tag{8}$$

donde $RMSE$, que se utiliza para obtener el $PSNR$, es el Error Cuadrado Medio calculado por la Ec. 9:

$$\text{RMSE} = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(I(i,j) - S(i,j) \right)^2}, \tag{9}$$

donde M, N son los tamaños de imagen, $I(i,j)$ es la imagen original y $S(i,j)$ es la imagen segmentada en un nivel particular. Valores altos en el PSNR indican mejores resultados en la segmentación por umbralización. El SSIM se calcula con la Ec. 10:

$$\text{SSIM} = \frac{(2\mu_I \mu_S + c_1)(2\sigma_{IS} + c_2)}{(\mu_I^2 + \mu_S^2 + c_1)(\sigma_I^2 + \sigma_S^2 + c_2)}, \tag{10}$$

donde I y S son la imagen original y la imagen segmentada, respectivamente. μ_I y μ_S son las medias de la imagen original y la imagen segmentada, respectivamente. σ_I^2 y σ_S^2 son las varianzas, σ_{IS} es la covarianza. Las constantes $c_1 = (0.01 \times 255)^2$ y $c_2 = (0.03 \times 255)^2$ se emplean para estabilizar el indicador [21]. Un $SSIM$ cercano a uno indica buenos resultados en la segmentación.

5. Resultados

Para los resultados se corrió el algoritmo en cada imagen 500 iteraciones, esto se realizó para cada imagen y para cada umbral. Se plantearon 4 niveles de umbralización ($2, 3, 4$ y, 5) en la segmentación de las seis imágenes. El Cuadro 1 muestra los resultados obtenidos con el EDA.

En la primera columna se hace referencia a la imagen que se segmento del conjunto de imágenes de la Fig. 1, la siguiente columna indica la cantidad de umbrales que se utilizaron, en la tercera columna se muestran los resultados del EDA en la función objetivo, las columnas cuatro y cinco muestran dos de las métricas que se mencionan (PSNR y SSIM) y, en la última columna se muestra la combinación de umbrales con que se tiene un valor máximo de Otsu.

Se puede observar en el Cuadro 1 que al incrementar la cantidad de regiones en la imagen el valor de la función Otsu incrementa, al igual que el $PSNR$ y el $SSIM$. Las imágenes segmentadas utilizando los valores de umbralización con que se optimiza Otsu y sus histogramas se muestran en la Figuras 3, 4, 5, 6, 7 y 8.

Cada imagen tiene un histograma único y el EDA segmentó las diferentes regiones de las imágenes probadas. La característica principal del EDA es que no requiere de parámetros para su funcionamiento, el algoritmo propuesto genera un modelo de probabilidad con el que genera las poblaciones que convergen en el óptimo de la función objetivo.

La contribución de este trabajo es utilizar un algoritmo que no utilice hiper-parámetros para obtener la solución del problema de segmentación por umbralización multinivel. Otros algoritmos utilizan hiper-parámetros en sus operaciones, algunos ejemplos son: los AG utilizan operadores de cruza y mutación, en ellos se deben elegir las probabilidades de cruza y mutación, en el DE se define un factor de escala y una probabilidad de cruza, entre otros.

Al tener hiper-parámetros en sus operaciones, se deben elegir de forma cuidadosa para obtener resultados óptimos, mientras que en el EDA no es necesario realizar esa optimización.

6. Conclusiones

El EDA es un algoritmo que no utiliza hiper-parámetros para operar. La operación de selección y el muestreo hacen que el EDA tenga la capacidad de explotación del espacio de búsqueda porque la selección por torneo elige a los mejores individuos del conjunto que conforma la población.

Con base en estos individuos que se eligen se realiza el muestreo tomando en cuenta la tabla de frecuencias que se genera por cada umbral. Por último, la operación de reemplazo ayuda a que el EDA considere las poblaciones anteriores. Se observa que el EDA resuelve el problema de segmentación por umbralización utilizando un modelo probabilístico en la generación de soluciones.

La función objetivo y las dos métricas ($PSNR$ y $SSIM$) son mayores cuando la cantidad de regiones a segmentar aumenta. Los resultados de la segmentación de las imágenes con el EDA se muestran con un mapa de colores en la imagen que logra

separar las diferentes regiones de un conjunto de seis imágenes que se extraen de una base de datos de la Universidad de Berkeley.

Referencias

1. Abd Elaziz, M., Ewees, A. A., Oliva, D.: Hyper-heuristic method for multilevel thresholding image segmentation. Expert Systems with Applications, vol. 146, pp. 113201 (2020) doi: 10.1016/j.eswa.2020.113201
2. Abd Elaziz, M., Lu, S.: Many-objectives multilevel thresholding image segmentation using Knee evolutionary algorithm. Expert systems with Applications, vol. 125, pp. 305–316 (2019) doi: 10.1016/j.eswa.2019.01.075
3. Abd Elaziz, M., Lu, S., He, S.: A multi-leader whale optimization algorithm for global optimization and image segmentation. Expert Systems with Applications, vol. 175, pp. 114841 (2021) doi: 10.1016/j.eswa.2021.114841
4. Agrawal, S., Panda, R., Bhuyan, S., Panigrahi, B. K.: Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. Swarm and Evolutionary Computation, vol. 11, pp. 16–30 (2013) doi: 10.1016/j.swevo.2013.02.001
5. Al-Amri, S. S., Kalyankar, N., Khamitkar, S.: Image segmentation by using edge detection. International Journal on Computer Science and Engineering, vol. 2, no. 3, pp. 804–807 (2010)
6. Bhandari, A. K.: A novel beta differential evolution algorithm-based fast multilevel thresholding for color image segmentation. Neural Computing and Applications, vol. 32, no. 9, pp. 4583–4613 (2020)
7. Chauhan, R., Joshi, R.: Comparative evaluation of image segmentation techniques with application to MRI segmentation. In: Proceedings of International Conference on Machine Intelligence and Data Science Applications, pp. 521–537 (2021) doi: 10.1007/978-981-33-4087-9_44
8. Dinkar, S. K., Deep, K., Mirjalili, S., Thapliyal, S.: Opposition-based laplacian equilibrium optimizer with application in image segmentation using multilevel thresholding. Expert Systems with Applications, vol. 174, pp. 114766 (2021) doi: 10.1016/j.eswa.2021.114766
9. Ewees, A. A., Abd Elaziz, M., Al-Qaness, M. A., Khalil, H. A., Kim, S.: Improved artificial bee colony using sine-cosine algorithm for multi-level thresholding image segmentation. IEEE Access, vol. 8, pp. 26304–26315 (2020) doi: 10.1109/ACCESS.2020.2971249
10. He, B., Hu, W., Zhang, K., Yuan, S., Han, X., Su, C., Zhao, J., Wang, G., Wang, G., Zhang, L.: Image segmentation algorithm of lung cancer based on neural network model. Expert Systems, vol. 39, no. 3, pp. e12822 (2022) doi: 10.1111/exsy.12822C
11. Hilali-Jaghdam, I., Ishak, A. B., Abdel-Khalek, S., Jamal, A.: Quantum and classical genetic algorithms for multilevel segmentation of medical images: A comparative study. Computer Communications, vol. 162, pp. 83–93 (2020) doi: 10.1016/j.comcom.2020.08.010
12. Houssein, E. H., Helmy, B. E., Oliva, D., Elngar, A. A., Shaban, H.: A novel black widow optimization algorithm for multilevel thresholding image segmentation. Expert Systems with Applications, vol. 167, pp. 114159 (2021) doi: 10.1016/j.eswa.2020.114159
13. Jha, D., Riegler, M. A., Johansen, D., Halvorsen, P., Johansen, H. D.: Doubleu-net: A deep convolutional neural network for medical image segmentation. In: IEEE 33rd International Symposium on Computer-Based Medical Systems, pp. 558–564 (2020) doi: 10.48550/arXiv.2006.04868
14. Larranaga, P.: A review on estimation of distribution algorithms. Estimation of Distribution Algorithms, pp. 57–100 (2002) doi: 10.1007/978-1-4615-1539-5_3

15. Oliva, D., Abd-Elaziz, M., Hinojosa, S.: Metaheuristic algorithms for image segmentation: Theory and applications (2019) doi: 10.1007/978-3-030-12931-6
16. Otsu, N.: A threshold selection method from gray-level histograms. In: IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, pp. 62–66 (1979) doi: 10.1109/TSMC.1979.4310076
17. Suresh, S., Lal, S.: An efficient cuckoo search algorithm based multilevel thresholding for segmentation of satellite images using different objective functions. Expert Systems with Applications, vol. 58, pp. 184–209 (2016) doi: 10.1016/j.eswa.2016.03.032
18. Teoh, T. T., Rong, Z.: Python for data analysis. Artificial Intelligence with Python, pp. 107–122 (2022) doi: 10.1007/978-981-16-8615-3_7
19. Tilton, J. C.: Image segmentation by region growing and spectral clustering with a natural convergence criterion. In: IEEE International Geoscience and Remote Sensing, vol. 4, pp. 1766–1768 (1998) doi: 10.1109/IGARSS.1998.703645
20. Wang, W., Duan, L., Wang, Y.: Fast image segmentation using two-dimensional Otsu based on estimation of distribution algorithm. Journal of Electrical and Computer Engineering, vol. 2017 (2017) doi: 10.1155/2017/1735176
21. Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P.: Image quality assessment: from error visibility to structural similarity. In: IEEE Transactions on Image Processing, vol. 13, pp. 600–612 (2004) doi: 10.1109/TIP.2003.819861

ISSN 1870-4069

Identificación de plagas y enfermedades en cultivos de citrus latifolia usando aprendizaje profundo

Alonso Hernández-Mora[1], Roberto Ángel Meléndez-Armenta[1], Carlos Alberto Ochoa-Ortiz[2], Irahan Otoniel José-Guzmán[1]

[1] Tecnológico Nacional de México,
Instituto Tecnológico Superior de Misantla,
División de Estudios de Posgrado e Investigación,
México

[2] Universidad Autónoma de Ciudad Juárez,
México

{222t0539, ramelendeza, iojoseg}@itsm.edu.mx,
alberto.ochoa@uacj.mx

Resumen. México es el segundo mayor productor de limón en el mundo; sin embargo, la mayoría de los agricultores no tienen el conocimiento suficiente para poder realizar la correcta identificación de plagas y enfermedades en este tipo de cultivos, por lo que son propensos a cometer errores y realizar acciones equivocadas que resulten en posibles daños en sus cultivos y con esto pérdidas económicas. Es por esto por lo que el objetivo del presente estudio es generar un modelo de aprendizaje profundo que sea capaz de identificar plagas y enfermedades en los cultivos de citrus latifolia. Para lograr este objetivo se utilizaron dos modelos de aprendizaje profundo preentrenados, se compararon y se eligió el que demostrara mejor rendimiento en cuanto a clasificación. Los resultados obtenidos por ambos modelos fueron satisfactorios, sin embargo, se eligió el modelo MobilNet ya que obtuvo una precisión del 99% en la clasificación de las distintas plagas y enfermedades tratadas en el estudio.

Palabras clave: Aprendizaje profundo, citrus latifolia, aprendizaje por transferencia, detección de enfermedades.

Using Deep Learning to Identify Pest and Disease Detection in Citrus Latifolia Crops

Abstract. México is the second largest producer of lemons in the world; however, most farmers do not have enough knowledge to be able to correctly identify pests and diseases in this type of crops, so they are prone to make mistakes and perform wrong actions that result in possible damage to their crops and thus economic losses. For this reason, the objective of the study is to generate a deep learning model capable of identifying pests and diseases in citrus latifolia crops. To achieve this, two pre-trained deep learning models were used, compared and the one with the best classification performance was chosen. The results obtained by both models were satisfactory; however, the MobilNet model

was chosen since it obtained an accuracy of 99% in the classification of the different pests and diseases treated in the study.

Keywords: Deep learning, citrus latifolia, transfer learning, disease detection.

1. Introducción

México ocupa el segundo lugar a nivel mundial en la producción de limón, con una producción cercana a 2.9 millones de toneladas en 2021 de acuerdo con el Servicio de Información Agroalimentaria y Pesquera (SIAP).

La producción de limón en México se divide principalmente en tres variedades: el limón agrio (citrus aurantifolia), el limón italiano (citrus lemon) y el limón persa (citrus latifolia). El limón persa es la variedad más cultivada y la que genera la mayor cantidad de ingresos económicos, con una producción de poco más de 1.5 millones de toneladas en 2021 [1].

No obstante, su producción está expuesta a diversas plagas y enfermedades, como la fumagina, huanglongbing, wood pocket, araña roja, ácaros [2], entre otras, lo que representa un desafío significativo para los agricultores. Por lo tanto, es importante detectar estas plagas y enfermedades en etapas tempranas para minimizar las pérdidas y mejorar la calidad de los frutos.

Sin embargo, la detección suele depender de la experiencia de los agricultores y puede ser propensa a errores, ya que algunas plagas o enfermedades no presentan síntomas visibles en las primeras etapas. Por esta razón, es crucial desarrollar herramientas confiables para detectar plagas y enfermedades, con el fin de tomar medidas efectivas para prevenir y controlar su propagación.

En la detección de plagas y enfermedades en cítricos, como el limón persa, es común utilizar lesiones visibles para poder identificarlas correctamente, dichas lesiones pueden presentarse como manchas oscuras, decoloraciones o moteados [2]. Para apoyar a los agricultores en esta tarea, se han utilizado diferentes enfoques de inteligencia artificial, como Máquinas de Vectores de Soporte (SVMs), Redes Neuronales Convolucionales (CNNs) y K-Nearest-Neighbor KNN, entre otros.

En particular, las CNNs son ampliamente utilizadas en tareas de visión artificial que implican imágenes [3], lo que las convierte en una herramienta común en la detección de plagas y enfermedades en cítricos. Por ejemplo, en [4, 5] se utilizaron modelos de CNN para detectar enfermedades en las hojas y frutos de los cítricos, obteniendo resultados superiores en rendimiento en comparación con otros modelos estudiados.

Además, en [6], se utilizó la arquitectura del modelo ResNet50 junto con el algoritmo AMSR para mejorar la calidad de las imágenes y desarrollar un nuevo modelo de aprendizaje profundo, que demostró una precisión de hasta el 97.95% en pruebas.

Por lo tanto, en este estudio se propone el desarrollo de una aplicación móvil que realice la identificación de plagas y enfermedades en los frutos de citrus latifolia con ayuda de modelos de aprendizaje profundo, con el fin de poder tomar las medidas necesarias de control sobre la plaga o enfermedad identificada y coadyuvar en la toma de decisiones para combatirlas y así reducir las pérdidas económicas en los agricultores.

Tabla 1. Resumen de los algoritmos.

Referencia	Algoritmo	Resultados
[4]	CNN	El modelo propuesto realiza la detección de enfermedades en hojas y frutos de los cítricos, con una precisión del 94,55%.
[5]	CNN	El modelo propuesto demuestra ser mejor que otros modelos como DenseNet y MobileNet, alcanzando una precisión de hasta el 95%.
[6]	CNN	Se entrenó un modelo para clasificar enfermedades en hojas de los cítricos, obteniendo una precisión de hasta el 95,6%.
[7]	MF-RANet	El modelo propuesto obtiene una precisión del 96% en la identificación de enfermedades de los cítricos, esto gracias al uso de un algoritmo de mejora de imágenes.
[8]	MIB	El modelo propuesto identifica distintas enfermedades en frutos y hojas de los cítricos con una precisión de hasta el 98%.
[9]	CNN	Se compararon dos modelos preentrenados y demuestra que los modelos de aprendizaje profundo tienen una sensibilidad mayor al 95% en la identificación de HLB.
[10]	CNN	Se utilizaron modelos preentrenados como AlexNet y VGG junto al optimizador SGDM para la detección de enfermedades en cítricos. La precisión obtenida fue del 94,3%.
[11]	MobileNet	Se utilizó el modelo preentrenado MobileNet junto con el algoritmo IWOA, logrando obtener una precisión de hasta el 99,7% en la identificación de enfermedades de los cítricos.

2. Trabajos relacionados

En la tabla 1 se muestra un resumen de los sistemas de identificación de enfermedades en cítricos basados en técnicas de aprendizaje profundo. Como se puede observar la mayor parte de los algoritmos revisados en la literatura hacen uso de redes neuronales convolucionales.

Los métodos de aprendizaje profundo han sido utilizados por distintos autores en el reconocimiento de enfermedades en las hojas y frutos de los cítricos tal como se puede observar en la tabla anterior. La identificación de plagas y enfermedades en hojas y frutos en cultivos es un tema que ha sido estudiado durante mucho tiempo.

Para esto se han utilizado una amplia variedad de métodos de aprendizaje automático y algoritmos de preprocesamiento de imágenes que pueden aumentar la precisión con la que se identifican estas enfermedades.

En estudios como [3], se menciona que los métodos de aprendizaje profundo brindan soluciones innovadoras a problemas relacionados con la agricultura y que, para este

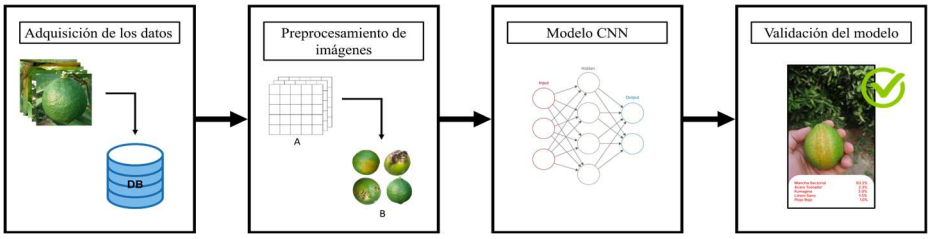


Fig. 1. Metodología propuesta (Fuente: Elaboración propia).

tipo de tareas, las redes neuronales convolucionales son la piedra angular (CNN). En [4, 5, 6] se muestran modelos de CNNs propuestos para la clasificación de enfermedades en hojas y frutos de los cítricos.

Por otro lado, en [7] se hace uso de algoritmos de mejora de imágenes como AMSR, que está basado en el algoritmo retinex multiescala, este fue utilizado para extraer la mayor cantidad de características de las imágenes. El modelo CNN utilizado en este estudio utiliza el modelo ResNet50 para la extracción de características del conjunto de imágenes de enfermedades de cítricos.

Así mismo en [8], se propone un modelo de aprendizaje profundo que hace uso de extracción de características gaussianas para la extracción de capas de varias imágenes, este modelo obtuvo excelentes resultados en la tarea de clasificación de enfermedades en imágenes de cítricos.

Por otro lado, a pesar de que los modelos de CNN representan herramientas que pueden ofrecer buenas soluciones a problemas de clasificación de imágenes, estos requieren de grandes cantidades de datos y consumen demasiados recursos computacionales.

Para situaciones en donde los datos son pocos se suelen utilizar modelos preentrenados [9, 10, 11, 12]. Este tipo de modelos se emplea para evitar que el modelo obtenga un bajo rendimiento al no contar con los datos necesarios para el entrenamiento.

Es importante mencionar que las CNN son muy utilizadas en la identificación de enfermedades en distintos cultivos como arroz [13], tomate [14] y maíz [15]. Además, que se suelen utilizar modelos preentrenados, ya que muchas veces obtener imágenes de enfermedades específicas, en este tipo de cultivos, se vuelve una tarea complicada.

3. Materiales y métodos

3.1. Herramienta Tensorflow

Tensorflow es una biblioteca de código abierto que permite construir y entrenar modelos de aprendizaje automático para distintas tareas, como clasificación de imágenes, procesamiento de lenguaje natural, reconocimiento de voz y detección de objetos [16].

También proporciona acceso a modelos de aprendizaje automático preentrenados, permitiendo la reutilización de modelos entrenados en grandes cantidades de datos y

Tabla 2. Distribución del número de imágenes por clase.

Enfermedad / plaga	Fumagina	Acaro Tostador	Mancha Sectorial	Piojo rojo	Limón sano
Ejemplo					

así reducir el tiempo y recursos necesarios que toma desarrollar nuevas soluciones de aprendizaje automático [17].

Para desarrollar herramientas orientadas a dispositivos móviles, tensorflow ofrece la biblioteca tensorflow lite. Esta biblioteca permite desarrollar modelos de aprendizaje automático que pueden ser utilizados en dispositivos con recursos limitados, como teléfonos inteligentes, tablets, dispositivos IoT (internet de las cosas) y otros sistemas integrados [18].

3.2. Android Studio

Android Studio es un entorno de desarrollo integrado (IDE) utilizado en la creación de aplicaciones móviles para el sistema operativo Android. Este IDE proporciona distintas características y herramientas que facilitan el proceso de desarrollo de aplicaciones. Algunas de las herramientas que ofrecen son el editor de código, depurador, emulador de dispositivos, diseñador de interfaces, plantillas, entre otras [19]. Esto lo convierte en una herramienta indispensable para los desarrolladores, ya que permite crear aplicaciones de manera efectiva y eficiente.

3.3. Procedimiento

En la figura 1, se observan los pasos de la metodología propuesta para el desarrollo de la aplicación móvil y el modelo de aprendizaje profundo. Obtención de los datos: En la primera etapa se realizó la recolección de las imágenes necesarias para poder llevar a cabo los experimentos.

El conjunto de imágenes utilizado en el estudio fue recolectado manualmente de distintas huertas de la región de Martínez de la Torre, Veracruz. Este conjunto estaba compuesto por un total de 5000 imágenes, de 5 categorías distintas (1000 imágenes por cada una), a color con un tamaño de 1280x1280 pixeles y en formato JPEG.

Preprocesamiento de los datos: Para el preprocesamiento de las imágenes, se retiró el fondo de cada una de ellas para evitar que afectara la extracción de características y se redimensionaron en dos tamaños distintos.

Durante el preprocesamiento, el conjunto de datos resultante se dividió en dos partes: la primera parte contenía el 80% de las imágenes totales del conjunto original para el entrenamiento del modelo, y la segunda parte contenía el 20% restante para realizar pruebas. Modelo CNN: Para el modelo de CNN se tomaron en cuenta dos modelos preentrenados proporcionados Tensorflow Hub que son MobileNet e Inception [17].

acaro_tostador	199	0	0	0	1
fumagina	0	199	0	0	1
limon_sano	0	0	200	0	0
mancha_sectorial	0	0	0	200	0
piojo_rojo	0	0	1	0	199

acaro_tostador	198	1	0	0	1
fumagina	0	198	0	0	2
limon_sano	0	0	199	0	1
mancha_sectorial	0	0	1	199	0
piojo_rojo	1	0	0	2	197

Fig. 2. Matriz de confusión de MobileNet e Inception.

Estos han sido utilizados en otros estudios para la identificación de enfermedades en cultivos como cítricos [12], arroz [13], tomate [14], entre otros [15]. Los resultados que estos modelos obtuvieron en la clasificación de enfermedades en estudios anteriores demostraron que pueden ser de gran utilidad y proporcionar herramientas que ayuden a los agricultores a realizar este tipo de tareas.

Para desarrollar el modelo de CNN se utilizó el lenguaje de programación Python, debido a que es un lenguaje que ofrece muchas herramientas para este tipo de problemas y además es compatible con la librería de tensorflow.

En esta etapa se realizaron ajustes a los modelos, esto para poder identificar las clases dentro de nuestro conjunto de imágenes. Dentro de los ajustes realizados se redimensionaron las imágenes a un tamaño específico para cada uno de los dos modelos, por ejemplo, para MobileNet las imágenes se redimensionaron a un tamaño de 224x244 pixeles, mientras que para Inception 299x299 pixeles.

Validación: En la última etapa de la metodología se realizan pruebas de los modelos utilizando una aplicación Android, que realiza la identificación de plagas y enfermedades en tiempo real. Esta muestra en pantalla la clase a la que pertenece la imagen capturada y muestra el porcentaje de precisión.

4. Resultados

Como se mencionó anteriormente en este estudio se utilizaron MobileNet e Inception. Ambas arquitecturas deberían ser capaces de identificar las plagas y enfermedades en los frutos de citrus latifolia con una precisión aceptable. Es importante mencionar que los resultados obtenidos por estos modelos se compararon, esto con el fin de poder elegir el que demuestre una mejor precisión en la clasificación.

Los datos de entrenamiento fueron los mismos para ambos algoritmos (tabla 2), aunque durante la etapa de preprocesamiento, el redimensionamiento de las imágenes fue distinto en cada modelo, ya que cada uno recibe como entrada imágenes en tamaños distintos por ejemplo MobileNet recibe imágenes de 224 x 224 píxeles e Inception imágenes de 299 x 299 píxeles. Esto es importante, pues de no hacer una correcta redimensión, se pueden generar errores a la hora de entrenar los modelos.

Con los ajustes necesarios, se procedió a realizar el entrenamiento de los modelos durante 25 épocas. Tras el entrenamiento, ambos modelos obtuvieron resultados satisfactorios, los modelos fueron evaluados con ayuda de una matriz de confusión, en la figura 2 se pueden observar las matrices resultantes de cada modelo. A partir de esta

Tabla 3. Métricas de evaluación.

Modelo	Exactitud	Precisión	Especificidad
MobileNet	0.9975	0.997	0.998
Inception	0.9945	0.991	0.992

se obtuvo la precisión de los modelos tal como se observa en la tabla 3. Se puede observar que el modelo MobileNet obtuvo una precisión superior al modelo Inception.

A partir de la matriz de confusión se obtuvieron las métricas para evaluar a cada uno de los modelos están son exactitud, precisión y especificidad.

Los resultados demuestran que el modelo MobileNet es superior a Inception en todas las métricas, a pesar de que ambos modelos obtienen resultados similares se considera mejor utilizar MobileNet, ya que en tiempo de entrenamiento este utiliza aproximadamente 2 horas e Inception demora hasta 8 horas.

El modelo MobileNet se implementó en la aplicación móvil, ya que obtuvo mejores resultados en la tarea de clasificación. La figura 3 muestra los resultados obtenidos por el modelo al capturar imágenes en tiempo real desde un dispositivo Android. En la imagen se puede apreciar que el funcionamiento de la aplicación es bastante simple, pues solo consta de una pantalla en donde se realiza la captura de imágenes en tiempo real y se muestra la clase a la que se asocia dicha imagen junto al porcentaje de precisión.

La aplicación desarrollada en Android Studio en conjunto con el modelo seleccionado representa una herramienta muy útil para ayudar a los agricultores a identificar plagas y enfermedades en los cultivos de citrus latifolia con una precisión de hasta el 99.7%.

5. Conclusiones y trabajo a futuro

Los resultados de la investigación sobre la clasificación de imágenes de plagas y enfermedades en frutos de citrus latifolia son muy prometedores. Se ha demostrado que el uso combinado de una aplicación móvil y un modelo de aprendizaje profundo, en particular una red neuronal convolucional, permite identificar con gran precisión plagas y enfermedades que afectan a los frutos de citrus latifolia.

El modelo MobileNet ha alcanzado una precisión impresionante del 99.7% en esta tarea. Estos hallazgos pueden tener importantes implicaciones para la industria agrícola, ya que una herramienta eficaz para la identificación temprana y el control de plagas y enfermedades en los cultivos de citrus latifolia puede aumentar significativamente la producción y reducir los costos.

MobileNet e Inception son dos arquitecturas de redes neuronales convolucionales utilizadas para la clasificación de imágenes en aplicaciones de visión por computadora.

Si bien ambas arquitecturas han demostrado ser efectivas en la clasificación de imágenes, es notable que MobileNet es una mejor opción si se desea realizar una implementación en una aplicación móvil, pues esta se diseñó para ser más eficiente dentro de entornos en donde los recursos son limitados.

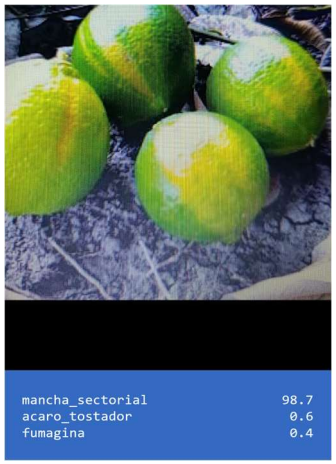


Fig. 3. Precisión de clasificación en la App.

Como trabajo futuro se pretende realizar una aplicación móvil más completa, que sea multiplataforma para poder llegar a más agricultores y que pueda ofrecer recomendaciones a los agricultores sobre qué acciones realizar para poder llevar un mejor control de las plagas y enfermedades identificadas en sus cultivos de citrus latifolia.

Otro punto importante que se desea tratar en un futuro es en relación con las plagas y enfermedades que puede identificar la aplicación móvil junto con el modelo, se podría expandir el conjunto de datos a otras plagas y enfermedades que afectan tanto los frutos de citrus latifolia, como sus hojas. Con esto se podría brindar una mejor herramienta a los agricultores.

Referencias

1. Servicio de información agroalimentaria y pesquera (SIAP).: Anuario estadístico de la producción agrícola 2021 (2022) https://nube.siap.gob.mx/cierreagricola/
2. Secretaria de agricultura y desarrollo rural (SADER): Plagas y enfermedades comunes del limón (2021) https://www.gob.mx/agricultura/es/articulos/plagas-y-enfermedades-comunes-del-limon
3. Dhanya, V. G., Subeesh, A., Kushwaha, N. L., Vishwakarma, D., Kumar, T. N., Ritika, G., Snigh, A. N.: Deep learning based computer vision approaches for smart agricultural applications. Artificial Intelligence in Agriculture, vol. 6, pp. 211–229 (2022) doi: 10.1016/j.aiia.2022.09.007
4. Khattak, A., Asghar, M. U., Batool, U., Asghar, M. Z., Ullah, H., Al-Rakhami, M., Gumaei, A.: Automatic detection of citrus fruit and leaves diseases using deep neural network model. IEEE Access, vol. 9, pp. 112942–112954 (2021) doi: 10.1109/ACCESS.2021.3096895
5. Janarthan, S., Thuseethan, S., Rajasegarar, S., Lyu, Q., Zheng, Y., Yearwood, J.: Deep metric learning based citrus disease classification with sparse data. IEEE Access, vol. 8, pp. 162588–162600 (2020) doi: 10.1109/ACCESS.2020.3021487
6. Ramadhan, M. I., Suyanto, S.: Detection of disease in citrus plants through leaf images using a convolutional neural network. In: 2021 3rd International Conference on Electronics

Representation and Algorithm (ICERA), pp. 71–76 (2021) doi: 10.1109/ICERA53111.2021.9538757
7. Yang, R., Liao, T., Zhao, P., Zhou, W., He, M., Li, L.: Identification of citrus diseases based on AMSR and MF-RANet. Plant Methods, vol. 18, no. 113 (2022) doi: 10.1186/s13007-022-00945-4
8. Brindha, G. M., Karishma, K. K., Nivetha, J., Vidhya, B.: Automatic detection of citrus fruit diseases using MIB classifier. In: 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 1111–1116 (2022) doi: 10.1109/ICESC54411.2022.9885702
9. Gómez-Flores, W., Garza-Saldaña, J. J., Varela-Fuentes, S. E.: A Huanglongbing detection method for orange trees based on deep neural networks and transfer learning. In IEEE Access, vol. 10, pp. 116686–116696 (2022) doi: 10.1109/ACCESS.2022.3219481
10. Elaraby, A., Hamdy, W., Alanazi, S.: Classification of citrus diseases using optimization deep learning approach. Computational Intelligence and Neuroscience, vol. 2022 (2022) doi: 10.1155/2022/9153207
11. Hassam, M., Khan, M. A., Armghan, A., Althubiti, S. A., Alhaisoni, M., Alqahtani, A., Kadry, S., Kim, Y.: A single stream modified MobileNet V2 and whale controlled entropy based optimization framework for citrus fruit diseases recognition. IEEE Access, vol. 10, pp. 91828–91839 (2022) doi: 10.1109/ACCESS.2022.3201338
12. Berger, J., Preussler, C., Agostini, J. P.: Identification of Huanglongbing symptoms in citrus leaves by deep learning techniques. Electronic Journal of SADIO, vol. 18, no. 2, pp. 2–20 (2019)
13. Masykur, F., Adi, K., Nurhayati, O. D.: Classification of paddy leaf disease using MobileNet model. In: IEEE 8th International Conference on Computing, Engineering and Design (ICCED), pp. 1–4 (2022) doi: 10.1109/ICCED56140.2022.10010535
14. Baheti, H., Thakare, A., Bhople, Y., Darekar, S., Dodmani, O.: Tomato plant leaf disease detection using inception V3. Intelligent Systems and Applications. Lecture Notes in Electrical Engineering, vol. 959, pp. 49–60 (2023) doi: 10.1007/978-981-19-6581-4_5
15. Saeed, Z., Raza, A., Qureshi, A. H., Haroon-Yousaf, M.: A multi-crop disease detection and classification approach using CNN. In: International Conference on Robotics and Automation in Industry (ICRAI), pp. 1–6 (2021) doi: 10.1109/ICRAI54018.2021.9651409
16. Tensorflow: Crea modelos de aprendizaje automático de nivel de producción con TensorFlow (2023) https://www.tensorflow.org/?hl=es-419
17. Tensorflow Hub (2023) https://tfhub.dev/
18. TensorFlow: Implementa modelos de aprendizaje automático en dispositivos móviles y perimetrales Tensorflow Lite (2023) https://www.tensorflow.org/lite?hl=es-419
19. Android Studio.: Conceptos básicos de la plataforma. Developers (2023) https://developer.android.com/about?hl=es-419

ISSN 1870-4069

Estimación del mapa de anisotropía fraccional y difusividad media en materia blanca utilizando transformers

Daniel Bandala Álvarez, Jorge Perez Gonzalez

Universidad Nacional Autónoma de México,
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
México

ing.dbandala@gmail.com, jorge.perez@iimas.unam.mx

Resumen. Los algoritmos de visión computacional se han convertido en una tecnología clave en muchos campos, desde sistemas de asistencia al conductor para automóviles hasta interacciones de usuario en videojuegos. Así, un campo que ha encontrado aplicaciones importantes y que ha impulsado todo una nueva área en la ingeniería es el procesamiento de imágenes médicas, ya sea para simplemente limpiar las componentes de ruido en las muestras obtenidos o para extraer e inferir conocimiento nuevo a partir de estas, tal como en la detección temprana de cáncer de mama, el diagnostico exacto de retinopatía diabética o detección y localización de tumores. Esto último respalda la opción de utilizar algoritmos de aprendizaje computacional y redes neuronales profundas para la asistencia e investigación médica. En el presente trabajo se propone utilizar una red neuronal tipo autoencoder basada en módulos Transformer para la estimación de los mapas de anisotropía fraccional y difusividad media. Se presentan los resultados obtenidos sobre un conjunto de sujetos control y se demuestra que el uso de arquitecturas basadas en autoatención son capaces de modelar estructuras altamente complejas como las fibras de axones en el cerebro.

Palabras clave: Tensor de difusión, materia blanca, anisotropía fraccional, difusividad media, transformers.

Fractional Anisotropy and Mean Diffusivity Maps Estimation in White Matter Using Transformers

Abstract. Computer vision algorithms have become a key technology in many fields, from driver assistance systems for automotive to user interactions in video games. Among these fields, medical image processing has particularly benefited from the application of these techniques. By utilizing computer vision algorithms, it is possible to enhance medical images by removing noise components and extracting valuable information from them. This, in turn, enables early detection of breast cancer, precise diagnosis of diabetic retinopathy, and accurate detection and localization of tumors. Consequently, the integration of machine learning

algorithms and deep neural networks has become pivotal in both medical care and research. In this study, we propose a novel approach to medical image processing utilizing an autoencoder neural network. The key innovation lies in the incorporation of Transformer modules within the network architecture for the estimation of fractional anisotropy and mean diffusivity maps. The results obtained on a set of control subjects are presented and it is demonstrated that the implementation of self-attention based architectures are able to model highly complex structures such as axon fibers across the brain tissue.

Keywords: Diffusion tensor, white matter, fractional anisotropy, mean diffusivity, transformers.

1. Introducción

Las neuroimágenes son una herramienta estándar en el estudio clínico de pacientes con patologías neurodegenerativas o en el análisis exploratorio prequirúrgico para cirugías cerebrales. En el área de la investigación, se han realizado avances contundentes debido al desarrollo de técnicas de extracción y procesamiento de datos a partir de imágenes estructurales y funcionales del cerebro [19].

En concreto, las imágenes ponderadas por difusión (MRI-DWI) permiten estudiar la estructura del tejido de la materia blanca debido a su alta sensibilidad al desplazamiento de las moléculas de agua con una precisión del orden de micras. Debido a la energía térmica, estas moléculas se encuentran moviéndose constantemente en direcciones aleatorias en el tejido cerebral en un proceso denominado autodifusión [8, 9].

Y, debido a que las estructuras celulares pueden impedir el movimiento del agua a una escala microscópica, estas señales actúan como evidencia de la microestructura del tejido. Existen varias metodologías para modelar la difusión en sistemas biológicos, cada uno con distintas suposiciones y nivel de complejidad.

El modelo más simple es asumir una difusión libre, caracterizada por un único coeficiente de difusión. Sin embargo, debido a que las medidas de la difusividad claramente dependen de los parámetros experimentales, se incorporó el uso del coeficiente de difusión aparente (ADC) [5], en donde se calcula la difusividad relativa a una señal sin dirección de difusión del mismo escáner.

Por otra parte, la materia blanca cerebral se compone principalmente de axones, que son responsables de transmitir información eléctrica y química a través del cerebro. Son largas proyecciones celulares que se extienden desde las células nerviosas, o neuronas, y permiten que las señales eléctricas se propaguen entre terminales nerviosas, donde se comunican con otras neuronas o células del cuerpo.

Estos axones cerebrales están organizados en fascículos o haces, que a menudo se denominan tractos. Estos tractos están recubiertos por una capa de mielina, un material graso que actúa como un aislante eléctrico y acelera la velocidad de transmisión de los impulsos nerviosos. Además, se ha demostrado que la difusión en la materia blanca cerebral es dependiente de la dirección [15], lo que contiene información con la orientación de las fibras de axones dentro de las regiones de materia blanca en conjuntos de fibras coherentes.

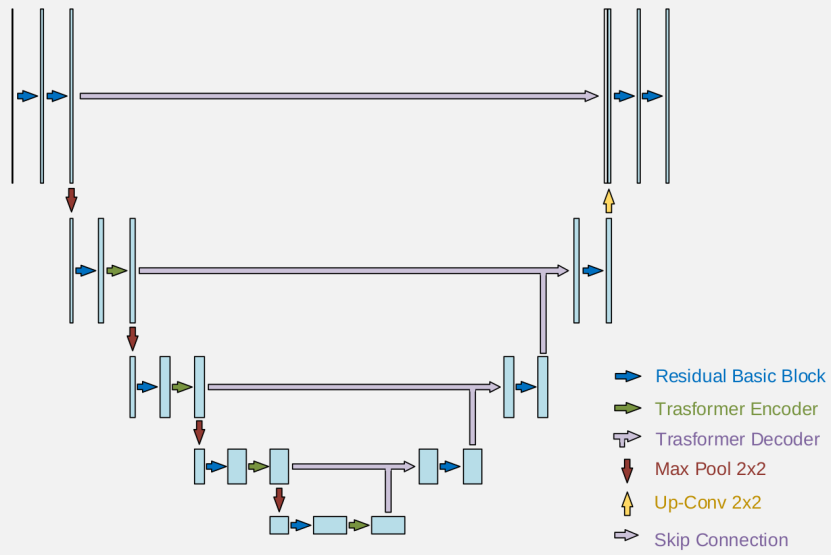


Fig. 1. Arquitectura UTNet implementada para la estimación de los mapas del tensor de difusión [4].

Esta información direccional de las señales de difusión presenta la única posibilidad de inferir las fibras de materia blanca dentro de un cerebro de un paciente vivo y de forma completamente no invasiva. Por tanto, el modelo del tensor de difusión se ha introducido como una extensión del modelo del coeficiente de difusión aparente con la capacidad de describir la difusión anisotrópica, es decir, la difusión que no es igual en todas las direcciones, y del cual se pueden extraer medidas como la difusividad media, la anisotropía fraccional y las direcciones principales de la difusión [11]. Gracias a la simplicidad de esta técnica, es utilizada como un estándar desde hace un par de décadas en especial en el área clínica.

1.1. Estado del arte

En general, los algoritmos para el seguimiento de los tractos o tractografía son simples, aunque sujetos a un número significante de limitaciones, por lo tanto resulta interesante el desarrollo de nuevos modelos o técnicas computacionales que permitan aumentar las capacidades y reducir la complejidad de los métodos convencionales para el ajuste del tensor de difusión en la microestructura de tejidos biológicos [20].

En [18] se presenta una arquitectura que utiliza tres modelos de redes neuronales convolucionales para la estimación de la orientación principal de la difusión, la segmentación de los tractos de interés y la segmentación de los puntos de inicio y finalización del seguimiento de la tractografía.

En [7] se presenta un modelo similar basado en redes neuronales convolucionales que ajusta la relación no lineal entre las imágenes ponderadas por difusión y sus correspondientes mapas de difusión, utilizando únicamente 6 señales de difusión.

De manera similar, en [10] proponen una red neuronal tipo encoder-decoder llamada AGYnet para segmentar la materia blanca en el cerebro, esta toma como información de entrada imágenes estructurales ponderadas en tiempo T1 y los mapas de las direcciones principales de difusión. La arquitectura cuenta con compuertas de atención en la parte de decoder que permite combinar las características obtenidas de cada tipo de entrada.

Esto permite mantener las características originales y resaltar aquellas que realicen una mayor contribución de información a la estimación del modelo [13]. En consecuencia, gracias a estos mecanismos de auto atención ha surgido una alternativa a las redes neuronales convolucionales para el procesamiento de imágenes: los Transformers [16].

Así, se han presentado propuestas donde se combinan bloques convolucionales y bloques tipo Transformers para la segmentación de imágenes médicas [4, 1].En ese contexto, Karimi et al. [6] han propuesto una arquitectura basada únicamente en Transformers para la estimación de los coeficientes del tensor de difusión obteniendo buenos resultados.

Y, de la misma forma, en el presente trabajo se propone un modelo de redes neuronales basado en Transformers y bloques convolucionales para la estimación de los mapas de anisotropía fraccional y difusividad media. A su vez, en comparación al método convencional, este modelo reduce significativamente el tiempo de procesamiento y el número de señales de difusión necesarias para la estimación de estos mapas.

2. Materiales y métodos

En esta sección se presenta la metodología de desarrollo del presente trabajo de investigación. Se describe la base de datos utilizada para la construcción del modelo y se muestran las características de la arquitectura implementada para la estimación de los mapas del tensor de difusión. Finalmente, se mencionan las métricas utilizadas para la evaluación de la arquitectura.

2.1. Base de datos

La base de información de imágenes ponderadas por difusión se obtienen de dos proyectos incluidos en el Laboratorio de Neuroimágenes en la Universidad del Sur de California (USC): El Proyecto de Conectoma Humano (Human Connectome Project, HCP) y la Iniciativa de Neuroimágenes de la Enfermedad de Alzheimer (Alzheimer's Disease Neuroimaging Initiative, ADNI). La base de información de la Iniciativa de Neuroimágenes para la Enfermedad de Alzheimer cuenta con más de tres mil imágenes cerebrales de Resonancia Magnética, Tensor de Difusión, Tomografía Computarizada y por Emisión de Positrones.

De este conjunto de información se han recopilado muestras de 30 sujetos control con 38 señales de difusión cada una. Cada volumen contiene 80 imágenes de $256x256$ píxeles. Por otra parte, el Proyecto de Conectoma Humano (HCP) contiene muestras de 35 sujetos control en un rango de edad entre 20 y 89 años.

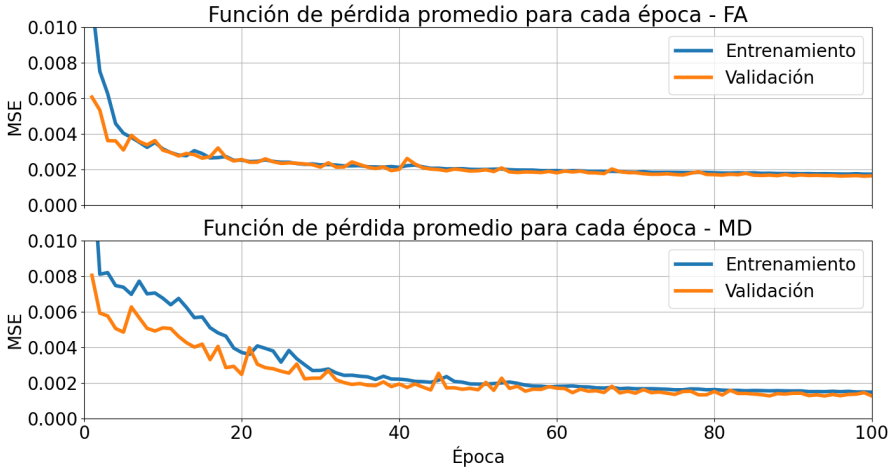


Fig. 2. Función de pérdida de las primeras 100 épocas durante el entrenamiento del modelo para la estimación de los mapas del tensor de difusión.

Cada volumen contiene 96 imágenes de $140x140$ píxeles, en donde las señales de difusión se han adquirido en cortes axiales oblicuos y con gradientes de difusión monopolares. Por tanto, en total se utilizan las muestras de 65 sujetos control para el entrenamiento, validación y evaluación del modelo.

2.2. Procesamiento con FSL

Para la construcción de la base de datos de referencia se utilizó el procesamiento estándar con la librería FSL (FMRIB Software Library) [3]. Este procedimiento consiste en extraer el conjunto de todas las imágenes que han sido obtenidas con un único valor de gradiente b, filtrar artificios generados por el propio escáner y, finalmente, ajustar el tensor de difusión mediante una método de mínimos cuadrados.

Además, se extrae todo el tejido no cerebral de las imágenes, mediante una mascara binaria, de modo que se ajusta el tensor de difusión para la región del cerebro únicamente. Así, a partir de cada imagen de tensor de difusión se ha extraído el mapa de fracción de anisotropía y difusividad media, los cuales servirán como referencia para el entrenamiento y validación del modelo propuesto.

2.3. Arquitectura de red neuronal

El diseño del modelo propuesto para la estimación de los mapas del tensor de difusión en imágenes cerebrales se basa en la arquitectura UTNet para segmentación binaria, presentada en [4], la cual se compone de bloques convolucionales, módulos Transformer, bloques de submuestreo y capas totalmente conectadas. Este modelo contiene una parte de contracción o compresión, como se muestra en la Figura 1, donde se reduce la dimensión espacial de las imágenes y se incrementa el número de canales o características; y una fase de expansión, empleada para la reconstrucción de las imágenes de referencia a sus dimensiones espaciales originales.

En la etapa de codificación de las imágenes se encuentran dos bloques convolucionales con conexión residual que extraen las primeras características locales de las señales de difusión y aumentan en un factor de 4 el número de canales de entrada de las imágenes. Se extraen estas últimas características con un submuestreo del máximo valor, utilizando una ventana de tamaño $2x2$, y se introducen estos tensores de características en el primer codificador tipo Transformer.

Este toma las dimensiones espaciales de la imagen para proyectarlas sobre una dimensión $x_{t_n} \in \mathbb{R}^{HW \times d}$, donde HW representa el tamaño de la imagen y d es la dimensión del mapa de características, obteniendo así una secuencia de características de píxeles que capturan las principales propiedades de las imágenes.

Se repite este proceso desde el submuestreo hasta el codificador Transformer en 3 niveles y, en este punto, las dimensiones espaciales de las imágenes se ha reducido en un factor de 16, mientras que el número de canales o características ha aumentado en la misma proporción.

Entonces, a partir del espacio latente a la salida de la etapa de codificación del modelo, se comienza la reconstrucción de las imágenes de salida utilizando decodificadores tipo Transformer, el cual toma dos tensores de entrada, correspondientes a las características del espacio latente de la etapa de contracción y las características de mayor resolución mediante una conexión residual.

Así, en lugar de integrar el módulo de autoatención sobre los mapas de características que se han extraído de manera automática o manual como se plantea en la arquitectura ViT [2] y TransUNet [1], se aplica el módulo Transformer a cada nivel del codificador y decodificador para modelar la dependencia de largo alcance en múltiples escalas.

Además, dado que las imágenes son datos muy estructurados, la mayoría de los píxeles de los mapas de características de alta resolución dentro de una ventana local comparten características similares, excepto en el caso de las regiones limitantes. En consecuencia, el cálculo de la atención por pares entre todos los píxeles es ineficiente y redundante. Desde una perspectiva teórica, la autoatención es esencialmente de bajo rango para las secuencias largas [17], lo que indica que la mayor parte de la información se concentra en los valores singulares más grandes.

Por tanto, se utilizan los bloques de multiatención eficientes empleados en [14], en donde se aplican dos transformaciones para proyectar la clave y el valor del mapa de características $K, V \in \mathbb{R}^{n \times d}$ en una codificación de menor dimensión $\overline{K}, \overline{V} \in \mathbb{R}^{k \times d}$, en donde $k = hw \ll n, h$ y w es el tamaño espacial reducido del mapa de características después del submuestreo. Entonces, la autoatención eficiente presentada en [4] se formula como:

$$\mathrm{EfficientAttention}(Q, \overline{K}, \overline{V}) = \mathrm{softmax}\left(\frac{Q\overline{K}}{\sqrt{d}}\right)\overline{V}. \tag{1}$$

De esta manera, la complejidad computacional del cálculo de los mapas de atención se reduce a $O(nkd)$. En particular, la proyección a una menor dimensión puede ser mediante cualquier operación de muestreo descendente. En este caso, se emplea una convolución con tamaño de kernel $1x1$ para reducir la muestra del mapa de características.

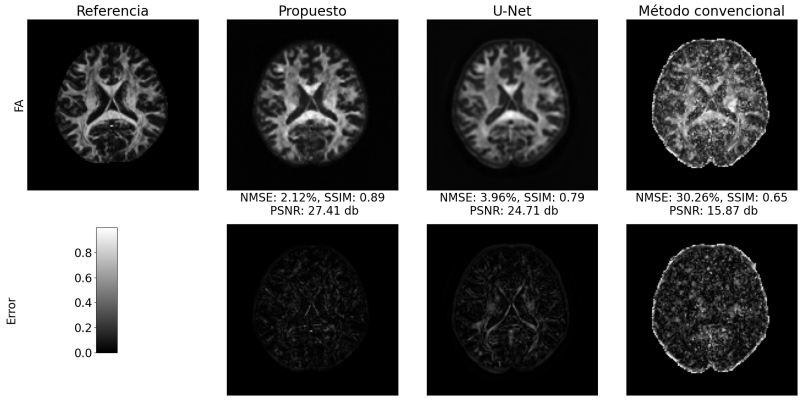


Fig. 3. Ejemplo de resultados obtenidos para mapa de anisotropía fraccional.

2.4. Validación

Para evaluar el desempeño del modelo se utilizan las métricas descritas en [7], que son el error cuadrático medio normalizado respecto al máximo valor de todos los errores para un mapa del tensor de difusión (NMSE); el índice de similitud estructural (SSIM), el cual se basa en comparar la luminancia, el contraste y la correlación estadística de los píxeles de una imagen con una imagen de referencia; y la relación de señal a ruido máxima (PSNR), utilizado para medir el desempeño en la limpieza de ruido de las imágenes de entrada del modelo.

3. Resultados y discusión

Todo el conjunto de imágenes de la base de datos que se ha recopilado se ha procesado utilizando la herramienta FSL. Entonces, para cada uno de los 65 casos control se han obtenido los volúmenes de anisotropía fraccional y difusividad media. Se ha utilizado el $80\,\%$ de los datos para el entrenamiento del modelo, el $10\,\%$ para validación y $10\,\%$ para una evaluación final.

En la implementación del modelo y la obtención de los resultados que se muestran a continuación, se ha utilizado el servicio de computación en la nube de Google Colabs, el cual proporciona un entorno Linux que cuenta con $32GB$ de memoria RAM y una tarjeta gráfica GPU Nvidia Tesla T4 con $16GB$ de memoria. Una ventaja de utilizar este servicio es la posibilidad de comunicar el entorno con cualquier repositorio público o privado, por lo que no es necesario copiar la base de datos utilizada para el entrenamiento al entorno del servicio.

Entonces, de acuerdo a la arquitectura del modelo que se ha implementado, es necesario realizar un acondicionamiento de la información original de la base de datos. Para empezar, se realiza un reordenamimiento de las dimensiones de los volúmenes de las señales de difusión obtenidas por el escáner de resonancia magnética, los cuales consisten originalmente en un tensor de 4 dimensiones $[h, w, d, c]$, en donde h es la altura de las imágenes, w es el ancho de las imágenes, d es la profundidad del volumen y c es el número de señales de difusión.

Tabla 1. Métricas promedio obtenidas para la inferencia del conjunto de evaluación.

Mapa - Método	NMSE (%)	SSIM	PSNR (db)	Tiempo (min)
FA				
Propuesto	8.5307	0.8552	27.8626	2.74
UNet	10.8327	0.8112	26.6207	47.61
Método convencional	–	0.7708	17.2302	18.98
MD				
Propuesto	5.1540	0.9094	27.9955	3.02
UNet	7.1707	0.9166	29.0652	53.89
Método convencional	11.0139	0.8789	27.0625	18.98

Esta transformación se efectúa de manera que el tensor resultante tiene las dimensiones $[d, c, h, w]$. En cambio, el tensor de difusión tiene solo 6 grados de libertad, por lo que teóricamente es necesario únicamente 6 señales de difusión para calcular los coeficientes del tensor en cada punto o voxel del volumen. Por consiguiente, se toman un total de 7 señales de difusión para la estimación de los mapas del tensor de difusión: 6 señales de difusión en direcciones equitativamente distribuidas en la esfera y una imagen sin dirección de gradiente de magnetización.

El modelo se ha implementado utilizando la biblioteca de aprendizaje automático PyTorch, que se basa en la librería Torch de Linux Foundation umbrella. Para ajustar los parámetros del modelo se ha empleado un algoritmo de optimización estocástico basado en gradiente. Asimismo, se utiliza la desviación cuadrática media entre la salida del modelo y la referencia como función de pérdida a optimizar por el algoritmo antes mencionado. Así, la función de pérdida durante el entrenamiento se encuentra definida por la siguiente expresión:

$$\mathrm{MSE}(\Theta) = \frac{1}{n}\sum_{t=1}^{n} ||F(x^t;\Theta) - y^t||^2, \tag{2}$$

donde $F(\cdot)$ representa la función del modelo de la red neuronal y Θ denota los parámetros de la red que se ajustan durante el entrenamiento. El modelo se entrena durante 140 épocas con un ritmo de aprendizaje inicial de 0.0002 y se reduce en un factor de 0.9 cada que la función de perdida no disminuye durante una época. Asimismo, se ha realizado un aumento de datos durante el entrenamiento en donde se han rotado o reflejado horizontalmente y verticalmente las imágenes de manera aleatoria.

También, se ha entrenado el modelo individualmente para ajustar ambos mapas del tensor de difusión. En las gráficas de la Figura 2 se muestran la función de pérdida de las primeras 100 épocas durante el entrenamiento del modelo para el ajuste de los mapas de anisotropía fraccional y difusividad media.

Por otro lado, para evaluar el efecto de utilizar módulos Transformer en una arquitectura tipo autoencoder, se ha implementado una arquitectura UNet [12], la cual se compone únicamente de módulos convolucionales, para la estimación de los mapas del tensor de difusión. Y, además, se han obtenido estos mapas utilizando el método convencional pero con las mismas 7 señales de difusión que utiliza el modelo propuesto.

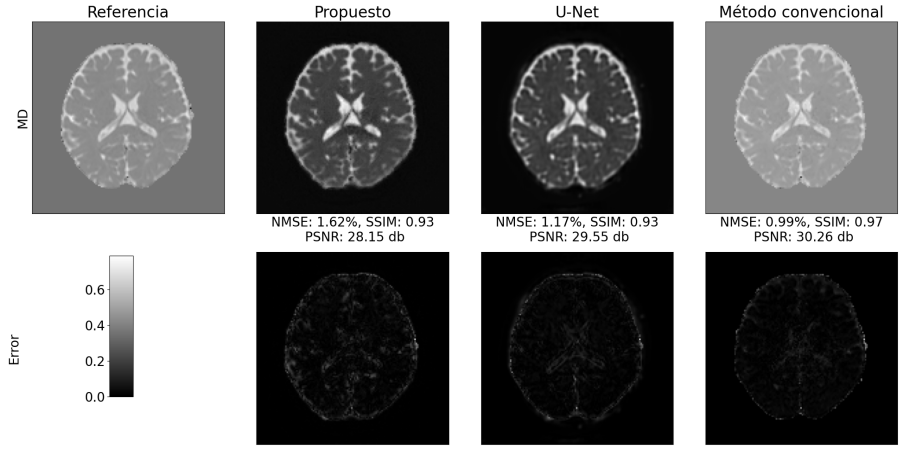


Fig. 4. Ejemplo de resultados obtenidos para mapa de difusividad media.

En la Figura 3 se presenta un ejemplo del mapa de anisotropía fraccional obtenido por el modelo propuesto, el modelo UNet y el método convencional, junto con los valores obtenidos de las métricas de desempeño y su correspondiente mapa de error absoluto para cada resultado. Es evidente, en este caso, que el modelo propuesto proporciona el mejor desempeño para la estimación de este mapa del tensor de difusión.

En la Figura 4 se muestra un ejemplo del mapa de difusividad media obtenido con el modelo propuesto y las dos metodologías de comparación mencionadas anteriormente, junto con sus correspondientes métricas de desempeño. En esta instancia, el modelo basado en módulos convolucionales ha logrado superar el desempeño del modelo propuesto y, aun más, el método convencional ha arrojado el mejor resultado para la estimación de este mapa, con un índice de similitud estructural de 0.97.

Esto último sugiere que la estimación del mapa de difusividad media es una tarea de menor complejidad que la estimación del mapa de anisotropía fraccional y no es necesaria la implementación de módulos basados en autoatención para mejorar la estimación de este mapa. De manera similar, en la Tabla 1 se presentan los valores promedio de las métricas de desempeño y el tiempo de procesamiento de un volumen completo obtenidos con los tres métodos a partir de todas las imágenes del conjunto de evaluación.

Entonces, para la estimación del mapa de anisotropía fraccional se ha obtenido un índice de similitud estructural promedio de 0.85, superando el modelo basado únicamente en bloques convolucionales y el método convencional, de los cuales se ha obtenido un índice de similitud estructural promedio de 0.81 y 0.65, respectivamente.

Además, el modelo propuesto es el que menor tiempo requiere para procesar un volumen completo de un paciente utilizando CPU, lo que abre la posibilidad de emplear este modelo en el área clínica. Por el contrario, para la estimación del mapa de difusividad media el modelo UNet ha generado el mejor desempeño, con un índice de similitud estructural promedio de 0.91, lo que sugiere que no es necesario el uso de bloques de autoatención para modelar imágenes con características altamente homogéneas, como es el caso de la segmentación binaria de imágenes.

En particular, la microestructura del fibras de materia blanca forman una complicada red en todo el cerebro con características altamente estructuradas y, debido a esto, el modelar mapas de características a partir de imágenes de resonancia magnética ponderadas por difusión es todo un reto computacional. Asimismo, es importante destacar que la estimación del mapa de anisotropía fraccional y difusividad media es una tarea crucial en el procesamiento de imágenes de DTI, ya que estos mapas son una representación visual de la microestructura de los tejidos biológicos y proporcionan información importante sobre la integridad de las fibras nerviosas y la conectividad del cerebro.

Por tanto, el uso de redes neuronales profundas basadas en autoatención para la estimación de estos parámetros tiene varias ventajas en comparación con los métodos tradicionales de procesamiento de DTI, puesto que el modelo ha sido capaz de modelar características complejas y no lineales en la estructura de los mapas a partir de únicamente 6 señales de difusión, aún cuando la base de información utilizada para optimizar el modelo ha sido limitada. Esto representa un avance en el uso e implementación de modelos basados en Transformers para extracción de imágenes médicas en entornos clínicos, ya que la precisión y velocidad en la estimación de estos mapas es crucial para la detección temprana de enfermedades cerebrales y para el monitoreo de la progresión de estas mismas.

4. Conclusiones

En el presente trabajo se ha desarrollado la implementación de una variante del modelo UTNet para la estimación del mapa de anisotropía fraccional y la difusividad media en la materia blanca cerebral. Los resultados obtenidos demuestran que el modelo propuesto es una herramienta prometedora para la estimación de los mapas del tensor de difusión y, en general, para el procesamiento de imágenes médicas. Además, las arquitecturas compuestas por Transformers tienen la ventaja de ser altamente escalables y eficientes en términos de recursos computacionales. Esto resulta ideal para aplicaciones en tiempo real y el procesamiento de grandes conjuntos de datos.

Agradecimientos. Este trabajo ha sido apoyado por el programa UNAM-PAPIIT IA104622.

Referencias

1. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A. L., Zhou, Y.: TransUNet: Transformers make strong encoders for medical image segmentation (2021) doi: 10.48550/ARXIV.2102.04306
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16×16 words: Transformers for image recognition at scale (2021) doi: 10.48550/ARXIV.2010.11929

3. Esteban, O., Markiewicz, C. J., Blair, R. W., Moodie, C. A., Isik, A. I., Erramuzpe, A., Kent, J. D., Goncalves, M., DuPre, E., Snyder, M., Oya, H., Ghosh, S. S., Wright, J., Durnez, J., Poldrack, R. A., Gorgolewski, K. J.: fMRIPrep: a robust preprocessing pipeline for functional MRI. Nature Methods, vol. 16, no. 1, pp. 111–116 (2018) doi: 10.1038/s41592-018-0235-4
4. Gao, Y., Zhou, M., Metaxas, D.: Utnet: A hybrid transformer architecture for medical image segmentation (2021) doi: 10.48550/ARXIV.2107.00781
5. Jones, D. K., Knösche, T. R., Turner, R.: White matter integrity, fiber count, and other fallacies: The do's and don'ts of diffusion MRI. NeuroImage, vol. 73, pp. 239–254 (2013) doi: 10.1016/j.neuroimage.2012.06.081
6. Karimi, D., Gholipour, A.: Diffusion tensor estimation with transformer neural networks. arXiv:220105701v1, pp. 102330 (2022) doi: 10.1016/j.artmed.2022.102330
7. Li, H., Liang, Z., Zhang, C., Liu, R., Li, J., Zhang, W., Liang, D., Shen, B., Zhang, X., Ge, Y., Zhang, J., Ying, L.: SuperDTI: Ultrafast DTI and fiber tractography with deep learning. International Society for Magnetic Resonance in Medicine, pp. 3334–3347 (2021) doi: 10.1002/mrm.28937
8. Mansouri, F. A., Koechlin, E., Rosa, M. G. P., Buckley, M. J.: Managing competing goals - a key role for the frontopolar cortex. Nature Reviews Neuroscience, vol. 18, no. 11, pp. 645–657 (2017) doi: 10.1038/nrn.2017.111
9. McLaren, D. G., Kosmatka, K. J., Oakes, T. R., Kroenke, C. D., Kohama, S. G., Matochik, J. A., Ingram, D. K., Johnson, S. C.: A population-average MRI-based atlas collection of the rhesus macaque. NeuroImage, vol. 45, no. 1, pp. 52–59 (2009) doi: 10.1016/j.neuroimage.2008.10.058
10. Nelkenbaum, I., Tsarfaty, G., Kiryati, N., Konen, E., Mayer, A.: Automatic segmentation of white matter tracts using multiple brain MRI sequences. In: IEEE 17th International Symposium on Biomedical Imaging, pp. 368–371 (2020) doi: 10.1109/ISBI45749.2020.9098454
11. Ozyurt, O., Dincer, A., Yildiz, M. E., Peker, S., Yilmaz, M., Sengoz, M., Ozturk, C.: Integration of arterial spin labeling into stereotactic radiosurgery planning of cerebral arteriovenous malformations. Journal of Magnetic Resonance Imaging, vol. 46, no. 6, pp. 1718–1727 (2017) doi: 10.1002/jmri.25690
12. Ronneberger, O., Fischer P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention, vol. 9351, pp. 234–241 (2015) doi: 10.48550/arXiv.1505.04597
13. Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., Rueckert, D.: Attention gated networks: Learning to leverage salient regions in medical images. Medical Image Analysis, vol. 53, pp. 197–207 (2019) doi: 10.48550/arXiv.1808.08114
14. Shen, Z., Zhang, M., Zhao, H., Yi, S., Li, H.: Efficient attention: Attention with linear complexities (2020) doi: 10.48550/arXiv.1812.01243
15. Tournier, J.: Diffusion MRI in the brain – theory and concepts. Progress in Nuclear Magnetic Resonance Spectroscopy, vol. 112–113, pp. 1–16 (2019) doi: 10.1016/j.pnmrs.2019.03.001
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6000–6010 (2017) doi: 10.48550/arXiv.1706.03762
17. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: ECA-net: Efficient channel attention for deep convolutional neural networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020) doi: 10.1109/cvpr42600.2020.01155
18. Wasserthal, J., Neher, P., Hirjak, D., Maier-Hein, K.: Combined tract segmentation and orientation mapping for bundle-specific tractography. Medical Image Analysis, vol. 58, pp. 101559 (2019) doi: 10.1016/j.media.2019.101559

19. Young, P. N. E., Estarellas, M., Coomans, E., Srikrishna, M., Beaumont, H., Maass, A., Venkataraman, A. V., Lissaman, R., Jiménez, D., Betts, M. J., McGlinchey, E., Berron, D., O'Connor, A., Fox, N. C., Pereira, J. B., Jagust, W., Carter, S. F., Paterson, R. W., Schöll, M.: Imaging biomarkers in neurodegeneration: current and future practices. Alzheimer's Research and Therapy, vol. 12, no. 49 (2020) doi: 10.1186/s13195-020-00612-7
20. Zhao, H., Tsai, C. C., Zhou, M., Liu, Y., Chen, Y. L., Huang, F., Lin, Y. C., Wang, J. J.: Deep learning based diagnosis of parkinson's disease using diffusion magnetic resonance imaging. Brain Imaging and Behavior, vol. 16, no. 4, pp. 1749–1760 (2022) doi: 10.1007/s11682-022-00631-y

ISSN 1870-4069

Sistema de interpretación de imágenes para el reconocimiento y traducción del lenguaje de señas

Dora María Calderón Nepamuceno, Gabriela Kramer Bustos, Efrén González Gómez

Universidad Autónoma del Estado de México,
Centro Universitario Nezahualcóyotl,
México

{dmcalderonn, gkramerb, egonzalezg}@uaemex.mx

Resumen. El lenguaje de señas, es utilizado por las personas sordas, con la finalidad de comunicarse. Se compone de movimientos y expresiones realizadas especialmente con las manos, existe una gran ausencia de tecnologías al interpretar este lenguaje. Como sociedad es importante llevar a cabo iniciativas que mejoren la calidad de vida de la comunidad sordomuda del país. En el presente trabajo se muestra el proceso y la creación del diseño e implementación del sistema de reconocimiento de gestos, atraves del lenguaje de programación en Python, junto con librerías como OpenCV (librería que permite detección de movimientos, reconocimiento de objetos a partir de imágenes), Numpy (Especializada en el cálculo numérico y el análisis de datos), Imutils (Funciones para realizar tareas básicas de procesamiento de imágenes de una manera más fácil). El sistema ha sido creado para realizar tareas básicas de procesamiento de imágenes de una manera más fácil, permite la visualización de la imagen adquirida y su traducción al lenguaje de señas.

Palabras clave: Python, procesamiento de imágenes, lenguaje de señas.

Image Interpretation System for Sign Language Recognition and Translation

Abstract. Sign language is used by deaf people to communicate. It is made up of movements and expressions made especially with the hands, there is a great absence of technologies when interpreting this language. As a society it is important to carry out initiatives that improve the quality of life of the country's deaf community. This paper shows the process and the creation of the design and implementation of the gesture recognition system, through the Python programming language, together with libraries such as OpenCV (library that allows movement detection, object recognition from images), Numpy (Specialized in numerical computation and data analysis), Imutils (Functions to perform basic image processing tasks in an easier way). The system has been created to perform basic image processing tasks in an easier way, it allows the visualization of the acquired image and its translation into sign language.

Keywords: Python, image processing, sign language.

1. Introducción

En la actualidad existen en el mundo millones de personas que sufren algún tipo de discapacidad y de las cuales sufren discriminación, por ejemplo, en algunos casos les es imposible acudir a la escuela y realizar ciertas actividades, o incluso para ellos es difícil conseguir empleo. Actualmente el lenguaje de señas, es la manera más efectiva y básica para la comunicación verbal de las personas sordas, con dificultad auditiva y para hablar.

Cuando hay personas que hablan diferentes idiomas y se quieren comunicar, utilizan gestos o movimientos, como, por ejemplo, gestos faciales y corporales, al realizarlos podemos comprender lo que quiere expresar la otra persona sin pronunciar una palabra; un simple movimiento de la boca, ojos, cejas, manos o cualquier extremidad, puede significar algo [1].

Las lenguas de los signos son habladas (en silencio) por cien millones de personas sordas en todo el mundo. En total hay al menos 138 idiomas de señas vigentes según el catálogo de Etnología, y muchos de ellos son lenguas oficiales (nacionales) u oficiales de la comunicación.

Existen personas que nacieron sordas y que incluso, no son capaces de leer. Además de los lenguajes que ya existen también hay alfabetos que se utilizan para deletrear letras (nombres, palabras raras, signos desconocidos, etc.).

Las personas que sufren este tipo de discapacidad tienen la necesidad de utilizar este sistema de señas, buscan la alternativa de aprender este tipo de lenguaje, por lo que se convierte en un reto memorizar cada una de las señas que conforman este sistema de comunicación, y sobre todo para las personas que no requieran descifrar algún mensaje.

Es por esto, que se requiere desarrollar una herramienta, que permita reconocer los gestos que son realizados por personas con discapacidad auditiva, con el fin de brindar instrumentos didácticos, que les facilite el aprendizaje para este tipo de lenguaje y lo puedan hacer de una manera más interactiva.

Por desgracia, aunque existan estas herramientas no todas las personas con esta discapacidad podrían obtenerlas, y esto hace que sigan dependiendo de la interpretación del lenguaje de señas mediante el apoyo de personas, para que tengan la facilidad de comunicarse con otras personas.

El objetivo de realizar este proyecto, surge por los problemas de comunicación que existe entre las personas sordas y el resto del mundo, y también por la discriminación que éstas reciben. Por ejemplo, hoy en día, es posible que la población sorda pueda acceder a instituciones educativas universitarias de forma virtual [3].

El uso de la tecnología ha permitido que la población discapacitada acceda a programas de la facultad de la Universidad Pedagógica y Tecnológica, de manera remota y con apoyo de traductores digitales.

Según estadísticas arrojadas por el Departamento Administrativo Nacional de Estadística, el 16.4% de los habitantes de la ciudad, tiene limitaciones permanentes para oír, es decir, que, de cada 100 personas con esta condición, 17 presentan algún tipo de discapacidad permanente auditiva [4]. “El lenguaje de señas se caracteriza por ser visual y corporal, es decir la comunicación se establece con el cuerpo, en un espacio determinado” [5, 6].

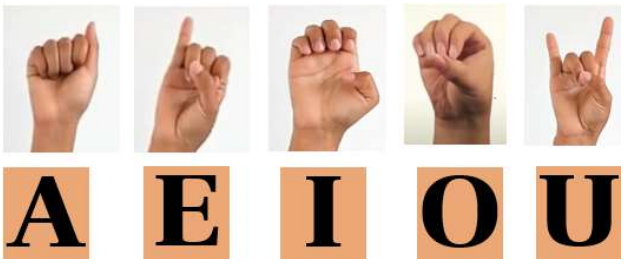


Fig. 1 Caracteres del lenguaje de señas.

Tabla 1. Materiales empleados para el desarrollo del sistema.

Librerías:	**Descripción**
Python	Es un lenguaje de alto nivel de programación, se utiliza para desarrollar aplicaciones de todo tipo, Además, se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar software sin límites.
OpenCV:	Es una librería para la detección de movimiento, reconocimiento de objetos, reconstrucción 3D a partir de imágenes, son sólo algunos ejemplos de aplicaciones
Numpy:	Biblioteca Especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos
Imutils:	Es creado para realizar tareas básicas de procesamiento de imágenes de una manera más fácil.
Cámara	Es el dispositivo empleado para la captura de imágenes.

2. Materiales y métodos

Para poder desarrollar el proyecto se necesitaron materiales y métodos para la implementación del desarrollo del sistema, a continuación, se realiza la descripción de los materiales y métodos empleados.

Entre estos trabajos, se encuentra la propuesta de García Incertis [1] que reconoce algunas letras del alfabeto de la Lengua de Signos; por su parte, Razo Gil [2] reconoce imágenes de la mayoría de las letras del alfabeto del Lenguaje de Señas. Los materiales que se utilizaron para la creación del sistema, son indicados en la Tabla 1, con el que cada uno tiene su propia descripción, por lo cual, es útil para el desarrollo de dicho sistema, y para que sea más sencilla la comprensión de usuarios que son ajenos a este desarrollo.

3. Desarrollo

Para el desarrollo de este sistema, se realizó el reconocimiento de las imágenes del alfabeto del lenguaje de señas, el cual, se enfocará en el cálculo de las variantes para poder identificar las imágenes capturadas e identificar las señales obtenidas y hacer su

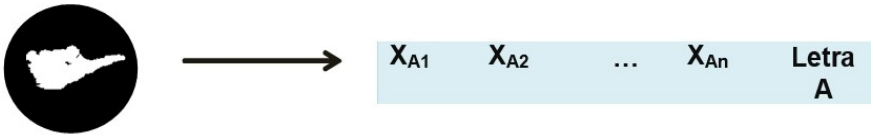


Fig. 3. Señal de la letra A.

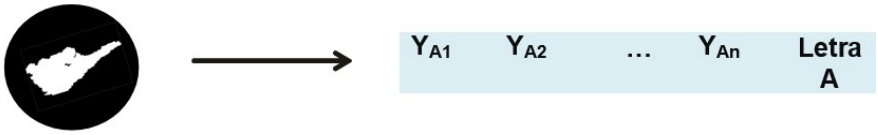


Fig. 4. Señal de la letra A con una variación.

correspondiente traducción. Se mostrarán las etapas, para llevar a cabo el sistema de reconocimiento de las imágenes:

1. Uso de una cámara digital.
2. Iluminaciones controladas.
3. Uso de imágenes estáticas.
4. Enfocado en transformaciones.
5. Las letras del alfabeto son realizadas con las manos.

3.1. Realización de la captura de la imagen

Se captura la imagen con una cámara digital, el formato de la imagen será jpg a color, la iluminación se dará por la lámpara del dispositivo, la imagen será convertida a una matriz.

3.2. Captura de la imagen

La corrección de la iluminación se realiza mediante un proceso morfológico, este proceso se usa para disminuir objetos claros sobre fondos oscuros. La imagen original es mejorada mediante la resta de la imagen de apertura.

3.3. Segmentación

De la imagen obtenida con la cámara se realizar un segmentado, en el proceso se realiza un umbral de reconocimiento.

3.4. Filtro morfológico

El filtro morfológico, facilita la interpretación de la imagen removiendo estructuras brillantes de la imagen.

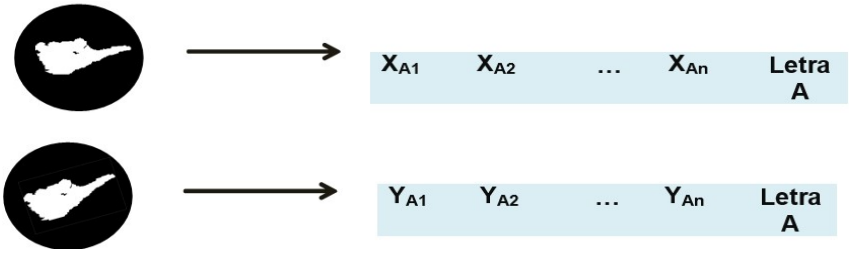


Fig. 5. Resultado del proceso en una matriz evolutiva.

Fig. 6. Imagen del resultado del proceso.

3.5. Arquitectura de las etapas del reconocimiento de imágenes

En la figura 2 se muestra la arquitectura del proceso del reconocimiento de una imagen.

3.5.1. Segmentación de la mano

Se realizan cálculos hasta encontrar uno o más objetos.La imagen procesada es una imagen binaria y los píxeles que contienen información tienen valor de 1.

3.5.2.Escalamiento de la mano

Transforma la imagen a una forma cuadrada, este proceso se realiza dando un ajuste a un tamaño definido de 200 x 200 píxeles. El paralelogramo obtenido se escala al tamaño con el fin de minimizar la variación de tamaño de las manos.

3.5.3.Matrices evolutivas

Se usa una matriz evolutiva para almacenar los patrones del conjunto de letras del alfabeto y hacer su identificación mediante un programa de computadora.

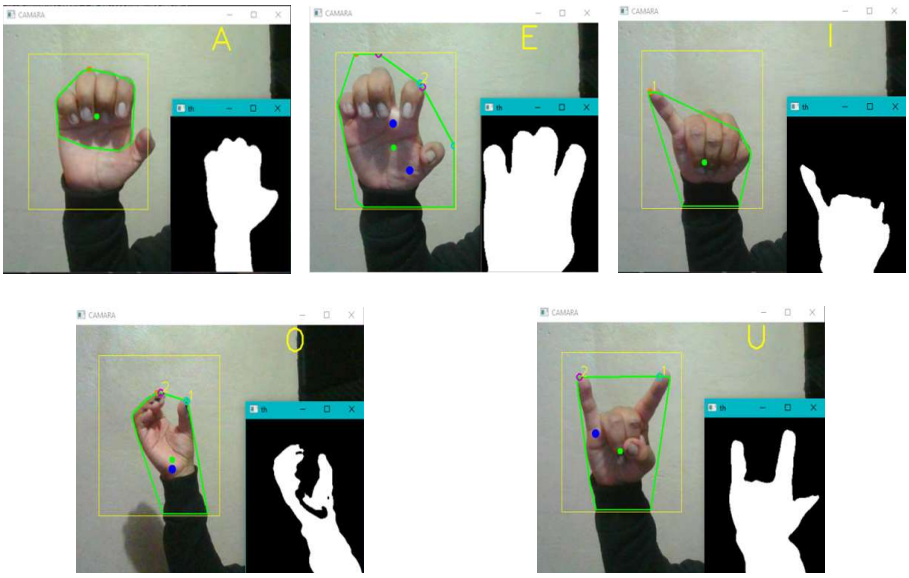


Fig. 7. Resultado del reconocimiento de las vocales en forma de texto.

Fig. 8. Resultado del reconocimiento de Hasta Pronto (despedida) en forma de texto.

3.5.4.Ejemplo de matriz evolutiva

a) Se procesa señal de la letra A como se muestra en la figura 3.

b) Se procesa otra señal de la letra A con una variación cómo se muestra en la figura 4.

c) El resultado se almacena con el significado de la letra A en la matriz evolutiva figura 5.

4. Resultados

Los resultados obtenidos con el sistema, reconocen las señas del alfabeto, de lenguaje de señas identificadas en imágenes capturadas mediante la cámara. Para realizar Las operaciones se requiere capturar la imagen realizando la señal del alfabeto que se desea. A la imagen se le aplica el proceso de umbralado dando como resultado la Figura 6.

El resultado del proceso final del reconocimiento de la imagen se muestra en forma de texto tal como se muestra en las figuras 7, donde se ejemplifica las 5 vocales y en la figura 8 se muestra la ejemplificación de una palabra.

5. Conclusiones y trabajo a futuro

El sistema de reconocimiento, alcanzó a distinguir las 5 vocales del alfabeto del lenguaje de señas que son: A, E, I, O, U. Para ello las principales características son extraídas de la imagen y estas deben ser detectables con el ruido y la iluminación que contenga. El uso de la matriz evolutiva y los procesos de normalización, permiten mayor similitud en la detección de las imágenes.

El reforzamiento disminuye patrones con los que se hace la comparación para el reconocimiento de las señas hechas con la mano. Con base en los experimentos y dado el parecido de algunas señas se detectó que el umbral debe de ser menor o igual al 10% para garantizar una adecuada relación entre los patrones.

Los cambios de iluminación se deben controlar ya que de ello depende la ejecución correcta de los resultados que obtengamos del sistema. OpenCV cuenta con una extensa documentación, así como la integración de varios lenguajes de programación, es una herramienta de software libre, con lo cual se permitió realizar dicho sistema.

Referencias

1. González-Riveros, C. G., Yimes-Inostroza, F. J.: Sistema de reconocimiento gestual de lengua de señas Chilena mediante cámara digital. Informe final del proyecto para optar al título profesional de ingeniero de ejecución en informática, Pontificia Universidad Católica de Valparaíso Facultad de Ingeniería Escuela de Ingeniería Informática (2016) opac.pucv. cl/pucv_txt/txt-0500/UCC0990_01.pdf
2. Rojas-Rojas, L. M., Arboleda-Toro, N., Pinzón-Jaime, L. J.: Caracterización de población con discapacidad visual, auditiva, de habla y motora para su vinculación a programas de pregrado a distancia de una universidad de Colombia. Revista Electrónica Educare, vol. 22, no. 1, pp. 1 (2017) doi: 10.15359/ree.22-1.6
3. Ballesta-Perez, J. L.: Diseño e implementación de sistema de interpretación. Obtenido de traducción de gestos asociados a preguntas, necesidades y saludos básicos del lenguaje de señas colombiano. Universidad de Córdoba Facultad de Ingenierías Programa de Ingeniería de Sistemas y Telecomunicaciones, pp. 1–109 (2015) https://repositorio.unicordoba. edu.co/handle/ucordoba/280
4. Visual studio code: Code editing, redefined. Edición de Código (2021) code.visual studio.com/

5. Mejía, H.: Lengua de Señas Colombiana: Segundo Tomo, Santa Fe de Bogotá, Federación Nacional de Sordos de Colombia (1996) https://fenascol.org.co/wp-content/uploads/2022/01/Tomo%202%20LSC%20Fenascol.pdf
6. Seijas, L. M.: Reconocimiento de patrones utilizando técnicas estadísticas y conexionistas aplicadas a la clasificación de dígitos manuscritos. Tesis Doctoral, Universidad Nacional de Buenos Aires, Facultad de Ciencias Exactas y Naturales (2011) https://bibliotecadigital. exactas.uba.ar/ collection/tesis/document/tesis_n4997_Seijas

ISSN 1870-4069

Clasificación automática de sentimientos en textos de canciones en idioma español

Omar García-Vázquez, Tania Alcántara, Grigori Sidorov, Hiram Calvo

Instituto Politécnico Nacional, Centro de Investigación en Computación, México

omar.gava@hotmail.com, {talcantaram2020, sidorov, hcalvo}@cic.ipn.mx

Resumen. Los sentimientos son el estado afectivo de ánimo, los cuales son producidos en el cerebro y son provocados por una emoción. Este sentimiento se ha trasladado de múltiples maneras, como textos, pinturas o música. La música transmite diferentes emociones, lo cual hace aún más importante saber que tipo de sentimientos se encuentra dentro de una canción, que pueden ser, entre muchos otros, positivos, negativos o neutrales. A través de este trabajo se analizó y clasificó los textos de canciones del idioma español, esto a través de un extractor de características basado en co-ocurrencias y la aplicación de modelos más modernos como las Redes Neuronales Convoluciones y una *Long Short-Term Memory*, donde se obtuvieron resultados competitivos en el estado del arte.

Palabras clave: Procesamiento de lenguaje natural, clasificación de textos, songs, CNN, LSTM.

Automatic Sentiment Classification in Spanish Language Song Texts

Abstract. Feelings are the affective state of mind, which are produced in the brain and are caused by an emotion. This feeling has been transfered in multiple ways, such as texts, paintings or music. Music transmits different emotions, which makes it even more important to know what kind of feelings are found within a song, which can be, among many others, positive, negative or neutral. Through this work, the texts of songs in the Spanish language were analyzed and classified, this through a feature extractor based on co-occurrences and the application of modern models such as Convolutional Neural Networks and a *Long Short-Term Memory*, where competitive results were obtained in the state of the art.

Keywords: Machine learning, classification, natural language processing, songs, CNN, LSTM.

1. Introducción

La opinión, los sentimientos y todos los conceptos que los rodean; como el afecto, los estados de ánimo y la actitud siempre se han basado en las creencias de cada

persona. Estos aspectos de la individualidad humana hacen que siempre las acciones que realizamos estén influenciadas por otras.

La inserción y el rápido crecimiento del análisis de sentimiento coincide con lo mostrado en redes sociales, foros de discusión y blogs, ya que, por primera vez en la historia humana, se tiene un gran volumen de datos de opinión en medios digitales [1]. Los datos recolectados a través de internet a menudo presentan opiniones, y es por eso que el análisis de sentimientos se ha convertido en una de las principales herramientas empleadas en el análisis de redes sociales.

El análisis de sentimientos es un campo de investigación dentro del Procesamiento de Lenguaje Natural (PLN o NLP por sus siglas del inglés *Natural Language Processing*), el cual utiliza técnicas de Aprendizaje Máquina (AP o ML por sus siglas del inglés *Machine Learning*).

Por otro lado, la búsqueda de emociones en las oraciones no es tan sencilla, ya que suelen ser oraciones subjetivas, las cuales enuncian hechos, porque las opiniones y los sentimientos son inherentemente subjetivos. Hoy en día, estos sentimientos han sido plasmados de diferente manera, desde libros, poemas y hasta en música.

La música es capaz de activar áreas emocionales e inclusive evocar recuerdos, a través de las partituras rítmicas. Pero el ritmo no es el único conducto de emociones, sino los textos y aquellas palabras que utilizan los autores para expresar tristeza, felicidad o emoción. La clasificación de este tipo de emociones puede realizarse a través de PLN.

Durante este trabajo se exploró el AS en textos de canciones, trabajando con un conjunto de datos en español. Esto por medio de la extracción de características con *embeddings*, combinado con clasificadores poco usuales en textos como las Convolutional Neural Network y las Long Short-Term Memory.

2. Marco teórico

2.1. Análisis de sentimientos

EL AS es una técnica de PLN, que se enfoca en identificar y extraer la emoción representada en un texto, como positiva, negativa o neutral [2]. La manera de abordar el SA tiene varios enfoques, desde los basados en reglas hasta el aprendizaje profundo.

De acuerdo con la página *QuestionPro*[1]: "El análisis de sentimiento utiliza tecnologías avanzadas de inteligencia artificial, como PLN, análisis de texto y ciencia de datos, para identificar, extraer y estudiar información subjetiva. En términos más simples, clasifica un texto como positivo, negativo o neutral". Para determinar esa polaridad, se puede hacer de técnicas de aprendizaje automático, aprendizaje profundo o del análisis semántico [3].

[1] Análisis de sentimiento. ¿Qué es y cómo realizarlo? https://www.questionpro.com/blog/es/herramienta-de-analisis-de-sentimientos/

Tabla 1. Distribución del corpus *Textos de canciones en español* [13].

Sentimiento	No. de oraciones	Porcentaje
S	97	6.67 %
P	780	52.80 %
N	600	40.53

2.2. Aprendizaje profundo

Las Redes Neuronales Profundas o mejor conocido como Deep Learning (DL) son una rama de la Inteligencia Artificial (IA o AI por sus siglas en inglés *Artificial Intelligence*).

La principal diferencia entre una red de DL y las Redes Neuronales (RN o NN por sus siglás en inglés Neural Network) clásicas, radica en la complejidad de la arquitectura [4]. Se puede describir de manera sencilla la estructura de una red neuronal profunda [4]:

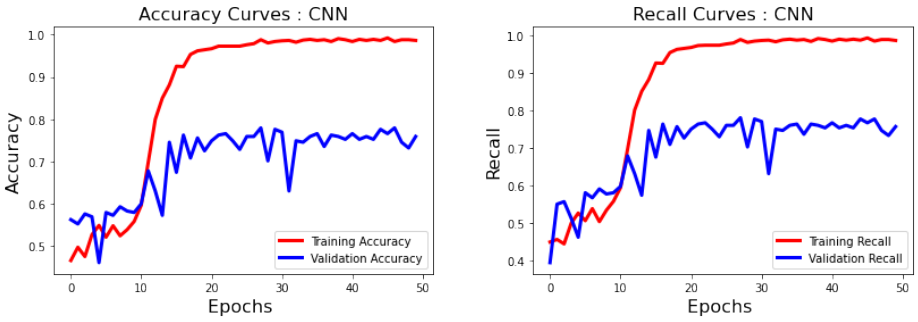
1. **Capa de entrada:** Son las neuronas que representan los datos de entrada.
2. **Capas ocultas:** La red neuronal profunda contendrá al menos una capa, los parámetros mínimos son el número de neuronas, la función de activación y la dimensión de los datos de entrada.
3. **Capa de salida:** Es la capa que da la respuesta codificada, se tendrán tantas salidas como entradas y se interpretara cada una como la probabilidad de que el dato de parámetros mínimos son el número de neuronas.

Existen varios tipos de DL, pero, este trabajo se centra principalmente en las siguientes [5]:

1. **Redes neuronales convolucionales (CNN, por sus siglas en inglés *Convolutional Neural Network*):** Constan de una o varias capas llamadas "convoluciones", en donde se aplican filtros a la entrada para extraer las principales características. Estos filtros son matrices pequeñas aplicando una operación de multiplicación y sumando los resultados para producir un mapa de características.
2. **Memoria prolongada de corto plazo (LSTM, por sus siglas en inglés *Long Short-Term Memory):*** Este tipo de modelo utiliza una estructura de celdas de memoria con puertas de entrada, salida y olvidar, esto con el fin de capturar la información a largo plazo de una secuencia de palabras.

2.3. Extractores de características basados en *embeddings*

Los extractores de características utilizados con *embedding* son modelos pre-entrenados para representar textos a vectores numéricos de alta dimensión [6]. Un ejemplo de un *embedding* es ***GloVe*** *(Global Vectors for Word Representation*), este es capaz de capturar información semántica y sintáctica de la palabra. GloVe utiliza una matriz de co-ocurrencia, la cual realiza la representación de la frecuencia de una palabra [7]. Después de la obtención de la matriz, se realiza una factorización para obtener los vectores de palabras finales, que capturen la co-ocurrencia de manera distribuida [7].



(a) Accuracy (b) Recall

Fig. 1. Curvas de métricas *accuracy* y *recall* para CNN.

3. Estado del arte

Existen múltiples trabajos en torno a la clasificación de sentimientos. En [8] se propone una clasificación de polaridad, con un conjunto de canciones tailandesas, basándose puramente en el texto de las mismas.

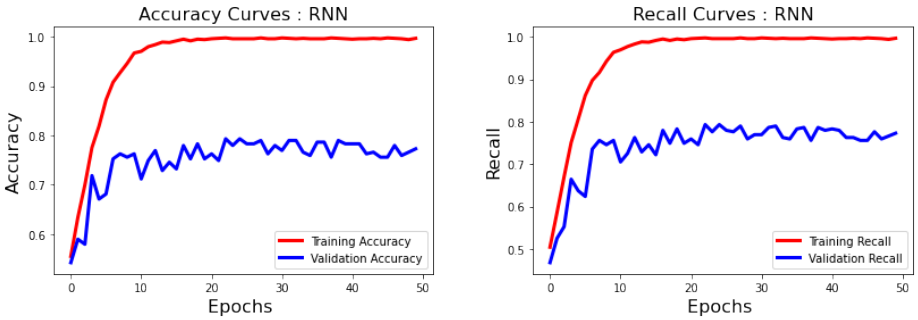
Proponen la utilización de un lexicón y utilización de técnicas de aprendizaje automático tradicional. De las diferentes partes de una canción (título, versos, estribillo, pre-coro, coro y puente) en donde solo se decidió utilizar el coro y los versos como corpus, ya que, se piensa que en estas dos partes es donde hay más probabilidad de que se encuentre el tema de la canción.

En el caso de [9] se utilizaron tres géneros como etiqueta (inspirador, divertido y romántico) para así poder utilizar minería de asociación en los datos de entrenamiento y encontrar a qué etiqueta pertenecen las palabras claves del texto de la canción. Posterior a esta tarea, se utilizó el modelo Naive Bayes para el cálculo de probabilidad. Encuentran una maximización de la probabilidad de observar las palabras que realmente se encontraron en los textos de ejemplo, mejorando así la habitual independencia de Naive Bayes.

Para [10] se utilizó una ontología llamada *SentiWordNet*, la cual incluye puntajes relacionados con los aspectos positivos o negativos de las palabras. La ontología fue utilizada para la extracción de características de sentimientos, todo esto en los textos de canciones, para encontrar el estado de ánimo al que pertenecen dichas canciones.

Los experimentos fueron desarrollados en un corpus de 185 canciones y se utilizaron tres diferentes algoritmos de clasificación; Naive Bayes, *K-Nearest Neighbor* y Máquinas de Vectores de Soporte (SVM por sus siglas en inglés *Support Vector Machine*).

En [11] se compara el rendimiento de algunos modelos de *Word embedding*, previamente entrenados en análisis de letras de canciones y tareas de polaridad de reviews de películas. Los resultados muestran que los *tweets* son lo mejor para el análisis de letras de canciones, mientras que Google *News* y *Common Crawl* son los mejores para el análisis de películas, ya que el vocabulario que se utiliza en estos portales es muy parecido en ambos casos. Los modelos entrenados con GLoVe superan ligeramente a los entrenados con Skip-gramas.



(a) Accuracy (b) Recall

Fig. 2. Curvas de métricas *accuracy* y *recall* para LSTM.

Por otro lado, se encontró que existen combinaciones de modelos que no son comúnmente usados para clasificación, como CNN [12]. Este modelo combina CNN y redes LSTM bidireccionales para capturar características relevantes del texto en diferentes niveles. Estos experimentos realizados en diferentes conjuntos demostraron un *accuracy* del 90.66 %.

4. Conjunto de datos

El conjunto de datos, denominado *Textos de canciones en español* [13], se trata de un corpus privado de PLN desarrollado en el Laboratorio de Procesamiento de Lenguaje Natural del Centro de Investigación en Computación del Instituto Politécnico Nacional[2].

El conjunto está formado por 91 canciones, todas escritas en el idioma español y con ritmos variados (bachata, pop, balada, entre otros). Cada canción fue seccionada en pequeños párrafos de manera manual, siguiendo un sentido de la oración, es decir, no se tienen ideas incompletas u oraciones que terminen en palabras de parada, lo que da un total de 1,477 datos.

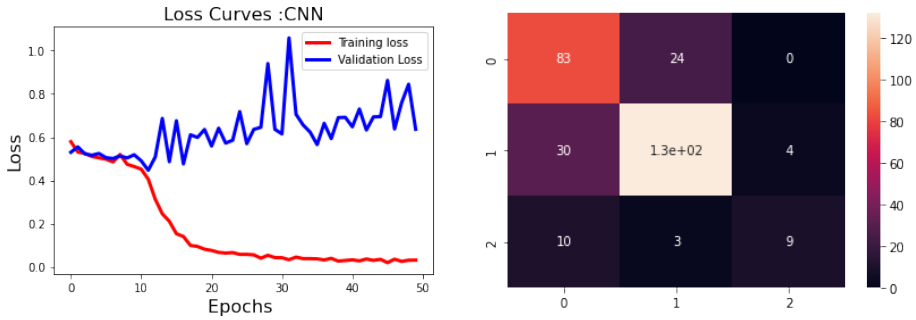
Para el etiquetado de canciones se utilizó un método desarrollado por los autores del conjunto. Para el etiquetado, se consideraron en 3 principales emociones: S, neutral; P, positivos; N, negativos. El cuadro 1 representa la distribución de los datos del conjunto, se puede notar que se trata de un conjunto desbalanceado. Para este trabajo, el conjunto de datos fue divido en 80 % para el entrenamiento y en 20 % para validación.

5. Propuesta de solución

5.1. Preprocesamiento

Para obtener los mejores resultados, es necesario preparar los datos de un texto con los mecanismos clásicos de preprocesamiento de datos, por ejemplo:

[2] Conjunto de datos *Textos de Canciones en español*, para consultarlo o acceder escriba a sidovor@cic.ipn.mx



(a) Función de perdida de CNN (b) Matriz de Confusión

Fig. 3. Función de perdida y matriz de confusión de CNN.

– Paso de mayúsculas a minúsculas.
– Remoción:
  1. Corrección ortográfica.
  2. Palabras gramaticales (*stop words*).
  3. Diagonales y diagonales invertidas.
  4. Números.
  5. Saltos de línea.
  6. Paréntesis.
  7. Dobles espacios en blanco.
– Tokenización.
– Lematización.
– Obtención de raíces (*Stemming*).

5.2. Extracción de características

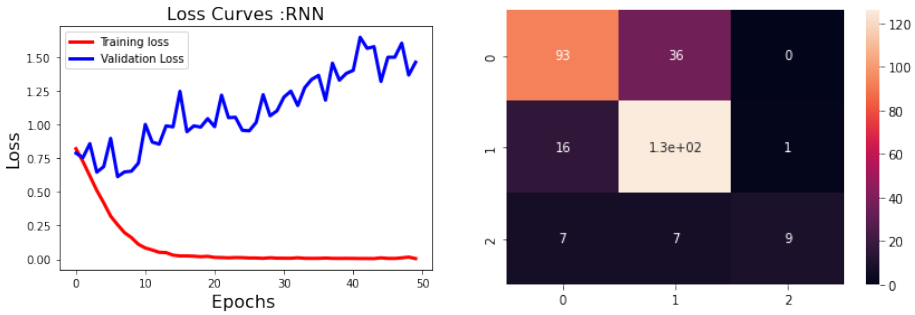
Se realizó la extracción de características por medio de GLoVe. La cual crea una matriz con la co-orcurrencia de la similitud de palabras dentro de una ventana (puede ser el número de palabras cercanas). Dicha matriz está conformada con la probabilidad de que dos palabras aparezcan. Para este trabajo se eligió una ventana de 10.

La matriz se transforma en otra por medio de la ponderación. La matriz, ahora, se pondera, es decir, a partir de una factorización matricial se hace la reducción de la dimensión de la matriz. Esta matriz se descompone en dos matrices para realizar representaciones vectoriales: Una para representar las co-ocurrencias y otra para contextos. Al finalizar estos procesos, las matrices se combinan, para obtener los *embeddings* finales.

5.3. Clasificación

Aprendizaje profundo

– **LSTM:** En la capa de entrada se define la máxima longitud de secuencia que aceptara la red, la segunda capa le pertenece al *embedding* GloVe, las siguientes



(a) Función de perdida de LSTM (b) Matriz de Confusión

Fig. 4. Función de perdida y matriz de confusión de LSTM.

capas pertenecen al algoritmo LSTM bidireccional, capas densas y al final la capa de salida en forma categórica. Posteriormente, se define la siguiente configuración: Función de perdida, *categorical crossentroy*; Optimizador, *ramsprop*;*Batch size*, 16; No. épocas, 50.

– **CNN:** Para este modelo se utilizaron tres capas convolucionales, tres capas de *MaxPooling* de una dimensión, una capa plana, una capa densa y por último, la capa de salida. La configuración fue la siguiente: Función de perdida, Binary crossentropy; Optimizador, rmsprop; Learning rate, 0.001.

6. Resultados

Se realizaron experimentos con clasificadores LSTM y CNN, acompañados de la extracción de características GLoVe. En las siguientes gráficas se muestran los mejores resultados de cada uno de los clasificadores.

En las figuras 1 para CNN y 2 para LSTM, se observa la evolución de las métricas por medio de las épocas.

En la figura 3a se observa que durante las primeras 15 épocas se obtuvo un comportamiento desfavorable en cuanto a la perdida, pero a partir de la época 16 esta mejoro llegando a un valor muy cercano a cero en la época 50. Para la figura 4a se observa que en el caso del entrenamiento para LSTM, existe un buen comportamiento, ya que, existe una curva favorable sobre la función de perdida, la cual llegó a valores muy próximos al cero.

En las figuras 3b para CNN y 4b, muestran las matrices de confusión, representadas como mapas de calor, donde los colores más claros representan resultados más favorables.

En el cuadro 2 se muestran los resultados para la métrica de *accuracy*. Se eligió reportar únicamente la métrica de *accuracy*, ya que es la más utilizada en el estado del arte.

En el cuadro 3 se muestran los resultados para la métrica *accuracy*, para los clasificadores en el estado del arte. En letras negritas, se puede observar el posicionamiento de los clasificadores descritos en este artículo.

Tabla 2. Resultados del metodo propuesto.

Clasificador	***Accuracy***
CNN	78.4 %
LSTM	80.1 %

Tabla 3. Resultados comparativo con los métodos del estado de arte.

Clasificador	***Accuracy***
CNN-BiLSTM [12]	90.66 %
Genre Classification [9]	85 %
LSTM (nuestro)	**80.1 %**
CNN (nuestro)	**78.4 %**
SentiWordNet 2 [10]	71 %
SentiWordNet 1 [10]	69 %
Thai Songs [8]	62 %
QWE [11]	61 %

Es importante mencionar que la comparación no puede ser 100 % directa, ya que el estado del arte, ni la propuesta en este artículo, utilizan el mismo conjunto de datos, pero es importante resaltar los resultados promedio en tareas similares, y así determinar una mejora en un trabajo a futuro.

7. Conclusiones y trabajo futuro

En este artículo se presentó la clasificación de sentimientos a través del extractor de características GLoVe, el cual extrae las características mediante las co-ocurrencias de palabras. Esta extracción de características, sirvió de entrada para clasificarlos por medio de LSTM y CNN.

Con los resultados obtenidos, se puede determinar un excelente punto de partida, ya que, aunque no se contemplaron modelos que consideran el contexto o modelos de atención, la graficas muestran que modificando los elementos que se toman para el entrenamiento se podrían mejorar los resultados obtenidos con creces.

Aunado a lo anterior, también se observa que se requiere de un modelo capaz de poder manejar cadenas de texto más largas, ya que el modelo LSTM tiene un rendimiento excelente solo con cadenas cortas. Por último, se utilizaron modelos de entrenamiento pequeños donde su coste computación es muy bajo, dando pie a que con modelos más robustos se incrementaría el valor de las métricas.

Se debe apreciar que involucrar la clasificación con una CNN no es usual, así que los resultados para la tarea de análisis de sentimientos, se demostró que con el preprocesamiento correcto y añadiendo capas adicionales de convolución, se podrían obtener resultados muy favorables.

Como trabajo a futuro se proponen diferentes enfoques: 1. Aplicar un extractor de características basado en el contexto y combinar las CNN con LSTM; 2. Aplicar mecanismos de atención que contemplen el contexto de las frases, poniendo especial enfoque en BERT o T5.

Referencias

1. Poria, S., Cambria, E., Hazarika, D., Majumder, N., Zadeh, A., Morency, L. P.: Context-dependent sentiment analysis in user-generated videos. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 873–883 (2017) doi: 10.18653/v1/P17-1081
2. Wang, C., Zhang, Q., Liu, W., Liu, Y., Miao, L.: Facial feature discovery for ethnicity recognition. WIREs Data Mining and Knowledge Discovery, vol. 9 (2019) doi: 10.1002/widm.1370
3. Mártinez-Cámara, E., Martín-Valdivia, M., Ureña, L. A.: Análisis de sentimientos. In: IV Jornadas TIMM Tratamiento de la Información Multilingüe y Multimodal (2011)
4. Rivera, M.: Perceptrón multicapa en Tensorflow-Keras. Aprendizaje Automático CIMAT (2022) personal.cimat.mx:8181/~mrivera/cursos/aprendizaje_profundo/mlp/mlp.html
5. Tariq, U., Tariq, S., Ahmad, R.: Deep Learning: A review of the state-of-the-art with an insight into Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM), pp. 71614–71630 (2018) doi: 10.1109/ACCESS.2018.2870225
6. Géron, A.: Hands-On machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Inc (2nd ed.) (2019)
7. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014) doi: 10.3115/v1/D14-1162
8. Srinilta, C., Sunhem, W., Tungjitnob, S., Thasanthiah, S.: Lyric-based sentiment polarity classification of Thai songs. In: Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1 (2017)
9. Giras, A., Advirkar, A., Patil, C., Khadpe, D., Pokhare, A.: Lyrics based song genre classification. Journal of Computing Technologies, vol. 3, no. 2 (2014) http://jctjournals.com/feb2014/v4.pdf
10. Kumar, V., Minz, S.: Mood classification of lyrics using SentiWordNet. In: 2013 International Conference on Computer Communication and Informatics, pp. 1–5 (2013) doi: 10.1109/ICCCI.2013.6466307
11. Çano, E., Morisio, M.: Quality of word embeddings on sentiment analysis tasks. Natural Language Processing and Information Systems, vol. 10260 (2017) doi: 10.1007/978-3-319-59569-6_42
12. Rhanoui, M., Mikram, M. Yousfi, S., Barzali, S.: CNN-BiLSTM model for document-level sentiment analysis. Machine learning and knowledge extraction, vol. 1, no. 3, pp. 832–847 (2019) doi: 10.3390/make1030048
13. Sidorov, G., Soto-Osorio, D., Chanona-Hernandez, L., Núñez-Prado, C. J.: Corpus "Textos de canciones en español". Laboratorio de Procesamiento de Lenguaje Natural, Centro de Investigación en Computación (2019)

ISSN 1870-4069

Un asistente virtual como prueba de concepto para el emprendimiento disruptivo

Adolfo Alejandro Romero-Angeles[1], Rosa Leonor Ulloa-Cazarez[2]

[1] Maestría en Desarrollo y Dirección de la Innovación,
Sistema de Universidad Virtual,
Universidad de Guadalajara,
México

[2] Sistema de Universidad Virtual,
Universidad de Guadalajara,
México

adolfo.romero@udgvirtual.udg.mx

Resumen. La presentación y análisis de solicitudes de inversión requieren mayor eficiencia debido a la adopción de procesos cada vez más estrictos por las firmas de inversión de riesgo. No se conocen asistentes virtuales con características tecno-pedagógicas para estimular el aprendizaje en sus usuarios y guiar en el diseño de un modelo de negocio o un proyecto de inversión. En este trabajo se presenta una propuesta para un asistente virtual que brinde esta guía a emprendedores, en la preparación de su discurso de negocio y que éstos puedan elevar posibilidades de conectar con inversionistas. Se presenta un prototipo utilizando una plataforma comercial y se aplica la metodología COSMIC para medir el esfuerzo del diseño y sustentar decisiones para reducir la huella del carbono en el diseño de un asistente virtual de inversión.

Palabras clave: COSMIC, asistentes virtuales, ingeniería de software, mejora de procesos, planeación.

A Virtual Assistant as a Proof of Concept for Disruptive Entrepreneurship

Abstract. Presenting and analyzing investment solutions require greater efficiency due to more rigorous procedures implemented by venture investment companies. Until now, no documented Virtual Assistants with techno-pedagogical features to encourage learning in its users' applications have been developed to direct and assist a person in the business model or investment project creation. This work is a design concept for an application that guides entrepreneurs in preparing their business speeches and improve their chances of getting in touch with investors. The COSMIC methodology supports the creation of a virtual investment assistant and sustainable decisions.

Keywords: COSMIC sizing, virtual assistants, software engineering, process improvement, planning.

1. Introducción

En México las PyMes conforman el 99.8% de las empresas en el país, generando el 72% del empleo y el 52% del PIB [1], sin embargo, enfrentan diversos obstáculos para solventar sus gastos, elevar su competitividad y aumentar su productividad, considerando que en México el 72.1% de las MiPyMES cerraron sus operaciones antes de cumplir 10 años [2].

En el contexto de la inversión de capital de riesgo en empresas emergentes durante el año 2022 se registró una caída histórica: el volumen de inversión se redujo un 40% en Latinoamérica y un 24% en México respecto al año anterior [3].

En este sentido, se vuelve crucial para las partes interesadas (emprendedores e inversionistas) hacer eficiente sus procesos para presentar y analizar solicitudes de inversión; debido a la adopción de requerimientos cada vez más estrictos por parte de las firmas de inversión de riesgo en cuanto a la realización de auditorías financieras y legales dentro de su portafolio de prospección, esto sumado a que de acuerdo con reportes recientes [4], los inversionistas destinan cada vez menos tiempo a la revisión de las presentaciones de solicitud de inversión (2 minutos con 42 segundos para un total de 19 diapositivas), un 24% menos de tiempo que en 2021.

La automatización de procesos y el uso de asistentes virtuales se han extendido a una variedad interesante de aplicaciones, que van desde brindar apoyo para recibir un servicio o atención, hasta asistentes que guían en los procesos de compra electrónica.

Hasta el año 2022, estas aplicaciones denominadas chatbot, entablaban conversaciones con el usuario, muy acotadas en términos de eficiencia comunicativa, dirigiendo el proceso y la interacción hacia contenidos predefinidos, y dejando a los agentes humanos, las comunicaciones que excedían estas precondiciones. Luego, en noviembre del 2022 apareció el chat de la empresa OpenAI, ChatGPT [5].

Aunque en los últimos años se habían desarrollado diversas aplicaciones de chatbot con base en tecnologías y modelos de procesamiento de lenguaje natural [6, 7], fue hasta el lanzamiento del ChatGPT que estas tecnologías mostraron su potencial de impacto gracias a la gran campaña mercadológica: el ChatGPT promete servicios y soluciones que requieren de una comunicación concreta, dirigida y acotada.

Sin embargo, hasta el día de hoy, no se han conocido aplicaciones de estos desarrollos para guiar u orientar en procesos concretos, por ejemplo, para guiar a una persona, en el diseño de un modelo de negocio o un proyecto de inversión. En este documento se describe una propuesta de diseño de una aplicación que brinde una guía a los emprendedores emergentes para la preparación de su modelo y discurso de negocio y que puedan elevar sus posibilidades de conectar con inversionistas que apoyen sus emprendimientos.

El diseño de la aplicación, en la forma de un asistente virtual de inversiones especializado, se fundamenta en principios de la enseñanza en línea de habilidades emprendedoras para estudiantes de áreas científico-tecnológicas, así como en la adecuación de resúmenes de patentes para simplificar su transferencia en el contexto de las solicitudes de financiamiento.

En la primera etapa de desarrollo de esta herramienta, nuestro interés radicó en elegir las plataformas que permitan una forma de trabajo eficiente y con el menor esfuerzo humano posible, relacionado con el costo de desarrollo de la aplicación.

En el primer prototipo, se integraron herramientas de Watson de la empresa IBM para dar forma a un asistente virtual de inversiones. Para tener una medición adecuada del esfuerzo y los costos del desarrollo de la herramienta, se utilizó la metodología del Common Software Measurement International Consortium, COSMIC [8].

El resto de este trabajo se desarrolla en las siguientes cuatro secciones. En la sección dos, de Trabajos Relacionados, presentamos las bases que sustentan el diseño de la aplicación de chatbot en cuanto a los aspectos tecno-pedagógicos, así como el resumen de los avances en el uso de asistentes virtuales con diferentes aplicaciones.

En la sección 3, del Método, se presentan los principios de la metodología de Ingeniería de Software en la que basamos la propuesta de diseño que se apega a la metodología del COSMIC [9].

La sección de Resultados, es la presentación de la aplicación de la metodología COSMIC para la medición del esfuerzo, en puntos de función, en el desarrollo del asistente y describe el prototipo desarrollado. Finalmente, en la sección de discusiones se presentan las limitaciones, el trabajo futuro y las implicaciones de esta propuesta.

2. Trabajos relacionados

El emprendedurismo como contenido de aprendizaje presenta dificultades prácticas que han llevado a la definición de bases teóricas para el desarrollo de estándares y listas de verificación con efectividad pedagógica. En esta línea, se han aplicado estos principios para el diseño de juegos de simulación que cumplen con cuatro atributos fundamentales [10, 11].

1. Escenarios reales y posibles.
2. Comunicación clara y concreta.
3. Factibilidad técnica.
4. Evaluación costo-beneficio.

Los juegos, han probado su eficacia en la enseñanza de negocios [11], incrementando el nivel de compromiso de los estudiantes.

Los chatbot son tecnologías que gamifican las experiencias de los usuarios y han sido utilizados con propósitos educativos, comprobando su impacto en la motivación del aprendizaje y el desempeño de estudiantes implementando la técnica del micro-aprendizaje, es decir, actividades educativas que deben completarse en una duración de 10 minutos, mediante recursos audiovisuales como textos, imágenes y videos [12].

Así, en el estudio realizado por Kohnke [11], se comparó los desempeños de los estudiantes que recibieron instrucción a través del chatbot, y los que la recibieron en formato de enseñanza tradicional, en el aula. Las conclusiones llevan a la identificación de cuatro aspectos relevantes para la evaluación de estas aplicaciones:

1. Tensión-presión.
2 Elección percibida.
3 Competencia percibida.
4 Valor percibido.

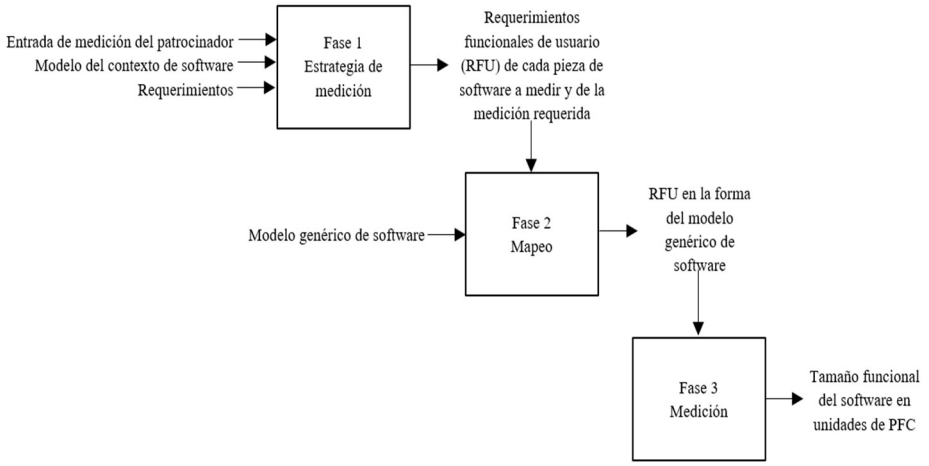


Fig. 1. El proceso de medida de COSMIC, adaptado de The COSMIC measurement process: https://cosmic-sizing.org/cosmic-sizing/intro/

Las tecnologías para chatbot han evolucionado rápidamente y algunos estudios han comprobado su eficacia para proporcionar apoyo a las personas, incluso en aspectos sociales e íntimos [13].

A través de distintas metodologías, se han explorado los potenciales de los chatbot para establecer interacciones significativas en términos de comunicación con las personas [14, 15].

3. Métodos

En este trabajo, se muestra el proceso de diseño de un asistente virtual o chatbot que brinda apoyo a emprendedores de base tecnológica para generar un modelo de negocio que incremente sus posibilidades de conectar con inversionistas que apoyen sus emprendimientos.

Como base de este diseño, se toma en consideración el estándar COSMIC [8] que ofrece una serie de principios generales para garantizar la madurez y productividad de los proyectos de software.

3.1. El método COSMIC

En el método COSMIC se definen los principios y reglas para medir una pieza de software a partir de los requerimientos de software [10], donde cada movimiento de datos significa un punto de función COSMIC (PFC) sea una entrada, una lectura, una escritura o una salida de datos.

El Modelo Genérico de Software COSMIC describe los principios fundamentales de la Ingeniería de Software, donde cada pieza de software puede ser analizada en procesos funcionales únicos. Incluye tres fases:

COSMIC es un estándar internacional ISO (ISO 19761) que ha demostrado que la escala de medida PFC es adecuada para los propósitos en que se diseñó, por ejemplo,

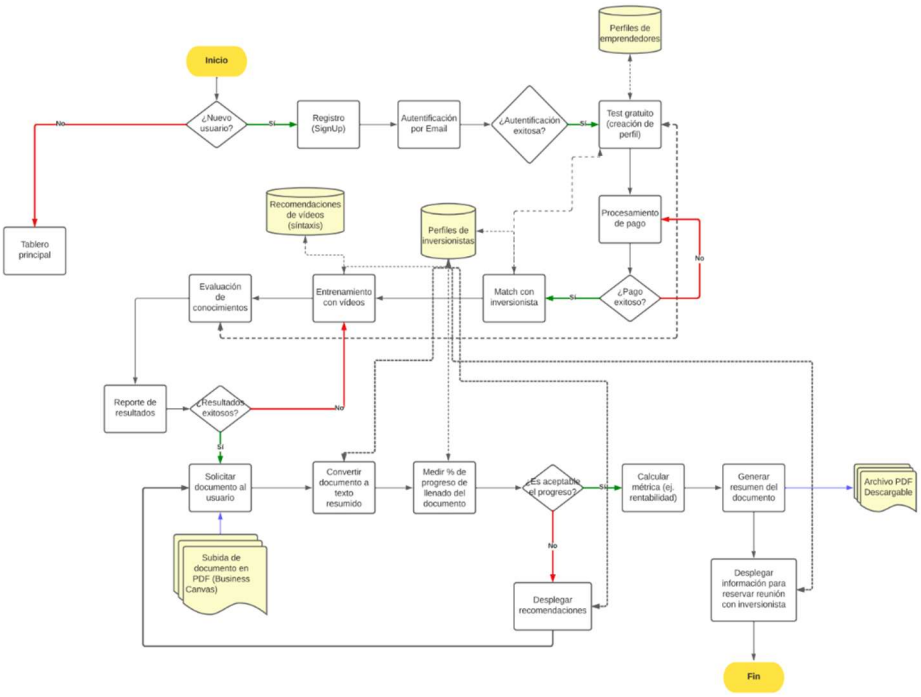


Fig. 2. Descripción del proceso que realiza el emprendedor en la primera sesión con el asistente virtual.

para medir el desempeño del software y como medición para la estimación del esfuerzo del proyecto [16]. Asimismo, ha demostrado utilidad para reducir los riesgos y garantizar la madurez, la validez y la consistencia del software.

3.2. Fase 1: Estrategia de medición

La estrategia de medición se conforma de cinco parámetros:

1. El propósito de la medición. En este proyecto, el propósito es estimar el tamaño funcional del software de un asistente virtual de inversiones con el objetivo de comparar su desempeño con el proceso de consultoría tradicional, esto es, sin automatizarse [17]. Asimismo, será de utilidad para estimar el esfuerzo requerido para la construcción del asistente virtual.
2. El alcance de la medición. Este parámetro se relaciona con los requerimientos funcionales del usuario (RFU), que en este caso es el emprendedor. Los RFU se detallan en la sección 3.3 de Requerimientos Funcionales.
3. El nivel de descomposición de las piezas de software. En nuestro caso, se trata de la aplicación completa, es decir, del asistente virtual de inversión, identificando algunas interacciones que se describen separadamente.

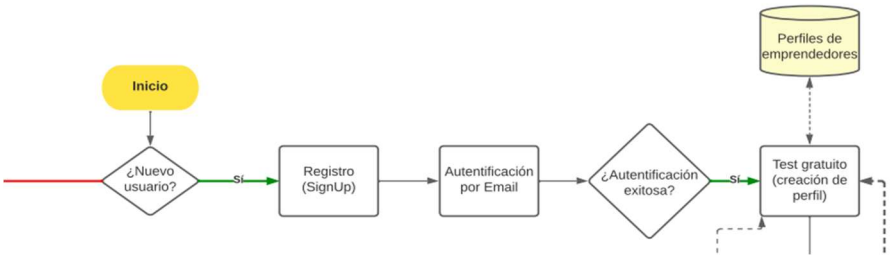


Fig. 3. Descripción del proceso que realiza el emprendedor en la primera sesión con el asistente virtual.

4. Los usuarios funcionales son dos, los primeros y principales, descritos como los emprendedores (UF1), son quienes generan las entradas y reciben las salidas de datos del chatbot. El segundo usuario funcional (UF2) es el sistema de validación de la información de intercambio.
5. Las capas de la arquitectura del software. Cada pieza de software se restringe a una sola capa.

3.3. Requerimientos funcionales

3.3.1. Contexto

El Asistente virtual de inversión, es una aplicación de chatbot que guía a emprendedores de base tecnológica, a generar su discurso de negocio de manera que cumpla con las características que le hagan atractivo para la inversión por un tercero. El proceso que lleva a cabo un usuario por primera vez, se describe en la Figura 2.

Desde el punto de vista del proyecto se pretende buscar alternativas de mejora al proceso de consultoría de pre-inversión para negocios de base científico-tecnológica, tal que se reduzcan las interacciones de la empresa con sus clientes.

Por lo que se espera que con esta propuesta de solución los clientes puedan acceder de manera más óptima a los servicios que ofrece la empresa, en comparación a la modalidad de consultoría tradicional. En cuanto a los movimientos tradicionales, se busca disminuir tales como la gestión de archivos físicos y las visitas de oficina.

Se propone que la interacción de los usuarios con el asistente virtual automatice los pasos intermedios entre la adquisición del servicio, las reuniones con inversionistas y la evaluación del servicio.

Esto es, se automaticen pasos del proceso de consultoría empresarial tales como: el registro del cliente en la plataforma, cobranza, desarrollo de soluciones (plan de entrenamiento y propuesta de emparejamiento con inversionistas afines), presentación de soluciones, revisiones de propuestas, capacitación especializada, entrega de resultados, llenado de solicitudes de inversión, mejora de proyectos y reservación con inversionistas potenciales.

Comúnmente, los asistentes virtuales establecen interacción directa y en distintos niveles con sus usuarios humanos, permitiéndoles diferentes niveles de modificación de las interfaces y las funciones. El asistente virtual de inversión, presenta solo un tipo de interacción con el emprendedor que se representa como un intercambio de datos.

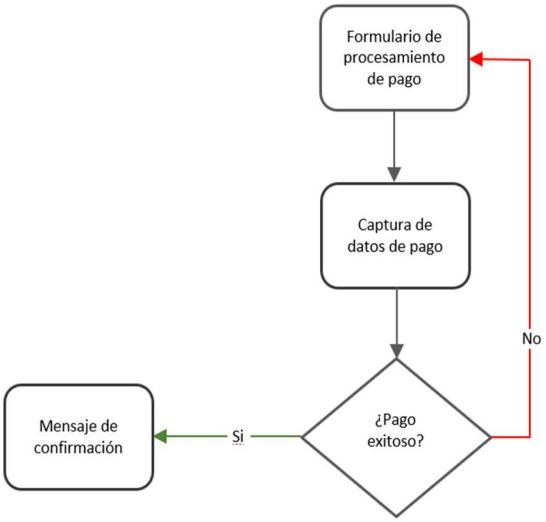


Fig. 4. Descripción del proceso de pago.

Esta limitación en la interacción del emprendedor con el asistente virtual de inversión, da lugar a los siguientes RFU.

3.3.2. Interfaces

La primera interfaz que se presenta al emprendedor es la de registro. Esta implica la entrada de información por parte del usuario, que en respuesta, recibe la solicitud de autenticación del registro a través de una cuenta de correo electrónico, y genera una respuesta de validación.

Considerando que son tres tareas, esta interfaz se mide con tres (3) PFC: registro de información de entrada, mensaje de solicitud de validación y respuesta del usuario (Figura 3).

La segunda interfaz del sistema es la del Test gratuito que implica la solicitud de información a través de un formulario, donde el usuario captura los datos para la creación de su perfil de emprendedor y de su proyecto. Esta información, se registra en la base de datos de perfiles, misma que genera un mensaje de confirmación al usuario. Dadas las tareas que se realizan (4), esta interfaz se mide en 4 PFC.

La conclusión de la interfaz dos, lleva al emprendedor al procesamiento de pago (Figura 4). En esta interfaz, se requiere la captura de los datos de pago, por ejemplo, el registro de una tarjeta bancaria.

Una vez concluido el proceso de registro de pago, el sistema registra la información en la base de datos de la página web que integra el servicio del asistente virtual y genera un mensaje de confirmación del proceso, o en su defecto, señala el error o la omisión en la captura y lleva al usuario al formulario. Este proceso implicó 5 PFC.

En el caso de que el registro de pago sea exitoso, la siguiente interfaz permite al emprendedor las siguientes decisiones: configurar cuenta, iniciar tutorial rápido o salir del sistema. Si la interacción del emprendedor continúa, una vez que se documenta el perfil, se le presenta la interfaz para el diseño del modelo de negocio. Esta sección

Tabla 1. Mapeo de los eventos y procesos funcionales del asistente virtual.

Interacción	Proceso funcional	Movimiento de datos
Primera interacción	Formulario de registro	6
Segunda interacción	Test gratuito	3
Tercera interacción	Procesamiento de pago	5
Cuarta interacción	Entrenamiento	7
Quinta interacción	Evaluación de conocimientos	3
Sexta interacción	Documento de proyecto	11
Séptima interacción	Match con inversionistas	6
Octava interacción	Reservar reunión de inversión	7

permite al usuario acceder a contenido educativo y generar solicitudes de inversión optimizadas con inteligencia artificial.

Al concluir, el asistente le entrega un resumen de resultados descargable, a su vez recomendaciones de estudios y motivación. Por último, le da la opción de probar una rutina nueva, ver su registro de actividad o salir del sistema.

3.4. Fase de mapeo

En la Tabla 1 "Mapeo de los procesos funcionales" se presenta el resumen de tareas y los PFC derivados de los RFU anteriores.

3.5. Fase de medición

Con base en la metodología COSMIC, cada interacción del UF1 con el asistente virtual, se considera una pieza de software. Las piezas de software pueden ser entradas o salidas de información, escritura de información en el sistema o base de datos, lectura de información. La Figura 5 describe este proceso en las interacciones del UF1 con el asistente virtual en el proceso de registro.

Para fines de este trabajo se consideran métricas como la cantidad de interacciones y movimientos de datos dentro del sistema propuesto con el objetivo de medir el tamaño de software, sin embargo, se contemplan requerimientos funcionales tales como la navegabilidad de la interfaz de usuario y usabilidad de la plataforma.

Por otro lado, se incluirán métricas para evaluar el desempeño del asistente virtual mediante métricas como la tasa de interacción, puntuación de satisfacción, duración de las conversaciones y precisión además de la tasa de éxito de los emprendedores. Esto último, con la finalidad de disminuir la incidencia de errores y garantizar la validez de la información.

Sin dudas, todas estas métricas en conjunto facilitarán optimizar la efectividad energética (PUE) del servicio digital desarrollado.

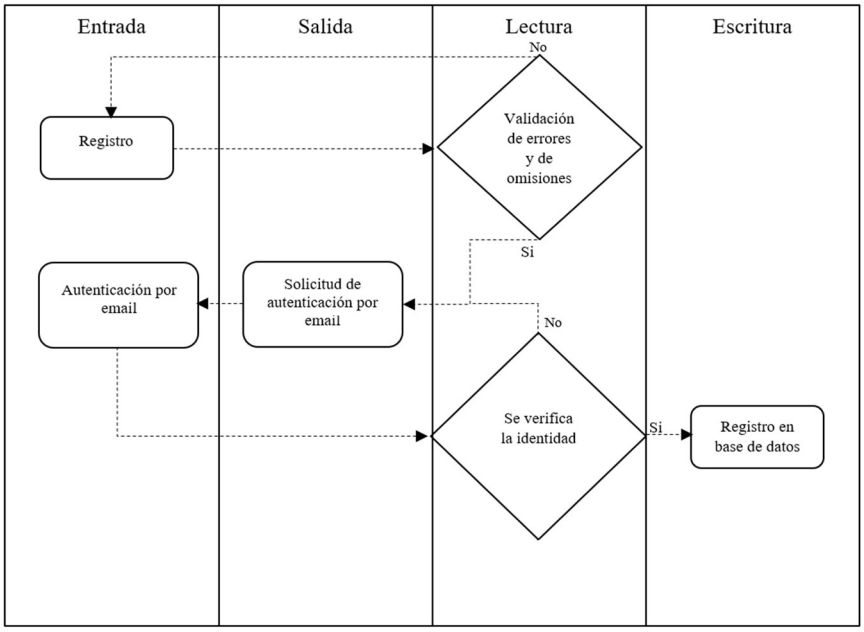


Fig. 5. Mapeo de los RFU por PFC.

Cada proceso funcional tiene su propia configuración y mapeo de RFU por PFC. En el caso mostrado en la Figura 4, se contabilizaron 6 PFC. Con base en esto, se contabilizan 48 PFC para la totalidad del asistente virtual de inversión. La regla para el cálculo del tamaño funcional del asistente virtual se calcula con la regla representada en la Ecuación 1:

$$\text{Tamaño funcional} = \text{Entradas} + \text{Salidas} + \text{Escrituras} + \text{Lecturas}. \quad (1)$$

4. Resultados

Preliminarmente se encontró un tamaño funcional equivalente a 14 puntos de movimientos de datos en el modelado de los procesos iniciales para el uso del asistente virtual que se propone como prueba de concepto (registro, evaluación y procesamiento de pago). En cuanto a los resultados de PFC para el prototipo completo basado en Watson de IBM, se obtuvieron para estos mismos procesos, un total de 48 PFC.

En la Figura 6 se muestra una interfaz conversacional que se construyó con tecnología de IBM Watson Assistant (plan Lite gratuito), plataforma que fue seleccionada dado que se contaba previamente con las credenciales de acceso y de disponibilidad de recursos dentro del proyecto.

Se seleccionó la localización de servicio en la nube en Dallas, Texas; por ser la más próxima a México, la sede del presente proyecto con la presuposición de que el gasto energético para el procesamiento computacional podría ser menor al de otras ubicaciones.

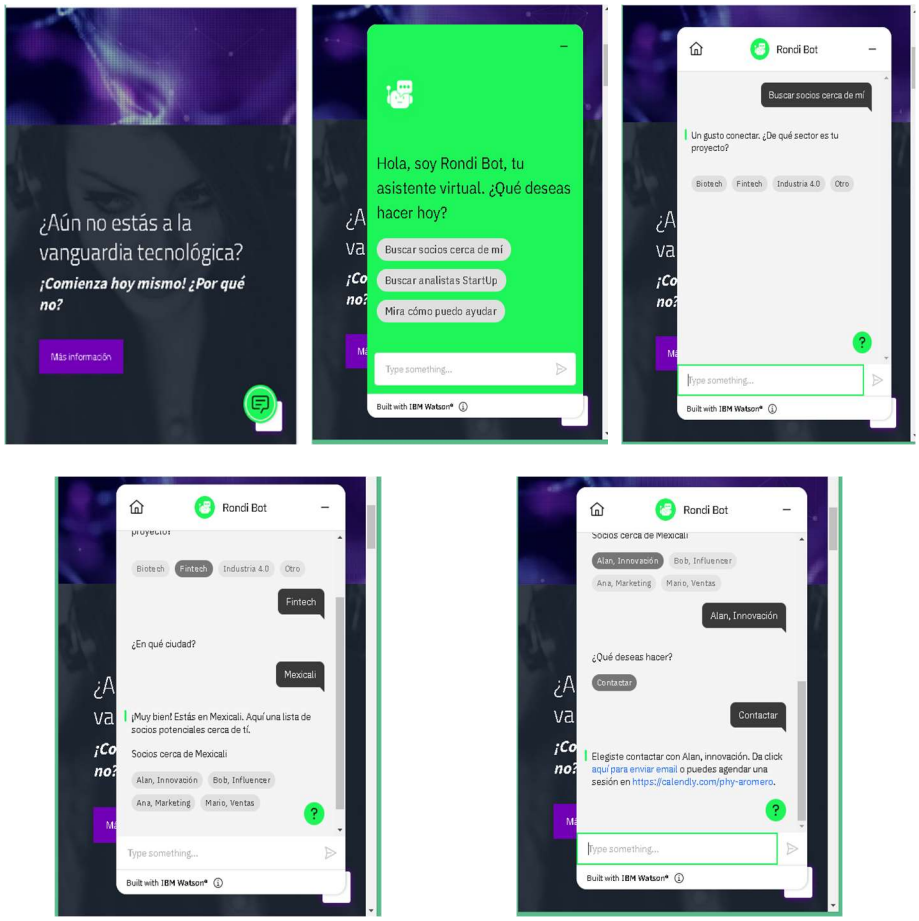


Fig. 6. Captura de pantalla de la implementación inicial del asistente virtual generado con IBM

Se creó un diálogo entrenado con 9 intenciones, 26 entidades y 261 nodos. En este modelo, las intenciones representan los verbos (las acciones que los usuarios quieren realizar) y consisten en un conjunto de ejemplos compuestos por oraciones previamente etiquetadas manualmente para la detección del contexto contenido en el texto que introduce el usuario en el asistente virtual cuando interactúan, por ejemplo, se definió la intención *#registrar_ciudad* con 19 ejemplos distintos previamente etiquetados, de los casos de texto que pudiera escribir el usuario al llenar el formulario de registro cuando se le pregunta "¿en qué ciudad?" al seleccionar el servicio de búsqueda de conectar con socios inversionistas cercanos.

Lo mismo se realizó para la gestión de números para operaciones aritméticas básicas por el usuario con la finalidad de registrar sus métricas de negocios actuales (punto de equilibrio de ventas y retorno de inversión). En el caso de las entidades, son los sustantivos a los que hace referencia el usuario para lograr dichas acciones, por ejemplo: la entidad *@ciudad* sirve para extraer el lugar de residencia del usuario al registrarse.

Por último, los nodos se refieren al número total de respuestas (condicionadas por el reconocimiento de entidades) del asistente virtual dentro del flujo del diálogo en todos los casos de conversación, en este caso inicial: para la atención de cliente y búsqueda de socios. De esta manera el asistente virtual puede identificar los textos de entrada del usuario y desplegar la información requerida.

Como parte de esta investigación se indaga sobre la efectividad del uso de energía (PUE) de cada proveedor de tecnologías de cómputo en la nube, la PUE es la relación entre la energía total consumida por el centro de datos dividida por la energía consumida por el equipo de TI.

En el caso de los centros de datos de Google tienen un PUE promedio de 1.10 (90.90% de efectividad), Amazon 1.11 (90.09%), Microsoft 1.12 (89.28%) e IBM 1.53 (65.35%) [18, 19, 20, 21]. Como se puede observar, una PUE más cercana a 1.0 significa que la mayor parte de la energía se utiliza óptimamente para la computación de los procesos en los centros de datos.

Dado que uno de los objetivos de este trabajo es la sustentabilidad del desarrollo, se quiere generar alternativas que reduzcan la cantidad requerida de movimientos de datos, pero a la vez facilite la experiencia de usuario promoviendo una participación activa dentro del entorno virtual de aprendizaje.

Los resultados obtenidos en este caso de estudio sirven como punto de partida para este proyecto de investigación como estándar de calidad en los procesos de planeación (selección de servicios en la nube), evaluación y mejora continua del servicio. De las 12 interacciones -requeridas por la consultoría tradicional- se lograron reducir a tan solo 4 en total, esto mediante la inclusión de la interacción con el chatbot.

La fase de medición con COSMIC facilitó la identificación de áreas de oportunidad en el diseño de la experiencia de usuario, ya que se evidenció cuantitativamente que la propuesta actual pudiera ser optimizada para reducir la huella de carbono del sistema propuesto al tener mecanismos más eficientes de navegación.

Lo anterior permitirá estimar y documentar las emisiones equivalentes de dióxido de carbono asociadas al entrenamiento final del modelo computacional de todo el sistema a lo largo de su ciclo de vida, así como definir los requerimientos energéticos por cada consulta del usuario que se recibe en tiempo real [22] y siguiendo las prácticas de las 4M's para reducir el consumo energético computacional [23], por lo que en consecuencia se planea buscar alternativas que reduzcan el tamaño funcional mediante un proceso de registro y pagos distinto al convencional que se proponía, es decir, se integren dichos procesos de manera unificada.

Esta forma de pago y registro puede ser unificada mediante la inclusión de una funcionalidad de visión artificial que evalúe la información de la solicitud de inversión y como consecuencia genere una métrica criptográfica que asocie una cuenta al usuario y créditos en forma de tokens intercambiables a lo largo de la experiencia gamificada.

5. Conclusiones y trabajo a futuro

En este trabajo, se muestra la aplicación de la metodología COSMIC para estimar el esfuerzo de desarrollo en PFC, de un asistente virtual de inversión. Con este análisis, es posible determinar proyectos de productividad y costos del desarrollo, así como

también evaluar su funcionalidad en términos de los objetivos que persigue el desarrollo del asistente virtual y de los requerimientos de los usuarios.

La estimación del esfuerzo de un proyecto de software es una necesidad del mercado de desarrollo de software que representa una ventaja competitiva y es una estrategia de supervivencia para las empresas de base tecnológica. El diseño del Asistente Virtual de Inversión, estimado en PFC ayuda a tener una medida más precisa de los costos de desarrollo del proyecto y permite sistematizar el cálculo de los costos habilitando su replicación.

También los PFC son de ayuda al determinar los recursos humanos y tecnológicos que se requieren para el desarrollo de cada pieza de software. Finalmente, los PFC se proponen como herramienta para identificar oportunidades de disminución de la huella de carbono, por lo que será útil proseguir con el análisis de viabilidad sobre la efectividad energética tanto de la infraestructura computacional a utilizar como de los algoritmos de aprendizaje automático.

El desarrollo y los diagramas presentados, no son documentos estáticos y dependen de la perspectiva del usuario funcional que se describe, y de cada persona involucrada en el desarrollo y operación de la herramienta. Por tanto, no describen exhaustivamente las operaciones que el asistente virtual realiza aunque es una aproximación que permite observar el efecto de las métricas en la atención de los requerimientos de usuarios específicos.

Como trabajo futuro, se analiza el servicio de emparejamiento con inversionistas utilizando técnicas de análisis multivariable para extraer información de perfiles de inversores (bases de datos propias) y emparejarlos con oportunidades de inversión adecuadas a sus preferencias y objetivos. Se consideran parámetros como la intención de inversión, sectores de interés y condiciones de inversión para reducir riesgos y evaluar oportunidades de manera efectiva.

Asimismo, se propone comparar los PFC que involucran el desarrollo de la aplicación desde el inicio, sin el apoyo de herramientas como Watson IBM, o Google, así como el desarrollo de tecnologías propias de Inteligencia Artificial para potenciar las funcionalidades del asistente virtual.

Referencias

1. Instituto mexicano de competitividad: Desarrollando las PyMES que requiere México (2009) imco.org.mx/pymes_que_requiere_mexico_2009
2. Instituto nacional de estadística: Demografía de los negocios. INEGI (2019) www.inegi.org.mx/temas/dn/
3. cuantico vc: Estado de la industria VC en Latinoamérica en 2023 (2023) cuantico.la/estado-de-la-industria-vc-en-latinoamerica-en-2023
4. Dropbox DocSend: The startup fundraising playbook (2022) www.docsend.com/index/startup-fundraising/
5. OpenAI: Introducing ChatGPT (2022) openai.com/blog/chatgpt
6. Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., Duvenaud, D.: Ffjord: free-form continuous dynamics for scalable reversible generative models. In: Seventh International Conference on Learning Representations (2018) doi: 10.48550/ARXIV.1810.01367

7. Radford, A., Wu, J., Amodei, D., Amodei, D., Clark, J., Brundage, M., Sutskever, I.: Better language models and their implications. OpenAI (2019) openai.com/research/better-language-models
8. COSMIC group: The most reliable way to measure software (2022) cosmic-sizing.org
9. COSMIC group: COSMIC method: Introduction (2021) cosmic-sizing.org/cosmic-sizing/intro/
10. Hindle, K.: A grounded theory for teaching entrepreneurship using simulation games. Simulation and Gaming, vol. 33, no. 2, pp. 236–241 (2002) doi: 10.1177/1046878 102332012
11. Kohnke, L.: A pedagogical chatbot: A supplemental language learning tool. Regional Language Centre of Choice Journal (2022) doi: 10.1177/00336882211067054
12. Yin, J., Goh, T., Yang, B., Xiaobin, Y.: Conversation technology with micro-learning: The impact of chatbot-based learning on students' learning motivation and performance. Journal of Educational Computing Research, vol. 59, no. 1, pp. 154–177 (2020) doi: 10.1177/073 5633120952067
13. Shum, H., He, X., Li, D.: From Eliza to Ciaoice: Challenges and opportunities with social chatbots. Frontiers of Information Technology and Electronic Engineering, vol. 19, no. 1, pp. 10–26 (2018) doi: 10.1631/fitee.1700826
14. Croes, E. A. J., Antheunis, M. L.: Can we be friends with Mitsuku? A longitudinal study on the process of relationship formation between humans and a social chatbot. Journal of Social and Personal Relationships, vol. 38, no. 1, pp. 279–300 (2020) doi: 10.1177/026540 7520959463
15. Purington, A., Taft, J. G., Sannon, S., Bazarova, N. N., Taylor, S. H.: Alexa is my new BFF. In: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (2017) doi: 10.1145/3027063.3053246
16. Abran, A., Vogelezang, F. W.: Early software sizing with COSMIC, practitioners. Practitioners Guide (2020) cosmic-sizing.org/publications/early-software-sizing-with-cosmic-practitioners-guide/
17. Valdés, F., Pedraza-Coello, R., Olguín-Barrón, F. C.: Cosmic sizing of RPA software: A case study from a proof of concept implementation in a banking organization. In: International Workshop on Statistical Modelling-Mensura, vol. 2725 (2020)
18. Google: Eficiencia. Google Data Center (2022) google.com/about/datacenters/efficiency
19. Sehgal, A., McDonnell, D.: Four trends driving global utility digitization. Amazon Web Services for Industries (2020) aws.amazon.com/es/blogs/industries/four-trends-driving-global-utility-digitization/
20. Walsh, N.: Sharing the latest improvements to efficiency in Microsoft's datacenters. Microsoft Cloud for Sustainability (2022) azure.microsoft.com/en-us/blog/sharing-the-latest-improvements-to-efficiency-in-microsoft-s-datacenters/
21. IBM: Energy and climate (2022) www.ibm.com/about/environment/energy-climate
22. Luccioni, A. S., Viguier, S., Ligozat, A.: Estimating the carbon footprint of bloom, a 176b parameter language model (2022) doi: 10.48550/ARXIV.2211.02001
23. Patterson, D., Gonzalez, J., Holzle, U., Le, Q., Liang, C., Munguia, L., Rothchild, D., So, D. R., Texier, M., Dean, J.: The carbon footprint of machine learning training will plateau, then shrink. Computer, vol. 55, no. 7, pp. 18–28 (2022) doi: 10.1109/mc.2022.3148714

ISSN 1870-4069

Aplicación del algoritmo SCAN en el agrupamiento de imágenes de trayectorias espermáticas: Identificación de la heterogeneidad de la respuesta espermática a la Ketanserina

Eder Alejandro Rodríguez Martínez[1], Cindy Ursula Rivas Arzaluz[2], Andrés Aragón Martínez[2]

[1] University of Picardie Jules Verne, France

[2] Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Iztacala, México

armandres@gmail.com

Resumen. La identificación de subpoblaciones cinemáticas es de vital importancia para comprender la naturaleza biológica de la heterogeneidad de los espermatozoides. Actualmente, los datos de parámetros de movilidad obtenidos mediante un sistema de análisis espermático asistido por computadora (CASA) se han utilizado como entrada para diferentes algoritmos con el fin de identificar subpoblaciones cinemáticas. En contraste, las imágenes de las trayectorias solamente se han representado como ejemplos de los patrones de movilidad en cada subpoblación. En este estudio, se escribió un código en Python para reconstruir las imágenes de las trayectorias a partir de sus coordenadas; estas se utilizaron como entrada para el algoritmo de SCAN; y, se describió estadísticamente el efecto de la ketanserina en la cinemática de cada grupo identificado. Las imágenes de las trayectorias en cada grupo se mostraron de una manera que denominamos gráfico de Pollock. Se trataron muestras de semen de cerdo con distintas concentraciones de ketanserina y se utilizaron muestras no tratadas como control. La movilidad espermática en cada muestra se analizó a los 0 y 30 minutos de incubación. Se obtuvieron seis grupos (subpoblaciones). La subpoblación 2 presentó los valores más altos de velocidad espermática a los 0 y 30 minutos. Después de 30 minutos de incubación, la ketanserina indujo un aumento de los valores de la velocidad curvilínea a altas concentraciones, mientras que la linealidad y la velocidad rectilínea disminuyeron. Nuestro flujo de trabajo permite una mejor identificación de las subpoblaciones cinemáticas que el enfoque tradicional y proporciona información sobre la heterogeneidad de la respuesta a la ketanserina. Por lo tanto, este podría impactar significativamente en la investigación de la relación entre la heterogeneidad de los espermatozoides y la fertilidad. Adicionalmente, el código se puede encontrar en[1].

Palabras clave: Subpoblaciones cinemáticas, imágenes de trayectorias espermáticas, análisis espermático asistido por computadora, patrones de movilidad.

Eder Alejandro Rodríguez Martínez, Cindy Ursula Rivas Arzaluz, Andrés Aragón Martínez

Application of the SCAN Algorithm in the Clustering of Sperm Trajectory Images: Identification of the Heterogeneity of the Sperm Response to Ketanserin

Abstract. Identifying kinematic subpopulations is of vital importance for understanding the biological nature of sperm heterogeneity. Currently, mobility parameter data obtained through a computer-assisted sperm analysis system (CASA) has been used as input for different algorithms to identify kinematic subpopulations. In contrast, trajectory images have only been represented as examples of mobility patterns in each subpopulation. In this study, Python code was written to reconstruct trajectory images from their coordinates; then, the trajectory images were used as input for the SCAN algorithm; finally, the effect of ketanserin on the kinematics of each identified group was statistically described. The trajectory images in each group were displayed in a way we called Pollock plots. Pig semen samples were treated with different concentrations of ketanserin and untreated samples were used as controls. Sperm motility in each sample was analyzed at 0 and 30 minutes of incubation. Six clusters (subpopulations) were obtained. Subpopulation 2 had the highest sperm velocity values at 0 or 30 minutes. After 30 minutes of incubation, ketanserin induced an increase in curvilinear velocity values at high concentrations, while linearity and straight-line velocity decreased. Our workflow allows for better identification of kinematic subpopulations than the traditional approach and provides information on the heterogeneity of the response to ketanserin. Therefore, this could significantly impact research on the relationship between sperm heterogeneity and fertility. Additionally, the code can be found at[1].

Keywords: Sperm subpopulations, images of sperm trajectories, computer assisted sperm analysis, patterns of motility.

1. Introducción

Actualmente, el procedimiento estadístico para identificar subpoblaciones cinemáticas utiliza los datos de parámetros de movilidad, obtenidos de un sistema de análisis de espermatozoides asistido por computadora (CASA), para identificar grupos de datos con base en su similitud [1, 2].

Una vez que se identifican y describen estadísticamente las subpoblaciones, generalmente se presentan imágenes representativas de las trayectorias seguidas por algunos espermatozoides, en cada subpoblación [3, 4]. Sin embargo, estos enfoques no utilizan toda la información potencial proporcionada por los sistemas CASA, es decir, los valores de los parámetros de movilidad y las trayectorias seguidas por los espermatozoides.

El software de los sistemas CASA rastrea cada espermatozoide siguiendo el centroide de los píxeles que definen la cabeza; posteriormente, un algoritmo calcula las velocidades y otros parámetros de movilidad de cada espermatozoide, basándose en las coordenadas del centroide.

[1] github.com/eder1234/traj_img_clusters

A continuación, diferentes técnicas estadísticas utilizan estos parámetros de movilidad para identificar subpoblaciones cinemáticas [5]. Aunque estos parámetros proporcionan información relevante relacionada con la movilidad de los espermatozoides, no son suficientes para, de manera inversa, reconstruir la trayectoria representada por las coordenadas a lo largo del tiempo.

En este sentido, podemos afirmar, entonces, que los parámetros de movilidad están asociados con la trayectoria del espermatozoide y que la trayectoria puede reconstruirse a partir de las coordenadas. En principio, las imágenes de las trayectorias espermáticas (también llamadas patrones de movilidad) [5] son en realidad una colección heterogénea de trayectorias, como se puede apreciar en las imágenes de trayectorias y en los valores de los parámetros de movilidad asociados a ellas [5, 6].

Hasta ahora, las imágenes de las trayectorias no se habían considerado como una entrada para algoritmos de aprendizaje de máquinas, que permiten identificar subpoblaciones espermáticas. Esta situación abre una ventana de oportunidad para desarrollar nuevos enfoques que permitan comprender, cuantitativamente, los patrones de movilidad espermática.

En consecuencia, la información biológica subyacente en los datos obtenidos de los sistemas CASA podría explotarse aún más. En primer lugar, describiendo finamente la cinemática de las subpoblaciones y, en segundo lugar, mejorando la interpretación de la movilidad de los espermatozoides en condiciones fisiológicas y experimentales.

La identificación de subpoblaciones cinemáticas proporciona información valiosa sobre la fisiología de los espermatozoides. Sin embargo, en la mayoría de los estudios en los que se han utilizado sistemas CASA no se han identificado subpoblaciones cinemáticas. Por otra parte, se ha reportado que los espermatozoides expresan proteínas relacionadas con la comunicación serotoninérgica, como receptores, transportadores y proteínas metabolizadoras [7]. En ese artículo, los autores describieron un aumento en las velocidades promedio de los espermatozoides debido a la exposición a la serotonina.

En otros estudios se describieron los valores promedio de los parámetros de movilidad de los espermatozoides expuestos a agonistas o antagonistas de los receptores de serotonina [8, 9]. Actualmente, se desconoce la estructura de las subpoblaciones cinemáticas de espermatozoides expuestos a sustancias que regulan la comunicación serotoninérgica. Sin embargo, si la serotonina estimula la movilidad de los espermatozoides, entonces el bloqueo de la comunicación serotoninérgica podría cambiar la estructura de las subpoblaciones cinemáticas.

En un acercamiento más reciente, el problema se abordó por medio de un agrupamiento de imágenes de trayectorias ya que estas últimas contienen información relevante que no es completamente descrita por los parámetros de movilidad [10]. Específicamente, la tarea de agrupamiento la llevó a cabo el algoritmo aglomerativo jerárquico, y el efecto de la ketanserina fue analizado estadísticamente en cada subpoblación espermática.

Con respecto a otros métodos de agrupamiento llevado a cabo por expertos [10], el aglomerativo jerárquico es un algoritmo objetivo y automático que no requiere de intervención humana. Sin embargo, una objeción que puede hacerse al uso del algoritmo aglomerativo jerárquico es que no es robusto a la rotación.

Por lo tanto, en este trabajo se optó por la implementación del algoritmo de Agrupamiento Semántico por Adaptación de los Vecinos más Próximos (o SCAN, por sus siglas en inglés) [11] para agrupar las imágenes de las trayectorias espermáticas. El

algoritmo SCAN es robusto a transformaciones de imágenes deseadas, por ejemplo, la rotación, la escala y la translación, entre otras.

Esto se debe a que la función de costo es capaz de destruir la información no deseada relativa a las transformaciones de imagen. En consecuencia, los agrupamientos resultantes son más homogéneos que los propuestos por el algoritmo aglomerativo jerárquico.

El objetivo de este trabajo fue desarrollar un método computacional, basado en algoritmos de aprendizaje profundo, para identificar subpoblaciones basadas en imágenes de trayectorias espermáticas. Este método computacional permite identificar los cambios estructurales en las subpoblaciones cinemáticas de espermatozoides de cerdo expuestas a ketanserina.

2. Materiales y métodos

Se utilizaron diez dosis de semen, dos dosis de cada uno de cinco cerdos fértiles; las dosis se obtuvieron del Centro de Enseñanza, Investigación y Extensión en Producción Porcina de la Facultad de Medicina Veterinaria y Zootecnia (UNAM). Las dosis de semen se produjeron inicialmente para inseminación artificial, fueron transportadas y procesadas de acuerdo con [12].

La unidad experimental en este trabajo fue la alícuota. De cada dosis de semen, se tomaron cuatro alícuotas (N = 40 alícuotas); 10 de estas dosis no se trataron y sirvieron como control, y el resto se trató con distintas concentraciones de ketanserina, según se especifica a continuación.

Se utilizó un sistema CASA de código abierto. La movilidad fue evaluada siguiendo el método descrito en [12]. De manera similar a [13], se evaluó visualmente la trayectoria de cada espermatozoide identificado y registrado en cada campo antes del análisis de la secuencia de seguimiento. Esta consideración se llevó a cabo para eliminar posibles restos celulares y disminuir el riesgo de incluir trayectorias poco claras en el análisis.

Se registraron los valores cinemáticos de cada uno de los siguientes parámetros de movilidad para cada espermatozoide analizado: velocidad promedio del recorrido (VAP, μm/s), velocidad curvilínea (VCL, μm/s), velocidad en línea recta (VSL, μm/s), frecuencia de cruce de batido (BCF, Hz), linealidad (LIN, VSL/VCL), coeficiente de rectitud (STR, VSL/VAP), amplitud de desplazamiento lateral de la cabeza (ALH, μm) y bamboleo (WOB, VAP/VCL).

El plugin CASA_bgm genera una hoja de datos con los valores cinemáticos de cada espermatozoide y muestra una imagen donde se describen las trayectorias de todos los espermatozoides analizados en una secuencia de video [13].

El plugin permite guardar las coordenadas de cada espermatozoide analizado. Para automatizar la construcción de los dataset para las coordenadas de los espermatozoides y los datos cinemáticos individuales, se escribió un script en Python que toma los datos de la hoja de resultados como entrada y luego construye dos archivos csv: uno para las coordenadas y otro para los parámetros de movilidad (el script en Python se nombró log_motility_data-6.py y está disponible en doi: 10.7910/DVN/YI3N4Q). El archivo de coordenadas se utilizó como entrada para otros scripts en Python, que convirtieron las coordenadas en imágenes individuales.

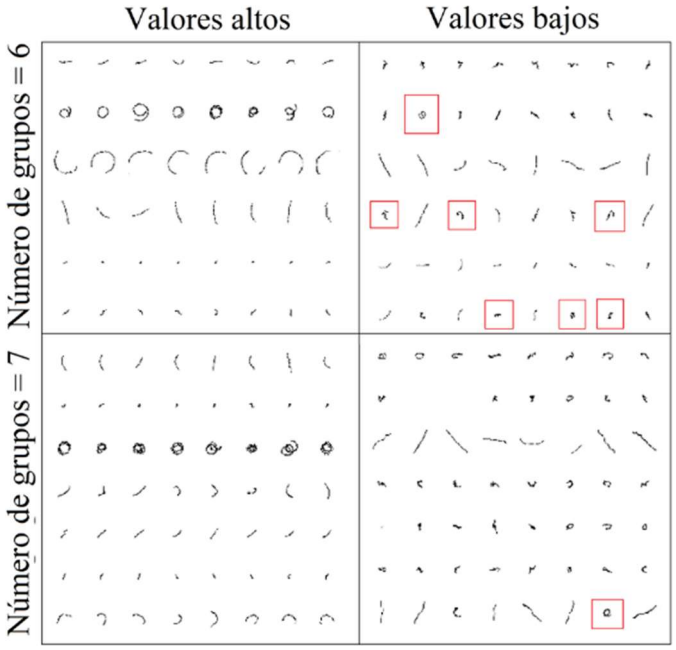


Fig. 1. Eficacia del agrupamiento de imágenes de trayectorias realizadas por el algoritmo SCAN. En vertical, se muestran los resultados del agrupamiento realizado por SCAN, con un ajuste a 6 y 7 grupos. En horizontal, se muestran las imágenes más cercanas (Valores altos) y más alejadas (Valores bajos) del centro de cada grupo al que pertenecen. Los cuadrados rojos señalan a las trayectorias semánticamente mal agrupadas.

Las imágenes de trayectorias se generaron como imágenes png utilizando matplotlib versión 3.6. Se utilizó el módulo de imagen Pillow Imaging Library para convertir cada trayectoria en un vector binario; este módulo tiene la opción "L" con la que los píxeles se asignan a su límite más cercano, ya sea negro (0) o blanco (256) [14].

Se verificó que los datos fueran binarios y se establecieron todos los valores como 1; por lo tanto, nuestros datos de entrada fueron un conjunto de vectores binarios con valores de 0 o 1. A partir de ahí, las imágenes individuales se almacenaron en una carpeta (el script se llamó traj-ah-6-full-0.ipynb y está disponible como un Jupyter notebook en doi:10.7910/DVN/CBMKVA).

Una vez que las imágenes de trayectorias fueron generadas y almacenadas, el siguiente paso consistió en clasificarlas o agruparlas automáticamente. En resumen, el problema consiste en agrupar las imágenes de forma que cada imagen sea semánticamente consistente con las otras que pertenecen al mismo grupo.

En la literatura, existen diferentes algoritmos de agrupamiento de imágenes, ej. Agrupamiento Selectivo Pseudo-etiquetado (o SPC, por sus siglas en inglés) [15], Agrupamiento Pseudo-etiquetado de Imágenes (o SPICE, por sus siglas en inglés) [16], SCAN [11], entre otros.

En este artículo, se optó por utilizar este último algoritmo debido a tres principales razones. La primera razón está relacionada a que se desconocen completamente las

etiquetas y el número de grupos. La segunda razón está relacionada con la función de costo de SCAN que lo hace robusto a transformaciones de imagen.

El agrupamiento debe de ser robusto a la rotación; sin embargo, debido a que el tamaño de la trayectoria es relevante, el agrupamiento debe ser sensible a la escala. La transformación de desplazamiento no debe ser considerada, ya que todas las trayectorias están centradas en la imagen (Figura 1).

La tercera razón está relacionada a que el algoritmo calcula un valor, relacionado con la distancia de cada imagen al centro de su grupo, que es de gran utilidad para calcular el número mínimo de grupos. El algoritmo SCAN se compone de tres fases: aprendizaje auto-supervisado, agrupamiento y auto etiquetado [11].

Durante la primera etapa, el algoritmo aprende información relevante de una gran base de datos que contiene imágenes, ej. CIFAR-10 [17]. Después, SCAN realiza un agrupamiento de imágenes de trayectorias utilizando el algoritmo de los K vecinos más próximos y una función de costo que destruye la información de bajo nivel, ej. información pixel, según las transformaciones de imagen definidas.

Por último, SCAN realiza un auto-etiquetado considerando las imágenes de trayectoria mejor agrupadas para aumentar su precisión. En la Tabla 1 se puede encontrar la descripción del algoritmo SCAN en pseudocódigo. Basado en el número de grupos previamente propuesto en [10], el primer agrupamiento de SCAN se llevó a cabo con 6 grupos.

Tabla 1. Pseudocódigo del algoritmo SCAN. Agrupamiento Semántico por Adaptación de los Vecinos más Próximos (SCAN).

1. **Entrada**: Base de datos D, Grupo C, Tarea τ, Red Neuronal Φ_θ y Φ_η, Vecinos $N_D = \{\}$.
2. Optimizar Φ_θ con la tarea τ. (Tarea de pretexto)
3. **Para** $X_i \in D$, **hacer**
4. $N_D \leftarrow N_D \cup N_{Xi}$, con N_{Xi} = K vecindario muestra de Φ_θ (X_i)
5. **Fin del ciclo**
6. **Mientras** SCAN-pérdida decremente **hacer** (Agrupamiento)
7. Actualizar Φ_η con SCAN-pérdida.
8. **Fin de ciclo**
9. M**ientras** longitud(Y) incremente **hacer** (Autoetiquetado)
10. $Y \leftarrow (\Phi_\eta (D) >$ umbral)
11. Actualizar Φ_η con pérdida de entropía cruzada, i.e. $H(\Phi_\eta(D), Y)$
12. **Fin de ciclo**
13. **Regresar**: Φ_η (D). (D es dividido entre los grupos C)

Sin embargo, al observar las imágenes peores agrupadas (es decir, que se localizan lejos del centro del grupo), se observó que sería necesario un grupo adicional. Entonces, se volvió a ejecutar el algoritmo con 7 grupos y notamos que todas las imágenes, incluyendo aquellas lejanas al centro de su grupo, estaban satisfactoriamente agrupadas (Figura 1).

Aquí es importante notar que es redundante continuar agrupando las imágenes con 8 grupos o más. Esto se debe a que los grupos adicionales podrían ser agrupados en

super grupos, de tal forma que el número mínimo de grupos (7, en este caso) es mantenido [10]. Salvo el número de grupos, ningún hiper-parámetro fue modificado de la versión original del código.

El algoritmo se implementó en un notebook Jupyter (versión JupyterLab 3.3.2) bajo Anaconda versión 3.21.5 ejecutando Python versión 3.8.10 y la librería Scikit-learn versión 1.0.2 [18], pandas versión 1.4.2, numpy versión 1.21.5 y matplotlib versión 3.6. Para el análisis estadístico se compararon los valores promedio de cada parámetro de movilidad en cada concentración de ketanserina contra el control en cada grupo y tiempo.

Los efectos de la ketanserina en cada subpoblación se examinaron mediante un análisis de varianza unidireccional (ANOVA) en un modelo de contrastes lineales (contrastes ortogonales). Los datos se presentan como M ± SEM. Para todas las pruebas, se consideró significativo $p < 0.05$. Las ANOVA se realizaron utilizando el paquete R car versión 2.1-0 [19]. Todos los datos fueron analizados estadísticamente con el software R versión 3.4.3 [20].

3. Resultados

La opción de imprimir las coordenadas de cada espermatozoide del complemento CASA_bgm [13] proporcionó los datos de una manera complicada de seguir. Por lo tanto, se modificó el código del plugin. La nueva versión del complemento BGM permite obtener correctamente las coordenadas de cada espermatozoide detectado, junto con los datos cinemáticos para cada espermatozoide (este complemento se llamó "CASA_RA" y está disponible en DOI: 10.7910 / DVN / YI3N4Q).

Se construyeron dos archivos con los resultados del CASA_RA: uno para las coordenadas (llamado archivo "traj", que contiene 1,634,950 filas) y otro para los parámetros de movilidad (llamado "parámetros de movilidad", que contiene 20,805 filas, 11,394 en tiempo 0 y 9411 en tiempo 30; ambos dataset están disponibles en DOI: 10.7910 / DVN / JDJVBT).

La construcción de los archivos fue tediosa y consumió mucho tiempo; por lo tanto, se escribió un script en Python para automatizar el proceso y construir los archivos de manera iterativa (el script en Python, log_motility_data-6.py e instrucciones de uso están disponibles en doi: 10.7910 / DVN / YI3N4Q).

Los archivos de los parámetros de movilidad y de las trayectorias fueron tomados por otro script en Python para construir una instancia de la clase dataframe, la cual se nombró "master dataframe".En el master dataframe, las filas corresponden a los espermatozoides individuales; dos columnas contienen datos categóricos para las diferentes concentraciones de ketanserina y tiempo, respectivamente; ocho columnas contienen valores flotantes para cada uno de los ocho parámetros de movilidad evaluados; finalmente, una columna contiene las listas de coordenadas.

Un script de Python tomó las coordenadas en el master dataframe como entrada para construir la imagen de cada trayectoria y las guardó en un nuevo directorio (el script está disponible como un notebook Jupyter llamado "traj-ah-6-full-0.ipynb" en doi: 10.7910 / DVN / CBMKVA).

Las imágenes de trayectorias sirvieron como entrada para el algoritmo SCAN.

Tabla 2. Efecto de las diferentes concentraciones de ketanserina en las subpoblaciones, identificadas por algoritmos de aprendizaje de máquinas SCAN. Las mediciones de la movilidad espermática se realizaron inmediatamente después de poner ketanserina o después de 30 minutos de incubación. Los datos representan la $media\_$ +SE.

Tiempo	Sub-population	Ketanserin (nM)	VCL (μm/s)	VAP (μm/s)	VSL (μm/s)	LIN (%)	STR (%)	WOB (%)	BCF (Hz)	Scan_values	n
0	5	0	99.08 ± 1.36	61.23 ± 0.77	43.66 ± 0.74	0.46 ± 0.01	0.72 ± 0.01	0.63 ± 0.00	39.54 ± 0.24	0.65 ± 0.01	353
		10	107.11 ± 1.74*	64.88 ± 0.96*	44.81 ± 0.90	0.44 ± 0.01	0.70 ± 0.01	0.62 ± 0.01	39.34 ± 0.27	0.65 ± 0.01	314
		100	105.97 ± 2.18*	62.71 ± 1.06	43.65 ± 0.92	0.44 ± 0.01	0.71 ± 0.01	0.61 ± 0.01*	39.79 ± 0.27	0.63 ± 0.01	284
		1000	111.67 ± 2.24*	67.87 ± 1.09*	49.62 ± 0.92*	0.48 ± 0.01	0.75 ± 0.01	0.63 ± 0.01	38.93 ± 0.27	0.69 ± 0.01*	294
	6	0	103.39 ± 1.49	48.59 ± 0.68	28.06 ± 0.60	0.28 ± 0.01	0.59 ± 0.01	0.48 ± 0.00	38.31 ± 0.20	0.44 ± 0.01	593
		10	116.62 ± 1.98 *	54.05 ± 0.95*	31.04 ± 0.82*	0.28 ± 0.01	0.58 ± 0.01	0.47 ± 0.00	37.75 ± 0.24	0.47 ± 0.01*	512
		100	113.31 ± 2.09 *	52.69 ± 1.00*	30.10 ± 0.83	0.27 ± 0.01	0.59 ± 0.01	0.47 ± 0.00	38.46 ± 0.24	0.48 ± 0.01*	469
		1000	129.44 ± 1.88*	58.95 ± 0.89*	31.77 ± 0.75*	0.25 ± 0.01*	0.55 ± 0.01*	0.46 ± 0.00*	38.13 ± 0.21	0.50 ± 0.01*	661
30	5	0	99.21 ± 1.73	65.31 ± 0.96	51.15 ± 0.93	0.54 ± 0.01	0.79 ± 0.01	0.67 ± 0.00	38.90 ± 0.27	0.70 ± 0.01	382
		10	102.36 ± 1.86	65.29 ± 0.99	45.87 ± 0.85*	0.48 ± 0.01*	0.72 ± 0.01*	0.65 ± 0.01*	39.05 ± 0.27	0.67 ± 0.01	303
		100	107.53 ± 1.95*	65.86 ± 1.06	45.91 ± 0.98*	0.45 ± 0.01*	0.71 ± 0.01*	0.62 ± 0.00*	39.40 ± 0.27	0.66 ± 0.01*	328
		1000	110.65 ± 1.75*	68.74 ± 0.92*	50.96 ± 0.85	0.48 ± 0.01*	0.75 ± 0.01*	0.64 ± 0.00*	38.61 ± 0.24	0.68 ± 0.01*	462
	6	0	100.32 ± 2.81	49.95 ± 1.49	34.78 ± 1.30	0.35 ± 0.01	0.71 ± 0.01	0.50 ± 0.01	38.17 ± 0.39	0.41 ± 0.01	215
		10	101.50 ± 2.99	50.89 ± 1.69	32.69 ± 1.38	0.33 ± 0.01	0.67 ± 0.02	0.50 ± 0.01	38.84 ± 0.43	0.44 ± 0.01*	179
		100	104.75 ± 2.47	48.05 ± 1.26	32.29 ± 1.16	0.31 ± 0.01*	0.67 ± 0.01	0.46 ± 0.01*	39.30 ± 0.33*	0.42 ± 0.01	264
		1000	109.21 ± 2.33*	51.88 ± 1.20	33.77 ± 1.03	0.32 ± 0.01*	0.66 ± 0.01*	0.48 ± 0.01*	38.00 ± 0.30	0.45 ± 0.01*	336

*Indica diferencia significativa (P<0.05, ANOVA de una vía) vs el control en el mismo cluster. VCL, velocidad curvilínea; VAP, velocidad promedio; VSL, velocidad rectilínea; LIN, linealidad; STR, rectitud; WOB, bamboleo; BCF, frecuencia de batido; n, es el número de espermatozoides en cada cluster.

Después del agrupamiento, el script de Python creó una nueva columna con los identificadores de los grupos (subpoblaciones) para cada fila en el master dataframe original. Finalmente, la descripción estadística de las subpoblaciones de imágenes de trayectorias, en cada tratamiento con ketanserina y tiempo, se realizó en los subdataset de los parámetros de movilidad.

Después del agrupamiento, el script de Python creó una nueva columna con los identificadores de los grupos (subpoblaciones) para cada fila en el master dataframe original. Finalmente, la descripción estadística de las subpoblaciones de imágenes de trayectorias, en cada tratamiento con ketanserina y tiempo, se realizó en los subdataset de los parámetros de movilidad.

Estos subconjuntos se obtuvieron del master dataframe con base en los datos categóricos y los identificadores de subpoblaciones. El algoritmo SCAN permitió agrupar las imágenes de las trayectorias espermáticas en siete subpoblaciones, numeradas de 0 a 6 para cada tiempo de evaluación. Para obtener una representación gráfica de las imágenes de trayectorias en cada subpoblación, construimos gráficos "Pollock" (Figura 2). El agrupamiento de las imágenes en las subpoblaciones 5 y 6 corresponden a trayectorias largas curvadas.

Las trayectorias con curvas más cerradas, incluso formando bucles se observaron en la subpoblación 6. El porcentaje de espermatozoides en los grupos 0-6 en el tiempo 0 fue de 2.39, 19.94, 16.30, 17.21, 12.92, 11.16 y 20.04, respectivamente; mientras que en el tiempo 30 fue de 3.06, 25.59, 15.83, 16.48, 12.60, 15.77 y 10.63, respectivamente.

A continuación, se describirá el efecto de la ketanserina en los espermatozoides de las subpoblaciones 5 y 6 puesto que son aquellas que presentaron las velocidades más altas en los controles, lo cual es biológicamente relevante; sin embargo, los efectos de la ketanserina en el resto de las subpoblaciones se encuentran en la Tabla 2.

Notablemente en las subpoblaciones 5 y 6 también se observaron los valores más altos del índice de clasificación de SCAN (Fig. 3).

Los efectos de la ketanserina en las trayectorias de los espermatozoides en la subpoblación 5 y 6 se muestran en la Figura 3. Al tiempo 0 las velocidades de los controles fueron significativamente mayores en los tratamientos con altas concentraciones de ketanserina (VCL, VAP y VSL); mientras que la linealidad de las trayectorias (LIN y STR) no se modificó.

Sin embargo, a los 30 minutos se observó una disminución significativa en los valores de linealidad en los tratamientos con ketanserina, principalmente en la subpoblación 5. Los índices de SCAN aumentaron en el tiempo 0, pero tienden a disminuir al tiempo 30 (Tabla 2). Gráficamente estos cambios se aprecian como trayectorias más curvadas (Fig. 3).

Al tiempo 0 los índices promedio de SCAN en la subpoblación 5 aumentó significativamente en la concentración más alta de ketanserina; pero aumentó significativamente en todas las concentraciones de ketanserina en la subpoblación 6. Al tiempo 30 los valores de SCAN disminuyeron significativamente en los tratamientos con ketanserina en la subpoblación 5, pero aumentaron significativamente en la subpoblación 6 (Fig. 3).

4. Conclusiones y trabajos a futuro

Hasta ahora, la identificación estadística de subpoblaciones cinemáticas se desarrolló en función de los parámetros de movilidad, obtenidos de los sistemas CASA. Una vez identificadas las subpoblaciones cinemáticas, algunos autores proporcionan imágenes representativas de las trayectorias seguidas por uno o unos pocos espermatozoides en cada subpoblación.

Hasta donde sabemos, este es el primer trabajo en el que se reconstruye la trayectoria de cada espermatozoide evaluado en un sistema CASA, para utilizarlas como entrada para un algoritmo de agrupamiento de aprendizaje automático; identificando y describiendo así subpoblaciones cinemáticas.

Los patrones de movilidad mencionados son la representación gráfica de las trayectorias de los espermatozoides observadas durante un período determinado. Aunque la dimensión tiempo estuvo ausente en el análisis de las imágenes de trayectorias, se recuperó parcialmente cuando los valores de los parámetros de movilidad se asociaron con las imágenes de trayectorias. Anteriormente, se ha informado sobre la importancia de algunos patrones de movilidad en la fertilización [21].

En contraste con la noción convencional de que los espermatozoides fértiles siempre son vigorosos, móviles y más lineales, los resultados en toros han demostrado que los patrones sinuosos están asociados con la fertilidad [21]. En nuestro trabajo, encontramos subpoblaciones caracterizadas por la presencia de trayectorias sinuosas (subpoblaciones 5 y 6) y no lineales.

Esto concuerda con el hecho de que porcentajes más significativos de espermatozoides rápidos y no lineales parecen tener una mayor capacidad de fertilización [22]. El manejo estadístico de los datos impacta en la predicción de la fertilidad. Por ejemplo, en el análisis de subpoblaciones de toros asociado con el

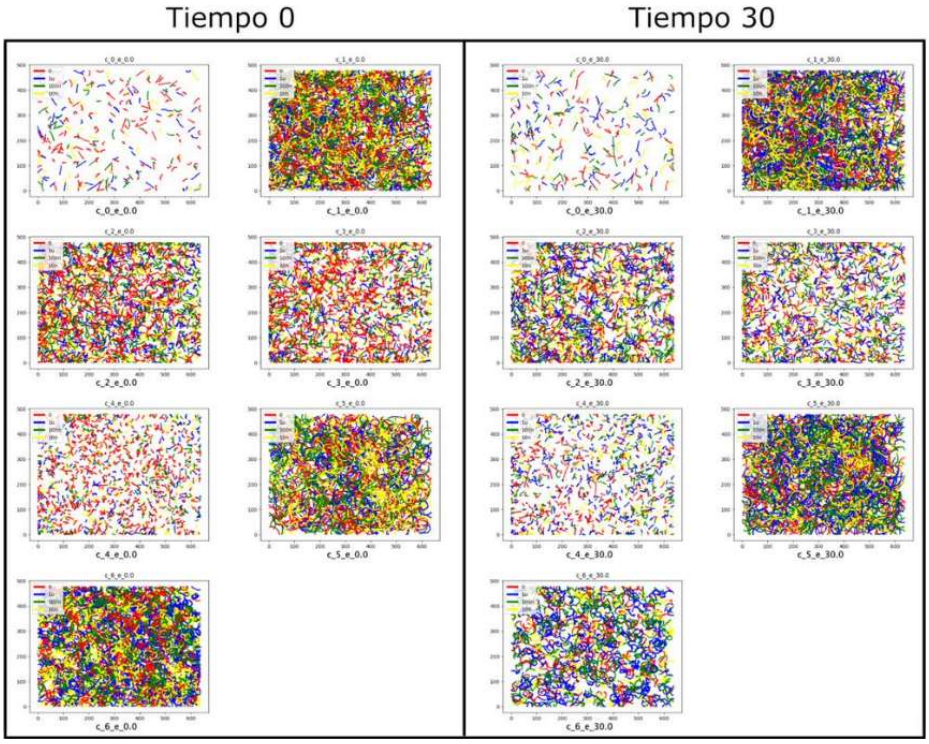


Fig. 2. Las imágenes de las trayectorias en cada tratamiento con ketanserina se encuentran en todos los grupos obtenidos por el algoritmo SCAN. En cada gráfico de Pollock se dibujó la trayectoria de los espermatozoides individuales en cada subpoblación. Las trayectorias en cada subpoblación lucen muy similares; mientras que la distribución de las trayectorias, coloreadas por tratamiento parecen distribuirse aleatoriamente. Las subpoblaciones 5 y 6 presentaron trayectorias largas y curvadas.

método de agrupamiento se puede considerar efectivo para predecir una mayor fertilidad [22].

Sin embargo, en cerdos, se agrupan los eyaculados, y el promedio de cada una de las variables cinemáticas tiene una capacidad predictiva limitada con respecto al tamaño de la camada y las variables de fertilidad [23]. Estudios similares a los previamente descritos, podrían aprovechar el método computacional que se describe aquí.

En este trabajo, se utilizaron espermatozoides de dosis de semen listas para inseminar. Por lo tanto, el semen se encontraba diluido, pero los espermatozoides no estaban lavados; por lo que, posiblemente la serotonina, entre otras sustancias presentes en el medio, podrían afectar la movilidad espermática.

La presencia de serotonina en el semen está respaldada por el hecho de que la serotonina está presente en las vesículas seminales [24] y en la próstata [25], y la emisión de espermatozoides desde el testículo va acompañada de líquido seminal de las vesículas seminales y la próstata. En este trabajo, las subpoblaciones se formaron con las imágenes de trayectorias y se describieron cuantitativamente con los valores de los parámetros de movilidad asociados con las trayectorias.

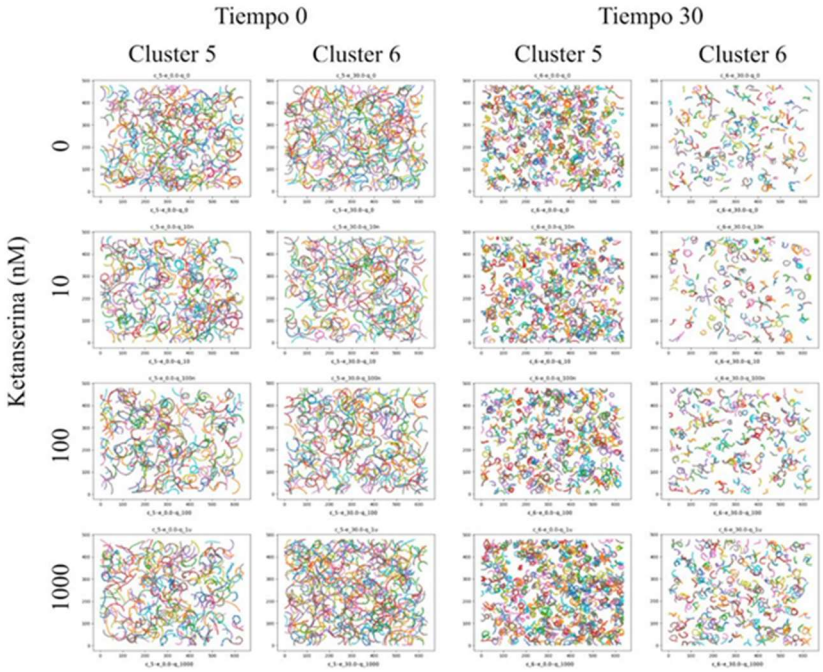


Fig. 3. La ketanserina indujo cambios en las trayectorias espermáticas. El dataset de imágenes de trayectorias sirvió como entrada para el algoritmo SCAN. Se identificaron 7 subpoblaciones. Los identificadores de tiempo de incubación y tratamiento con ketanserina se utilizaron para obtener los subdataset en cada subpoblación. Los gráficos de Pollock indican los cambios en las trayectorias espermáticas en las diferentes concentraciones de ketanserina al tiempo 0 y 30. Los espermatozoides en las subpoblaciones 5 y 6 presentaron las velocidades más altas y las trayectorias más curvas. La curvatura de las trayectorias tiende a aumentar después de 30 minutos de incubación con ketanserina. Los colores de las trayectorias son aleatorios.

Aquí, se proporciona un modelo computacional para discriminar imágenes de trayectorias, sin el sesgo inducido por el técnico. Nuestro modelo computacional sirvió para obtener una descripción detallada de las subpoblaciones cinemáticas después de la exposición a la ketanserina.

Este enfoque novedoso para manejar datos de movilidad podría proporcionar nuevas ideas sobre el papel de la movilidad espermática en la producción animal y la investigación básica sobre la regulación fisiológica de la movilidad espermática.

Las imágenes de las trayectorias tienen diferencias sutiles entre sí; estas diferencias están ocultas en los parámetros promedio de movilidad en cada subpoblación cinemática. Por otro lado, la forma de las trayectorias es diferente en longitud, grado de curvatura de la trayectoria completa, amplitud, frecuencia y forma.

El metabolismo de la ketanserina podría ayudar a explicar los resultados del efecto en VSL después de 30 minutos de incubación. La ketanserina en plasma decae triexponencialmente con vidas medias secuenciales de 0.13, 2 y 14.3 horas [26], lo cual concuerda con los cambios en los efectos en VCL y VSL observados después de 30 minutos en nuestro trabajo.

Por otro lado, el rápido aumento de las velocidades de los espermatozoides en los grupos 5 y 6 parece intrigante, puesto que se esperaba un efecto opuesto. Sin embargo, si la serotonina estuviera presente en el eyaculado y otros receptores 5-HT también están presentes en el espermatozoide de cerdo, entonces el antagonismo en los receptores 5-HT2 por la ketanserina favorecería la señalización por estos otros receptores.

Los diferentes resultados entre el trabajo de [10] y nuestro estudio podrían explicarse por factores como la especie, el uso de medios capacitantes durante la incubación, el tiempo de incubación antes de realizar las mediciones de movilidad, el antagonista distinto del receptor de serotonina utilizado entre trabajos y la estrategia de análisis de datos; es decir, el uso de valores promedio de parámetros de movilidad en [10] y el uso de un algoritmo de aprendizaje de máquinas, especializado en el agrupamiento de imágenes, utilizado en este trabajo.

En conclusión, el modelo computacional de agrupamiento de imágenes de trayectorias, desarrollado en este trabajo, proporciona un enfoque muy sutil para identificar el efecto de la ketanserina en los patrones de movilidad. La identificación de subpoblaciones cinemáticas siguiendo nuestro enfoque podría ayudar a identificar proporciones de espermatozoides relevantes para las técnicas de reproducción artificial.

Nuestro enfoque podría adoptarse para datos de otras especies (por ejemplo, humanos) y para reevaluar datos de trabajos anteriores, donde únicamente se utilizaron los parámetros de movilidad para identificar subpoblaciones cinemáticas (solamente se necesitarían los valores de coordenadas y parámetros de movilidad).

Agradecimientos. Este proyecto se desarrolló gracias al apoyo DGAPA-PAPIIT de la UNAM número IT201021.

Referencias

1. Martínez-Pastor, F.: What is the importance of sperm subpopulations? Animal Reproduction Science, vol. 246, p. 106844 (2022) doi: 10.1016/j.anireprosci.2021.106844
2. Ramón, M., Martínez-Pastor, F.: Implementation of novel statistical procedures and other advanced approaches to improve analysis of CASA data. Reproduction, Fertility and Development, vol. 30, no. 6, pp. 860–866 (2018) doi: 10.1071/RD17479
3. Ibănescu, I., Leiding, C., Bollwein, H.: Cluster analysis reveals seasonal variation of sperm subpopulations in extended boar semen. Journal of Reproduction and Development, vol. 64, no. 1, pp. 33–39 (2018) doi: 10.1262/jrd.2017-083
4. Henning, H., Petrunkina, A. M., Harrison, R. A. P., Waberski, D.: Cluster analysis reveals a binary effect of storage on boar sperm motility function. Reproduction, Fertility and Development, vol 26, no. 5, pp. 623–632 (2014) doi: 10.1071/RD13113
5. Goodson, S. G., Zhang, Z., Tsuruta, J. K., Wang, W., O'Brien, D. A.: Classification of mouse sperm motility patterns using an automated multiclass support vector machines model. Biology of reproduction, vol. 84, no. 6, pp. 1207–1215 (2011) doi: 10.1095/biolre prod.110.088989
6. Gacem, S., Valverde, A., Catalán, J., Yánez Ortiz, I., Soler, C., Miró, J.: A new approach of sperm motility subpopulation structure in donkey and horse. Frontiers in Veterinary Science, vol. 8 (2021) doi: 10.3389/fvets.2021.651477
7. Jimenez-Trejo, F., Tapia-Rodriguez, M., Cerbon, M., Kuhn, D. M., Manjarrez-Gutierrez, G., Mendoza-Rodriguez, C. A., Picazo, O.: Evidence of 5-HT components in human sperm:

implications for protein tyrosine phosphorylation and the physiology of motility. Reproduction, vol. 144, no. 6, pp. 677–685 (2012) doi: 10.1530/REP-12-0145
8. Fujinoki, M.: Serotonin-enhanced hyperactivation of hamster sperm. Reproduction, vol. 142, no. 2, pp. 255–266 (2011) doi: 10.1530/REP-11-0074
9. Sakamoto, C., Fujinoki, M., Kitazawa, M., Obayashi, S.: Serotonergic signals enhanced hamster sperm hyperactivation. Journal of Reproduction and Development, vol. 67, no. 4, pp. 241–250 (2021) doi: 10.1262/jrd.2020-108
10. Rodríguez-Martínez, E. A., Rivas, C. U., Ayala, M. E., Blanco-Rodríguez, R., Juarez, N., Hernandez-Vargas, E. A., Aragón, A.: A new computational approach, based on images trajectories, to identify the subjacent heterogeneity of sperm to the effects of ketanserin. Cytometry Part A (2023) doi: 10.1002/cyto.a.24732
11. Van-Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., Van-Gool, L.: Scan: learning to classify images without labels. In: 16th European Conference on Computer Vision, pp. 268–285 (2020) doi: 10.1007/978-3-030-58607-2_16
12. Rivas, C., Ayala, M. E., Aragón, A.: Effect of various pH levels on sperm kinetic parameters of boars. South African Journal of Animal Science, vol. 52, no. 5, pp. 693–704 (2022) doi: 10.4314/sajas.v52i5.13
13. Giaretta, E., Munerato, M., Yeste, M., Galeati, G., Spinaci, M., Tamanini, C., Mari, G., Bucci, D.: Implementing an open-access CASA software for the assessment of stallion sperm motility: Relationship with other sperm quality parameters. Animal Reproduction Science, vol. 176, pp. 11–19 (2017) doi: 10.1016/j.anireprosci.2016.11.003
14. Lundh, F., Clark, J. A.: Image module (2022) pillow.readthedocs.io/en/stable/reference/Image.html
15. Mahon, L., Lukasiewicz, T.: Selective pseudo-label clustering. In: Advances in Artificial Intelligence: 44th German Conference on AI, Virtual Event, pp. 158–178 (2021) doi: 10.1007/978-3-030-87626-5_12
16. Niu, C., Shan, H., Wang, G.: Spice: Semantic pseudo-labeling for image clustering. IEEE Transactions on Image Processing, vol. 31, pp. 7264–7278 (2022) doi: 10.1109/TIP.2022.3221290
17. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine learning in python. The Journal of Machine Learning Research, vol. 12, pp. 2825–2830 (2011)
19. Fox, J., Weisberg, S.: An R companion to applied regression. Sage Publications (2011)
20. R core team: R: A language and environment for statistical computing (2015) www.R-project.org/
21. Nagata, M. P. B., Endo, K., Ogata, K., Yamanaka, K., Egashira, J., Katafuchi, N., Yamanouchi, T., Matsuda, H., Goto, Y., Sakatani, M., Hojo, T., Nishizono, H., Yotsushima, K., Takenouchi, N., Hashiyada, Y., Yamashita, K.: Live births from artificial insemination of microfluidic-sorted bovine spermatozoa characterized by trajectories correlated with fertility. In: Proceedings of the National Academy of Sciences, vol. 115, no. 14, pp. E3087–E3096 (2018) doi: 10.1073/pnas.1717974115
22. Tsunokawa-Hidalgo, M. M., Marques-de-Almeida, A. B., Zito-de-Moraes, F. L., Palhaci-Marubayashi, R. Y., Ferreira-de-Souza, F., Rigo-Barreiros, T. R., Mello-Martins, M. I.: Sperm subpopulations influence the pregnancy rates in cattle. Reproduction in Domestic Animals, vol. 56, no. 8, pp. 1117–1127 (2021) doi: 10.1111/rda.13955
23. Barquero, V., Roldan, E. R. S., Soler, C., Vargas-Leitón, B., Sevilla, F., Camacho, M., Valverde, A.: Relationship between fertility traits and kinematics in clusters of boar ejaculates. Biology, vol. 10, no. 7, p. 595 (2021) doi: 10.3390/biology10070595.
24. Hsieh, J. T., Liu, S. P., Hsieh, C. H., Lai, M. K., Cheng, J. T.: An ex vivo evaluation of regulatory role of biogenic amines in rat seminal vesicle after pharmacological

manipulation. Life Sciences, vol. 63, no. 15, pp. PL221–PL229 (1998) doi: 10.1016/S0024-3205(98)00403-2

25. Mota, P., Barbosa-Martins, J., Moura, R. S., Lima, E., Miranda, A., Correia-Pinto, J., Carvalho-Dias, E.: Effects of testosterone replacement on serotonin levels in the prostate and plasma in a murine model of hypogonadism. Scientific Reports, vol. 10, no. 1, p. 14688 (2020) doi: 10.1038/s41598-020-71718-z
26. Persson, B., Heykants, J., Hedner, T.: Clinical pharmacokinetics of ketanserin. Clinical Pharmacokinetics, vol. 20, no. 4, pp. 263–279 (1991) doi: 10.2165/00003088-199120040-00002

ISSN 1870-4069

Hacia una categorización en el problema de asignación de recursos en logística humanitaria y su resolución utilizando aprendizaje automático

Galo Ruiz-Soto, Miguel González-Mendoza,
Jaime Mora-Vargas

Instituto Tecnológico y de Estudios Superiores de Monterrey,
Escuela de Ingeniería y Ciencias,
México

{A01799399, mgonza, jmora}@tec.mx

Resumen. El principal reto de la logística humanitaria es entregar los suministros apropiados en las cantidades adecuadas en el momento preciso y en el lugar correcto, en una situación de emergencia para aliviar el sufrimiento de los supervivientes. Las primeras 72 horas son críticas, no sólo para este fin, sino también para encontrar sobrevivientes que pudieron quedar atrapados después de un desastre natural. Identificamos dos problemas dentro de la logística humanitaria. El primero es el rescate oportuno de sobrevivientes. El segundo problema es la distribución eficiente, efectiva, y justa de los suministros de emergencia. En ambos, identificamos un problema de asignación de recursos y consideramos que el aprendizaje por refuerzo puede ser utilizado para resolverlo. El aprendizaje por refuerzo busca maximizar la recompensa obtenida por acciones en un ambiente en el largo plazo, por lo que es importante para el agente no solo considerar la recompensa inmediata por una acción sino también explorar nuevas acciones esperando obtener el mayor valor. Este es un trabajo exploratorio que busca encontrar una primera categorización del problema de la asignación de recursos y buscamos utilizar técnicas de aprendizaje por refuerzo en las 72 horas críticas posteriores a un desastre natural.

Palabras clave: Logística humanitaria, aprendizaje por refuerzo, aprendizaje automático, desastre natural, problema de asignación de recursos.

Towards a Categorization in the Resource Application Problem in Humanitarian Logistics and its Resolution Using Machine Learning

Abstract. The main challenge of Humanitarian Logistics is to deliver the appropriate supplies, in suitable quantities, in the precise moment and the right place in an emergency to relieve human suffering. The first 72 hours are critical not only to achieve this objective but also to find survivors in the aftermath of a natural disaster. We identified two problems in humanitarian logistics. The first one is the rescue of survivors under the rubble after a building collapse. The second problem is the efficient, effective, and equal distribution of emergency

supplies. We have identified a Resource Allocation Problem (RAP) in both cases, and we consider that Reinforcement Learning can be applied to solve the problem. Reinforcement Learning is about maximizing the reward obtained by actions taken in an environment in the long run, therefore it is important for the agent not only to consider the immediate reward but also to explore new actions to maximize the value. This is an exploratory work towards a categorization in the RAP and we intend to use reinforcement learning techniques in the critical 72 hours in the aftermath of a natural disaster.

Keywords: Humanitarian logistics, reinforcement learning, machine learning, natural disaster, resource allocation problem.

1. Introducción

La logística humanitaria debe de tratarse de manera separada a la logística comercial en varios aspectos, sobre todo tomando en cuenta los objetivos de una logística humanitaria post desastre. La logística humanitaria puede definirse como "un amplio rango de operaciones incluyendo la distribución de suministros médicos para la prevención de enfermedades de rutina, suministros alimenticios para luchar contra el hambre, y suministros críticos en la secuela de un desastre" [3].

Sin embargo, otros factores podrían considerarse en la definición tales como la velocidad de la distribución de los recursos, la asignación de políticas implementadas por las autoridades y la necesidad de encontrar sobrevivientes después de un desastre. "Cuando un desastre mayor golpea, una respuesta oportuna para salvar vidas y mitigar los sufrimientos de la población afectada se vuelve crítica.

De hecho, las primeras 72 horas de un esfuerzo de alivio de un desastre son críticas ya que las oportunidades de supervivencia posteriores a esa ventana de tiempo sin agua ni comida decrecen drásticamente. El reto es entregar los suministros apropiados de emergencia en cantidades suficientes exactamente cuándo y dónde son necesarios" [11].

El Problema de la Asignación de Recursos (RAP por sus siglas en inglés) en la logística humanitaria plantea que "el objetivo es entregar los recursos escasos de forma que se maximice la efectividad de la ayuda" [6].

La investigación se dirige hacia resolver el RAP utilizando técnicas de aprendizaje automático y en particular se explorará el aprendizaje por refuerzo y se contrastarán los resultados con otros modelos. La Fig. 1 presenta una representación gráfica del RAP. Es posible que se reciban donaciones (medicinas, agua, alimentos no perecederos, cobijas) de particulares, empresas u organizaciones. Estas donaciones se llevan a un centro de recursos para su clasificación y de ahí se distribuyen a refugios temporales.

De acuerdo con Sutton y Barto [9], el aprendizaje por refuerzo se refiere a aprender qué hacer para obtener la mayor recompensa. Sin embargo, no existe una guía para indicarle a un agente qué hacer, sino que debe de descubrirlo por sí mismo pensando que sus acciones pueden afectar no sólo la recompensa más inmediata sino también las recompensas siguientes. "Estas dos características, ensayo y error y recompensa retrasada, son las dos características distintivas más importantes del aprendizaje por refuerzo" [9].

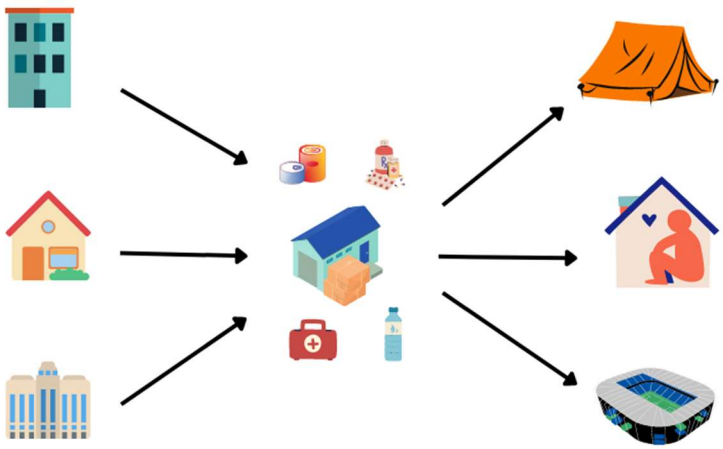


Fig. 1. Representación general del problema de la asignación de recursos en una situación de desastre.

Sin embargo, de acuerdo con Sutton y Barto, hay un juego entre explotar y explorar ya que "el agente tiene que explotar lo que ha experimentado anteriormente para obtener recompensa, pero también debe de explorar para lograr mejores elecciones en el futuro.

El agente debe de intentar una variedad de acciones y progresivamente favorecer aquellas que aparentemente son las mejores…Todos los agentes de aprendizaje por refuerzo tienen objetivos específicos, pueden sentir aspectos de su entorno, y pueden escoger acciones para influir en sus ambientes. Además, se suele asumir desde el inicio que el agente tiene que operar a pesar de una incertidumbre significativa sobre el ambiente al que se enfrenta" [9].

En el aprendizaje por refuerzo se cuentan cuatro elementos, una política, una señal de recompensa, una función de valor además de un modelo del ambiente. La política habla del comportamiento de un agente en un momento dado. La señal de recompensa define los buenos y malos eventos para el agente por lo que la política puede modificarse. En cada etapa el agente recibe un número que será conocido como la recompensa, haciendo que el objetivo del agente sea maximizar la recompensa total en el largo plazo.

Los valores indican el atractivo de los estados del ambiente después de tomar en cuenta aquellos estados que pueden seguir y las recompensas que se pueden encontrar ahí. Los valores son importantes para lograr mayor recompensa. En la evaluación de las decisiones se buscan acciones que provean el mayor valor y no la mayor recompensa dado que estas acciones obtendrán el mayor monto de recompensa sobre el tiempo [9].

2. Trabajos relacionados

Fan et al. [1] utilizan Deep Q Network (DQN) para la distribución de suministros a gran escala. Consideran centros locales de recursos que se encargan de distribuir

suministros a las distintas áreas afectadas (AA), sin embargo, opinan que las necesidades de recursos pueden ser distintas en cada una de las AA.

Su objetivo es "obtener la política optima de distribución de suministros de emergencia que minimice el costo económico mientras se minimiza el sufrimiento de los sobrevivientes" [1]. Utilizan tres criterios, la eficiencia reflejada en el costo de transporte de los suministros, la efectividad manifestada en costo de la privación de los suministros de emergencia, y la equidad reflejada por el costo de la distribución injusta de los suministros.

Yu et al. [10] también consideran las métricas de eficiencia (el costo de entrega basado en accesibilidad), efectividad (fundamentada en el costo inicial de los suministros recibidos además de la demanda), y equidad (reflejada en el costo de que todos los sobrevivientes obtengan las mismas condiciones), aplicando un algoritmo de Q-learning.

Sin embargo, consideran refugios temporales como AA y "sirven como el punto donde se reciben los recursos desde fuera del AA y se distribuyen a los supervivientes al interior del AA" [10]. Toman en cuenta un Centro Local de Respuesta (CLR) que recibe recursos desde un centro mayor de distribución usualmente administrado por el gobierno local. Utilizan accesibilidad en lugar de distancia para medir el costo de entrega [10]. De acuerdo con su investigación, su algoritmo de Q-learning supera al algoritmo de programación dinámica.

A pesar de que Hachiya et al. [2] utilizan técnicas de aprendizaje por refuerzo para la distribución de suministros de emergencia, solo consideran vehículos aéreos no tripulados (UAV, por sus siglas en inglés), los cuales pueden ser muy útiles en el transporte en caso de que vías terrestres estén bloqueadas después de un desastre, sin embargo, su capacidad de carga es relativamente pequeña por lo que para hacer una distribución justa de los suministros se tendrían que realizar varios viajes.

Mora-Ochomogo et al. [5] aplican el proceso de decisión de Markov para representar las operaciones en CLR que reciben donaciones. Los CLR deciden qué donaciones recibir por lo que el costo de equidad no se aplica de la misma manera que lo hacen otros autores. La tabla 1 presenta un resumen de trabajos relacionados a la resolución del RAP con distintos modelos o marcos de trabajo.

3. Hacia una categorización en el problema de asignación de recursos en logística humanitaria

Se pueden identificar dos problemas distintos en logística humanitaria. Por un lado, es esencial poder transportar suministros de emergencia a las áreas afectadas. Es una tarea de la "logística humanitaria explorar una estrategia eficiente, efectiva y justa de distribución de suministros para reducir el sufrimiento humano causado por la escasez de materiales en las áreas influenciadas" [1]. Este es usualmente el problema que se considera en el RAP.

Sin embargo, hay un segundo flujo que es el rescate de sobrevivientes que puedan estar atrapados bajo los escombros después del colapso de un edificio o estructura. Dado que los recursos son escasos, los rescatistas deben de decidir en qué sitios poner los esfuerzos de rescate para encontrar sobrevivientes.

Tabla 1. Trabajos relacionados.

Referencia	Problema	Modelo/Marco	Observaciones
Yu et al. (2021)	Asignación de recursos	Q-learning	Requerimientos iguales en cada AA
Fan et al (2022)	Asignación de recursos	DQN	Requerimientos diferentes en cada AA
Hachiya et al. (2022)	Asignación de recursos	Q-learning	Vehículos aéreos no tripulados
Mora-Ochomogo et al. (2022)	Asignación de recursos	Proceso de decisión de Markov	No hay costo de equidad

Las primeras 72 horas son cruciales para la oportunidad de rescatar gente debajo de los escombros y asignar los recursos críticos como agua y comida para los sobrevivientes. También deben de tomarse en cuenta otro tipo de recursos como pueden ser palas mecánicas para remover escombros de manera más rápida. Consideramos que este segundo problema ha sido menos estudiado en la literatura y debería de ser considerado en el RAP.

Mencionamos anteriormente que el RAP en la logística humanitaria indica que se busca distribuir los recursos maximizando la efectividad de la ayuda. En el caso de la presente investigación, se consideran distintas AA después de un desastre natural por lo que el objetivo es maximizar la efectividad de la ayuda, ya sean suministros o rescatistas, en cada una de las AA.

Sin embargo, consideramos, al igual que Fan et al. [1] que las necesidades de recursos en las AA no tienen por qué ser las mismas en cada una de ellas. El número de edificaciones afectadas después de un desastre natural puede hacer variar las necesidades de recursos.

La técnica que se pretende utilizar para atacar el RAP es aprendizaje por refuerzo. Se realizó una primera categorización del RAP, encontrando dos problemas distintos en la logística humanitaria, y ambos pueden ser abordados con aprendizaje por refuerzo.

El primer problema en nuestra categorización, identificado en la logística humanitaria, es el rescate de sobrevivientes que puedan estar atrapados bajo los escombros después del colapso de un edificio o estructura.

En este caso, consideramos que existe un RAP ya que se deben de considerar recursos que deben de ser entregados de manera eficiente y justa buscando maximizar la efectividad de la ayuda y facilitar el rescate de sobrevivientes. Estos recursos pueden incluir un número adecuado de rescatistas entrenados, perros preparados para rescate, grupos de topos, palas mecánicas para remover escombros, herramienta especializada, entre otros.

Consideramos que este primer problema puede ser abordado con aprendizaje por refuerzo. En los últimos años, la sociedad civil ha contribuido de manera importante en el rescate de sobrevivientes en edificios colapsados.

La sociedad civil se auto organiza para establecer centros de rescate no gestionados por el gobierno. Por lo tanto, existen dos vertientes que buscan lograr el mismo fin, pero la gestión es distinta. Los agentes estarán buscando la asignación de recursos dedicados a rescatar sobrevivientes.

Tabla 2. Problemas identificados en la logística humanitaria.

Problema	Subproblema	Agentes involucrados
Estrategia eficiente, efectiva y justa de distribución de suministros	1) Distribución de recursos a una AA después del colapso de un edificio	Donantes de recursos
	2) Distribución de recursos a un albergue temporal	Distribuidores y donantes de recursos
Rescate de sobrevivientes		Rescatistas, tomadores de decisiones.

La recompensa que obtendrán de manera inmediata será el rescate de personas bajo los escombros, sin embargo, hay agentes que tendrán que buscar maximizar el valor en el tiempo dependiendo de los recursos escasos dedicados a tareas de rescate.Un ejemplo de lo anterior es la asignación de una pala mecánica para remover escombro dado que es un recurso muy escaso y debería de ser asignada a una AA donde se pueda maximizar el número de sobrevivientes rescatados. El segundo problema que se categorizó habla de encontrar una estrategia eficiente y efectiva de distribución de suministros de emergencia.

Sin embargo, consideraremos que la distribución no deberá de ser justa en el sentido de que todos los CRL recibirán la misma cantidad de insumos, sino que las necesidades serán diferentes en cada uno de ellos dado que la situación de emergencia seguramente será distinta en cada AA. Por ejemplo, dependerá de si hubo un colapso de edificio en el caso de un temblor, o si hay un refugio que está recibiendo gente que no puede volver a casa. En el caso de un colapso de un edificio, usualmente la sociedad civil se organiza para recibir suministros de emergencia tanto para posibles afectados por el derrumbe, pero también se recibe comida preparada o bebidas para los rescatistas que estarán en busca de sobrevivientes debajo de los escombros.

Usualmente la literatura considera un único centro de recursos centralizado (CRC) el cual se encargaría de distribuir suministros de emergencia a las AA. Sin embargo, debemos de analizar más a fondo si atacar el RAP con este mismo supuesto o considerar un CR local (CRL) en cada una de las AA debido a la propia organización de la sociedad civil ya que los suministros pueden ser enviados de manera directa al CRL sin pasar por un CRC, usualmente manejado por el gobierno.

En este caso que estamos considerando, los agentes no recibirán instrucciones de qué suministros ni las cantidades que tendrán que llevar al CRL, ni tampoco de cómo transportar los insumos. Usualmente estos agentes llevan los insumos a los CRL cercanos a su domicilio. Recibirán recompensas inmediatas en caso de que el CRL acepte los suministros que estén llevando

Sin embargo, llegará un momento donde el CRL no pueda recibir ya ciertos insumos por lo que los agentes dejarán de recibir recompensa. Mientras tanto, otros agentes

estarán dispuestos a explorar y asistir a un CRL distinto o incluso al CRC obteniendo una recompensa tardía, pero haciendo mayor el valor en el tiempo.

En el caso de personas afectadas por un desastre natural y que no quedaron atrapadas bajo los escombros de un edificio serían trasladadas a un refugio temporal para recibir asistencia médica y alimentos. En este caso, usualmente se reciben donaciones de particulares, organizaciones o empresas y éstas se llevan a un CRC.

En el centro de recursos se clasifican y posteriormente se distribuyen a los refugios, ya sean centralizados o en las AA. Este caso es similar al anterior, sin embargo, se agregan agentes que requieren distribuir los recursos a los refugios temporales dentro del período crítico. Es posible algunos caminos para llegar del CRC a los refugios se encuentren bloqueados por escombros, o incluso pueda haber réplicas del desastre natural que compliquen la tarea de distribución.

Los agentes no recibirán instrucciones de qué caminos tomar o cómo entregar los suministros de emergencia, pero recibirán recompensan inmediatas si entregan dentro del período crítico de 72 horas, incluso si transitan por áreas peligrosas o con obstáculos en el camino. Algunos de estos agentes estarán dispuestos a explorar rutas sostenibles en el tiempo para poder distribuir los recursos de manera más eficiente a los albergues temporales, retardando así la recompensa, pero el valor pudiera ser maximizado en el tiempo.

4. Conclusiones y trabajo a futuro

En el presente trabajo revisamos el problema ya clásico de la logística humanitaria que es entregar los suministros apropiados en las cantidades adecuadas en el momento preciso y en el lugar correcto en una situación de emergencia para aliviar el sufrimiento de los supervivientes en las primeras 72 horas críticas. Para lograr ese fin, se debe de abordar el problema de asignación de recursos.

Sin embargo, pretendemos establecer una categorización del problema ya que hemos distinguido dos vertientes importantes y consideramos que una de ellas debe de ser considerada como prioritaria. Por un lado, está el rescate de sobrevivientes y que no ha sido abordado de manera extensa en la literatura. Por otro lado, está la distribución de suministros de emergencia a las áreas afectadas o a albergues temporales y que ha sido abordado de manera más extensa en la literatura.

Pretendemos aplicar técnicas de aprendizaje por refuerzo donde los agentes estarán en busca de recompensas inmediatas como lo es encontrar sobrevivientes o entregar suministros suficientes en las primeras horas críticas posteriores a un desastre natural. Dado que los recursos son escasos, es crucial la búsqueda de recompensas tardías para maximizar el valor en el tiempo. Este es un trabajo exploratorio y hemos presentado el planteamiento de la investigación que pretendemos realizar en el corto plazo.

En el presente trabajo presentamos una categorización del problema de la asignación de recursos y pretendemos atacar de manera inicial el rescate de sobreviviente. En una segunda etapa abordaremos el problema de entrega de suministros de emergencia. Como trabajo a futuro, consideramos que podemos extender la categorización con nuevas vertientes del problema conforme avance nuestra investigación.

Referencias

1. Fan, J., Chang, X., Mišić, J., Mišić, V. B., Kang, H.: DHL: Deep reinforcement learning-based approach for emergency supply distribution in humanitarian logistics. Peer-to-Peer Networking and Applications, vol. 15, no. 5, pp. 2376–2389 (2022) doi: 10.1007/s12083-022-01353-0
2. Hachiya, D., Mas, E., Koshimura, S.: A reinforcement learning model of multiple UAVs for transporting emergency relief supplies. Applied Sciences, vol. 12, no. 20, pp. 10427 (2022) doi: 10.3390/app122010427
3. Holguín-Veras, J., Jaller, M., Wassenhove, L. N. V., Pérez, N., Wachtendorf, T.: On the unique features of post-disaster humanitarian logistics. Journal of Operations Management, vol. 30, no. 7-8, pp. 494–506 (2012) doi: 10.1016/j.jom.2012.08.003
4. Holguín-Veras, J., Pérez, N., Jaller, M., Wassenhove, L. N. V., Aros-Vera, F.: On the appropriate objective function for post-disaster humanitarian logistics models. Journal of Operations Management, vol. 31, no. 5, pp. 262–280 (2013) doi: 10.1016/j.jom.2013.06.002
5. Mora-Ochomogo, I., Serrato, M., Mora-Vargas, J., Akhavan-Tabatabaei, R.: Application of a Markov decision process in collection center operations. Humanitarian Logistics from the Disaster Risk Reduction Perspective, pp. 407–428 (2022) doi: 10.1007/978-3-030-90877-5_14
6. Pérez-Rodríguez, N., Holguín-Veras, J.: Inventory-allocation distribution models for postdisaster humanitarian logistics with explicit consideration of deprivation costs. Transportation Science, vol. 50, no. 4, pp. 1261–1285 (2016) doi: 10.1287/trsc.2014.0565
7. Sheu, J.: An emergency logistics distribution approach for quick response to urgent relief demand in disasters. Transportation Research Part E: Logistics and Transportation Review, vol. 43, no. 6, pp. 687–709 (2007) doi: 10.1016/j.tre.2006.04.004
8. Sphere: Humanitarian charter and minimum standards in disaster response. Technical report, The Sphere Project, Geneva, Switzerland (2018) www.spherehandbook.org/
9. Sutton, R. S., Barto, A. G.: Reinforcement learning: An introduction (Adaptive computation and machine learning). The MIT Press, Cambridge, 2nd Edition (1998)
10. Yu, L., Zhang, C., Jiang, J., Yang, H., Shang, H.: Reinforcement learning approach for resource allocation in humanitarian logistics. Expert Systems with Applications, vol. 173, pp. 114663 (2021) doi: 10.1016/j.eswa.2021.114663
11. Zeimpekis, V., Ichoua, S., Minis, I.: Humanitarian and relief logistics. Operations Research/ Computer Science Interfaces Series (2013) doi: 10.1007/978-1-4614-7007-6

ISSN 1870-4069

Reconocimiento de acciones de empaquetado usando redes CNN-biLSTM y optimización bayesiana

Alberto Angulo Landeros[1], Luis A. Castro[1], Jessica Beltrán-Márquez[2]

[1] Instituto Tecnológico de Sonora, Ciudad Obregón, México

[2] Universidad Autónoma de Coahuila, Centro de Investigación en Matemáticas Aplicadas, México

luis.castro@acm.org, alberto.angulo242400@potros.itson.edu.mx, jessicabeltran@uadec.edu.mx

Resumen. En la actualidad, el empaquetado de productos aún depende principalmente de trabajadores manuales. Para garantizar una respuesta rápida a las demandas cambiantes de los clientes, se espera que la tendencia continúe. Por lo tanto, cuantificar el trabajo realizado es de suma importancia para la optimización de procesos. La heterogeneidad tanto del tamaño y forma de los productos a empacar, como la variabilidad de cómo los empleados empacan artículos dificultan el proceso de reconocimiento de la actividad en esta área. Para resolver este problema de reconocimiento de actividad humana (Human Activity Recognition) se han utilizado varios enfoques. Recientemente, se han utilizado métodos de aprendizaje profundo como RNN y LSTM para esta tarea. Sin embargo, estas arquitecturas no logran obtener buenos resultados cuando se trata de capturar dependencias a largo plazo en datos de series temporales, más aún cuando las actividades son secuenciales. En este trabajo, se propone una arquitectura de red convolucional con memoria a largo y corto plazo bidireccional para reconocer acciones de empaquetado en un entorno industrial. Además, se implementó una optimización bayesiana para lograr encontrar la mejor configuración de hiperparámetros. La arquitectura se evaluó utilizando el conjunto de datos Openpack logrando 93.15% de Valor F1, superando los resultados de las arquitecturas de referencia.

Palabras clave: HAR, empaquetado de productos, optimización bayesiana, redes neuronales convolucionales, redes de memoria a largo y corto plazo.

Recognition of Packet Actions Using CNN-biLSTM Networks and Bayesian Optimizationet

Abstract. Currently, product packaging processes still depend mainly on manual workers. To ensure a quick response to changing customer demands, the trend is expected to continue. Therefore, quantifying the work done is paramount to

optimizing processes. The heterogeneity of the size and shape of the packed products and the variability of how employees pack items make it difficult to recognize the activity in this area. Various approaches have been used to solve this Human Activity Recognition (HAR) problem. Recently, deep learning methods such as RNN and LSTM have been used for this task. However, these architectures fail to perform well when it comes to capturing long-term dependencies on time series data, even more so when activities are sequential. This work proposes a convolutional network architecture with bidirectional long short-term memory to recognize packaging actions in an industrial environment. In addition, a Bayesian optimization was implemented to find the best hyperparameter configuration. The architecture was evaluated using the Openpack data set, achieving 93.15% F1-Value, surpassing the results by the reference architectures.

Keywords: HAR, product packaging, Bayesian optimization, convolutional neural networks, long short-term memory.

1. Introducción

El reconocimiento de actividades humanas (Human Activity Recognition, HAR) es un campo enfocado en identificar las acciones realizadas por una persona [1]. Debido al progreso en tecnologías de sensores y computación ubicua, el HAR ha expandido su aplicación en diversas áreas. Actualmente, el HAR se utiliza ampliamente en sectores como la atención médica [2], monitoreo de empleados [3], interfaces hombre-máquina [4, 5], entrenamiento deportivo [6], vigilancia [7], entre otros.

El avance acelerado en las tecnologías de sensores e informática ubicua ha impulsado la creciente popularidad del uso de datos de sensores en el reconocimiento de actividades humanas. En el entorno industrial, HAR tiene varias aplicaciones, entre estas se encuentra la identificación de la manera en que los empleados realizan las actividades, lo que puede servir para la mejora y optimización de procesos.

Por ejemplo, el personal que labora en centros logísticos en el área de empaquetado realiza una serie de actividades manuales secuenciales para empacar artículos. Algunos de los aspectos que pueden ser de interés es la identificación de posturas inadecuadas de los empleados del área de empaquetado, que puede llevar a lesiones y afectaciones consecuentes en la productividad.

Mediante HAR, es posible reconocer acciones y la forma en que se realizan, para identificar a los empleados que no realizan adecuadamente el empaquetado, señalar los errores que se cometen y generar recomendaciones para mejorar las posturas. Asimismo, mediante HAR se pueden identificar otras acciones relacionadas con la eficiencia de la producción, como el correcto seguimiento de los protocolos de empaquetado, las anomalías en el empaquetado, las técnicas de empaquetado más eficientes, entre otras.

Recientemente se han hecho investigaciones tratando de abordar este problema. Por ejemplo, en [8] se propuso un enfoque para reconocer cuatro actividades diferentes utilizando datos obtenidos de un acelerómetro triaxial y un giroscopio. Las actividades fueron martillar, atornillar con un destornillador, usar una llave inglesa y fijar tornillos con un taladro eléctrico. Se utilizó el método del vecino más cercano (kNN) para clasificar los datos.

Tabla 1. Detalles de conjunto de datos Openpack para reconocimiento de actividades humanas.

Elementos	Detalles
Tipo	Reconocimiento de trabajo de empaquetado
Participantes	16
Tasa de muestreo	IMU (Acc, Gyro, Ori): 30Hz
	Empática E4 (BVP, EDA): 64HZ y 4Hz
	Sensores Empática E4 (ACC): 32Hz
Actividades (clases)	10 principales + 32 secundarias
No. de datos obtenidos	20,129 operaciones de trabajo
	52,529 acciones
Modalidad	D+Keypoints+LiDAR+Acc+Gyro+Ori+EDA+BVP+Temp
Duración de la grabación	53h50m

D: Depth, Acc: acelerómetro, Gyro: giroscopio, Ori: sensor de orientación, EDA: actividad electrodérmica, BVP: pulso de volumen sanguíneo, Temp: temperatura. Keypoints: puntos cla del sensor Kinect.

En otro trabajo se implementó una arquitectura de redes convolucionales para identificar acciones realizadas en un contexto de logística [9]; Los autores evaluaron el modelo utilizando un conjunto de datos propio llamado LARa. Para ello, aplicaron una técnica de ventana deslizante junto con redes neuronales convolucionales combinada con capas lineales.

En la última capa de la arquitectura, se implementaron dos funciones de activación: softmax y sigmoid. Finalmente, se utilizó la función de pérdida de entropía cruzada, fusionando ambas salidas para optimizar el rendimiento del modelo obteniendo un Valor-F1 de 64.43%.

Algunos de los retos existentes en el área de HAR son el reconocimiento de actividades a partir de datos multimodales, el reconocimiento adecuado de actividades formadas por acciones secuenciales, y el reconocimiento de actividades en diferentes granularidades.

En el caso específico de problemas orientados a empaquetado de artículos, un reto importante se debe a la complejidad de las actividades realizadas, a la heterogeneidad del tamaño y forma de los artículos a empaquetar, así como la variabilidad de la manera en que los empleados empaquetan artículos.

Asimismo, otro reto se debe a la similitud en la manera en que se realizan actividades distintas. Por ejemplo, cuando las personas cierran una caja es similar a cuando se agrega una etiqueta a la caja. Este trabajo se centra en la utilización de técnicas de HAR orientadas a reconocer actividades de empaquetado en un entorno industrial utilizando el conjunto de datos Openpack [10].

Este conjunto de datos incluye datos multimodales como datos de sensores IMU (Acc, Gyro, Ori), puntos clave de sensor Kinect, visión de profundidad y provee una granularidad de categorías de actividades mayor en comparación con otros conjuntos de datos disponibles como InHARD [11] y LARa [9]. Para la clasificación de las

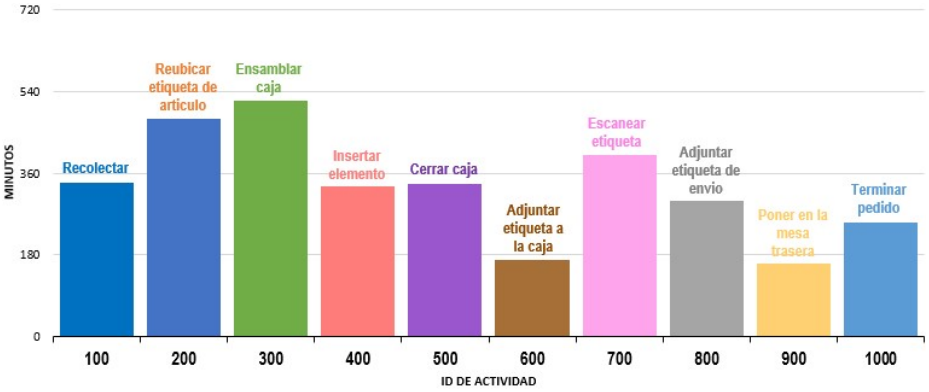


Fig. 1. Distribución de duración total de grabación de cada actividad. Imagen basada en [10].

actividades, se propone una variante de red convolucional profunda bidireccional CNN-biLSTM que se aplica al escenario de empaquetado de productos.

Además, se utiliza Optimización Bayesiana para encontrar la mejor arquitectura e hiperparámetros de la red propuesta. Finalmente, se evaluó la red propuesta utilizando diferentes combinaciones de sensores y conjuntos de usuarios.

2. Trabajo relacionado

El reconocimiento de la actividad humana es un problema basado en series temporales. La evaluación y el análisis de estas señales para el reconocimiento de actividades humanas es de especial interés para realizar optimizaciones en la industria donde el trabajo manual sigue siendo dominante.

Se han desarrollado métodos para clasificar los movimientos humanos. Un ejemplo de aporte en el área de HAR es un trabajo orientado a reconocer las actividades y movimientos humanos en la preparación de pedidos [12]. En [12], los autores utilizaron datos de tres unidades de medición inercial (IMU) que usaron los trabajadores mientras hacían actividades en dos escenarios de preparación de pedidos comparables (A y B).

Ambos escenarios se operaron de manera manual, las mercancías se almacenan en estantes, el trabajador se encargaba de recolectar los pedidos por el almacén y cada artículo se almacenó en una caja dedicada. En el escenario A, los pedidos se proporcionaron en papel y en B se utilizaron dispositivos portátiles con conexión a WiFi.

Se utilizaron características estadísticas en el dominio del tiempo en segmentos utilizando el enfoque de ventana deslizante. Se evaluaron tres clasificadores: una máquina de vectores de soporte, un clasificador Naive Bayes y un bosque aleatorio utilizando validación de 3 k-fold, los resultados fueron 67.3, 67.7 y 72.6 respectivamente, mostrando que el bosque aleatorio muestra el rendimiento más estable.

En otro trabajo, se propuso una arquitectura de red neuronal profunda utilizando datos secuenciales de múltiples unidades de medida inercial (IMU) [13]. Se evaluaron los datos de acelerómetro, giroscopio y magnetómetro. Los autores reportaron una

Tabla 2. Detalles de escenarios en conjunto de datos Openpack.

ID	Descripción
ES01	Los participantes siguieron las instrucciones lo más fielmente posible. La lista de artículos en un pedido se basó en hojas de pedidos reales, pero se limitó la variedad de artículos en un pedido a 54.
ES02	Los participantes tuvieron libertad para alterar el procedimiento de las operaciones según su criterio. También se redujeron las probabilidades de incluir artículos muy grandes o pequeños en un pedido en comparación con la ES01, y se agregaron 21 artículos nuevos.
ES03	Se introdujeron situaciones/acciones irregulares al ES02, como cajas de envío ya ensambladas que podían ser utilizadas por los trabajadores, la inclusión de artículos pequeños en bolsas de papel, y la posibilidad de que un sujeto llevara varios pedidos consecutivos de artículos pequeños de la mesa trasera al banco de trabajo al mismo tiempo.
ES04	Se implementó una alarma auditiva en el ES03 para simular un tiempo de trabajo ocupado y se establecieron alarmas periódicas (con un intervalo de 30-45 segundos) cuando el tiempo transcurrido de un periodo excedía el 80% de la duración promedio de un periodo de trabajo previamente registrado.

mejoría de hasta 2% de exactitud en la clasificación comparado con enfoques tradicionales (p.ej., bosque aleatorio), así como otras arquitecturas de redes neuronales.

Además, entre los métodos HAR que han sido propuestos, se encuentran los que son basados en redes convolucionales (Convolutional Neural Network, CNN) en donde la extracción de características es parte de la arquitectura de la red neuronal [14, 15]. Se propone un enfoque basado en CNN para la clasificación de actividades humanas utilizando datos provenientes de diferentes tipos de sensores colocados en el cuerpo de las personas.

El enfoque propuesto se probó en tres diferentes conjuntos de datos: Opportunity [16, 17], PAMAP2 [18] y Order Picking [19]. Los tres conjuntos de datos presentan un tipo y cantidad distinta de actividades y desbalance de clases. Los autores evaluaron la red propuesta utilizando series de tiempo multicanal adquiridas de sensores corporales IMU. Se logró un mejor resultado utilizando la red CNN-IMU propuesta comparado a una red CNN base.

Además, se investigó el efecto de utilizar operaciones de agrupación máxima, ya que esta operación podría no conservar la información como se sugiere en [20], para secuencias relativamente largas, las CNN que contienen operaciones de agrupación máxima muestran mejor resultados.

En [10] se propuso un conjunto de datos multimodal sobre el reconocimiento de actividades laborales en un entorno industrial. Además, se propuso un nuevo modelo de reconocimiento LTS-Net para la clasificación de series temporales que utiliza lecturas provenientes de dispositivos de internet de las cosas (IoT). Para las CNN es difícil extraer la dependencia a largo plazo dentro de una serie temporal, lo que dificulta mejorar el rendimiento del modelo.

Una propuesta de solución son las llamadas redes de memoria a largo y corto plazo (LSTM) [21], las cuales se han empleado en HAR debido a sus ventajas para extraer dependencias a largo plazo dentro de series temporales. En [22] se propuso una red

Tabla 3. Detalles de participantes y sesiones en conjunto de datos OpenPack.

Participantes					Sesiones				
ID	Sexo	Edad	Mano Dominante	Experiencia	S0100	S0200	S0300	S0400	S0500
U0101	F	-	D	-	ES01	ES01	ES01	ES01	ES01
U0102	F	-	D	-	ES01	ES01	ES01	ES01	ES01
U0103	F	50	D	6 meses	ES01	ES01	ES01	ES01	ES01
U0105	F	30	D	4 años	ES01	ES01	ES01	ES01	ES01
U0106	F	40	D	1 mes	ES01	ES01	ES01	ES01	ES01
U0107	F	40	D	3 años	ES01	ES01	ES01	ES01	ES01
U0109	M	30	I	6 meses	ES01	ES01	ES01	ES01	ES01
U0110	F	40	D	10 meses	ES01	ES01	ES01	ES01	ES01
U0111	F	50	D	2 años	ES01	ES01	ES01	ES01	ES01
U0205	F	30	D	4 años	ES02	ES02	ES03	ES03	ES04
U0202	F	40	D	3 años	ES02	ES02	ES03	ES03	ES04
U0210	F	50	D	3 meses	ES02	ES02	ES03	ES03	ES04

neuronal profunda que combina capas convolucionales con memoria a corto plazo para el reconocimiento de actividad humana, el enfoque propuesto es capaz de aprender la dinámica temporal en varias escalas de tiempo aumentando la exactitud obtenida.

El uso de capas convolucionales con memoria a corto y largo plazo ha ayudado a aumentar la precisión en diferentes conjuntos de datos HAR. Sin embargo, las investigaciones actuales se han enfocado en reconocer actividades de uso cotidiano como caminar, subir escaleras, bajar escaleras, o andar en bicicleta. Existe muy poca literatura relacionada a actividades en el área del empaquetado de productos en el área industrial [9, 10, 12, 13, 19].

3. Métodos

En este trabajo, se propone una red CNN-biLSTM que nos permite la extracción de la dependencia de tiempo tanto hacia atrás como hacia adelante y al tratarse de actividades secuenciales nos permite predecir la actividad de interés no solo de la actividad anterior, sino también de la siguiente actividad.

3.1. Conjunto de datos

Se seleccionó el conjunto de datos Openpack [10] para este estudio debido a que es considerado el conjunto de datos más completo en el reconocimiento de actividad humana en la industria, específicamente en el área del empaquetado de productos. Este conjunto de datos contiene una gran cantidad de datos de sensores y se compone de registros de 16 participantes realizando actividades de empaquetado en entornos industriales.

La tabla 1 proporciona más información detallada sobre los elementos que forman parte y describen el conjunto de datos Openpack. La construcción del conjunto de datos Openpack siguió un documento de instrucciones utilizado en un centro de logística real, el cual especifica la secuencia de acciones que debe realizar el trabajador durante el empaquetado de productos.

Tabla 4. Configuraciones utilizadas para evaluar arquitectura CNN-biLSTM para Entrenamiento (E), Validacion (V) y Pruebas (P).

	Subconjunto 1		**Subconjunto 2**	
	Participante	Sesión	Participante	Sesión
E	U0102	S0100, S0200, S0300	U0101, U0102, U0103, U0105, U0106, U0107, U0109, U0110	S0100, S0200, S0400, S0500
V	U0102	S0400	U0101, U0103, U0105, U0107, U0109, U0111, U0205	S0300
P	U0102	S0500	U0102, U0106, U0202, U0210	S0300

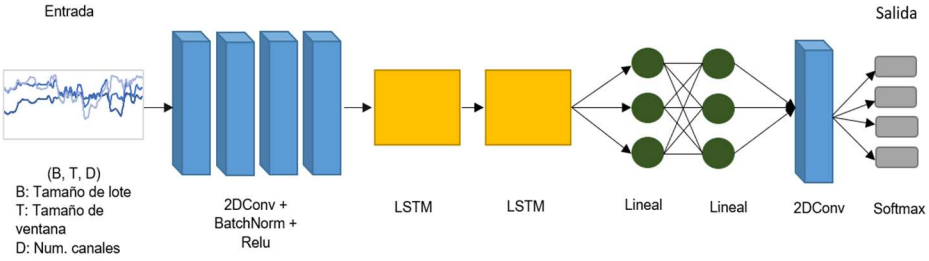


Fig. 2. Arquitectura del modelo para CNN-biLSTM.

Los creadores de Openpack utilizaron estas acciones para etiquetar el conjunto de datos. En la Fig. 1 se muestran las clases de operaciones de empaquetado, junto con la cantidad de minutos capturados correspondientes a cada una de ellas.

Como se observa en la Fig. 1, las clases están desbalanceadas. La actividad de reubicar la etiqueta del artículo y ensamblar caja son las actividades con mayor número de muestras. Las muestras son calculadas multiplicando los segundos de grabación y la tasa de muestreo de cada sensor.

Los participantes realizaron las acciones siguiendo 4 escenarios preestablecidos. En la Tabla 2 se muestran los detalles de escenarios utilizados para la obtención del conjunto de datos. En la Tabla 3 se muestra información sobre los participantes, las sesiones y los escenarios que realizaron.

La propuesta en este estudio se enfoca en utilizar los datos de los sensores Atr (acc, gyro, quaternion) y E4 (acc, BVP, EDA, temperatura) con los cuales se realizaron diferentes experimentos. Los sensores Atr se ubicaron en ambas muñecas y ambos brazos, los sensores E4 se ubicaron en ambas muñecas.

3.2. Preparación de datos

Se aplicaron diferentes técnicas de preprocesamiento sobre los datos. Para reducir el ruido, se aplicó el filtro Kalman utilizando la librería Pykalman[1] aplicando una covarianza de observación de 0.1 y covarianza de transición de 0.01.

[1] Pykalman, pykalman.github.io/, último acceso 17/04/2023

Tabla 5. Lista de hiperparámetros seleccionados.

Escenario	Hiperparámetros	Valores seleccionados
Procesamiento de datos	Tamaño de ventana	1800
Entrenamiento	Optimizador	Adam
	Tamaño de lote (Batch Size)	12
	Tasa de Aprendizaje (Learning Rate)	0.0006
	Caída de peso (Weight Decay)	0.0005
	Épocas	50

Tabla 6. Combinaciones de sensores utilizados para evaluación.

Combinación	C1	C2	C3	C4	C5	C6
Sensor	Acc, E4Acc	Gyro	Acc, E4Acc, Gyro	Acc, E4Acc, Gyro, Ori	Acc, E4Acc, Gyro, Ori, Bvp	Acc, E4Acc, Gyro, Ori, Bvp, Eda

Debido a que los datos fueron capturados en condiciones realistas y los sensores que se usan en los sujetos son inalámbricos, algunos datos fueron perdidos durante el proceso de recopilación y la sincronización. Para evitar esto, los datos se preprocesaron mediante la técnica de interpolación.

Otro problema que se detectó en los datos es un desfase en los tiempos de los datos obtenidos. Uno de los motivos se debe a que los sensores capturan datos usando diferentes frecuencias de muestreo. Otro motivo es que al momento de la captura algunos sensores empezaron a grabar antes y terminaron antes que los otros sensores en algunas sesiones.

Por ejemplo, en la sesión S0100 del usuario U0102, los datos del sensor Atr están adelantados un segundo, y en la sesión S0100, el sensor E4 está adelantado un segundo. Para abordar este problema, se realizó un preprocesamiento que consistió en remuestrear los datos a 25Hz y sincronizar los datos utilizando las épocas de Unix.

3.3. Separación de datos

Debido a la complejidad de los datos, se trabajó con un subconjunto de datos (denominado “Subconjunto 1”) para encontrar los mejores hiperpárametros, ya que el costo computacional de entrenamiento es elevado y además requiere una cantidad considerable de tiempo.

Posteriormente, se utilizó un conjunto de datos más amplio (denominado “Subconjunto 2”) para entrenar y evaluar la mejor configuración de hiperparámetros. En la Tabla 4 se detallan los participantes y sesiones que formaron parte de cada experimento. Es importante mencionar que el conjunto de datos Openpack incluye datos sin etiquetar o con numerosos valores faltantes, los cuales fueron excluidos.

3.4. Arquitectura propuesta

La estructura de la red propuesta en este trabajo (CNN-biLSTM) es una variación de la arquitectura descrita en [23], y combina capas convolucionales, lineales y

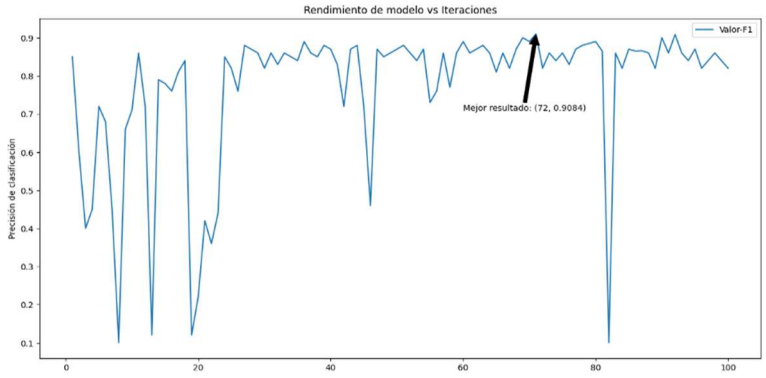


Fig. 3. Rendimiento del modelo vs iteraciones para obtener hiperparámetros de modelo propuesto utilizando el Valor-F1 con datos de configuración de Subconjunto 1 y sensor Gyro.

recurrentes. Las capas convolucionales se encargan de extraer características espaciales y proporcionar representaciones abstractas de los datos de entrada en mapas de características, mientras que las capas recurrentes aprenden las dependencias a largo plazo tanto hacia adelante como hacia atrás. La arquitectura propuesta consta de nueve capas (Fig. 2).

Como se observa en la Fig. 2, los datos preprocesados ingresan a la red CNN-BiLSTM propuesta, en donde el primer paso consiste en 4 capas que cuentan con 99 filtros encargados de extraer características espaciales. Entre cada capa, se encuentra una normalización por lotes estándar que actúa como regularizador, así como una función de activación ReLU.

Le siguen dos capas LSTM bidireccionales con 234 neuronas, que se encargan de obtener características temporales. Para evitar el sobreajuste se utilizan capas de abandono con 0.33 y 0.16 respectivamente. La segunda capa LSTM usa como entrada la salida de la capa anterior. Luego se agregan dos capas lineales totalmente conectadas con 330 y 223 neuronas, respectivamente.

Por último, la salida del modelo está dada por una capa de salida (una capa Conv con una función de activación softmax). Las variaciones implementadas a la arquitectura propuesta en comparación con la de referencia incluyen el uso de LSTM bidireccionales, lo cual permite predecir la actividad de interés tanto a partir de la actividad anterior cómo a partir de la actividad siguiente.

Además, se agregaron dos capas lineales, las cuales mejoran la capacidad del modelo para aprender representaciones más complejas de los datos. De esta manera, se logra mejorar la capacidad de generalización del modelo y evitar el sobreajuste.

3.5. Entrenamiento

La arquitectura propuesta se implementó en Pytorch y se utilizó la función de pérdida entropía cruzada para la optimización de la red, tomando en cuenta las 10 clases de actividad descritas en la Fig. 1.

La entrada de la red consiste en una secuencia de datos formada por series de tiempo extraídas desde los datos de los sensores Atr (Acc, Gyro, Ori) y E4 (Acc, BVP, EDA)

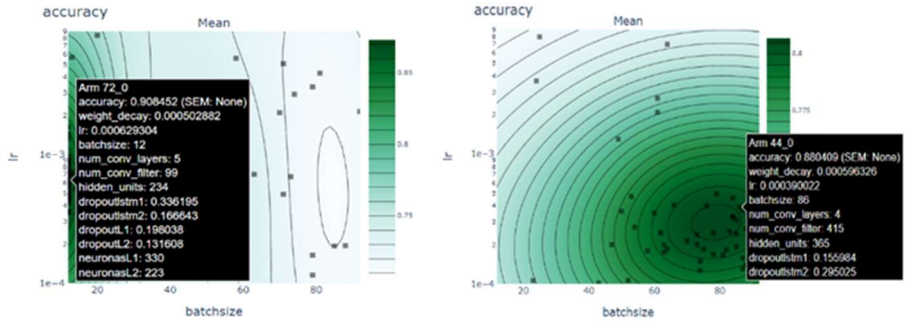


Fig. 4. Mejores hiper parámetros obtenidos utilizando optimización bayesiana para CNN-biLSTM (izquierda); Mejores hiperparámetros obtenidos utilizando optimización bayesiana para DeepConvLSTM (derecha).

utilizando un enfoque de ventana deslizante compuesto por varios canales de sensores. Para demostrar la eficiencia del modelo, tanto en el entrenamiento como en la prueba, los datos se segmentan en tamaños de lotes de 12 datos por segmento.

Los datos recibidos por la primera capa convolucional son de la forma (B, CH, T, 1) donde B es el tamaño del lote, CH es el número de canales de entrada, T es el tamaño de la ventana que en este caso fue de 30 muestras por segundo, dando un tamaño de 1800.

Después de ser procesados por las cuatro capas convolucionales, se eliminaron las dimensiones con entrada uno utilizando la función Squeeze[2] y se utilizó la forma resultante (B, T, CH) en las capas LSTM y lineales. Antes de la capa de salida, se regresó a la forma original (B, CH, T, 1) utilizando la función Unsqueeze[3], el modelo propuesto regresa la forma (B, N_CLASES, T, 1).

3.6. Optimización de hiperparámetros

Uno de los principales problemas al momento de entrenar un modelo de reconocimiento de actividades es seleccionar la mejor configuración de hiperparámetros. Esto se ha abordado utilizando diferentes técnicas como búsqueda en cuadrícula (GS), búsqueda aleatoria (RS) y la optimización de enjambre de partículas (PSO). Además, existe la forma manual que consta de realizar cambios manuales en los hiperparámetros y realizar pruebas.

Para reducir el tiempo que lleva encontrar la mejor configuración, se implementó la Optimización Bayesiana. Este método ha sido utilizado para la optimización de hiperparámetros en subconjuntos de big data [24]. La optimización se llevó a cabo utilizando las herramientas BoTorch [25] y Ax [26], debido a la facilidad que nos proporciona para realizar experimentos y la fácil integración que cuenta con Pytorch.

[2] Función Squeeze, https://pytorch.org/docs/stable/generated/torch.squeeze.html, último acceso 17/04/2023

[3] Función Unsqueeze, https://pytorch.org/docs/stable/generated/torch.unsqueeze.html, último acceso 17/04/2023

Tabla 7. Valor-F1 utilizando diferentes combinaciones de sensores utilizando datos del Subconjunto 1 y los sensores de ambas manos.

Combinación	C1	C2	C3	C4	C5	C6
# de canales	18	12	30	46	48	50
CNN-biLSTM (Sub 1)	**88.45%**	**89.86%**	**90.65%**	**89.32%**	**88.40%**	**89.41%**
DeepConvLstm	78.53%	84.70%	86.05%	81.53%	83.51%	85.23%

3.7. Evaluación

Para evaluar el desempeño de la arquitectura durante el entrenamiento, se utilizó la medida de exactitud de clasificación general multiclase de la biblioteca torchmetrics1, utilizando el optimizador Adam [27].

Al tratarse de un conjunto de datos desbalanceado, si el clasificador predice cada instancia como una clase mayoritaria y se utiliza la exactitud de clasificación general para evaluar el resultado, los resultados podrían lograr un alto rendimiento. Por lo tanto, la exactitud de clasificación general no es una medida apropiada para evaluar el modelo.

Por otro lado, el Valor-F1 (F1 score) toma en cuenta tanto los falsos positivos como los falsos negativos y muestra el equilibrio entre la precisión y recuperación. La precisión puede verse como $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$ y la recuperación como $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$ donde TP y FP son el número de verdaderos y falsos positivos. FP corresponde al número de falsos positivos. La fórmula del Valor-F1 está dada por:

$$\text{Valor-F1}= \sum_{i}^{\mathrm{N}} 2\times \mathrm{w}_i \frac{\text{precisión}_i .\ \text{recuperación}_i}{\text{precisión}_i + \text{recuperación}_i}, \tag{1}$$

donde $w_i = n_i/N$ es la proporción de muestra de la clase i, siendo n_i el número de muestras de la clase i-ésima y N el número total de muestras.

4. Resultados y discusión

Para evaluar el desempeño de la arquitectura propuesta se hizo una comparación contra DeepConvLSTM utilizando las configuraciones descritas por los autores en [23]. En la Tabla 5 se puede observar la lista de hiperparámetros. Se probó la arquitectura con diferentes combinaciones de sensores. En la Tabla 6 se pueden observar las combinaciones evaluadas. Para la evaluación se realizó un muestreo de los datos a 25 Hz.

Para obtener la mejor configuración de la arquitectura propuesta se utilizó optimización bayesiana, véase en la Fig. 3. Esta optimización se aplicó utilizando los datos de la configuración del Subconjunto 1, se realizaron 100 interacciones, los resultados se obtuvieron utilizando el Valor-F1.

Los mejores resultados se encontraron en la iteración 72 con un 90.84% de precisión. Los hiperparámetros obtenidos se pueden observar en la Fig. 4 (izquierda). Además, también se aplicó una optimización bayesiana a la arquitectura de referencia

Tabla 8. Valor-F1 utilizando datos de subconjunto 1 (Sub1) y subconjunto 2 (Sub2), así como los sensores de ambas manos aplicando filtro Kalman.

Combinación	C1	C2	C3	C4	C5	C6
# de canales	18	12	30	46	48	50
CNN-biLSTM (Sub 1)	89.60%	**90.71%**	90.67%	89.86%	87.88%	88.47%
CNN-biLSTM (Sub 2)	92.83%	**93.15%**	93.09%	92.90%	92.68%	93.12%

DeepConvLSTM, lo cual permitió identificar los hiperparámetros óptimos, que pueden observarse en la Fig. 4 (derecha).

Como se puede ver en la Tabla 7 la red propuesta CNN-biLSTM logra mejores resultados en todas las combinaciones evaluadas. Igualmente se puede notar que el mejor resultado obtenido es utilizando la combinación de sensores C3 (Acc, E4Acc, Gyro). Para reducir el ruido y las fluctuaciones registradas en las mediciones de los datos, se aplicó el filtro Kalman descrito en la sección 3.2.

Los resultados obtenidos por la arquitectura propuesta utilizando los datos del Subconjunto 1 se pueden observar en la Tabla 8. Se puede observar que el aplicar el filtro Kalman mejoró los resultados en las configuraciones donde se utilizan los datos del sensor Atr, sin embargo, se observó una disminución en el rendimiento al utilizar los datos del sensor E4, debido a que los datos de E4 tienen un rango de lectura más extenso y no estaban normalizados.

Los resultados obtenidos con la arquitectura propuesta utilizando los datos del Subconjunto 2 se pueden observar en la Tabla 8. Para el entrenamiento se utilizaron los hiperparámetros obtenidos en el Subconjunto 1. En comparación con el Subconjunto 1, el mejor resultado se obtuvo utilizando la combinación C2.

5. Conclusión y trabajo a futuro

En este artículo se propuso una nueva arquitectura de red convolucional profunda que combina capas convolucionales con LSTM para el reconocimiento de actividad humana en entornos industriales. Para probar la red se utilizó el conjunto de datos Openpack y se comparó contra una arquitectura de redes neuronales base. Se utilizó el Valor-F1 para comparar el desempeño. Finalmente, se logró un Valor-F1 de 93.12% con la configuración de Subconjunto 2.

Se realizaron diferentes experimentos para encontrar la mejor combinación de sensores. Además, también se exploró cómo el cambio de los hiperparámetros afecta el rendimiento del modelo, se aplicó la optimización bayesiana para obtener los mejores hiperparámetros. En comparación con los métodos propuestos en otros trabajos, CNN-biLSTM demostró un rendimiento superior cuando se trata de reconocer actividades en el área de la logística.

Si bien los resultados fueron buenos, solamente se realizaron experimentos utilizando los sensores IMU. En trabajos futuros se pretende adaptar la arquitectura propuesta para utilizar una combinación de los datos del sensor Kinect, visión profunda y la configuración actual, además de aplicar la normalización en los datos.

Agradecimientos. Este trabajo fue parcialmente financiado por Consejo Nacional de Ciencia y Tecnología (CONACYT) en México, con una beca al primer autor (1146285), así como por el Instituto Tecnológico de Sonora (ITSON) a través del programa PROFAPI.

Referencias

1. Ann, O. C., Theng, L. B.: Human activity recognition: A review. In: IEEE International Conference on Control System, Computing and Engineering, pp. 389-393 (2014) doi: 10.1109/iccsce.2014.7072750
2. Dinarevic, E. C., Husic, J. B., Barakovic, S.: Issues of human activity recognition in healthcare. In: 18th International Symposium INFOTEH-JAHORINA, pp. 1–6 (2019) doi: 10.1109/infoteh.2019.8717749
3. Malaise, A., Maurice, P., Colas, F., Ivaldi, S.: Activity recognition for ergonomics assessment of industrial tasks with automatic feature selection. IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1132–1139 (2019) doi: 10.1109/lra.2019.2894389
4. Gkournelos, C., Karagiannis, P., Kousi, N., Michalos, G., Koukas, S., Makris, S.: Application of wearable devices for supporting operators in human-robot cooperative assembly tasks. Procedia CIRP, vol. 76, pp. 177–182 (2018) doi: 10.1016/j.procir.2018.01.019
5. Ignatov, A.: Real-time human activity recognition from accelerometer data using convolutional neural networks. Applied Soft Computing, vol. 62, pp. 915–922 (2018) doi: 10.1016/j.asoc.2017.09.027
6. Cust, E. E., Sweeting, A. J., Ball, K., Robertson, S.: Machine and deep learning for sport-specific movement recognition: A systematic review of model development and performance. Journal of Sports Sciences, vol. 37, no. 5, pp. 568–600 (2018) doi: 10.1080/02640414.2018.1521769
7. Chen, L., Wei, H., Ferryman, J.: A survey of human motion analysis using depth imagery. Pattern Recognition Letters, vol. 34, no. 15, pp. 1995–2006 (2013) doi: 10.1016/j.patrec.2013.02.006.
8. Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P., Roning, J.: Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In: 2009 17th Mediterranean Conference on Control and Automation, pp. 401–405 (2009) doi: 10.1109/med.2009.5164574
9. Niemann, F., Reining, C., Rueda, F. M., Nair, N. R., Steffens, J. A., Fink, G. A., Hompel, M. T.: LARa: Creating a dataset for human activity recognition in logistics using semantic attributes. Sensors, vol. 20, no. 15 (2020) doi: 10.3390/s20154083
10. Yoshimura, N., Morales, J., Maekawa, T., Hara, T.: OpenPack: A large-scale dataset for recognizing packaging works in IoT-enabled logistic environments (2022) doi: 10.48550/ARXIV.2212.11152
11. Dallel, M., Havard, V., Baudry, D., Savatier, X.: InHARD-Industrial human action recognition dataset in the context of industrial collaborative robotics. In: 2020 IEEE International Conference on Human-Machine Systems, pp. 1–6 (2020) doi: 10.1109/ICHMS49158.2020.9209531
12. Feldhorst, S., Masoudenijad, M., Ten-Hompel, M., Fink, G., A.: Motion classification for analyzing the order picking process using mobile sensors – General concepts, case studies and empirical evaluation. In: Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods, vol. 1, pp. 706–713 (2016) doi: 10.5220/0005828407060713
13. Grzeszick, R., Lenk, J. M., Rueda, F. M., Fink, G. A., Feldhorst, S., Ten-Hompel, M.: Deep neural network based human activity recognition for the order picking process. In:

Proceedings of the 4th International Workshop on Sensor-based Activity Recognition and Interaction, no. 14, pp. 1–6 (2017) doi: 10.1145/3134230.3134231
14. Rahman, M. A., Mia, Y., Rahman-Masum, R., Hasan-Abid D. M., Islam, T.: Real time human activity recognition from accelerometer data using convolutional neural networks. In: 2022 7th International Conference on Communication and Electronics Systems, pp. 1394–1397 (2022) doi: 10.1109/ICCES54183.2022.9835797
15. Panwar, M., Ram Dyuthi, S., Chandra Prakash, K., Biswas, D., Acharyya, A., Maharatna, K., Gautam, A., Naik, G. R.: CNN based approach for activity recognition using a wrist-worn accelerometer. In: 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2438–2441 (2017) doi: 10.1109/EMBC.2017.8037349
16. Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Forster, K., Troster, G., Lukowicz, P., Bannach, D., Pirkl, G., Ferscha, A., Doppler, J., Holzmann, C., Kurz, M., Holl, G., Chavarriaga, R., Sagha, H., Bayati, H., Creatura, M., Millan, J. R.: Collecting complex activity datasets in highly rich networked sensor environments. In: Seventh International Conference on Networked Sensing Systems, pp. 233–240 (2010) doi: 10.1109/INSS.2010.5573462
17. Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S. T., Tröster, G., Millán, J. del R., Roggen, D.: The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters, vol. 34, no. 15, pp. 2033–2042 (2013) doi: 10.1016/j.patrec.2012.12.014
18. Reiss, A., Stricker, D.: Introducing a new benchmarked dataset for activity monitoring. In: 16th International Symposium on Wearable Computers, pp. 108–109 (2012) doi: 10.1109/ISWC.2012.13
19. Moya-Rueda, F., Grzeszick, R., Fink, G., Feldhorst, S., Hompel, M.: Convolutional neural networks for human activity recognition using body-worn sensors. Informatics, vol. 5, no. 2, pp. 1–17 (2018) doi: 10.3390/informatics5020026
20. Rippel, O., Snoek, J., Adams, R., P.: Spectral representations for convolutional neural networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 2 (2015) doi: 10.48550/arXiv.1506.03767
21. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Computation, vol. 31, no. 7, pp. 1235–1270 (2019) doi: 10.1162/neco_a_01199
22. Xia, K., Huang, J., Wang, H.: LSTM-CNN architecture for human activity recognition. IEEE Access, vol. 8, pp. 56855–56866 (2020) doi: 10.1109/ACCESS.2020.2982225
23. Ordóñez, F. J., Roggen, D.: Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors, vol. 16, no. 1, pp. 2–25 (2016) doi: 10.3390/s16010115
24. Klein, A., Falkner, S., Bartels, S., Hennig, P., Hutter, F.: Fast bayesian optimization of machine learning hyperparameters on large datasets. Artificial intelligence and statistics, pp. 528–536 (2017) doi: 10.48550/arXiv.1605.07079
25. Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., Bakshy, E.: BOTORCH: A framework for efficient monte-carlo bayesian optimization (2020) doi: 10.48550/arXiv.1910.06403
26. Baird, S. G., Liu, M., Sparks, T. D.: High-dimensional bayesian optimization of hyperparameters for an attention-based network to predict materials property: a case study on CrabNet using Ax and SAASBO. Computational Materials Science, vol. 211 (2022) doi: 10.1016/j.commatsci.2022.111505

ISSN 1870-4069

Pseudoetiquetado para el análisis de polaridad en tuits: Un primer acercamiento

Diana Jimenez, Marco A. Cardoso-Moreno,
Cesar Macias, Hiram Calvo

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Ciencias Cognitivas Computacionales,
México

{djimenezl2022, mcardosom2021,
cmaciass2021, hcalvo}@cic.ipn.mx

Resumen. El análisis de polaridad en textos es un tema de creciente interés, sobre todo en redes sociales, ya que ayuda a conocer si la opinión de las personas es negativa o positiva con respecto a un tema en particular en un ambiente libre, lo que nos permite conocer el impacto que productos y noticias, por mencionar algunos ejemplos, tienen en la sociedad. En este trabajo se realiza análisis de polaridad en tuits en español mexicano, mediante el uso de pseudoetiquetas generadas automáticamente, con el fin de intentar combatir la escasez de datos etiquetados, reto inherente a la tarea de análisis de polaridad dada la vasta cantidad de información disponible y lo complicado de asignar etiquetas adecuadas a la misma; además de técnicas de clasificación para tareas de procesamiento de lenguaje natural.

Palabras clave: Pseudoetiquetado, polaridad, tuits, PLN, aprendizaje automático.

Twits Pseudolabeling for Polarity Analysis: A First Approach

Abstract. Polarity analysis in texts is experiencing an interest growth, specially on social networks, due to its assitance in understanding if the opinion people have about certain topic, such as products or news, to name a few, is positive or negative, as well as the impact they have in societies. In this work we perform a polarity analysis on twits written in mexican spanish, by creating pseudolabels via machine learning tech- niques to try to overcome the issues related with the vast data available online and the enormous ammmount of efort that implies to correctly label it; in addition, we use classifcation techniques for natural language processing tasks.

Keywords: Polarity, pseudolabeling, twits, machine learning, NLP.

1. Introducción

Las redes sociales hoy en día forman una parte de la vida cotidiana para la población en general, ya sea para cuestiones de relaciones interpersonales, networking e incluso, para la consulta y diseminación de información [14, 17, 13] .

A partir de este incremento en el uso de redes sociales, intensificado en años recientes gracias a la pandemia de COVID-19 [6], es que estas plataformas se han vuelto parte del discurso público, ya que los algoritmos utilizados en las mismas permiten a sus usuarios interactuar con diversos grupos sociales, lo que los mantiene al tanto de los eventos y problemáticas actuales [4].

En particular, Twitter no presenta muchas restricciones sobre el contenido de las publicaciones que sus usuarios pueden efectuar por lo que, en general, suelen ser sobre cualquier tema, esta aparente libertad que la plataforma provee es la principal razón de que esta red social tiene preferencia entre los internautas para, en ella, mostrar sus opiniones [13].

Dentro de las áreas de procesamiento de lenguaje natural (PLN) y lingüística computacional existe la tarea de análisis de opiniones, que consiste en, mediante el análisis del texto donde un comentario opinión fue expresado, determinar el la opinión que una persona sobre el tema en cuestión [15]; el análisis de la polaridad en una opinión se considera, a su vez, una subtarea de este campo [10].

Determinar la polaridad de un texto se refiere, entonces, a clasificar, dado un texto, si la opinión que se ha vertido en este es positiva o negativa, es decir, qué tan polarizada resulta.

2. Revisión de la literatura

La clasificación de textos, por su parte, puede llevarse a cabo mediante estrategias de apredizaje automática, específicamente, aprendizaje supervisado. Estas técnicas han sido, y siguen siendo, ampliamente utilizadas en la clasificación de textos para diferentes tareas, siendo una de las más destacadas el análisis de sentimientos.

Por ejemplo, en [7] se hace uso de redes neuronales recurrentes (RNR), específicamente redes Bi-LSTM (del inglés Bi Long Short-Term Memory) para esta tarea; de manera similar, [1] utilizan Twitter como un medio donde la gente puede expresar síntomas de depresión que requieren ser reportados por un individuo con esta afectación psicólogica para detectar dicho padecimiento de manera temprana, para lo cual utilizaron RNR tradicionales, así como redes LSTM; en [18] utilizan el modelo de transforme RoBERTa-GRU (del inglés Robustly Optimized BERT Pretraining Approach y Gated Recurrent Units) para la clasificación de sentimientos en diversos datasets considerados como baselines; por su parte, [5] utilizan el clasificador Naive Bayes (NB) para la misma tarea, apoyandose del recurso léxico sentiwordnet para agregar a cada palabra un puntaje de sentimiento positivo, negativo u objetivo.

En cuanto a la tarea específica de análisis de polaridad en texto, uno de los primeros trabajos que se llevaron a cabo fue aquel de [16], en el cual se utilizaron clasificadores tradicionales, tales como: NB, Entropía Máxima y Máquinas de Soporte Vectorial (SVM, del inglés Support Vector Machines) para la clasificación de polaridad

Tabla 1. Las cuatro configuraciones utilizadas; para cada una de ellas se muestra si se removieron o no palabras auxiliares, y lematización.

Configuración	Remoción de palabras Auxiliares	Lematización
1	No	No
2	No	Sí
3	Sí	No
4	Sí	Sí

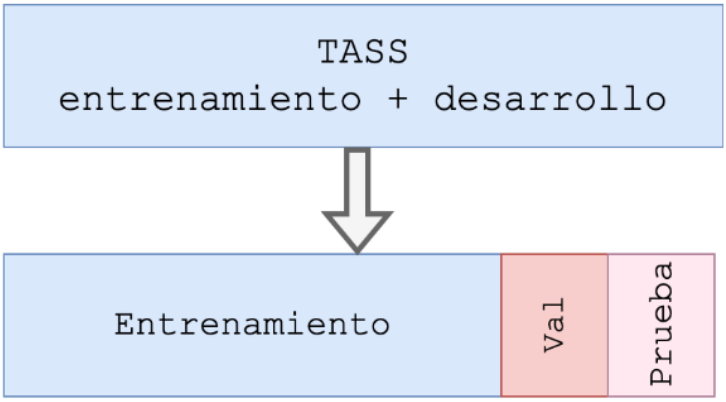


Fig. 1. Particiones sobre el dataset TASS.

de reseñas (en inglés) de películas; en [8] se realiza un estudio sobre el impacto de la negación en la clasificación de la polaridad en tuits en español, concluyendo que el tomar en cuenta dicho aspecto contribuye significativamente a una mejora en la clasificación de la polaridad; además, se encuentra el trabajo de [12], donde se utilizan múltiples clasificadores, tales como: Entropía Máxima, NB Multinomial (NBM), SVMs y BETO, un modelo BERT (del inglés Bidirectional Encoder Representations from Transformers) entrenado con un corpues en español, para obtener la polaridad de tuits en español, lo que incluía encabezados de noticias, los hilos de la conversación correspondiente a dichos encabezados, tuits citados y los hilos de conversación que se generaron a partir de éstos.

También podemos encontrar el trabajo de Arias et al. [2], quienes crearon un banco de datos mediante la extracción de tuits a través de la API1 de Twitter, con clases: positivo, negativo y neutro, durante las pruebas de clasificación los autores optaron por diversos modelos, como son: Random Forest (RF), K-nearest Neighbors (KNN), NB, Gradient Boosting (GB), Support Vector Classifier (SVC) y Extreme Gradient Boosting (XGBoost).

Por otro lado, se ha observado que los modelos de aprendizaje de máquina y aprendizaje profundo suelen ver afectado su desempeño cuando no se cuenta con suficientes datos, por lo que se suelen utilizar técnicas de aumento de datos y, en los casos en los que no se puede preservar las etiquetas, pseudoetiquetado, tal es el caso de [3], donde se utilizó dicha técnica para mejorar el desmepeño de diversas arquitecturas de redes neuronales para la tarea de detección de agresión en redes sociales.

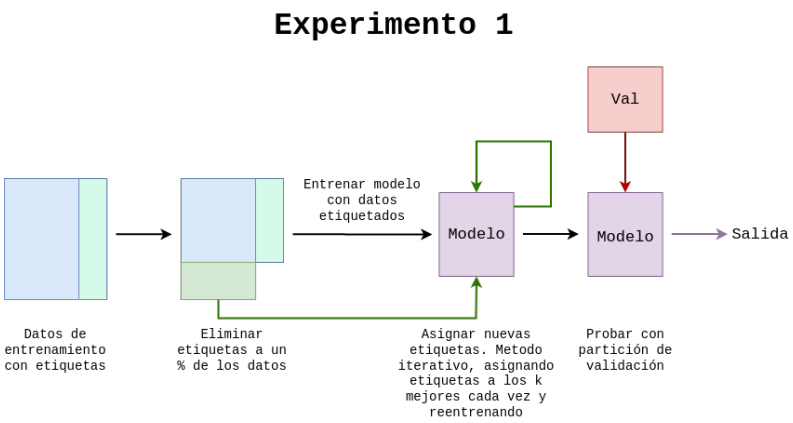


Fig. 2. Experimento 1.

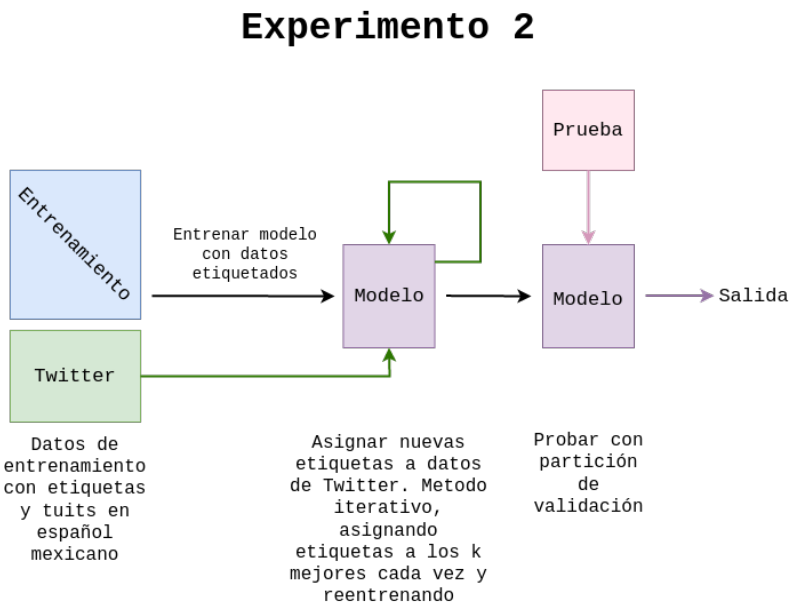


Fig. 3. Experimento 2.

También destaca el trabajo de [11], donde se hace uso del modelo DistilBERT en la tarea de clasificación de preguntas para su incorporación de sistemas tipo chatbots dedicados a responder preguntas, para contravenir la falta the datos etiquetados se utilizan técnicas de pseudoetiquetado, obteniendo como resultado que el modelo cuyo banco de datos de entrenamiento datos pseudoetiquetados presentó un mejor desempeño que aquellos con los cuáles no se utilizó dicha técnica para su entrenamiento.

Tabla 2. Resultados del experimento 1 con el modelo de RL.

Conf	% ignorado	Recall	Precisión	Acc	F1
	0	0.4981	0.5018	0.5822	0.4992
	20	**0.5297**	**0.5493**	**0.6151**	**0.5341**
1	40	0.5028	0.5142	0.5813	0.5048
	60	0.4829	0.4982	0.584	0.4825
	80	0.4222	0.437	0.584	0.4094
	0	**0.4992**	**0.49**	**0.5644**	**0.4557**
	20	0.4316	0.4777	0.544	0.4287
2	40	4339	0.4589	0.5333	0.4293
	60	0.4288	0.4742	0.5458	0.4191
	80	0.4038	0.4383	0.5124	0.3824
	0	**0.5262**	**0.5469**	**0.6044**	**0.5326**
	20	0.5105	0.5212	0.5884	0.5137
3	40	0.5036	0.5231	0.5862	0.5079
	60	0.4696	0.4908	0.5644	0.4694
	80	0.421	0.4607	0.5378	0.4076
	0	0.4731	0.4771	0.5422	0.4747
	20	**0.4953**	**0.5005**	**0.5667**	**0.4958**
4	40	0.4809	0.4878	0.5596	0.4815
	60	0.4539	0.4662	0.5489	0.4544
	80	0.4284	0.434	0.5307	0.4212

Por último, en [9] se hace uso del pseudoetiquetado para la mejora en los sistemas de detección de noticias, ya que el etiquetado manual de texto suele ser una tarea laboriosa, sobre todo dada la ingente cantidad de recursos disponibles en internet, lo que resulta en una falta de datos etiquetados disponibles; en este trabajo se observó un incremento en el desempeño de clasificación de entre el 2 % y 3 % cuando se agregaron nuevos datos cuya etiqueta fue asignada mediante algún algoritmo.

3. Desarrollo del proyecto

3.1. Datasets

Se utilizaron dos datasets: del TASS 2019 se seleccionaron las partes de entrenamiento y desarrollo para ser unidas en una sola partición, adicionalmente, se hizo un proceso de web scraping mediante la API versión 2 de Twitter, con lo que se extrajeron 6500 tuits en español mexicano.

3.2. Preproceso de datos

Como parte de la etapa del preprocesamiento de datos se llevaron a cabo los siguientes procedimientos:

– Entidades HTML: se remueven las entidades HTML que contenga el texto.

Tabla 3. Resultados del experimento 1 con el modelo de SVM.

Conf	% ignorado	Recall	Precisión	Acc	F1
	0	0.5039	0.5899	0.6444	0.4949
	20	**0.526**	**0.5379**	**0.6049**	**0.5291**
1	40	0.4979	0.5104	0.5884	0.4989
	60	0.4758	0.4952	0.5778	0.4749
	80	0.4275	0.4484	0.5449	0.4187
	0	**0.4538**	**0.6328**	**0.5956**	**0.4448**
	20	0.4416	0.4913	0.5547	0.4371
2	40	0.4317	0.4821	0.536	0.4237
	60	0.4312	0.4545	0.5422	0.4222
	80	0.4078	0.4428	0.5302	0.3874
	0	0.4764	0.7902	0.6311	0.4601
	20	**0.5211**	**0.5388**	**0.5969**	**0.5259**
3	40	0.5049	0.5187	0.5827	0.508
	60	0.4872	0.521	0.5756	0.4918
	80	0.4291	0.4786	0.5471	0.4224
	0	**0.4901**	**0.6983**	**0.6356**	**0.4822**
	20	0.4794	0.4848	0.5591	0.4804
4	40	0.4761	0.489	0.5578	0.4764
	60	0.4535	0.4679	0.5418	0.4535
	80	0.4326	0.4592	0.5329	0.4279

– Saltos de línea: se quitan los saltos de línea.
– Hashtags: En caso de haber hashtags, se separa el texto contenido en los mismos (p.e. #CiudadDeMexico → Ciudad De Mexico).
– Entidades de Twitter: se les dice así a las entidades que se utilizan propiamente en Twitter para denotar usuarios, etiquetas, hashtags y retuits, cada uno de estos tiene un identificador especial (@User, rt, #hashtag), se identifican estas entidades y se remueven del texto.
– URLs: se identifican y se remueven del texto.
– Transformar el texto a únicamente letras minúsculas.
– Palabras auxiliares: en caso de que así se requiera, se remueven las palabras auxiliares que contenga el texto.
– Lematización: si se requiere, las palabras son lematizadas utilizando la librería spaCy.
– Apóstrofes: tras la lematización, se remueven los apóstrofes del texto, conservando el caracter sin el apóstrofe (p.e concatenación, niño → concatenacion, nino).
– Puntuación: se remueven los caracteres utilizados para puntuar el texto (puntos, comas, punto y comas, etc.).
– Caracteres repetidos: en caso de que un caracter se repita más de tres veces, este se corta a dos repeticiones (p.e. Nooooo → Noo).
– Palabras alfanuméricas: si el texto contiene palabras compuestas por letras y números, como en el leet speaking, estas se remueven.
– Caracteres especiales: se remueven todos los caracteres especiales que no aportan nada al texto, signos de admiración, interrogación, etc.

Tabla 4. Resultados del experimento 1 con el modelo de NBM.

Conf	% ignorado	Recall	Precisión	Acc	F1
1	**0**	**0.4883**	**0.4883**	**0.4883**	**0.4883**
	20	0.4589	0.4550	0.6244	0.4235
	40	0.4356	0.4882	0.6031	0.3964
	60	0.4001	0.4511	0.5707	0.3455
	80	0.3498	0.4344	0.5218	0.2604
2	**0**	**0.4883**	**0.5459**	**0.5956**	**0.4406**
	20	0.4502	0.5855	0.5973	0.4343
	40	0.4271	0.5129	0.5836	0.3969
	60	0.4137	0.4854	0.5764	0.3723
	80	0.3856	0.4493	0.5551	0.3245
3	**0**	**0.5539**	**0.624**	**0.6667**	**0.5608**
	20	0.5201	0.6097	0.648	0.5127
	40	0.5099	0.5996	0.644	0.4969
	60	0.4671	0.5786	0.6076	0.448
	80	0.4327	0.4582	0.5636	0.3937
4	**0**	**0.5341**	**0.6427**	**0.6578**	**0.5388**
	20	0.5159	0.6908	0.6498	0.5131
	40	0.4913	0.6497	0.6329	0.4772
	60	0.4751	0.6585	0.6147	0.457
	80	0.3765	0.5051	0.5391	0.3169

– Espacios en blanco: en caso de que exista más de un espacio en blanco entre palabras, estos se remueven para homogeneizar el texto.

Durante este proceso se crearon cuatro configuraciones para los conjuntos de datos utilizados, las cuáles difieren únicamente en si se incluyó o no, tanto la remoción de palabras auxiliares como la lematización de los textos; es decir, las cuatro configuraciones utilizadas contienen todo el preproceso enlistado previa- mente, y solo difieren entre sí por la presencia o ausencia de palabras auxiliares y la lematización. La Tabla 1 muestra cada una de las configuraciones.

Por último, para la extracción de características se utilizó un método de bolsa de palabras binario, donde los vectores solo tienen valores 0 y 1, dependiendo de si la palabra está presente o no en un tuit dado.

3.3. Experimentos

En este trabajo se utilizan tres modelos: Regresión Logística (RL), NBM y una SVM con kernel lineal; además, creamos una partición de tres conjuntos sobre el dataset generado a partir de TASS: una para entrenamiento, que se conforma por el 70 % de los datos; además de dos particiones de validación y prueba, respectivamente, cada una formada por 15 %. La Figura 1 muestra gráficamente el proceso de partición.

Se realizaron dos tipos de experimentos, cada uno de ellos utilizando los 3 modelos seleccionados, para las 4 configuraciones de datos descritas en la Tabla 1.

Tabla 5. Resultados del experimento 2 con el modelo de RL.

Conf	% pseudo	Total	Orig	Pseudo	Recall	Precisión	Acc	F1
	0	1050	1050	0	0.5403	0.5526	0.6267	0.5405
	20	1312	1050	263	0.5404	0.5682	0.6262	0.5440
1	**40**	**1749**	**1050**	**700**	**0.5516**	**0.5764**	**0.6324**	**0.5557**
	60	2623	1050	1574	0.5451	0.5694	0.6248	0.5484
	80	5245	1050	4196	0.5495	0.5774	0.6328	0.5532
	0	**1050**	**1050**	**0**	**0.4217**	**0.4474**	**0.5333**	**0.4079**
	20	1312	1050	263	0.3642	0.4583	0.508	0.3035
2	40	1749	1050	700	0.3623	0.4656	0.5075	0.2955
	60	2623	1050	1574	0.3656	0.4369	0.5106	0.3013
	80	5245	1050	4196	0.3662	0.4571	0.5111	0.3033
	0	1050	1050	0	0.5034	0.507	0.5867	0.5021
	20	1312	1050	263	0.5226	0.5484	0.6004	0.5273
3	**40**	**1749**	**1050**	**700**	**0.5296**	**0.5471**	**0.604**	**0.533**
	60	2623	1050	1574	0.528	0.5495	0.6062	0.5315
	80	5245	1050	4196	0.5248	0.545	0.6027	0.5279
	0	**1050**	**1050**	**0**	**0.5155**	**0.5313**	**0.5956**	**0.5177**
	20	1312	1050	263	0.4913	0.5016	0.5796	0.4894
4	40	1749	1050	700	0.4864	0.4979	0.5733	0.4843
	60	2623	1050	1574	0.4818	0.4935	0.5702	0.4803
	80	5245	1050	4196	0.4919	0.5007	0.58	0.4885

En el primer experimento se utiliza la partición de entrenamiento para entrenar al modelo como pseudoetiquetador.

En primera instancia, mediante un proceso inspirado en k-fold cross validation, se eliminan las etiquetas de un porcentaje de los datos de entrenamiento (variando en cada iteración los datos a los cuales se les quitó su etiqueta); posteriormente se entrena al modelo con los datos que aún mantienen su etiqueta verdadera; toda vez que se ha entrenado al modelo, se procede a asignar etiquetas a los datos que carecen de ellas, aquí se conservan los k mejores resultados para reentrenar el modelo con los datos con etiquetas verdaderas y los nuevos k datos, este proceso continúa hasta terminar de asignar nuevas etiquetas; por último, se utiliza el conjunto de validación (Figura 1) para medir el desempeño del modelo, tanto como pseudoetiquetador y como clasificador al final del experimento. La Figura 2 muestra el procedimiento llevado a cabo.

En el segundo experimento se vuelve a trabajar con el conjunto de entrenamiento para utilizar al modelo como pseudoetiquetador, una vez que se ha entrenado, se procede a añadir etiquetas a los datos extraídos de Twitter (que no tienen una etiqueta asignada) mediante el mismo proceso del experimento 1, es decir, se asignan etiquetas a todos los datos, se conservan las k mejores para reentrenar al modelo y así sucesivamente, hasta terminar de etiquetar los datos de Twitter.

Una vez que se ha concluído esa primera etapa, se procede a medir el desmepeño del modelo con el conjunto de prueba (Figure 1). De igual modo, la Figura 3 muestra el procedimiento realizado en el experimento 2.

Tabla 6. Resultados del experimento 2 con el modelo de SVM.

Conf	% pseudo	Total	Orig	Pseudo	Recall	Precisión	Acc	F1
	0	1050	1050	**0**	0.4938	0.7688	0.6311	0.4599
	20	1312	1050	263	0.4979	0.7707	0.6329	0.4706
1	**40**	**1749**	**1050**	**700**	**0.4988**	**0.7713**	**0.6342**	**0.4713**
	60	2623	1050	1574	0.4933	0.7684	0.6307	0.4595
	80	5245	1050	4196	0.4938	0.7688	0.6311	0.4599
	0	**1050**	**1050**	**0**	**0.4135**	**0.4816**	**0.5644**	**0.3692**
	20	1312	1050	263	0.3904	0.4629	0.5458	0.3282
2	40	1749	1050	700	0.4054	0.4742	0.56	0.3512
	60	2623	1050	1574	0.4096	0.471	0.5627	0.3591
	80	5245	1050	4196	0.415	0.4729	0.5667	0.3689
	0	**1050**	**1050**	**0**	**0.4652**	**0.7853**	**0.6089**	**0.4335**
	20	1312	1050	263	0.4613	0.619	0.6067	0.4258
3	**40**	**1749**	**1050**	**700**	**0.4617**	**0.6507**	**0.6067**	**0.4265**
	60	2623	1050	1574	0.4648	0.785	0.6084	0.433
	80	5245	1050	4196	0.4652	0.7853	0.6089	0.4335
	0	**1050**	**1050**	**0**	**0.4592**	**0.4278**	**0.6044**	**0.4174**
	20	**1312**	**1050**	**263**	**0.4587**	**0.4274**	**0.604**	**0.4169**
4	40	1749	1050	700	0.4587	0.4274	0.604	0.4169
	60	2623	1050	1574	0.4592	0.4278	0.6044	0.4174
	80	5245	1050	4196	0.4592	0.4278	0.6044	0.4174

4. Resultados

En esta sección se presentan los resultados de ambos experimentos. En las Tablas 2, 3 y 4 se presentan los resultados obtenidos en el experimento 1 con los modelos RL, SVM y NBM, respectivamente.

Por su parte, las tablas Tablas 5, 6 y 7 muestran los resultados obtenidos en el experimento 2 con los modelos RL, SVM y NBM, respectivamente.

5. Conclusiones

En este trabajo se presentó un primer acercamiento en el uso de técnicas de pseudoetiquetado para el robustecimiento de modelos desarrollados para la tarea de análisis de polaridad en tuits.

En primera instancia, el método aquí propuesto mostró ayudar al desempeño de los modelos de RL y SVM que, en términos generales, crean una frontera de decisión a través de un hiperplano que separa el espacio de características; por su parte, nuestro método no refleja una mejora en los resultados del clasificador NBM, que realiza sus decisiones mediante computaciones probabilísticas.

Es importante hacer notar que en el experimento 1, para aquellos casos en los que hubo una mejora en la clasificación gracias al pseudoetiquetado, dicha mejora se presenta cuando el porcentaje de datos ignorados no excede la mitad de la cardinalidad del dataset original; es decir, una vez que se ignora la etiqueta de la mitad, o más, patrones, deja de haber una mejora en los resultados.

Tabla 7. Resultados del experimento 2 con el modelo de NBM.

Conf	% pseudo	Total	Orig	Pseudo	Recall	Precisión	Acc	F1
	0	**1050**	**1050**	**0**	**0.461**	**0.5766**	**0.5956**	**0.4343**
	20	1312	1050	263	0.4569	0.665	0.5964	0.4277
1	40	1749	1050	700	0.4356	0.533	0.5844	0.3921
	60	2623	1050	1574	0.4211	0.4577	0.5733	0.3727
	80	5245	1050	4196	0.3962	0.4833	0.5516	0.3365
	0	**1050**	**1050**	**0**	**0.3858**	**0.4469**	**0.5289**	**0.3372**
	20	1312	1050	263	0.3366	0.3983	0.4964	0.2272
2	40	1749	1050	700	0.3333	0.1644	0.4933	0.2202
	60	2623	1050	1574	0.3333	0.1644	0.4933	0.2202
	80	5245	1050	4196	0.3333	0.1644	0.4933	0.2202
	0	**1050**	**1050**	**0**	**0.4981**	**0.5527**	**0.6044**	**0.4874**
	20	1312	1050	263	0.4908	0.5452	0.5982	0.4824
3	40	1749	1050	700	0.4738	0.5413	0.5902	0.4570
	60	2623	1050	1574	0.4638	0.5367	0.5916	0.4384
	80	5245	1050	4196	0.4506	0.5183	0.5871	0.4143
	0	1050	1050	0	0.4608	0.4934	0.5778	0.4347
	20	**1312**	**1050**	**263**	**0.4679**	**0.5188**	**0.5849**	**0.4447**
4	40	1749	1050	700	0.4615	0.5085	0.588	0.4288
	60	2623	1050	1574	0.4571	0.5472	0.5924	0.4196
	80	5245	1050	4196	0.4505	0.4724	0.592	0.4102

Por último, se observa que el ignorar etiquetas de algunos patrones, para posteriormente asignarles a los mismos nuevas etiquetas mediante pseudoetique- tado, contribuyó a una mejora en el dessempeño de los modelos, lo que presenta una nueva veta de investigación que permita utilizar el pseudoetiquetado como un método de regularización.

Referencias

1. Apoorva, A., Goyal, V., Kumar, A., Singh, R., Sharma, S.: Depression detection on twitter using RNN and LSTM models. In: Advanced Network Technologies and Intelligent Computing: Second International Conference, ANTIC 2022, vol. 1798, pp. 305–319 (2023) doi: 10.1007/978-3-031-28183-9_22
2. Arias, F., Guerra-Adames, A., Zambrano, M., Quintero-Guerra, E., Tejedor-Flores, N.: Analyzing spanish-language public sentiment in the context of a pandemic and social unrest: The Panama case. International Journal of Environmental Research and Public Health, vol. 19, no. 16 (2022) doi: 10.3390/ijerph191610328
3. Aroyehun, S. T., Gelbukh, A.: Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC'08), pp. 90–97 (2018)
4. Bastick, Z.: Would you notice if fake news changed your behavior? An experiment on the unconscious effects of disinformation. Computers in Human Behavior, vol. 116, pp. 1–12 (2021) doi: 10.1016/j.chb.2020.106633
5. Goel, A., Gautam, J., Kumar, S.: Real time sentiment analysis of tweets using Naive Bayes. In: 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), pp. 257–261 (2016) doi: 10.1109/NGCT.2016.7877424

6. Greenhow, C., Staudt-Willet, K. B., Galvin, S.: Inquiring tweets want to know:# Edchat supports for# RemoteTeaching during COVID-19. British Journal of Educational Technology, vol. 52, no. 4, pp. 1434–1454 (2021) doi: 10.1111/bjet.13097
7. Jaca-Madariaga, M., Zarrabeitia-Bilbao, E., Rio-Belver, R. M., Moens, M. F.: Sentiment analysis model using Word2vec, Bi-LSTM and attention mechanism. In: IoT and Data Science in Engineering Management: Proceedings of the 16th International Conference on Industrial Engineering and Industrial Management and XXVI Congreso de Ingeniería de Organización, vol. 160, pp. 239–244 (2023) doi: 10.1007/978-3-031-27915-7_43
8. Jimenez-Zafra, S. M., Martin-Valdivia, M. T., Martinez-Camara, E., Urena-Lopez, L. A.: Studying the scope of negation for spanish sentiment analysis on twitter. IEEE Transactions on Affective Computing, vol. 10, no. 1, pp. 129–141 (2019) doi: 10.1109/TAFFC.2017.2693968
9. Jiménez, D., Gambino, O. J., Calvo, H.: Pseudo-labeling improves news identification and categorization with few annotated data. Computación y Sistemas, vol. 26, no. 1, pp. 183–193 (2022) doi: 10.13053/cys-26-1-4163
10. Juárez-Gambino, J. O.: Sentiment polarity prediction of twitter users' opinions to national newspapers news. Ph.D. thesis, Centro de Investigación en Computación (2019)
11. Kuligowska, K., Kowalczuk, B.: Pseudo-labeling with transformers for improving question answering systems. Procedia Computer Science, vol. 192, pp. 1162–1169 (2021) doi: 10.1016/j.procs.2021.08.119
12. Macias, C., Calvo, H., Gambino, O.: News intention study and automatic estimation of its impact. In: Advances in Computational Intelligence - 21st Mexican International Conference on Artificial Intelligence, MICAI'22, Proceedings, vol. 13613, pp. 83–100 (2022) doi: 10.1007/978-3-031-19496-2_7
13. Macias-Sánchez, C.: Estudio de la intención de noticias y estimación automática de su impacto. Master's thesis, Centro de Investigación en Computación, Instituto Politécnico Nacional (2022)
14. Madni, H. A., Umer, M., Abuzinadah, N., Hu, Y. C., Saidani, O., Alsubai, S., Hamdi, M., Ashraf, I.: Improving sentiment prediction of textual tweets using feature fusion and deep machine ensemble model. Electronics, vol. 12, no. 6 (2023) doi: 10.3390/electronics12061302
15. Mejova, Y.: Sentiment analysis: An overview. University of Iowa, Computer Science Department, pp. 1–34 (2009)
16. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pp. 79–86 (2002) doi: 10.3115/1118693.1118704
17. Pezoa-Fuentes, C., García-Rivera, D., Matamoros-Rojas, S.: Sentiment and emotion on twitter: The case of the global consumer electronics industry. Journal of Theoretical and Applied Electronic Commerce Research, vol. 18, no. 2, pp. 765–776 (2023) doi: 10.3390/jtaer18020039
18. Tan, K. L., Lee, C. P., Lim, K. M.: RoBERTa-GRU: A hybrid deep learning model for enhanced sentiment analysis. Applied Sciences, vol. 13, no. 6 (2023) doi: 10.3390/app13063915

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación