EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 151 No. 11
November 2022

Research in Computing Science

Research in Computing Science, Año 21, Volumen 151, No. 11, noviembre de 2022, 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 noviembre de 2022.

Research in Computing Science, year 21, Volume 151, No. 11, November 2022, is published monthly by the Center for Computing Research of IPN.

Volume 151(11)

Advances in Computing Science and Applications

Germán Ríos-Toledo
Fernando Pech-May (eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2022

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

Propuesta de control de acceso vehicular empleando aprendizaje profundo 5
Fernando Contreras-Perez, Oscar Chavez-Bosquez, Jose Hernandez-Torruco, and Betania Hernandez-Ocana

Deep Learning-based Methods for Face and Text Detection in Natural Images..... 17
Marco Lopez-Sanchez, Oscar Chavez-Bosquez, Betania Hernandez-Ocana, Jose Hernandez-Torruco

Segmentación semántica en cultivos de jitomate usando ResNet50..... 29
Juan Pablo Guerra Ibarra, Francisco Javier Cuevas de la Rosa, Oziel Arellano Arzola

Three Metrics for Build Association Rules Using Differential Evolution for Bacterial Vaginosis 45
Freddy Garcıa-Fuentes, Juana Canul-Reich, Rafael Rivera-Lopez, Efren Mezura-Montes, and Erick De-La-Cruz-Hernández

Sistema de control de acceso mediante identificacion y verificacion facial 55
Luis Antonio Lopez Gomez, Jorge Magana Govea, Fernando Pech May

ISSN 1870-4069

Propuesta de control de acceso vehicular empleando aprendizaje profundo

Fernando Contreras-Pérez, Oscar Chávez-Bosquez,
José Hernández-Torruco, Betania Hernández-Ocaña

Universidad Juárez Autónoma de Tabasco,
División Académica de Ciencias y Tecnologías de la Información,
México

212H13001@alumno.ujat.mx,
{oscar.chavez,jose.hernandezt,betania.hernandez}@ujat.mx

Resumen. La vigilancia y el control de acceso en instituciones públicas o privadas son elementos muy importantes para contribuir a la seguridad de las personas, vehículos e inmuebles que se encuentren dentro del área o lugar custodiado. Si tomamos en cuenta que la inseguridad es uno de los problemas que más afectan al país y que en el estado de Tabasco se ha presentado una tendencia a la alza en los robos y hurtos a instituciones educativas durante los últimos años, entonces tiene sentido contribuir en un sistema que apoye a la vigilancia. Este trabajo propone un sistema de identificación automática de vehículos empleando aprendizaje profundo y reconocimiento óptico de caracteres para identificar y reconocer los caracteres de las matrículas de vehículos. El sistema propuesto cumple ser simple en su implementación y sus elementos son económicos, con el propósito de ser un modelo replicable para diferentes áreas o lugares de interés.

Palabras clave: Redes neuronales convolucionales, raspberry pi, openCV AI kit.

Proposal for Vehicle Access Control Using Deep Learning

Abstract. Abstract. Surveillance and access control in public or private institutions are key elements to ensuring the security of people, vehicles, and facilities within a monitored area. Considering that insecurity is one of the most pressing problems in the country, and that in the state of Tabasco there has been an upward trend in thefts and burglaries in educational institutions in recent years, it makes sense to contribute to a system that supports surveillance efforts. This work proposes an automatic vehicle identification system using deep learning and optical character recognition (OCR) to detect and recognize the characters on vehicle license plates. The proposed system is designed to be simple to implement and composed of cost-effective components, with the goal of being a replicable model for different areas or locations of interest.

Keywords: Convolutional neural networks, raspberry pi, openCV AI kit.

1. Introducción

La inseguridad es un problema social muy presente en México. Todos los días se cometen robos, asaltos, hurtos, secuestros, entre otros delitos. Este problema ha causado una serie de efectos en la sociedad. Según [28], existen cambios en las costumbres para salir de casa, para transportarse y pasear, lo cual desincentiva o limita las actividades sociales y políticas.

La inseguridad en el sector educativo no es la excepción. De acuerdo a los datos de las fiscalías estatales y las secretarías de Educación, entre marzo de 2020 y marzo de 2021 las escuelas de México han sufrido casi 7000 robos [18]. Por su parte, la Secretaría de Eduación del estado de Tabasco (SETAB) reportó en 2021 que se habían registrado un total de 32 robos a escuelas públicas, el número más alto en los últimos cuatro años [6].

Con respecto al robo de autos particulares, en [21] se menciona que este delito tuvo un crecimiento del 40 % de 2015 a 2016 en Tabasco, pasando de 2250 a 3508 casos. De estos 3508 robos, el 75 % se registraron en la región Centro-Chontalpa. Justamente en esta región se ubica el Campus Chontalpa de la Universidad Juárez Autónoma de Tabasco (UJAT). Además, se sabe que el robo de autos es una actividad central en la economía criminal y tiene una relación íntima con otros delitos como el secuestro.

Por otro lado, durante muchos años la construcción de un sistema de reconocimiento de características requería de una cuidadosa ingeniería y vasta experiencia en el área para poder transformar los datos de su forma a cruda a una representación adecuada para los sistemas de aprendizaje automático. En cambio, los métodos de aprendizaje profundo son métodos de aprendizaje con múltiples niveles de representación. Esta representación se va transformando por niveles, desde la forma cruda de los datos hacia niveles superiores cada vez más abstractos. Además, las capas de características significativas se van aprendiendo de manera automática sin necesidad de ser diseñadas por ingenieros humanos. Este enfoque es muy útil, especialmente para descubrir características complicadas o confusas en datos de gran dimensión y ha batido récords en diferentes campos de la inteligencia artificial [14].

El aprendizaje profundo se basa en redes neuronales artificiales, las cuales poseen por naturaleza una capacidad de aprendizaje. De acuerdo a [4], los términos aprendizaje profundo y red neuronal profunda se refieren a una red neuronal artificial con múltiples capas. Además también menciona que el aprendizaje profundo es considerado una de las herramientas más poderosas y populares en la literatura gracias a su capacidad de manejar grandes volúmenes de datos. El interés en tener capas ocultas más profundas está empezando a superar el rendimiento de los métodos clásicos, especialmente en el campo del reconocimiento de patrones.

Debido a lo anterior, en este artículo se propone un sistema de identificación automática de vehículos en tiempo real empleando aprendizaje profundo que busca mejorar y agilizar la entrada de vehículos al Campus Chontalpa de la UJAT. Otro de los objetivos del sistema es el de contribuir a la seguridad de los vehículos, del personal de la universidad, estudiantes, trabajadores y comerciantes, y de las instalaciones en general. Esto mediante la clasificación e identificación automática de los vehículos de profesores, estudiantes, y en general, de los vehículos que ingresen al campus.

2. Trabajos relacionados

Existen muchas formas de identificar vehículos de manera automática, ya sea empleando técnicas de inteligencia artificial o técnicas de otras áreas de las Ciencias de la Computación [3]. Dentro del estado del arte de la identificación automática de vehículos, se han utilizado principalmente tres enfoques: uso de sensores, uso de algoritmos clásicos de visión computacional y uso de aprendizaje profundo.

Con respecto al uso de sensores, en [9,8] utilizaron redes de sensores con etiquetas de identificación de radio frecuencia (RFID, del inglés *Radio-Frecuency Identification*) y el protocolo de largo alcance (LoRa, del inglés *Long Range*) para la identificación de vehículos y la comunicación con un servidor en la nube. Además, propusieron una arquitectura para una red de sensores que aborda el problema de monitorear las redes de tráfico. Las principales desventajas de esta propuesta es la adquisición de dispositivos y la integración de diferentes tecnologías. Además, el tiempo de vida útil de un sensor se ve afectado por diversos factores como sobrecalentamiento, humedad, desperfecto por vibraciones, entre otras.

En cuanto a las técnicas clásicas de visión computacional destacan [30,26,2], en los cuales utilizan el algoritmo HOG (*Histogram of oriented gradients*) para describir propiedades representativas del vehículo, como la apariencia general y la textura local. También se propone un sistema de identificación de vehículos que tome en cuenta no solo la forma del vehículo, sino también su matrícula. Por otro lado, también se abordó el problema del monitoreo de las actividades en las intersecciones de tráfico para detectar congestiones y luego predecir el flujo de vehículos, lo que ayuda a regular el tráfico.

Finalmente, existe la tendencia de emplear aprendizaje profundo para la identificación automática de vehículos. En ese sentido, en [23] se aborda este problema mediante un sistema dividido en tres partes: detección, segmentación y reconocimiento de caracteres. La parte de detección de los vehículos puede ser realizada mediante un modelo de red neuronal existente, modificado o propio. En [12,10,25,29] se utilizan diferentes versiones y modificaciones que mejoran uno de los modelos de red neuronal más utilizados para la detección de vehículos: YOLO [22] (del inglés *You Only Look Once)*. También se utilizan herramientas tecnológicas como Raspberry Pi3 y módulos de cámara Pi NoIR [12]. Algunos trabajos como [17,24] utilizan la identificación de vehículos con objetivos más

específicos como el monitoreo de las reglas de tránsito, la vigilancia del robo de vehículos o la estimación dinámica de flujos origen-destino. En estos trabajos se toman en cuenta distintos aspectos y detalles clave tales como imágenes en color y escala de grises, la calidad de las imágenes, la velocidad de movimiento de los vehículos, la iluminación, las condiciones climáticas, las variaciones en ángulos, entre otros. Además se presta atención a las características de la placa que varían según el estado o país de origen tales como colores, tamaños, tipos de alfabeto, diseño de los dígitos y caracteres escasos.

Un trabajo muy importante dentro del estado del arte es [27]. En éste se realiza una revisión de las principales arquitecturas de aprendizaje profundo utilizadas en la detección de vehículos, destacando principalmente RetinaNet [15], gracias a su precisión de detección bastante alta, que a su vez, es consecuencia de una función de pérdida que puede reducir efectivamente el peso de las muestras fáciles de clasificar y así centrarse en las muestras difíciles en la fase de entrenamiento.

Además por medio de experimentos comparativos, se encuentra que el valor de la métrica *recall* de SSD [16] (del inglés *Single Shot MultiBox Detector*) es bajo, y hay una gran tasa de detección errónea. Por su parte, YOLOv3 se comporta de manera contraria, con un valor de *recall* más alto y un valor de *precision* más bajo.

Por otro lado, en las pruebas realizadas en escenarios reales, se descubre que SSD también posee una excelente capacidad de generalización ya que la construcción del modelo es menos propensa al sobreajuste y cumple ser robusto.

3. Descripción del problema

El Campus Chontalpa de la UJAT se encuentra ubicado en el municipio de Cunduacán en el estado de Tabasco. Este campus alberga tres divisiones académicas de la Universidad: la División Académica de Ciencias Básicas, la División Académica de Ciencias y Tecnologías de la Información y la División Académica de Ingeniería y Arquitectura.

Las instalaciones del campus cuentan con varios edificios de diferentes tipos, entre ellos: aulas de clases, oficinas administrativas, cubículos para profesores, centros de investigación, laboratorios, centros de cómputo, una biblioteca, salas audiovisuales, papelerías, entre otros. Además de estos edificios, que normalmente son de uso exclusivo de alumnos, profesores y trabajadores, el campus también cuenta con instalaciones de negocios y áreas comunes como un Centro de lenguas extranjeras abierto al público en general, auditorios, una cancha de pasto sintético, canchas de baloncesto, una cancha de fútbol rapido, y un campo de fútbol. También se encuentra ubicado dentro del campus el Centro de Investigación de Ciencia y Tecnología Aplicada de Tabasco (CICTAT). Debido a lo anterior, y al hecho de que a menudo se realizan eventos de distinta índole dentro del campus, actualmente se requiere una mayor atención y control sobre el acceso de los vehículos.

Fig. 1. Vista satelital del acceso al campus.

La cantidad de alumnos que asisten al Campus Chontalpa es de aproximadamente 8000 entre las diferentes licenciaturas y posgrados de las tres Divisiones Académicas. La cantidad de profesores e investigadores es de aproximadamente 600, además de los trabajadores de vigilancia, intendencia y de los comercios dentro del campus es de aproximadamente 500. No es una tarea simple la identificación de esta cantidad de vehiculos por solo 2 o 3 personas encargadas de la vigilancia del portón de acceso al campus. Una persona podría recordar el color y probablemente hasta el modelo de un conjunto de vehículos, pero difícilmente recordará las matrículas.

Es importante mencionar que para el acceso y salida de vehículos del Campus Chontalpa de la UJAT se tiene un solo punto, el cual cuenta con un puesto de control y vigilancia (Figura 1). En este punto de acceso siempre se mantiene al menos un trabajador de vigilancia, esto con el fin de llevar el control de los vehículos que ingresan al campus. Por otra parte, el hecho de que exista solamente una entrada, genera como resultado una fila de vehículos que esperan pasar por el control del empleado de vigilancia. Esto representa un problema que empeora durante las horas pico, las cuales representan los horarios en que el número de ingresos de vehículos es significativamente mayor al resto del día. En las horas más concurridas, la fila de vehículos que se forma en la entrada del campus llega a invadir la carretera (ver Figura 2). Esta invasión y obstrucción ocasiona que el tránsito del municipio también se vea afectado, además de aumentar el riesgo de accidentes entre los automovilistas.

Debido a lo anterior, la implementación de un sistema de control de acceso vehícular es factible, ya que bastará con tener un solo dispositivo de acceso en este punto para monitorear la entrada y salida de los vehículos.

4. Materiales y método

4.1. Materiales

Para el desarrollo de nuestra propuesta se requiere el siguiente equipo de hardware y software:

Fig. 2. Control de acceso para ingresar al Campus Chontalpa de la UJAT.

Sensor OAK Lite. El kit OpenCV de inteligencia artificial con profundidad[1] (OAK-D, del inglés *OpenCV Artificial Intelligence Kit with Depth*) cuenta con tres cámaras integradas que implementan la visión estereoscópica y RGB, que a su vez están acopladas a un procesador Intel llamado Myriad X VPU (del inglés *Vision Processing Unit*) capaz de ejecutar modelos de aprendizaje profundo.

Mobilenet SSDv2. Este modelo de red neuronal convolucional, al igual que la mayoría de los modelos de redes ligeras, utiliza Mobilenet-v2 como red troncal. Esta incluye una capa convolucional estándar y 17 módulos residuales inversos. Por su parte, cada módulo residual inverso contiene una capa convolucional 1x1, una capa convolucional separable `Dwise` 3x3, funciones de normalización por lotes (BN, del inglés *Batch Normalization*) y función de activación `Relu6`.

TensorFlow Object Detection API. Es una biblioteca de código abierto para cálculo numérico y que usa como forma de programación grafos de flujo de datos. El código fue liberado como software libre e incluyó mejoras, como potenciar el rendimiento gracias al uso de unidades de procesamiento gráfico (GPU, del inglés *Graphics Processing Unit*) [1]. Además, TensorFlow facilita y acelera la investigación y la aplicación de modelos de redes neuronales y otros modelos de aprendizaje automático. La TensorFlow Object Detection API sirve para entrenar las últimas capas de una red neuronal convolucional (CNN, del inglés *Convolutional Neural Network*) con capas personalizadas. Además, facilita la construcción, el entrenamiento y el despliegue de modelos de detección de objetos [11].

PaddleOCR versión 2.6.0. Es un conjunto de herramientas de reconocimiento óptico de caracteres (OCR, del inglés *Optical Character*

[1] `https://docs.luxonis.com/projects/hardware/en/latest/pages/DM9095.html`

Recognition) multilingüe basadas en el framework PaddlePaddle[2] (del inglés *PArallel Distributed Deep LEarning*) que admite el reconocimiento de combinaciones de caracteres en diferentes idiomas de forma vertical u horizontal [31]. El objetivo de PaddleOCR es crear herramientas de OCR multilingües y prácticas que ayuden a los usuarios a entrenar mejores modelos y aplicarlos en la práctica.

Raspberry Pi 3 modelo B. Computadora de placa reducida o placa única (SBC – *SingleBoard Computer*) de bajo costo [5]. Utiliza un microprocesador con arquitectura ARM, memoria RAM y tarjeta gráa fica (GPU) en un solo chip, por tanto se trata de un sistema SoC (*System on a Chip*, Sistema en un chip). La Raspberry Pi 3 Modelo B es el primer modelo de la tercera generación de Raspberry Pi. Sustituyó a la Raspberry Pi 2 Model B en febrero de 2016.

OpenVINO. El (*Open Visual Inferencing and Neural Network Optimization*) [3] de Intel, es un kit de herramientas para desarrollar aplicaciones y soluciones basadas en el aprendizaje profundo. Entre estas tareas se encuentra la simulación de la visión humana, y el reconocimiento automático del habla. Además, proporciona un alto rendimiento y diversas opciones de implementación, desde cómputo en dispositivos integrados hasta cómputo en la nube.

4.2. Método

Para que el sensor identifique matrículas de vehículos en tiempo real y reconozca de qué vehículo se trata, planteamos la siguiente metodología:

- Recopilar un conjunto de imágenes de entrenamiento. Como primer paso usaremos el dataset de uso libre "Open Images Dataset V6"[4], el cual contiene una gran cantidad de matrículas de vehículos ya etiquetadas.
- *Data augmentation.* Aplicaremos transformaciones a cada imagen para exponer al modelo a variaciones de matrículas para así volverlo más robusto.
- *Transfer learning.* Entrenar las últimas capas de Mobilenet SSDv2 con nuestro conjunto de datos y así obtener el modelo final que reconozca matrículas de vehículos locales.
- Validación del modelo. Además de monitorear el desempeño con métricas de validación durante el entrenamiento, probaremos el modelo con imágenes de vehículos y matrículas desconocidas. Con base en estas pruebas, se calcularán las métricas *precision* y *recall*.
- *Deployment* en el dispositivo OAK Lite. Exportar el modelo de TensorFlow a DepthAI empleando la plataforma OpenVINO.

[2] `https://github.com/PaddlePaddle/PaddleOCR`

[3] `https://docs.openvino.ai/latest/openvino_docs_install_guides_overview.html`

[4] `https://storage.googleapis.com/openimages/web/visualizer/index.html?set=train&type=segmentation&r=false&c=%2Fm%2F01jfm_`

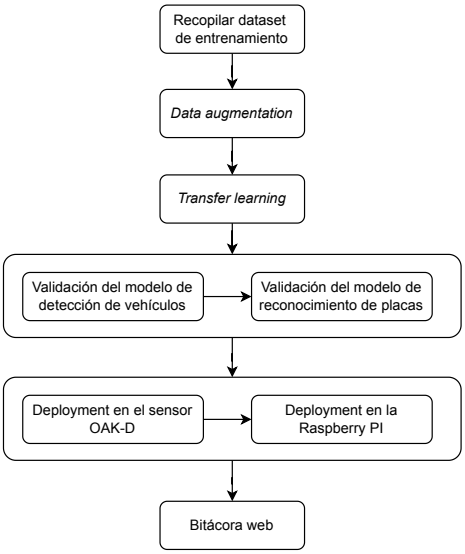


Fig. 3. Descripción del método.

– Crear el módulo OCR. Desarrollar el script para convertir la imagen de la matrícula a texto empleando PaddleOCR.
– Crear el servicio web de bitácora de acceso. Desarrollar el script para almacenar en una base de datos el número de matrícula, además de la hora y fecha en que se identificó el vehículo.

En la Figura 3 se muestran los pasos a seguir durante el desarrollo de la metodología.

5. Propuesta

En [7] se menciona que la inteligencia artificial es utilizada en ámbitos en los que se registran tareas repetitivas, el uso de grandes volúmenes de información, riesgo de vida y extrema complejidad en la resolución de problemas. Esta tendencia es la motivación para enfrentar el problema de la identificación de vehículos de manera automática mediante aprendizaje profundo con un desempeño aceptable.

El desempeño del sistema se analizará mediante las métricas *precision* y *recall*, que son utilizadas principalmente para la evaluación de métodos de detección de objetos[13]. Estas métricas serán calculadas tanto para el modelo de detección de matrículas como para el modelo de OCR de la matrícula. La métrica *precision* indica la habilidad del modelo para identificar solamente objetos relevantes, es decir, el porcentaje de predicciones positivas correctas [19]. Por su parte, *recall* mide la capacidad de un modelo para encontrar todos los

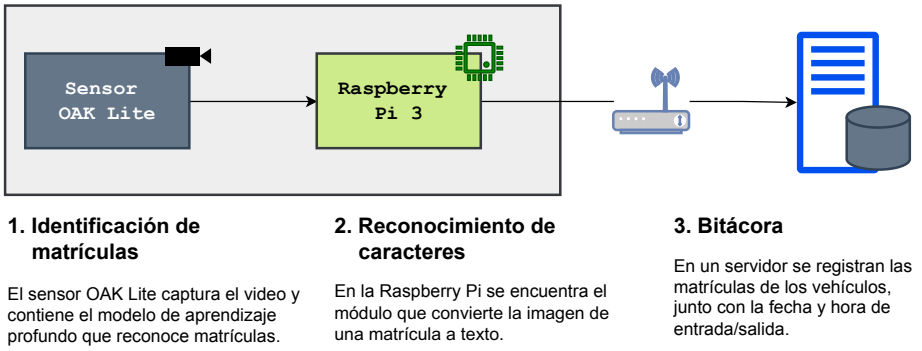


Fig. 4. Componentes y esquema de funcionamiento de la propuesta.

casos relevantes, en otras palabras, es el porcentaje de predicciones positivas correctas de todos los casos reales [20]. A continuación se puede observar el cálculo de *precision* y *recall*:

$$\text{Precision} = P = \frac{TP}{(TP + FP)} = \frac{TP}{\text{Todas las detecciones}}, \tag{1}$$

$$\text{Recall} = R = \frac{TP}{(TP + FN)} = \frac{TP}{\text{Todos los objetos reales}}. \tag{2}$$

Existen varios tipos de cámaras de video con diferentes características, todas cumpliendo la función de capturar y extraer información visual en tiempo real. Sin embargo, al utilizar el dispositivo OAK Lite en nuestro sistema, los datos obtenidos por el dispositivo pasarán directamente al modelo de red neuronal contenido dentro del mismo dispositivo. Es decir, no es necesario convertir el video obtenido por alguna cámara para poder ejecutar el modelo de detección de la placa del vehículo y así reconocer los caracteres mediante PaddleOCR.

El enfoque utilizado en este trabajo impacta en los siguientes rubros:

Computacional. Mediante la elección del aprendizaje profundo como enfoque para modelar el problema.

Social. Mediante el objetivo de procurar la seguridad y el orden entre los individuos y vehículos del lugar del proyecto.

Económico. Mediante la aportación de un sistema simple que requiere muy poca tecnología y equipo.

El Campus Chontalpa cuenta con cobertura de WiFi en todas sus áreas, por lo que es viable colocar el sensor OAK Lite conectado a una Raspberry Pi para poder enviar los resultados obtenidos (caracteres de la matrícula, fecha y hora de acceso) a un servidor dentro de la red UJAT. En la Figura 4 se muestra un esquema general del funcionamiento de nuestra propuesta.

El modelo propuesto estará limitado a identificar solo vehículos, no a sus pasajeros, esto con el fin de garantizar la privacidad. El sensor OAK Lite estará

fijo justo en la entrada del campus. Debido a los horarios de trabajo y de clases, el sistema se entrenará y funcionará durante el día, con luz solar. Es decir, no se utilizará ningún tipo de luz artificial.

Finalmente, mediante la combinación de las bondades del aprendizaje profundo y del OCR se aborda el problema de la identificación de vehículos con fines de vigilancia y control.

6. Conclusiones

Con la implementación de esta propuesta esperamos obtener beneficios no solo en seguridad, sino también en la facilidad para el control de acceso vehicular en el Campus Chontalpa de la UJAT. Se pretende que la implementación del sistema sea simple y la adquisición de materiales sea accesible económicamente, de tal manera que pueda ser replicado en diferentes instituciones, fraccionamientos, carreteras, parques industriales, entre otros.

Agradecimientos. Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo a la Maestría en Ciencias de la Computación de la Universidad Juárez Autónoma de Tabasco.

Referencias

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., , Zheng, X.: TensorFlow: A System for Large-Scale Machine Learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). pp. 265–283 (2016)
2. Ahmad, I. S., Boufama, B.: Automatic Vehicle Identification through Visual Features. In: Proceedings of the 17th International Conference on Advances in Mobile Computing & Multimedia. pp. 185–194. MoMM2019, Association for Computing Machinery, New York, NY, USA (2019) doi: 10.1145/3365921.3365938
3. Ahmed, W., Arafat, S. Y., Gul, N.: A Systematic Review on Vehicle Identification and Classification Techniques. In: 2018 IEEE 21st International Multi-Topic Conference (INMIC). pp. 1–6 (2018) doi: 10.1109/INMIC.2018.8595585
4. Albawi, S., Mohammed, T. A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). pp. 1–6. IEEE (2017) doi: 10.1109/ICEngTechnol.2017.8308186
5. Aldea, E. L.: Raspberry PI fundamentos y aplicaciones. Grupo Editorial RA-MA (2017)
6. Arias, N.: Se disparan los robos en escuelas públicas de Tabasco, (10 2021)
7. Azar, M. A., Tapia, M., García, J. L., Pérez, A. J. M.: Inteligencia artificial de las cosas. In: XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan) (2019)
8. Álvarez Bazo, F., Sánchez-Cambronero, S., Vallejo, D., Glez-Morcillo, C., Rivas, A., Gallego, I.: A Low-Cost Automatic Vehicle Identification Sensor for Traffic Networks Analysis. Sensors, vol. 20, no. 19 (2020) doi: 10.3390/s20195589

9. Griese, M. G., Kleinschmidt, J. H.: Performance Analysis of a System for Vehicle Identification Using LoRa and RFID. In: Miani, R., Camargos, L., Zarpelão, B., Rosas, E., Pasquini, R. (eds) Green, Pervasive, and Cloud Computing. pp. 115–127. Springer International Publishing, Cham (2019)
10. Hendry, Chen, R.-C.: Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning. Image and Vision Computing, vol. 87, pp. 47–56 (2019) doi: 10.1016/j.imavis.2019.04.007
11. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7310–7311 (2017) doi: 10.1109/CVPR.2017.351
12. Izidio, D. M., Ferreira, A. P., Barros, E. N.: An Embedded Automatic License Plate Recognition System using Deep Learning. In: 2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC). pp. 38–45. IEEE (2018) doi: 10.1109/SBESC.2018.00015
13. Kaur, J., Singh, W.: Tools, techniques, datasets and application areas for object detection in an image: a review. Multimedia Tools and Applications, pp. 1–55 (2022)
14. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature, vol. 521, no. 7553, pp. 436–444 (2015)
15. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Cision. pp. 2980–2988 (2017) doi: 10.1109/ICCV.2017.324
16. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A. C.: SSD: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
17. Mondal, M., Mondal, P., Saha, N., Chattopadhyay, P.: Automatic number plate recognition using CNN based self synthesized feature learning. In: 2017 IEEE Calcutta Conference (CALCON). pp. 378–381. IEEE (2017) doi: 10.1109/CALCON.2017.8280759
18. Montes, S.: Desde tuberías arrancadas a miles de pesos en equipo electrónico: Las escuelas mexicanas son saqueadas durante la pandemia, (05 2021)
19. Padilla, R., Netto, S. L., da Silva, E. A. B.: 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
20. Padilla, R., Passos, W. L., Dias, T. L. B., Netto, S. L., da Silva, E. A. B.: A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. Electronics, vol. 10, no. 3 (2021) doi: 10.3390/electronics10030279
21. Ramírez-Sánchez, R. D.: Del edén al infierno: inseguridad y construcción estatal en Tabasco. LiminaR, vol. 17, no. 2, pp. 196–216 (2019)
22. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016) doi: 10.1109/CVPR.2016.91
23. Selmi, Z., Halima, M. B., Alimi, A. M.: Deep learning system for automatic license plate detection and recognition. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR). vol. 1, pp. 1132–1138. IEEE (2017) doi: 10.1109/ICDAR.2017.187

24. Tang, K., Cao, Y., Chen, C., Yao, J., Tan, C., Sun, J.: Dynamic origin-destination flow estimation using automatic vehicle identification data: A 3D convolutional neural network approach. vol. 36, pp. 30–46. Wiley Online Library (2021)
25. Tourani, A., Shahbahrami, A., Soroori, S., Khazaee, S., Suen, C. Y.: A robust deep learning approach for automatic iranian vehicle license plate detection and recognition for surveillance systems. IEEE Access, vol. 8, pp. 201317–201330 (2020) doi: 10.1109/ACCESS.2020.3035992
26. Vibha, L., Shenoy, P., Venugopal, K., Patnaik, L.: Moving vehicle identification using background registration technique for traffic surveillance. In: Proceedings of the International MultiConference of Engineers and Computer Scientists. vol. 1, pp. 19–21. IMECS (2018)
27. Wang, H., Yu, Y., Cai, Y., Chen, X., Chen, L., Liu, Q.: A comparative study of state-of-the-art Deep learning algorithms for vehicle detection. IEEE Intelligent Transportation Systems Magazine, vol. 11, no. 2, pp. 82–95 (2019) doi: 10.1109/MITS.2019.2903518
28. Yanes Pérez, M., Canto Valdés, L. R., López López, M.: La percepción de la inseguridad pública en Cunduacán, Tabasco. Península, vol. 17, no. 1 (2022)
29. Yang, W., Zhang, J., Wang, H., Zhang, Z.: A vehicle real-time detection algorithm based on YOLOv2 framework. In: Real-Time Image and Video Processing 2018. vol. 10670, pp. 106700N. International Society for Optics and Photonics (2018)
30. Yang, X., Tang, Y. Y., Luo, H.-W., Wu, T., Sun, L., Li, L.: One sample based feature learning for vehicle identification. In: 2016 International Conference on Machine Learning and Cybernetics (ICMLC). vol. 2, pp. 1049–1054 (2016) doi: 10.1109/ICMLC.2016.7873024
31. Zhang, B., Huang, S., Zhang, L., Liu, X., Song, G., Qin, J., Li, M.: Method of railway shunting operation sheet information extraction guided by table header. IET Intelligent Transport Systems, vol. 2022, no. 16 (2022) doi: 10.1049/itr2.12213

ISSN 1870-4069

Deep Learning-based Methods for Face and Text Detection in Natural Images

Marco López-Sánchez,Oscar Chávez-Bosquez, Betania Hernández-Ocaña, José Hernández-Torruco

Universidad Juárez Autónoma de Tabasco,
División Académica de Ciencias y Tecnologías de la Información,
Mexico

{marco.lopezsanchez, oscar.chavez,
betania.hernandez,jose.hernandezt}@ujat.mx

Abstract. The automatic detection of elements within an image has been the subject of numerous investigations in Computer vision. Detecting the objects making up an image and their relationship provides information that helps to interpret the scene's meaning. In this work, five methods based on Deep learning were evaluated, two for face detection and three for text detection. The methods for face detection are Dlib (Library for Machine Learning) and MTCNN (Multi-task cascaded convolutional neuronal networks). On the other hand, the evaluated methods for text detection are TesseractOCR, EasyOCR, and PaddleOCR. Results obtained with the evaluation indicate that the best face detection method was MTCNN and the best text detection method was EasyOCR. After analyzing the results, we propose a model based on MTCNN and EasyOCR to identify faces and texts in natural images simultaneously.

Keywords: Face detection, text detection, deep learning.

1 Introduction

The automatic detection of faces in natural images is one of the most studied topics in Computer vision [13]. Human faces are unique and cannot be reproduced, and they also provide information about human identity [25]. Detection is the first step for all facial analysis methods, such as facial recognition, face modeling, face verification, and face tracking. [19]. Facial detection is also used in the entertainment market (video games [26], virtual reality [6], and photo galleries [24]).

Regarding automatic text detection, several methods have been developed for text detection in natural images, becoming an active research field due to the growing demand for solutions to some artificial vision problems.

Detecting texts in natural scenes is a more challenging than detecting texts in scanned documents. Detecting the locations of the texts in the scene is

complicated because they are present in a scattered way, and in some cases, the appearance of the text makes it difficult to segment it.

In this work, two different Deep learning methods used for face detection are evaluated: (1) the Dlib [11], and (2) the MTCNN [10]. For text detection, three state-of-the-art methods are evaluated: (1) TesseractOCR [2], (2) EasyOCR [1], and (3) PaddleOCR [5]. For this evaluation, three subsets derived from the public data sets of Flickr8k [21], and COCO-Text [27] have been used.

The rest of the article is organized in the following order: Section 2 briefly reviews the background and state-of-the-art Deep learning-based methods used for face detection and text detection on natural images. Section 3 introduces the materials and methods. The experimental design is described in Section 4. Results and discussion are part of Section 5. Finally, conclusions are presented in Section 6.

2 State of the Art

Between the '70s and '80s, templates and measurements of geometric features were used to detect and recognize faces [17]. Early face detection efforts were primarily based on the traditional approach. The features were handcrafted from the image and introduced into a classifier to detect likely face regions. For this, two classic methods were used: the Histogram of Oriented Gradients (HOG) [4] and the HAAR Cascades classifier [28]. Despite the success of these methods, in recent years, models based on Deep learning have obtained outstanding results. In [3], they propose a face detector based on YOLOv3 [22], including a more accurate regression loss function and more appropriate anchor frames for the face. In [9], they propose a method based on Complete Discriminative Features (DCF) to improve face detection speed. This method uses a CNN that performs face detection directly on feature maps. Finally, Zhang et al.[32] proposed the FANet framework to build a detector that achieves high performance detecting faces with varied scales and features.

On the other side, traditional methods for text detection are primarily based on the discriminating characteristics of text areas within an image. These methods were divided into two approaches: component-based methods [8,16,12] and window-runner-based methods [18,14,29].

Deep learning methods for text detection have recently been used to achieve outstanding results. For example, in [15], a text detector called TextBoxes++ uses an end-to-end convolutional network that detects arbitrarily oriented scene text with high efficiency and accuracy. In [31], a novel text detector called TextField is designed to detect texts from irregular scenes; this detector was also trained with a fully convolutional neural network. This article [23] presents a model based on convolution neural networks to identify the language of the detected scene texts.

3 Materials and Methods

3.1 Face Detection Methods

Dlib It is a deep learning-based method created specifically for face detection in images. It is based on the histogram of oriented gradients (HOG) and convolutional neural networks (CNN). This model extracts facial reference points to calculate the orientation of a face in the scene [11]. It was trained with 68 facial reference points that provide information about the mouth, eyes, and nose.

MTCNN Acronym of *Multi-task cascaded convolutional neuronal networks*, it detects faces using a cascade of convolution neural networks divided into three stages: detect candidate face windows, discard candidates in which there are no faces, and identifies in which of the candidates a face exists [33].It works identifying the positions of five facial landmarks, one at each eye, another at the tip of the nose, and the remaining two at the corners of the lips.

3.2 Text Detection Methods

TesseractOCR It is an open-source text recognition engine[1]. It uses an LSTM neural network-based OCR engine and started as a research project in HP labs, using it in their line of scanners. Then, it was adopted by Google and made available to the public as an open source project. It supports various image formats such as PNG, JPEG and, TIFF, and can recognize more than 100 languages.

EasyOCR It is an open-source library used for text detection in images and supports more than 42 languages for detection purposes[2]. It has a default Deep learning architecture that uses three different types of neural networks [30].

PaddleOCR It is a framework that offers a series of pre-trained models with Recurrent neural networks (RNN) and CNNs[3]. It is based on the PaddlePaddle (*(PArallel Distributed Deep LEarning)*) framework. It is used for the detection, classification, and recognition of texts. It supports more than 80 languages.

3.3 Dataset

To carry out this research, we create 3 subsets of data from two different datasets:

Faces dataset This subset of data was used to compare the performance of face detection methods; thus, it consists of pictures including one or more faces. It is composed of 60 images that we selected from the Flickr8k dataset [4]. This dataset includes 245 faces distributed among 60 images.

[1] `https://tesseract-ocr.github.io`
[2] `https://github.com/JaidedAI/EasyOCR`
[3] `https://github.com/PaddlePaddle/PaddleOCR/tree/release/2.2`
[4] `https://www.kaggle.com/adityajn105/flickr8k?select=Images`

Text dataset The second subset is used to compare the performance of the text detection methods. It comprises 60 images including texts in different orientations, diverse sizes, and different fonts. This subset derives from the COCO-Text data set [27]. A total of 355 words are distributed among 60 images.

Face-text dataset This subset contains 40 images, only considering pictures where both faces and texts were found in the scene. These images were extracted from the public data set COCO-Text [27]. A total of 126 faces and 211 words are included in the 40 images.

3.4 Evaluation Metrics

Following evaluation metrics [20] compute the performance of the methods employing the following results:

- *TP*: True Positive is when the real value is 1 (True), and the predicted value is also 1 (True). It represents the recognized elements in the image (faces or words).
- *FP*: False Positive is when the real value is 0 (False), and the predicted value is 1 (True). It represents false identifications. It occurs when the detector identifies a region of the image as a face or text, but none of the elements are present.
- *FN*: False Negative is when the real value is 1 (True), and the predicted value is 0 (False). It represents the elements (faces or words) included in the image but not identified by the detector.

Precision: It is the number of items correctly identified as positive out of a total of items identified as positive.

$$precision = \frac{TP}{TP + FP}.$$

Recall: It is the proportion of positive cases correctly identified by the detector.

$$recall = \frac{TP}{TP + FN}.$$

F-Score: It combines the precision and recall measures to return a more general quality measure of the model. It is calculated as the harmonic mean of the metrics mentioned above.

$$F\text{-}Score = 2\frac{precision \cdot recall}{precision + recall}.$$

4 Experimental Design

Experiments were conducted using the Python programming language, including implementations of the methods for face detection (Dlib and MTCNN) and text detection (TesseractOCR, PaddleOCR, and EasyOCR).

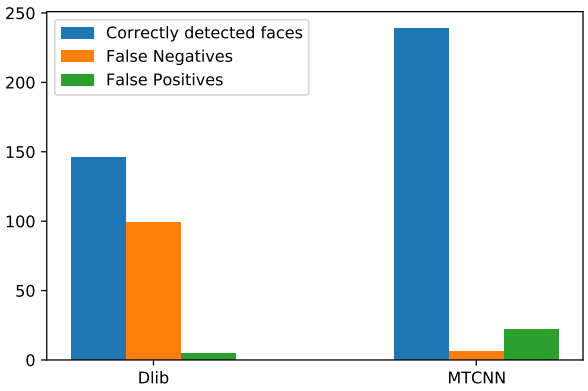


Fig. 1. Results obtained by Dlib and MTCNN in the Faces dataset.

Due the built-in support for implementing computer vision, we used the OpenCV [7] libraries. The following libraries were also used: `dlib 19.22.1`, `mtcnn 0.1.1`, `TesseractOCR 4.0.0`, `EasyOCR 1.6.2`, `PaddleOCR 2.6.0`, `Numpy 1.21.6` and `Matplotlib 3.2.2`. The default configuration of each detector was used.

Regarding the datasets, we performed manual labeling on each subset used in this work. Some faces found to be very blurred and difficult to identify by a human, so those images were not considered in the dataset. Also, incomplete or blurred words were not considered.

Three experiments were conducted to analyze the performance of the methods. In the first experiment, the methods for face detection were analyzed for all the elements of the Faces subset. In the second experiment, we analyze the three methods using all the elements of the Text dataset. In the third and last experiment, the proposed method was executed based on the methods that obtained the best performance. We use the Face-text dataset in this last experiment. All the experiments were conducted in Google Colab.

5 Results and Discussion

5.1 Face Detection Methods

Figure 1 shows the performance of the Dlib and MTCNN methods when evaluating the 60 images of the Faces dataset. We can notice that Dlib has a lower number of detected faces and a high number of False negatives, i.e., it could not detect 99 out of 245. On the other hand, MTCNN detected the most number of faces (239 out of 245), but it also detected 22 false positives (it detects faces where there are none).

Table 1. Face detection methods evaluation metrics.

Method	Precision	Recall	F-Score
Dlib	0.99	0.83	0.89
MTCNN	0.95	0.97	0.95

Table 2. Example of 3 images of the Faces dataset and corresponding results by Dlib and MTCNN.

Image	Total of faces	Dlib detection	MTCNN detection
	**4**	4	4
	**4**	2	4
	**3**	2	0

The best face detection method is highlighted in Table 1. Both Dlib and MTCNN methods were tested over the 60 images in the Faces dataset.

It should be noted that Dlib detected the fewest false positives, which is why it has the higher *precision*. However, it also detected the fewest true positives, which is why it has a lower recall. For this reason, the F-Score obtained by MTCNN is higher than that obtained by Dlib, indicating that MTCNN is a better detection method.

Table 2 shows 3 examples of the Faces dataset and the results obtained by the Dlib and MTCNN methods. We have included images with multiple faces, contrasting luminosity, and people in different scenes to test the face detection methods.

5.2 Text Detection Methods

Figure 2 shows the performance of the TesseractOCR, EasyOCR, and PaddleOCR methods when evaluating the 60 images of the Text dataset. We

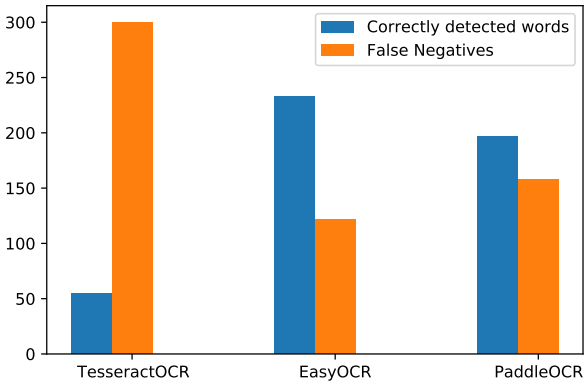


Fig. 2. Results obtained by TesseractOCR, EasyOCR, and PaddleOCR in the Faces dataset.

Table 3. Evaluation metrics results for the text detection methods.

Method	Precision	Recall	F-Score
TesseractOCR	1	0.54	0.70
EasyOCR	1	0.77	0.85
PaddleOCR	1	0.71	0.82

can notice that TesseractOCR has a lower performance, as it could only detect 55 out of 355 words. The best method was EasyOCR, detecting 233 out of 355 words, followed by PaddleOCR, with 197 out of 355 words.

The best text detection method is highlighted in Table 3. EasyOCR is the best of the three text detection methods evaluated with the Text dataset since the F-Score obtained exceeds the obtained by the other methods. None of the three methods detected false positives, so the result of the *precision* metric is 1. This means that 100 % of the words detected are found in the image, i.e., no false words are recognized in the scene. However, the *recall* metric indicates that EasyOCR indeed identifies more texts than the other methods, thus obtaining the highest *F-Score*.

Table 4 shows 3 samples of the Text dataset along with the results obtained by each text detection method. We notice that some words are not in a vertical orientation, yet EasyOCR obtained the best performance.

5.3 Proposed Face and Text Detection Method

The previous experiments allowed us to find the method with the best performance when identifying faces in natural images and the the best performance when identifying text in natural images to create a method to

Table 4. Example of 3 images of the Text dataset and corresponding results by TesseractOCR, EasyOCR, and PaddleOCR.

Image	Total of words	TesseractOCR detection	EasyOCR detection	PaddleOCR detection
	**4 words:** Clean Food Good Taste	0 words	2 words: - Food - Good	0 words
	**4 words:** WELCOME to our home	0 words	3 words: - to - our - home	3 words: - to - our - home
	**4 words:** Welcome to Kids Town	3 words: - Welcome - to - Kids	4 words: - Welcome - to - Kids - Town	0 words

Table 5. Result of the proposed method in the Face-text dataset.

Method	Precision	Recall	F-Score	Global score
MTCNN	1	0.71	0.82	0.84
EasyOCR	1	0.77	0.85	

detect both faces and texts in images. Ee implemented the EasyOCR method for text detection, and the MTCNN method for face detection. We tested our model with the Face-text dataset, using a threshold of 0.7. Table 5 shows the results where the overall *F-Score* (the average of the two methods) is highlighted.

Table 6 shows 3 examples from the Face-text dataset and their corresponding results. The images in the subset contain faces in different positions; likewise, the text appears in different orientations and is presented in different font types and colors. These conditions result in a challenge for detection models. However, both methods obtained acceptable results.

6 Conclusion and Future Work

In this work, 5 Deep learning methods were evaluated: 2 for detecting faces in images and 3 for detecting texts in images. The performance of the face and text detection methods was compared using data subsets derived from the publicly available Flickr8K and COCO-Text datasets.

Table 6. Example of text and face detection using our proposal.

Image	Number of faces	Number of words	Faces detected	Words detected
	**1**	**4 words:** PARIS DANI ALVES 32	1	1 word: - ALVES
	**3**	**7 words:** Cole WELCOME Harbour HOME OF SYDNEY CROSBY	3	4 words: - OF - CROSBY - Cole - SIDNEY - HOME
	**1**	**3 words:** FOR SALE GREEN	1	3 words: - GREEN - SALE - FOR

MTCNN obtained the best overall performance in face detection. it has the highest *recall* than Dlib, although the latter obtains better *precision*. We choose MTCNN because it detects more faces per image than Dlib.

On the other hand, EasyOCR obtained the best results; it can detect slated words and text in curved orientations. However, the PaddleOCR method was the method that detected the most considerable amount of horizontally oriented words. Therefore, we opted for the EasyOCR method because it detects text in different orientations.

Finally, We proposed a custom method for face and text detection adopting the best methods in each category (face detection and text detection), intending to have an efficient model that recognizes both faces and texts with the best possible performance.

Future work will try to apply our model to different applications (counting people at events or public transport), apply recognition of detected faces, or even automatically evaluate the emotions of a given person by analyzing their gestures.

Acknowledgments. To the Consejo Nacional de Ciencia y Tecnología (CONACYT) for supporting the Doctoral program in Computer Science at the Universidad Juárez Autónoma de Tabasco.

References

1. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9365–9374 (2019)
2. Breuel, T. M., Ul-Hasan, A., Al-Azawi, M. A., Shafait, F.: High-performance OCR for printed English and Fraktur using LSTM networks. In: 2013 12th international conference on document analysis and recognition. pp. 683–687. IEEE (2013)
3. Chen, W., Huang, H., Peng, S., Zhou, C., Zhang, C.: Yolo-face: a real-time face detector. The Visual Computer, vol. 37, no. 4, pp. 805–813 (2021)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol. 1, pp. 886–893. Ieee (2005)
5. Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., et al.: PP-OCR: A practical ultra lightweight OCR system. arXiv preprint arXiv:2009.09941, (2020)
6. Fuchter, S. K., Zucchi, S., Wortley, D.: Formative assessment of inquiry skills for responsible research and innovation using 3d virtual reality glasses and face recognition. In: Technology Enhanced Assessment: 21st International Conference, TEA 2018, Amsterdam, The Netherlands, December 10–11, 2018, Revised Selected Papers. vol. 1014, pp. 91. Springer (2019)
7. Gollapudi, S.: Learn computer vision using OpenCV. Springer (2019)
8. Greenhalgh, J., Mirmehdi, M.: Real-time detection and recognition of road traffic signs. IEEE transactions on intelligent transportation systems, vol. 13, no. 4, pp. 1498–1506 (2012)
9. Guo, G., Wang, H., Yan, Y., Zheng, J., Li, B.: A fast face detection method via convolutional neural network. Neurocomputing, vol. 395, pp. 128–137 (2020)
10. Jiang, B., Ren, Q., Dai, F., Xiong, J., Yang, J., Gui, G.: Multi-task cascaded convolutional neural networks for real-time dynamic face recognition method. In: International conference in communications, signal processing, and systems. pp. 59–66. Springer (2018)
11. King, D. E.: Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research, vol. 10, pp. 1755–1758 (2009)
12. Koo, H. I., Kim, D. H.: Scene text detection via connected component clustering and nontext filtering. IEEE transactions on image processing, vol. 22, no. 6, pp. 2296–2305 (2013)
13. Lal, M., Kumar, K., Arain, R. H., Maitlo, A., Ruk, S. A., Shaikh, H.: Study of face recognition techniques: A survey. International Journal of Advanced Computer Science and Applications, vol. 9, no. 6, pp. 42–49 (2018)
14. Lee, J.-J., Lee, P.-H., Lee, S.-W., Yuille, A., Koch, C.: Adaboost for text detection in natural scene. In: 2011 International conference on document analysis and recognition. pp. 429–434. IEEE (2011)
15. Liao, M., Shi, B., Bai, X.: Textboxes++: A single-shot oriented scene text detector. IEEE transactions on image processing, vol. 27, no. 8, pp. 3676–3690 (2018)
16. Mosleh, A., Bouguila, N., Hamza, A. B.: Image text detection using a bandlet-based edge detector and stroke width transform. In: BMVC. pp. 1–12 (2012)
17. Nixon, M.: Eye spacing measurement for facial recognition. In: Applications of digital image processing VIII. vol. 575, pp. 279–285. SPIE (1985)
18. Pan, Y.-F., Hou, X., Liu, C.-L.: A hybrid approach to detect and localize texts in natural scene images. IEEE transactions on image processing, vol. 20, no. 3, pp. 800–813 (2010)

19. Parekh, H. S., Thakore, D. G., Jaliya, U. K.: A survey on object detection and tracking methods. International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, no. 2, pp. 2970–2978 (2014)
20. Powers, D. M.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061, (2020)
21. Rashtchian, C., Young, P., Hodosh, M., Hockenmaier, J.: Collecting image annotations using Amazon's mechanical turk. In: Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk. pp. 139–147 (2010)
22. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, (2018)
23. Saha, S., Chakraborty, N., Kundu, S., Paul, S., Mollah, A. F., Basu, S., Sarkar, R.: Multi-lingual scene text detection and language identification. Pattern Recognition Letters, vol. 138, pp. 16–22 (2020)
24. Savchenko, A. V., Demochkin, K. V., Grechikhin, I. S.: Preference prediction based on a photo gallery analysis with scene recognition and object detection. Pattern Recognition, vol. 121, pp. 108248 (2022)
25. Simpson, E. A., Maylott, S. E., Leonard, K., Lazo, R. J., Jakobsen, K. V.: Face detection in infants and adults: Effects of orientation and color. Journal of experimental child psychology, vol. 186, pp. 17–32 (2019)
26. Solorzano Alcivar, N. I., Herrera Paltan, L. C., Lima Palacios, L. R., Paillacho Chiluiza, D. F., Paillacho Corredores, J. S.: Visual metrics for educational videogames linked to socially assistive robots in an inclusive education framework. In: Perspectives and Trends in Education and Technology, pp. 119–132. Springer (2022)
27. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: COCO-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140, (2016)
28. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. vol. 1, pp. I–I. Ieee (2001)
29. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International conference on computer vision. pp. 1457–1464. IEEE (2011)
30. Xiao, Z., Liang, P.: Chinese sentiment analysis using bidirectional lstm with word embedding. In: International Conference on Cloud Computing and Security. pp. 601–610. Springer (2016)
31. Xu, Y., Wang, Y., Zhou, W., Wang, Y., Yang, Z., Bai, X.: Textfield: Learning a deep direction field for irregular scene text detection. IEEE Transactions on Image Processing, vol. 28, no. 11, pp. 5566–5579 (2019)
32. Zhang, J., Wu, X., Hoi, S. C., Zhu, J.: Feature agglomeration networks for single stage face detection. Neurocomputing, vol. 380, pp. 180–189 (2020)
33. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, vol. 23, no. 10, pp. 1499–1503 (2016)

ISSN 1870-4069

Segmentación semántica en cultivos de jitomate usando ResNet50

Juan Pablo Guerra Ibarra[1], Francisco Javier Cuevas de la Rosa[1], Oziel Arellano Arzola[2]

[1] Centro de Investigaciones en Óptica A.C., México

[2] Instituto Tecnológico de Estudios Superiores de Zamora, México

{juangi,fjcuevas}@cio.mx, oziel.aa@zamora.tecnm.mx

Resumen. La aplicación de algoritmos de inteligencia artificial poco a poco van permeando en diferentes ámbitos de la vida cotidiana. Sirva de ejemplo, el uso de algoritmos de aprendizaje profundo han ido aumentado su presencia en diversos aspectos de las actividades de la sociedad en su día a día. Del mismo modo en las últimas dos décadas, la agricultura de precisión ha incrementando su impacto en los diferentes procesos de la labranza de la tierra, principalmente, en busca de aumentar su productividad mientras reduce su impacto al medio ambiente. La aplicación de de algoritmos de aprendizaje profundo en la agricultura, genera un área de estudio para el desarrollo de diferentes tecnologías, las cuales apoyan una actividad primordial para mantener y preferentemente aumentar la producción de alimentos, lo que aseguran la subsistencia presente y futura de la sociedad moderna. En el presente trabajo se utiliza una red neuronal convolucional, en específico el modelo $ResNet50$ para realizar la tarea de una segmentación semántica en imágenes de cultivos de jitomate en invernaderos, el objetivo es separar las hojas y frutos que estén presentes en la imagen. La segmentación de imágenes es la etapa inicial para el desarrollo de otros sistemas computacionales al eliminar el ruido presente y separar los elementos de interés, lo cual es básico para el desarrollo de mejores tecnologías aplicadas a los procesos agrícolas.

Palabras clave: Aprendizaje profundo, redes neuronales convolucionales, segmentación semántica, agricultura de precisión.

Semantic Segmentation in Tomato Crops Using ResNet50

Abstract. The application of artificial intelligence algorithms is gradually permeating various areas of everyday life. A clear example is the growing presence of deep learning algorithms in multiple aspects of

society's daily activities. Similarly, over the last two decades, precision agriculture has increased its impact on different farming processes, mainly aiming to enhance productivity while reducing its environmental impact. The use of deep learning algorithms in agriculture creates a research area for the development of different technologies that support a fundamental activity for maintaining—and ideally increasing—food production, ensuring the present and future subsistence of modern society. In this work, a convolutional neural network, specifically the ResNet50 model, is employed to perform semantic segmentation tasks on images of tomato crops in greenhouses. The goal is to separate the leaves and fruits present in the image. Image segmentation is the initial stage for developing other computational systems, as it eliminates noise and separates the elements of interest, which is essential for building better technologies applied to agricultural processes.

Keywords: Deep learning, convolutional neural networks, semantic segmentation, precision agriculture.

1. Introducción

La agricultura de precisión engloba un conjunto de tecnologías, las cuales combinan sensores, estadística, algoritmos clásicos, algoritmos inteligentes entre otros, con el objetivo de optimizar la producción de alimentos, tanto en cantidad como en calidad, esto se logra dando seguimiento al crecimiento de los cultivos y reduciendo costos [11,30,28]. Para lograr incrementar la producción de alimentos provenientes de los campos de cultivos, es importante realizar de manera oportuna y eficiente la detección de las necesidades que pudieran tener las plantas, algunas éstas pueden ser referentes a proceso de fertilización, riego é iluminación, por mencionar algunas.

La detección oportuna de las necesidades nutrimentales é hídricas es de gran relevancia en la agricultura protegida, ya que el riego y nutrición dependen de la eficiencia para determinar dichas necesidades [31,40,36]. Buscando apoyar la detección eficiente de necesidades en diferentes tipos de cultivos, se han realizado diferentes trabajos con técnicas de segmentación basadas en umbrales en formato de color RGB o en algún otro sistema de representación del mismo [8,42,32,7,15,37,29]. Los procedimientos de segmentación mencionados en los artículos [8,42,32,7,15,37,29] han reportado resultados exitosos en sus experimentos. Los trabajos citados tienen en común, el que las imágenes con las que se realizan los procesos de segmentado fueron tomadas en condiciones controladas; haciendo énfasis en controlar el fondo que se aprecia en las imágenes, con la variable de fondo controlada, se facilita en gran medida la determinación de los umbrales adecuados para llevar a cabo la separación o segmentación de los píxeles asignándolos a una clase particular.

En años recientes la rama de la inteligencia artificial llamada aprendizaje máquina, ha sido probada en diferentes espacios de la vida moderna, una de las actividades donde más impacto puede tener es en la agricultura de precisión

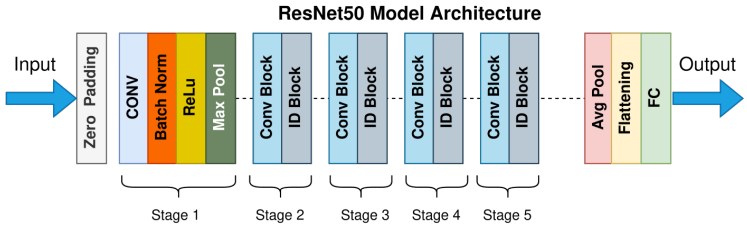


Fig. 1. Estructura de la RNC $ResNet50$.

[4,26,10,23], los algoritmos de aprendizaje profundo son los que mayor relevancia han tenido.

Los algoritmos de aprendizaje profundo han surgido como una alternativa para procesar grandes cantidades de información heterogénea, lo que facilita la tarea de realizar predicciones y clasificaciones complejas en ambientes agrícolas; actualmente existe una amplia diversidad o gama de sensores que generan grandes volúmenes de información, como lo son imágenes provenientes de diferentes tipos de dispositivos de captura, desde cámaras de teléfonos celulares hasta drones especializados en agricultura de precisión [28,35,16,6,25]. En la segmentación de imágenes adquiridas en ambientes agrícolas no controlados, con la finalidad de separar los elementos de estudio de interés del resto de la imagen, es una de las áreas donde se pueden explotar las ventajas de los algoritmos de aprendizaje profundo.

La segmentación semántica es la asignación de una clase o etiqueta a cada píxel que forman una imagen [12,22]. Llevar a cabo el proceso de etiquetado o marcado de los píxeles de una imagen mediante aprendizaje profundo ha tenido un crecimiento significativo en los años recientes [35]. Técnicas o herramientas de aprendizaje profundo que se ha empleado con éxito, son las redes neuronales convolucionales RNC [5,45,44], el éxito de este tipo de arquitectura se funda en su invariancia ante traslaciones y cambio de escalas en las entradas proporcionadas a los modelos de RNC, lo cual es excelente en problemas de alto nivel de ruido.

En la presente investigación se explora el uso de una RNC, en particular la $ResNet50$ [13] (ver Figura 1 tomada de [2]), para desarrollar un segmentador semántico, el cual separe los píxeles que conforman las hojas y frutos de cultivo de jitomate en imágenes tomadas en invernaderos. En la Figura 2 se ve un ejemplo de las imágenes a segmentar.

2. Metodología

2.1. Redes neuronales convolucionales (RNC)

Las RNC son algoritmos de aprendizaje profundo que procesan imágenes y entregan información espacial de las mismas [43]. Las redes neuronales simulan

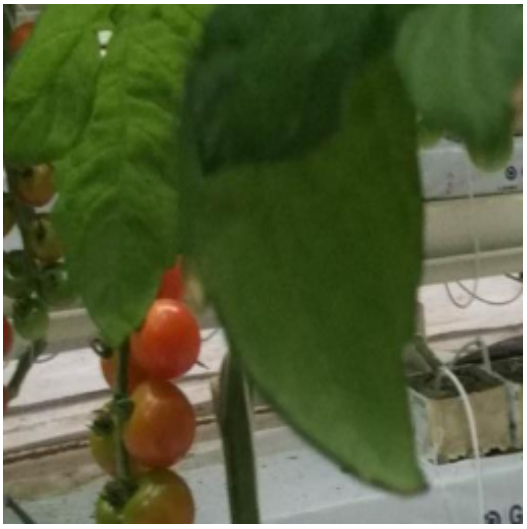


Fig. 2. Ejemplo de imagen a segmentar.

el sentido de la vista de los seres vivos, han sido desarrollados diferentes modelos con buenos resultados en las últimas dos décadas.

Los modelos desarrollados de RNC tienen en común que, en cada uno de ellos la cantidad de parámentros entrenables ha aumentado y la aparición de diferentes funciones de activación han posibilitado su uso en diferentes ámbitos y contextos, reportando buenos resultados en las tareas en que fueron probadas. En 1998 $Lenet-5$ [19] es presentada con 60 mil parámetros, presenta por primera vez los kernel de convoluciones como funciones de activación y la agrupación de capas. En 2012 $AlexNet$ [18] se presentó con 60 millones de parámetros e introdujo al mundo de las RNC las funciones de activación $ReLUs$ y $Dropout$. VGG-16 [34] hace su aparición en el 2014 con 138 millones de parámetros, en el mismo año $Inception-v1$ [38] que fue la primera red que uso bloques de capas. Para el año 2015 el modelo $ResNet50$ [14], con 44 millones de parámetros y popularizó la forma de realizar las conexiones actuales entre capas al disminuir el desvanecimiento del gradiente, logró 152 capas sin perder generalización en el descenso del gradiente y es de los modelos pioneros en usar "batch normalization".

El modelo de RNC $ResNet50$ tiene gran relevancia en el mundo del aprendizaje profundo, debido a su capacidad de entrenar redes con gran profundidad en lo referente al número de capas interconectas [13], esto ha permitido ser aplicada en el de reconocimiento de imágenes [33], detección de objetos [24], reconocimiento de rostros [21,20,39] y clasificación de imágenes [21]. En el ámbito agrícola la $ResNet50$ para la detección de plagas [9], detección de déficit nutrimentales en plantas [41].

El núcleo de este trabajo recae en una $RNC\ ResNet50$ que es la encargada de realizar la tarea de segmentado semántico en imágenes de cultivos de jitomate. Las etapas para implementar el segmentador semántico mediante la $RNC\ ResNet50$ para separar las hojas y frutos en imágenes con cultivos de jitomate, se muestran en la Figura 3. Cuenta con 6 etapas que son descritas a continuación:

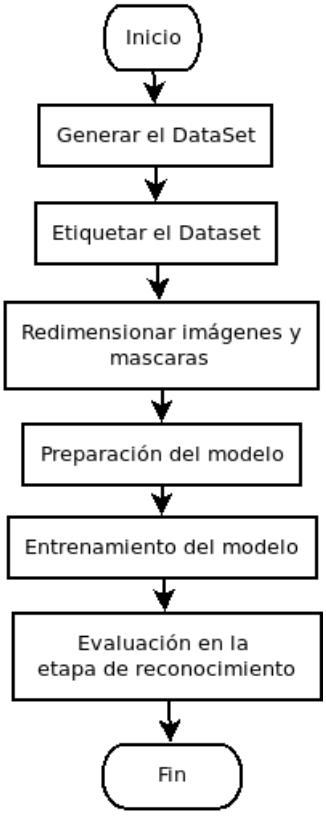


Fig. 3. Metodología de implementación del segmentador.

1. Dataset.
   Consta de 500 imágenes (ver ejemplo de las imágenes del dataset en la Figura 2), las que pertenecen a un dataset accesible para su descarga desde Internet [3]. Las imágenes del dataset tienen las siguientes características:formato de los archivos PNG, dimensiones de las imágenes (NxM): $400x500$ y $500x400$ píxeles. Del conjunto de imágenes del dataset se crearon dos subconjuntos, de entrenamiento y prueba; para ello se realizó una asignación aleatoria de las 500 imágenes. El conjunto de entrenamiento quedo conformado por 350

Fig. 4. Etiquetado manual de la Figura 2.

imágenes, lo que corresponde al 70 % del dataset, mientras que el conjunto de prueba está integrado por las 150 imágenes restantes, lo que corresponde al 30 % de las 500 imágenes que conforman el dataset. Los porcentajes mencionados fueron determinados por la convención usada de 60 % para el conjunto de entrenamiento, el restante 40 % para prueba y validación. Al no tener un conjunto de validación se repartió equitativamente en los conjuntos de entrenamiento y prueba.

2. Etiquetado manual de la imágenes del dataset.

   Para poder entrenar el segmentador semántico por medio de una *RNC ResNet*50, es necesario marcar o etiquetar los elementos que deseamos separar en las imágenes de los conjuntos de entrenamiento y de prueba, en nuestro caso: las hojas y frutos de cultivos de jitomate. El etiquetado de las imágenes se realizó con la herramienta "Computer Vision Annotation Tool" (CVAT) [1], la cual es una herramienta web gratuita. En CVAT se pueden realizar diferentes tipos de marcados o etiquetados para trabajar con diferentes algoritmos de aprendizaje profundo. En la Figura 4 se observa el etiquetado de los píxeles que corresponden a las hojas y frutos de la Figura 2, es importante mencionar que se realizó el etiquetado de las 500 imágenes del dataset, generando una máscara para cada imagen.

3. Redimensionar imágenes y máscaras.

   Para poder realizar el entrenamiento del modelo de la *RNC ResNet*50 es necesario transformar las imágenes y sus respectivas máscaras al tamaño de las entradas de nuestro modelo propuesto a utilizar.

4. Preparación del modelo.

   Se establecen las dimensiones de las entradas de nuestro modelo y la cantidad de clases a segmentar, para el modelo propuesto serán de $256x256$ las imágenes de entrada del conjunto de entrenamiento y clasificará 3 clases: hojas, frutos y fondo.

5. Entrenamiento del modelo de *RNC ResNet*50.

```
dim=(256,256)
img = cv2.imread(ruta_img)
imgr = cv2.resize(img, dim)
cv2.imwrite(archivo_destino,imgr)
```

Fig. 5. Ajuste de tamaño de imagen.

Se procede a realizar el entrenamiento del modelo propuesto con las 350 imágenes del conjunto de entrenamiento tomadas de aleatoriamente de las 500 imágenes del dataset..

6. Evaluación en la etapa de reconocimiento.
   Con las imágenes del conjunto de prueba se evalúa la segmentación realizada por el modelo $RNC\ ResNet50$ para medir su eficiencia en la tarea de separar las hojas y frutos de cultivos de jitomate.

3. Implementación

La implementación del modelo de $RNC\ ResNet50$ se realizó con el lenguaje de programación de propósito general "Python", se usaron las librerías de keras [17] (https://keras.io/) para implementar el modelo de aprendizaje profundo y OpenCV [27] (https://opencv.org/) para la manipulación de las imágenes de los conjuntos de entrenamiento y prueba. Las líneas de código principales implementadas para la realización del segmentador semántico se describen a continuación:

1. Redimensionar imágenes del dataset y sus máscaras.
   En la Figura 5 se muestran las líneas de código necesarias para cambiar las dimensiones de las imágenes de los conjuntos de entrenamiento y prueba a un tamaño especifico, en este caso es de $256x256$ píxeles. La tercera linea de código es la instrucción "cv2.resize" de "OpenCV',' la cual recibe dos parámetros, el primero es la imagen a cambiar de tamaño y mientras que el segundo son las nuevas dimensiones que tendrá la imagen. Es necesario cambiar el tamaño de las imágenes para que coincidan con las dimensiones de la $RNC\ ResNet50$.
2. Binarizado de máscaras de las imágenes del dataset.
   El binarizado de las máscaras es necesario para realizar el proceso de entrenamiento del modelo $RNC\ ResNet50$. El proceso de binarizado para las máscaras del conjunto de entrenamiento similares a la Figura 4 se realiza por medio del código que se muestra en la Figura 6.
3. Preparación del modelo de aprendizaje profundo de la $ResNet50$.
   Para implementar el modelo de $RNC\ ResNet50$ es necesario establecer los parámetros de configuración básicos, como lo son: 'n_classes", en este parámetro almacena la cantidad de clases a segmentar, 3 en este caso. Los parámetros dos y tres corresponden a las dimensiones de las imágenes con

```
img = cv2.imread(ruta_img)
mascara = np.zeros((ancho,alto,1))
for y in range(img.shape[0]):
      for x in range (img.shape[1]):
                if (img[y][x][1]>200):
                    mascara[y][x]=1
                else:
                    if (img[y][x][2]>200):
                        mascara[y][x]=2
cv2.imwrite(archivo_destino_mascara,mascara)
```

Fig. 6. Código de binarizado de máscaras.

las que se entrenará el modelo RNC, dichos parámetros son asignados a "input_height" e "input_width".
La generación del modelo de RNC $ResNet50$ se observa en el código de la Figura 7.

```
from keras_segmentation.models.unet import resnet50_unet

model = resnet50_unet(
n_classes=3 ,
input_height=256,
input_width=256  )
```

Fig. 7. Generación del modelo $ResNet50$.

La Figura 8 muestra la adaptación de la RNC $ResNet50$ para realizar el objetivo de segmentar las hojas y frutos en imágenes de cultivos de jitomate (ver Figura 2 para un ejemplo de las imágenes). La Figura 8 es generada usando la librería "visualkeras", las líneas de código para generar la Figura 8 se muestra en la Figura 9.

4. Entrenamiento del modelo.
   Para realizar el entrenamiento del modelo generado en la Figura 7, se requiere asignar los parámetros: 'train_images", se le asigna la ruta en donde se localizan las imágenes del conjunto de entrenamiento. El parámetro "train_annotations" le es asignada la ruta de las máscaras de las imágenes del conjunto de entrenamiento. El tercer parámetro "checkpoints_path" se le asigna la ruta donde se almacenan los pesos después de cada iteración o época. El cuarto parámetro "epochs" le es asignado el número de épocas que durará el entrenamiento.
5. Evaluación del conjunto de imágenes de prueba.
   En la Figura 11 se muestra la línea de código para realizar el proceso de segmentado de la imagen que se quiera separar sus hojas y frutos de cultivos de jitomate. Para realizar el proceso de segmentado semántico utilizado la RNC $ResNet50$, es necesario especificar en el parámetro "inp" la ruta a la imagen a segmentar y en el parámetro "out_fname" la ruta

Fig. 8. Modelo generado para el segmentador semántico.

```
import visualkeras
visualkeras.layered_view(model, to_file='output.png',
                                    legend=True,)
```

Fig. 9. Código para generar imagen del modelo $ResNet50$ de la Figura 8.

donde se almacena la imagen con los resultados del proceso de segmentación semántica.

4. Resultados

Los bloques de código mostrados en la etapa de **Implementación** desde la Figura 5 hasta la Figura 11, generan imágenes en las cuales se aprecian la segmentación semántica que se realizó por medio del modelo de RNC $ResNet50$ de las hojas y frutos de las plantas de jitomate.

El modelo tuvo en eficiencia de 98 % en las etapa de entrenamiento y un 87 % en fase de prueba, dichos porcentajes corresponden a la métrica "Accuracy", observe la la Figura 12.

Las Figuras 13, 14 y 15 muestran el resultado de la segmentación realizada por el modelo descrito anteriormente en la sección de **Implementación**. En cada Figura se pueden apreciar 4 imágenes, la primera es la toma original del cultivo de jitomate, la segunda es la máscara de dicha toma, la tercera y cuarta es la segmentación semántica de las hojas y frutos respectivamente.

5. Conclusiones y trabajos futuros

El presente articulo reporta la primera etapa de un proyecto de agricultura de precisión, que es la segmentación de hojas y frutos en imágenes de cultivos de jitomate, de un sistema generador de soluciones nutritivas para el cultivo de la fruta antes mencionada. El modelo de RNC $ResNet50$ descrito en las cuartillas anteriores, al tener un 87 % de eficacia con las imágenes del conjunto de pruebas, se considera como exitoso al momento realizar la segmentación semántica de las hojas y frutos de cultivos de jitomate. Un aspecto importante a considerar al momento de implementar una RNC para llevar a cabo un proceso de segmentado semántico, es la cantidad de tiempo que se requiere invertir para realizar el etiquetado de las imágenes. Otro factor a tomar en cuenta, es la poder de computo necesario para realizar el entrenamiento del modelo, ya que se requiere de tarjetas de vídeo para poder reducir los tiempos de entrenamiento.

```
model.train(
train_images =  "ruta_imagen_entrenamiento",
train_annotations = "ruta_etiquetado_entrenamiento",
checkpoints_path = "ruta_pesos_intermedios",
epochs=25
)
```

Fig. 10. Código para realizar el entrenamiento del modelo.

```
out = model.predict_segmentation(
inp=origen,
out_fname=destino)
```

Fig. 11. Evaluación de imágenes.

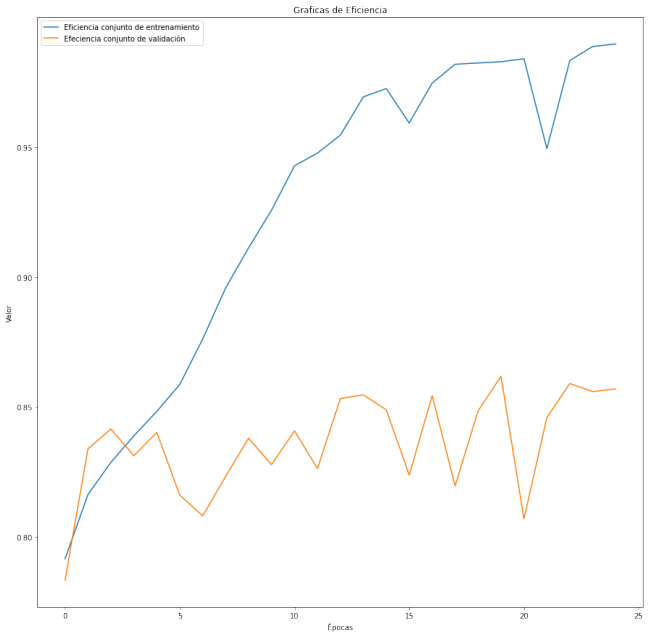
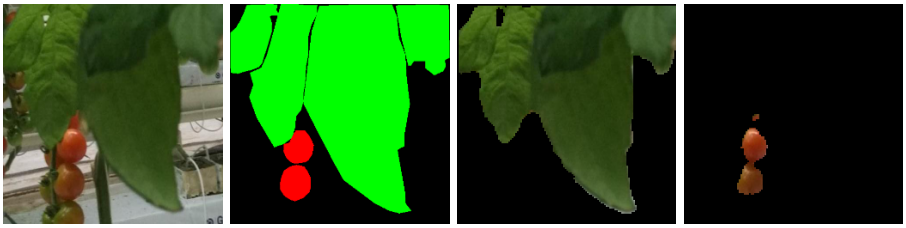


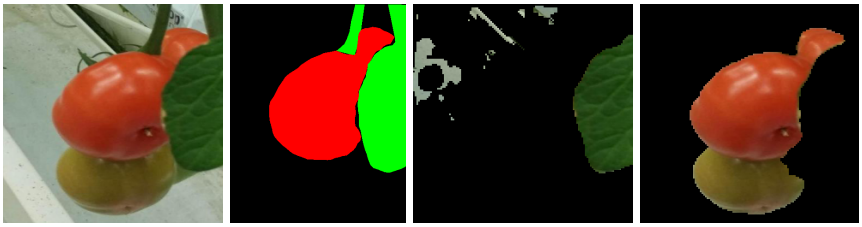
Fig. 12. Métricas de rendimiento en el entrenamiento "accuracy".

Las vertientes del trabajo futuro se dividen en: mejorar el rendimiento del segmentador semántico, llevar a cabo una comparativa de los resultados del segmentador semántico contra algún proceso de segmentación que implemente una técnica diferente y el desarrollo de un sistema use algún algoritmo heurístico para la creación de solución nutritiva que solventé la carencia detectada en las hojas o frutos segmentados.



(a) (b) (c) (d)

Fig. 13. Resultado del proceso de segmentación semántica ejemplo 1. a) Imagen Original b) máscara de la imagen c) Segmentación semántica de hojas d) Segmentación semántica de frutos



(a) (b) (c) (d)

Fig. 14. Resultado del proceso de segmentación semántica ejemplo 2. a) Imagen Original. b) máscara de la imagen. c) Segmentación semántica de hojas. d) Segmentación semántica de frutos.

(a) (b) (c) (d)

Fig. 15. Resultado del proceso de segmentación semántica ejemplo 3.
a) Imagen Original. b) máscara de la imagen. c) Segmentación semántica de hojas. d) Segmentación semántica de frutos.

Para mejorar el desempeño del segmentador semántico mostrado en la Figura 12, se probará con las siguientes alternativas: aumentar la cantidad de épocas de entrenamiento, incrementar el número de imágenes en el conjunto de entrenamiento y mejorar el etiquetado de las imágenes del dataset.

En la segunda vertiente de trabajo se desarrollará un segmentador de hojas y frutos de cultivos de jitomates basado en umbrales con algún modelo de representación de color, como lo son el RGB y HSV, con los resultados de ambas segmentaciones, la semántica y por umbrales se realizará una comparativa de los resultados.

La tercera vertiente de trabajo, pretende con los resultados de la segmentación de las hojas y frutos de cultivos de jitomates se alimentará un detector de déficit de nutrientes, para posteriormente proponer una solución nutritiva que corrija el déficit detectado.

Agradecimientos. Es importante agradecer a las Instituciones y empresas que hicieron posible el desarrollo del presente trabajo al destinar recursos de diferentes índoles para la realización de de la misma:

- Consejo Nacional de Ciencia y Tecnología (CONACYT).
- Centro de Investigaciones en Óptica A.C.
- Instituto Tecnológico de Estudios Superiores de Zamora.
- Empresa Opus Farms.

Referencias

1. CVAT, https://www.cvat.ai/
2. File:ResNet50.png - Wikimedia Commons, https://commons.wikimedia.org/wiki/File:ResNet50.png
3. Tomato Detection — Kaggle, https://www.kaggle.com/datasets/andrewmvd/tomato-detection
4. Aruul Mozhi Varman, S., Baskaran, A. R., Aravindh, S., Prabhu, E.: Deep Learning and IoT for Smart Agriculture Using WSN. 2017 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2017, pp. 1–6 (2018) doi: 10.1109/ICCIC.2017.8524140
5. Badrinarayanan, V., Kendall, A., Cipolla, R., Badrinarayanan, V., Cipolla, R., Zhuang, J., Yang, J., Gu, L., Dvornek, N.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2481–2495 (2017) doi: 10.5244/c.31.57
6. Bhargava, A., Bansal, A.: Fruits and vegetables quality evaluation using computer vision: A review. Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 3, pp. 243–257 (2021) doi: 10.1016/j.jksuci.2018.06.002
7. Chen, L., Lin, L., Cai, G., Sun, Y., Huang, T., Wang, K., Deng, J.: Identification of nitrogen, phosphorus, and potassium deficiencies in rice based on static scanning technology and hierarchical identification method. PLoS ONE, vol. 9, no. 11, pp. 1–17 (2014) doi: 10.1371/journal.pone.0113200

8. Cilia, C., Panigada, C., Rossini, M., Meroni, M., Busetto, L., Amaducci, S., Boschetti, M., Picchi, V., Colombo, R.: Nitrogen status assessment for variable rate fertilization in maize through hyperspectral imagery. Remote Sensing, vol. 6, no. 7, pp. 6549–6565 (2014) doi: 10.3390/rs6076549
9. Fuentes, A., Yoon, S., Kim, S. C., Park, D. S.: A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors (Switzerland), vol. 17, no. 9 (2017) doi: 10.3390/s17092022
10. Garcia-Lamont, F., Cervantes, J., López, A., Rodriguez, L.: Segmentation of images by color features: A survey. Neurocomputing, vol. 292, pp. 1–27 (5 2018) doi: 10.1016/j.neucom.2018.01.091
11. Gebbers, R., Adamchuk, V. I.: Precision agriculture and food security. Science, vol. 327, no. 5967, pp. 828–831 (2010) doi: 10.1126/science.1183899
12. Hao, S., Zhou, Y., Guo, Y.: A Brief Survey on Semantic Segmentation with Deep Learning. Neurocomputing, vol. 406, pp. 302–321 (2020) doi: 10.1016/j.neucom.2019.11.118
13. He, K.: Deep Residual Learning for Image Recognition. Indian Journal of Chemistry - Section B Organic and Medicinal Chemistry, vol. 45, no. 8, pp. 1951–1954 (2006) doi: 10.1002/chin.200650130
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 2016-Decem, pp. 770–778. IEEE Computer Society (12 2016) doi: 10.1109/CVPR.2016.90
15. Jia, L., Chen, X., Zhang, F., Buerkert, A., Roemheld, V.: Optimum nitrogen fertilization of winter wheat based on color digital camera images. Communications in Soil Science and Plant Analysis, vol. 38, no. 11-12, pp. 1385–1394 (2007) doi: 10.1080/00103620701375991
16. Kamilaris, A., Prenafeta-Boldú, F. X.: Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, vol. 147, no. July 2017, pp. 70–90 (4 2018) doi: 10.1016/j.compag.2018.02.016
17. KERAS: Keras API reference / Optimizers / RMSprop (2021), https://keras.io/api/
18. Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet classification with deep convolutional neural networks. Communications of the ACM, vol. 60, no. 6, pp. 84–90 (5 2017) doi: 10.1145/3065386
19. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2323 (1998) doi: 10.1109/5.726791
20. Lu, Z., Jiang, X., Kot, A.: Deep Coupled ResNet for Low-Resolution Face Recognition. IEEE Signal Processing Letters, vol. 25, no. 4, pp. 526–530 (4 2018) doi: 10.1109/LSP.2018.2810121
21. Mandal, B., Okeukwu, A., Theis, Y.: Masked Face Recognition using ResNet-50. CoRR, vol. abs/2104.0 (4 2021)
22. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D.: Image Segmentation Using Deep Learning: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 7, pp. 3523–3542 (7 2022) doi: 10.1109/TPAMI.2021.3059968
23. Mohanty, S. P., Hughes, D., Salathe, M.: Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, vol. 7, pp. 1–10 (4 2016) doi: 10.3389/fpls.2016.01419

24. Mukti, I. Z., Biswas, D.: Transfer Learning Based Plant Diseases Detection Using ResNet50. In: 2019 4th International Conference on Electrical Information and Communication Technology, EICT 2019. Institute of Electrical and Electronics Engineers Inc. (12 2019) doi: 10.1109/EICT48899.2019.9068805
25. Nanehkaran, Y. A., Zhang, D., Chen, J., Tian, Y., Al-Nabhan, N.: Recognition of plant leaf diseases based on computer vision. Journal of Ambient Intelligence and Humanized Computing, , no. 0123456789 (2020) doi: 10.1007/s12652-020-02505-x
26. Nyalala, I., Okinda, C., Nyalala, L., Makange, N., Chao, Q., Chao, L., Yousaf, K., Chen, K.: Tomato volume and mass estimation using computer vision and machine learning algorithms: Cherry tomato model. Journal of Food Engineering, vol. 263, no. July, pp. 288–298 (2019) doi: 10.1016/j.jfoodeng.2019.07.012
27. OpenCV: Home - OpenCV, https://opencv.org/
28. Patrício, D. I., Rieder, R.: Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. Computers and Electronics in Agriculture, vol. 153, pp. 69–81 (10 2018) doi: 10.1016/j.compag.2018.08.001
29. Philipp, I., Rath, T.: Improving plant discrimination in image processing by use of different colour space transformations. Computers and Electronics in Agriculture, vol. 35, no. 1, pp. 1–15 (2002) doi: 10.1016/S0168-1699(02)00050-9
30. Pierce, F. J., Nowak, P.: Aspects of precision agriculture. advances in Agronomy, vol. 67, pp. 1–68 (1999)
31. Pokrajac, D., Obradovic, Z.: Neural network-based software for fertilizer optimization in precision farming. Proceedings of the International Joint Conference on Neural Networks, vol. 3, no. I, pp. 2110–2115 (2001) doi: 10.1109/ijcnn.2001.938492
32. Rorie, R. L., Purcell, L. C., Karcher, D. E., King, C. A.: The assessment of leaf nitrogen in corn from digital images. Crop Science, vol. 51, no. 5, pp. 2174–2180 (2011) doi: 10.2135/cropsci2010.12.0699
33. Sai Bharadwaj Reddy, D. Sujitha Juliet: Transfer Learning with ResNet-50 for Malaria Cell-Image Classification. Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019, pp. 945–949 (2019)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (9 2015), http://arxiv.org/abs/1409.1556
35. Singh, A. A. K. A., Ganapathysubramanian, B., Sarkar, S., Singh, A. A. K. A.: Deep Learning for Plant Stress Phenotyping: Trends and Future Perspectives. Trends in Plant Science, vol. 23, no. 10, pp. 883–898 (2018) doi: 10.1016/j.tplants.2018.07.004
36. Smith, R., Baillie, J., McCarthy, A., Raine, S. R., Baillie, C. P.: Review of precision irrigation technologies and their application. National Centre for Engineering in Agriculture University of Southern Queensland Toowoomba, , no. November (2010)
37. Story, D., Kacira, M., Kubota, C., Akoglu, A., An, L.: Lettuce calcium deficiency detection with machine vision computed plant features in controlled environments. Computers and Electronics in Agriculture, vol. 74, no. 2, pp. 238–243 (2010) doi: 10.1016/j.compag.2010.08.010
38. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2016-Decem, pp. 2818–2826 (12 2015) doi: 10.1109/CVPR.2016.308

39. Theckedath, D., Sedamkar, R. R.: Detecting Affect States Using VGG16, ResNet50 and SE-ResNet50 Networks. SN Computer Science, vol. 1, no. 2, pp. 1–7 (2020) doi: 10.1007/s42979-020-0114-9
40. Tian, H., Wang, T., Liu, Y., Qiao, X., Li, Y.: Computer vision technology in agricultural automation —A review. Information Processing in Agriculture, vol. 7, no. 1, pp. 1–19 (2020) doi: 10.1016/j.inpa.2019.09.006
41. Tran, T. T., Choi, J. W., Le, T. T. H., Kim, J. W.: A comparative study of deep CNN in forecasting and classifying the macronutrient deficiencies on development of tomato plant. Applied Sciences (Switzerland), vol. 9, no. 8 (2019) doi: 10.3390/app9081601
42. Wang, Y., Wang, D., Zhang, G., Wang, J.: Estimating nitrogen status of rice using the image segmentation of G-R thresholding method. Field Crops Research, vol. 149, pp. 33–39 (2013) doi: 10.1016/j.fcr.2013.04.007
43. Wulandhari, L. A., Gunawan, A. A. S., Qurania, A., Harsani, P., Triastinurmiatiningsih, Tarawan, F., Hermawan, R. F.: Plant nutrient deficiency detection using deep convolutional neural network. ICIC Express Letters, vol. 13, no. 10, pp. 971–977 (2019) doi: 10.24507/icicel.13.10.971
44. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: BiSeNet: Bilateral segmentation network for real-time semantic segmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11217 LNCS, pp. 334–349 (2018) doi: 10.1007/978-3-030-01261-8{_}20
45. Zhuang, J., Yang, J., Gu, L., Dvornek, N.: Fully Convolutional Networks for Semantic Segmentation. Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019, pp. 847–856 (2019) doi: 10.1109/ICCVW.2019.00113

ISSN 1870-4069

Three Metrics for Building Association Rules Using Differential Evolution for Bacterial Vaginosis

Freddy García-Fuentes[1], Juana Canul-Reich[1], Rafael Rivera-López[2], Efrén Mezura-Montes[3], Erick De-La-Cruz-Hernández[4]

[1] Universidad Juárez Autónoma de Tabasco, DACYTI, Mexico

[2] Instituto Tecnológico de Veracruz, Mexico

[3] Universidad Veracruzana, IIIA, Mexico

[4] Universidad Juárez Autónoma de Tabasco, DAMC, Mexico

211H18004@alumno.ujat.mx,juana.canul@ujat.mx, rafael.rl@veracruz.tecnm.mx, emezura@uv.mx, erick.delacruz@ujat.mx

Abstract. This paper proposes a Differential-Evolution-based approach to building association rules describing the relationship between the elements triggering bacterial vaginosis. The differential evolution algorithm uses a population of real-valued vectors as candidate solutions representing an association rule, which is evaluated on the clinical dataset using a fitness function. This function includes three metrics to measure the quality rule to generalize the dataset. The dataset has binary attributes representing the absence or presence of the bacteria. The resulting rules indicate that this approach can build rules with biological significance.

Keywords: Bacterial vaginosis, differential evolution, association rules.

1 Introduction

Bacterial vaginosis (BV) is a vaginal infection characterized by a bacterial disorder due to a change in the vaginal flora. The anaerobic pathogens that increase concentration by 10 to 100 times are species of *Prevotella and Peptostreptococcus, Gardnerella vaginalis, Mobiluncus spp, Bacteroides spp, Peptostreptococcus spp, Urea-plasma urealyticum, and Mycoplasma hominis* [4]. BV is a silent health problem since the symptoms can go unnoticed, causing severe consequences such as premature delivery, post-abortion infection, pelvic inflammatory disease, and sexually transmitted diseases [8].

Data Mining is an exciting Artificial Intelligence area that allows for information analysis and knowledge discovery. Data Mining techniques such as association rule mining (AR) allow identifying correlation, association, and frequent patterns from a dataset [6]. In particular, the Apriori algorithm [14] is one of the most widely used algorithms for pattern discovery, using frequent itemsets to generate association rules [3]. A disadvantage of the Apriori algorithm is the combinatorial exploitation of the rules produced, so applying techniques to obtain a reduced set of high-quality rules is essential [7]. To address this problem, we propose using a Differential-Evolution-based approach to create meaningful association rules of high quality. The Differential Evolution (DE) algorithm is an effective method used in numerical optimization, which both explores and exploits a set of solutions in a continuous space based on an intelligent and robust combination scheme [1].

The rest of this manuscript is organized as follows: Section 2 describes the dataset and the methods used in this work. The elements used to implement the proposed method are detailed in Section 3, and Section 5 shows the preliminary results. Finally, conclusions and future work are described in Section 6.

2 Materials and Methods

2.1 Data

The dataset used in this work includes information from Mexican sexually active women between 18 and 50 years old. They underwent a gynecological examination at the Metabolic and Infectious Diseases Research Laboratory of the Universidad Juárez Autónoma de Tabasco [12]. The dataset comprises 201 observations and 11 attributes that led to a BV presence or absence diagnosis. Attributes used describes four lactoballici (*crispatus*, *gasseri*, *iners*, and *jensenni*) and seven bacteria (*atopobium*, *garnerella vaginalis*, *megasphaera*, *mycoplasma hominis*, *ureaplasma parvum*, *ureaplasma urealuytum*, and *mycoplasma genitalium*). Fifty-one positive and 134 negative vaginosis cases exist, and 16 indeterminate cases.

2.2 Association Rules

Association rule mining (AR) is a data mining technique that identifies associations between attributes from a dataset [2]. Association rule is formalized as the implication $if \ldots then \ldots$ in the form $A \Rightarrow B$, where A is the antecedent and B the consequent. To select an AR as a candidate, it must meet some quality measure. The measures most commonly used are Support, Confidence, and Lift. They are defined as follows:

$$\text{Support}(A \Longrightarrow B) = P(A \cup B), \tag{1}$$

$$\text{Confidence}(A \Longrightarrow B) = P(A|B), \tag{2}$$

$$\text{Lift}(A \Longrightarrow B) = \frac{\text{Confidence}(A \Longrightarrow B)}{\text{Support}(B)}. \tag{3}$$

2.3 Differential Evolution Algorithm

Differential Evolution (DE) is an efficient evolutionary algorithm for solving optimization problems in continuous spaces [13]. DE encodes candidate solutions through real-valued vectors and applies a difference vector to disrupt a population of these solutions. First, a population of candidate solutions is randomly created, then applying the DE evolutionary process that builds a new population using mutation, crossover, and selection operators at each iteration. Instead of implementing traditional crossover and mutation operators, DE applies a linear combination of several candidate solutions selected randomly to produce a new solution. Finally, DE returns the best candidate solution in the current population when the stop condition is fulfilled.

If for each $j \in 1, \ldots, |x^i|$, x_j^{min} and x_j^{max} are the minimum and the maximum values of the j-th parameter, respectively, the j-th value of x^i in the initial population is calculated as follows:

$$x_j^i = x_j^{min} + r(x_j^{max} - x_j^{min}), \tag{4}$$

where $r \in [0, 1]$ is a uniformly distributed random number.

Furthermore, the mutation, crossover, and selection operators are defined as follows:

- **Mutation:** Three randomly chosen individuals of the current population (x^{r_1}, x^{r_2} and x^{r_3}), being different from each other and also different from the target vector, are linearly combined to yield a *mutated vector* v^i, using a user-specified scale factor F to control the differential variation, as follows:

$$v^i = x^{r_1} + \mathrm{F}\big(x^{r_2} - x^{r_3}\big). \tag{5}$$

  Eq. 5 is related with the DE/rand/1 variant defined in [9].
- **Crossover:** The mutated vector is recombined with the target vector to build the trial vector u^i. For each $j \in \big\{1, \ldots, |x^i|\big\}$, either x_j^i or v_j^i is selected based on a comparison between a uniformly distributed random number $r \in [0, 1]$ and the crossover rate CR. The recombination operator also uses a randomly chosen index $l \in \{1, \ldots, |x^i|\}$ to ensure that u^i gets at least one value from v^i, as follows:

$$u_j^i = \begin{cases} v_j^i & \text{if } r \leq \text{CR or } j = l, \\ x_j^i & \text{otherwise.} \end{cases} \tag{6}$$

- **Selection:** A one-to-one tournament is applied to determine which vector, between x^i and u^i, is selected as a member of the new population.

An advantage of DE is that it uses a few control parameters: a crossover rate Cr, a mutation scale factor F, and a population size NP.

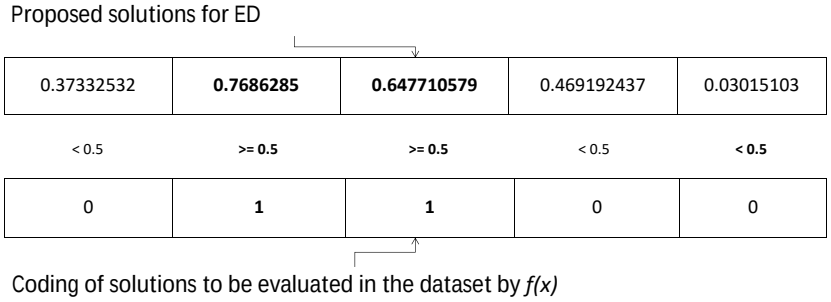


Fig. 1. Coding the candidate solution proposed by DE.

3 Implementation

DE evolves a population of randomly generated real-valued vectors. The evolutionary process is guided by a fitness function $f(x)$ determining the quality value of each individual in the population. In this work $f(x)$ is computed using several metrics searching for a maximum correlation between dataset attributes. The two crucial elements determining the success of the evolutionary process to create representative association rules to identify positive BV cases are defined as follows:

3.1 DE/rand/1/bin Version Implemented

DE/rand/1/bin is a classic DE variant, where *rand* indicates that base vectors are randomly chosen, 1 indicates that only one vector difference is used to form the mutated population, and the term *bin* (from binomial distribution) points out that uniform crossover is employed during the formation of the trial population.

3.2 Solution Encoding Scheme

Since the dataset's attributes are categorical, a scheme is required where the vector of real numbers can represent the selection or not of some attribute. The threshold-based scheme is the traditional approach to represent the selected attributes from a dataset: If the i-th parameter value is greater than 0.5, then the i-th attribute is chosen to build an association rule; otherwise, this attribute is discarded [10]. This scheme is used in this work, and Fig. 1 shows an example of this mapping scheme.

3.3 Fitness Function Definition

In research health, qualitative or dichotomous attributes are frequently used. In this research, we propose studying associations between bacteria that trigger

BV+. There are different statistical models to analyze qualitative data based on contingency tables. Thus, the objective function is made up of three statistical tests: Chi-squared, Yates and Fisher. These metrics are defined as follows:

- **Chi-squared Test:**

$$X^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}, \tag{7}$$

  where O_i is the observed value (the number of cases observed in a cell contingency table). E_i is the expected value, the number of expected cases in each cell of the contingency table.
- **Yates's formula:**

$$X^2(Yates) = \frac{n\left(\mid ad - bc \mid -\frac{n}{2}\right)^2}{(a+b)(c+d)(a+c)(b+d)}. \tag{8}$$

- **Fisher's test:**

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!}. \tag{9}$$

  In Eqs. 8 and 9, a, b, c, d are the expected frequencies in a 2 x 2 contingency table. $(a+b)$ is the sum of the two frequencies in row 1 and $(c+d)$ the sum of the frequencies in row 2. Similarly $(a+c)$ is the sum of the frequencies in column 1 and $(b+d)$ the sum of the frequencies in column 2. The total sum of all frequencies is given by n [15].

The fitness function is defined as follows:

$$f(x) = \begin{cases} p & \text{if expected frecuency} < 3 \\ X^2(Yates) & \text{if expected frecuency} > 3 \text{ and} < 5 \\ X^2 & \text{otherwise.} \end{cases} \tag{10}$$

3.4 Contingency Table

The contingency table plays an important role in the process of discovering the association between the attributes to be analyzed. The contingency table in Fig. 2, is composed of rows and columns, in the cells are recorded the absolute or relative frequencies where it is possible to analyze the correlation between two variables from the calculations that can be performed. From the set of solutions that integrate the initial population proposed by the evolutionary algorithm, they are coded in a binary format described in 3.2 , to select the combination of attributes of the study data set and placed in the contingency table.

To determine the existence of a correlation between the selected attributes, the fitness function *f(x)* is applied to the contingency table, and the results obtained will determine the existence of correlation at the 95% confidence level. The fitness function will guide the evolutionary process to determine the quality of each individual, for our case study, the fitness function is composed of three statistical tests under these restrictions:

Variable of study

VB+	1	1	0	1	1	1	1	1	0	0
Bacteria	1	1	0	0	0	0	1	1	1	1

	B	B'	
A	n(AB)	n(AB')	n(A)
A'	n(A´B)	n(A'B')	n(A')
	n(B)	n(B')	N

→

	Vaginosis		
Bacteria	VB+	VB-	Total
Present	4	2	6
Absent	3	1	4
Total	7	3	10

Fig. 2. The content of each cell is the number of frequencies that simultaneously satisfy the two conditions as $n(AB)$ [5] and N is total number of cases.

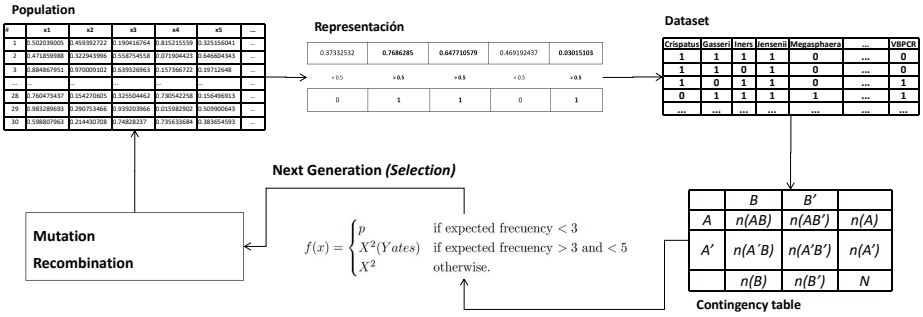


Fig. 3. Experimental design to build Association Rules using Differential Evolution.

– If 25% of the expected frequencies is >3 and <5, the Yates test is applied.
– If 25% of the expected frequencies is < 3, Fisher's exact text is used.

After calculating the existence of dependence between attributes, the strength of the association is measured with Pearson's *Phi* correlation function. If the result is close to one, it indicates a strong association, while if the results are close to zero, the association is very poor or non-existent.

4 Experimental Design

The following diagram details the steps to create rules with differential evolution.

The initial population is defined with random numbers within the continuous space. The solutions composing the population are evaluated with the objective function *f(x)* composed of the three metrics. To evaluate the solution, it is coded to a binary format to select the combination of attributes of the dataset corresponding to the coding. Subsequently, the result of the selected set of attributes from the dataset is placed in a contingency table and the function *f(x)* is applied under the criteria described in Section 3.4.

Table 1. Chi-squared test result.

No.	Rule	$f(x)$	Phi
1	Jensenni, Megasphaera, Atopobium $\Longrightarrow$ VB+	1.71×10^{-17}	0.625
2	Gasseri, Megasphaera, Atopobium $\Longrightarrow$ VB+	4.86×10^{-19}	0.655
3	Iner, Gardnerella $\Longrightarrow$ VB+	6.1×10^{-9}	0.427

Table 2. Yates's correction test result.

No.	Rule	$f(x)$	Phi
1	Iners, Jensenii, Megasphaera $\Longrightarrow$ VB+	7.32×10^{-10}	0.473
2	Gasseri, Iners, Jensenii, Atopobium, Gardnerella $\Longrightarrow$ VB+	2.01×10^{-9}	0.463
3	Gasseri, Iners, MH $\Longrightarrow$ VB+	1.39×10^{-4}	0.303

Table 3. Fisher's exact test result.

No.	Rule	$f(x)$	Phi
1	Crispatus, Gardnerella, MH, MG $\Longrightarrow$ VB+	5.0×10^{-6}	0.366
2	Gasseri, Iners, Megasphaera, Atopobium, MG, UP $\Longrightarrow$ VB+	2.1×10^{-5}	0.344
3	Crispatus, Gasseri, Iners, Megasphaera, MG, UP $\Longrightarrow$ VB+	8.8×10^{-5}	0.321

Finally, the solution with the best fitness is passed to the next generation, this process will continue until the stop condition is completed.

5 Results

The results obtained by minimizing the fitness function *f(x)* described in Section 3.3 are summarized below. *f(x)* evaluates the results under the p-value criterion, the null hypothesis is either approved or rejected with a confidence level of 95%. The null hypothesis H_0 is rejected if the p-value is less than 0.05 meaning that there is an association. Under this criterion, we minimize the fitness function *f(x)* and measure the strength of the association with the Pearson correlation coefficient Phi, of the analyzing variables.

The results are detalled in Table 1, 2 y 3. In this tables, the rules with the highest statistical value are depicted. The DE parameters are adjusted according to the recommendation suggested in the existing literature [13], as follows:

1. $F \in [0.5, 1.0]$
2. $CR \in [0.8, 1.0]$

and a population of 30 individuals.

The evolutionary process is stopped when 30 generations are reached.

In the results of the Chi-square test in Table 1, rule one indicates that *Jensenni, Megasphaera and Atopobium* bacteria trigger Bacterial Vaginosis with association strength Phi = 0.625. In the results of the Yates test in Table 2, rule one formed by bacteria *Iners, Jensenii, Megasphaera* indicated positive

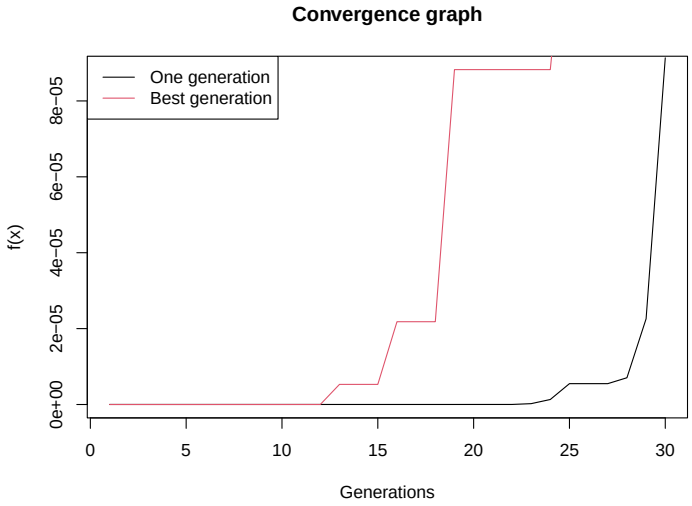


Fig. 4. Convergence of the evolution in the solutions of the fitness function.

for bacterial vaginosis with an association Phi= 0.473. Finally, in Table 3 the Fisher's test results for rule one *Crispatus, Gardnerella, Mycoplasma Hominis and Mycoplasma Genitalium* indicated positive for Baginosis with an association Phi = 0.366.

From the results, the rules that may be involved in the development of the infection are summarized under the criterion of highest statistical significance. The explanation of the cause of infection is so far complicated by the large number of bacteria that coexist in the vagina of women.

Furthermore, Figure 4 shows the fitness function convergence behavior to the best individual in the population.

6 Conclusion

BV is a health problem and should be treated early to avoid future risks in women. In this work, we implemented the differential evolution algorithm as a tool to discover strongly related association rules and avoid generating low-quality rules. The proposed approach is used to avoid the combinatorial explosion present when the Apriori algorithm is applied.

In this analysis, several tests were performed by modifying the parameters according to the limits suggested by the experts. The observed results allow knowing the bacteria associated with bacterial vaginosis with statistical values. This is an on-going research, and currently a function with a set of biological constraints is being implemented to guide the evolutionary search towards the

optimal result. Other bacteria causing the Bacterial Vaginosis infection will also be expected to be discovered.

References

1. Coello, C.: Introducción a la computación evolutiva (Notas de curso). CINVESTAV-IPN, México, DF (2004)
2. Ceglar, A., Roddick, J.: Association mining. ACM Computing Surveys. 38, 5 (2006)
3. Dongre, J., Prajapati, G.L., Tokekar, S.V.: The role of apriori algorithm for finding the association rules in data mining. In: ICICT 2014, pp. 657—660 (2014)
4. García, P.: Vaginosis bacteriana. Revista Peruana De Ginecología Y Obstetricia, 53, 167–171 (2007)
5. Geng, L., Hamilton, H.: Interestingness measures for data mining: A survey. ACM Computing Surveys (CSUR), 38, 9 (2006)
6. Hernández Orallo, J., et al.: Introducción a la Minería de Datos. Biblioteca Hernán Malo González (2004)
7. Kotsiantis, S., Kanellopoulos, D.: Association rules mining: A recent overview. GESTS Int. Trans. on Computer Science & Engineering, 32, 71–82 (2006)
8. Pérez-Gómez, J.F., Canul-Reich, J., Hernández-Torruco, J., Hernández- Ocaña, B.: Predictor selection for bacterial vaginosis diagnosis using decision tree and relief algorithms. Applied Sciences 10(9), 3291 (2020)
9. Price, K., Storn, R., Lampinen, J.: Differential evolution: a practical approach to global optimization Springer (2006)
10. Rivera-López, R., Mezura-Montes, E., Canul-Reich, J., Cruz-Chávez, M. A.: A permutational-based differential evolution algorithm for feature subset selection. Pattern Recognition Letters, 133, 86–93 (2020)
11. Saldaña, M.: La prueba chi-cuadrado o ji-cuadrado (2). Revista Enfermería Del Trabajo. 1, 31–38 (2011)
12. Sanchez-Garcia, E., et al.: Molecular epidemiology of bacterial vaginosis and its association with genital micro-organisms in asymptomatic women. Journal Of Medical Microbiology, 68, 1373–1382 (2019)
13. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal Of Global Optimization, 11, 341–359 (1997)
14. Wu, X., et al.: Top 10 algorithms in data mining. Knowledge & Information Systems, 14, 1–37 (2008)
15. Zar, J.: A fast and efficient algorithm for the Fisher exact test. Behavior Research Methods, Instruments, & Computers, 19, 413–414 (1987)

ISSN 1870-4069

Sistema de control de acceso mediante identificación y verificación facial

Luis Antonio López Gómez, Jorge Magaña Govea, Fernando Pech May

Tecnológico Nacional de México Campus de Los Ríos,
Mexico

fernando.pech@cinvestav.mx

Resumen. En la actualidad existen infinidad de aplicaciones orientadas al reconocimiento facial por computadora, una disciplina que combina técnicas de inteligencia artificial, matemáticas, entre otros y que forma parte de nuestro día a día. El presente artículo tiene como objetivo desarrollar un módulo de control de acceso biométrico capaz de identificar rostros humanos y garantizar altos niveles de seguridad y disminuir la vulnerabilidad ante accesos malintencionados. El uso de este moduló automatiza el proceso de identificación y validación mediante reconocimiento facial facilitando esta tarea con mayor rapidez y exactitud. Los usuarios se colocan frente a la cámara web de la computadora que realiza la detección y reconocimiento del rostro, almacenándolos y comparándolos con los existentes en la base de datos. En su desarrollo se usaron modelos preentrenados de detección de rostros como MTCNN y RetinaFace, y modelos de reconocimiento de rostros como DeepFace y Facenet. El lenguaje de programación fue Python, teniendo como framework TensorFlow y OpenCV.

Palabras clave: Reconocimiento facial, algoritmo de aprendizaje automático, redes neuronales.

Access Control System Using Facial Identification and Verification

Abstract. Currently there are countless applications oriented to facial recognition by computer, a discipline that combines artificial intelligence techniques, mathematics, among others, and that is part of our day to day. This article aims to develop a biometric access control module capable of identifying human faces and guaranteeing high levels of security and reducing vulnerability to malicious access. The use of this module automates the identification and validation process through facial recognition, facilitating this task with greater speed and accuracy. Users are placed in front of the webcam of the computer that performs face detection and recognition, storing them and comparing them with those in the database. Pre-trained face detection models such as MTCNN and RetinaFace, and face recognition models such as DeepFace and

Facenet were used in its development. The programming language was Python, with TensorFlow and OpenCV as framework.

Keywords: Artificial intelligence, deep learning, facial recognition.

1. Introducción

Los avances tecnológicos en inteligencia artificial, especialmente visión por computadora, ha crecido su importancia en múltiples áreas y se considera un campo de investigación muy importante en los últimos años. El reconocimiento facial es una técnica biométrica que permite identificar o verificar a un sujeto a través de una imagen, vídeo o cualquier elemento audiovisual de su rostro, debido a la seguridad que proporcionan cada vez tiene mayor presencia en la automatización de procesos de seguridad. Estos métodos son cada vez más precisos con el surgimiento de nuevas técnicas. El presente artículo se muestran los resultados de implementar un sistema de control acceso biométrico basado en el reconocimiento facial.

En el 2010 las redes sociales comienzan el uso del reconocimiento facial, Facebook comenzó a usar una función de reconocimiento facial que ayudaba a detectar personas con rostros destacados en las fotos actualizadas por sus usuarios. Al no tener un impacto negativo, cada día se cargan y etiquetan más de 350 millones de fotos utilizando el reconocimiento facial.

En el 2011, el auge del aprendizaje automático, las redes neuronales y el aprendizaje profundo junto con el reconocimiento facial generó un campo de oportunidades de posibles aplicaciones, agrupándolas en las categorías: seguridad, salud y comercio.

Actualmente, en el área de reconocimiento facial se han propuesto muchos enfoques que han llevado a diferentes algoritmos. Pero la mayoría de ellos se centran en 3 aspectos básicos: detección, extracción y reconocimiento. La Detección es el procedimiento de localizar caras en imágenes o videos. Entre los algoritmos destacados, se encuentran: eingenface, redes neuronales (NN), discriminante de Fisher y modelos oculto de Márkov (HMM).

La Extracción es el procedimiento de obtener información relevante de un rostro en una imagen, por ejemplo regiones de la cara, variaciones, ángulos o medidas del rostro. Entre los algoritmos se encuentran: histogramas de gradientes orientados (HOG), características similares de HAAR, patrones binarios locales (LBP) y redes neuronales convolucionales (CNN).

El Reconocimiento implica un método de comparación. Algoritmos de clasificación destacados son: K-vecinos más cercanos (KNN), máquinas de vectores de soporte (SVM), arboles de decisión, regresión logística, descenso de gradiente estocástico, bosque aleatorio.

La solución propuesta en este módulo usa herramientas de aprendizaje profundo, especialmente las redes neuronales convolucionales (CNN), las cuales se han convertido en una de las técnicas más populares para resolver problemas relacionados con clasificación de imágenes, detección de objetos, reconocimiento de rostros entre otros.

2. Trabajos relacionados

La forma de acceder a la información está siendo transformada radicalmente debido a las Tecnologías de la Información y las Comunicaciones. Para estar a la vanguardia, se debe tener en cuenta la investigación y el uso de innovaciones en la industria, robótica, automatización, inteligencia artificial, big data, entre otras. Estas innovaciones no solo ayudan a mejorar procesos, sino también al descubrimiento de nuevos conocimientos.

Muños et al. [1] desarrolló un sistema de control de acceso de usuarios basado en el reconocimiento facial empleando algoritmos de aprendizaje profundo deep learning, a través de la visión por computadora. Utilizó el algoritmo de aprendizaje profundo MTCNN, el modelo pre entrenado Facenet teniendo mayor respuesta en cuanto a la métricas rendimiento, acurracy y precisión. Los resultados muestran que el sistema de reconocimiento facial tiene un alto rendimiento en identificación facial cercanos al 100 %.

Ildefonso et al. [2] implementó un sistema de reconocimiento facial que hace uso de redes neuronales artificiales, el algoritmo utilizado para la detección ha sido el de histogramas de gradientes orientes (HOG), para ajustar la imagen se utilizó un algoritmo llamado estimación de punto de referencia que ha sido implementado en la librería Face Recognition, así mismo para la extracción de características se utilizó una red neuronal convolucional profunda. Este modelo presenta una precisión del 99,38 %.

Cayllahua et al. [3] desarrolló una sistema basado en una red neuronal convolucional de aprendizaje profundo para el reconocimiento facial y el control de acceso de estudiantes de la carrera de Ingeniería Mecatrónica; la metodología consistió en el entrenamiento de la red neuronal y extraer datos relevantes de los rasgos faciales en las fotografías tomadas. Utilizó una muestra de 426 fotografías correspondiente a 14 alumnos que utilizaron el Laboratorio de Control en el semestre 2019-I; igualmente, el software empleado para el entrenamiento de la red fue el MATLAB y su Toolbox Deep Learning. También, se realizaron pruebas para la selección de la red, iniciando con una capa de convolución, luego dos, y finalmente tres capas, las cuales dieron como resultados los siguientes porcentajes de precisión 15.63 %, 94.00 % y 67.13 %, respectivamente. De esta manera, optaron por elegir la red neuronal con dos capas de convolución, de 16 y 32 filtros, para realizar el reconocimiento facial.

Meza et al. [4] Desarrolló un sistema basado en RNA para mejorar la identificación de rostros de delincuentes en el distrito de Laredo, utilizó la metodología Jhon Durkin, para mejorar la identificación de rostros delictivos en el distrito de Laredo en apoyo a la policía nacional; realizó pruebas para determinar la normalidad en datos estadísticos a través de Shapiro-Wilk, la población es de 2553 delincuentes; tomando 334. Los resultados muestran en el primer indicador que el tiempo promedio en la identificación de rostros de delincuentes se redujo en un 91,66 % con una disminución de 414,85 segundos, en el segundo indicador el número de identificaciones de delincuentes se incrementó recientemente el número de delincuentes identificados en un 68.82 y en el tercer

indicador el tiempo promedio en alerta sobre los delincuentes identificados ante la policía se reducen en un 77,31 %.

3. Estado del arte

3.1. Modelos de detección

En la visión por computadora la clasificación de imágenes se toma de una imagen y se predice el objeto en una imagen, mientras que la detección de objetos no solo predice el objeto, sino que también se encuentra su ubicación en términos de cuadros delimitadores. Por ejemplo, cuando se construye un clasificador de rostros, se toma una imagen de entrada y se predice si contiene un rostro, mientras que un modelo de detección de objetos también indicaría la ubicación del rostro encontrado.
A continuación, se listan algunos detectores de rostros dentro del estado del arte.

3.2. MTCNN

MTCNN [5] (Redes convolucionales en cascada multitarea) es una red neuronal convolucional en cascada multitarea, que se utiliza para tratar simultáneamente la detección de rostros y el posicionamiento de puntos clave de rostros. MTCNN realiza tres etapas, para las que se necesita redimensionar la imagen a diferentes escalas para construir una pirámide de imágenes. En la primera etapa se utiliza una red convolucional que detecta ventanas de caras candidatas. A continuación, se utiliza otra red neuronal convolucional que descarta un gran número de candidatos en los que no existen rostros. Finalmente, una última red convolucional trata de identificar cualidades de los candidatos donde existe realmente un rostro, identificando las posiciones de cinco puntos de referencia faciales: uno en cada ojo, otro en la punta de la nariz y los dos restantes en las comisuras de los labios Ver Fig 1.

3.3. RetinaFace

RetinaFace [6] se presenta en el 2020 como un método de localización de rostros multinivel singleshot, en el que se unifica la predicción de cuadros faciales, la localización de puntos de referencia faciales en 2D y la regresión de vértices en 3D; con el fin de obtener la regresión de puntos en el plano de la imagen Ver Fig 2.

3.4. SCRFD

Sample and Computation Redistribution for Efficient Face Detection (SCRFD) [7]. El muestreo de datos de entrenamiento y las estrategias de distribución de cómputo son las claves para una detección de rostros eficiente y precisa. Debido a lo anterior, se presenta dos métodos simples pero efectivos:

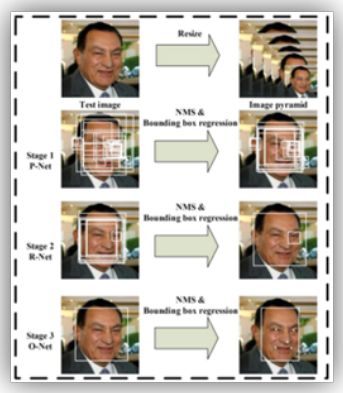


Fig. 1. Canalización del marco en cascada que incluye redes convolucionales profundas multitarea de tres etapas.

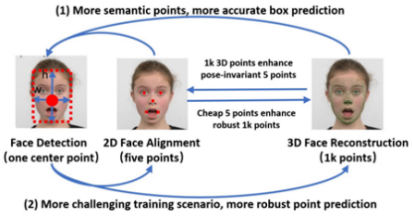


Fig. 2. Tres tareas de localización de rostros tienen diferentes niveles de detalle, pero comparten el mismo objetivo: predicción precisa de puntos en el plano de la imagen.

La redistribución de muestras (SR), que aumenta las muestras de capacitación para las etapas más necesarias, según las estadísticas de los conjuntos de datos de referencia; y la Redistribución de Computación (CR), que reasigna la computación entre la columna vertebral, el cuello y la cabeza del modelo, con base en una metodología de búsqueda meticulosamente definida.

3.5. Librerías para la detección de rostros

Para la detección de rostros, los modelos de aprendizaje profundo funcionan mejor. Anteriormente, los descriptores de características clásicos y los clasificadores lineales eran una muy buena solución para la detección de rostros, en esta ocasión veremos 3 librerías o frameworks que analizamos para el desarrollo de este módulo.

1. MediaPipe: Es una solución ultrarrápida de detección de rostros que viene con 6 puntos de referencia y compatibilidad con múltiples rostros. Se basa

en BlazeFace, un detector de rostros liviano y de buen rendimiento diseñado para la inferenciade GPU móvil (Google, s.f.). El rendimiento en tiempo real del detector permite que se aplique a cualquier experiencia de visor en vivo que requiera una región facial de interés precisa como entrada para otros modelos específicos.
2. OpenCV: Es una biblioteca libre de visión artificial originalmente desarrollada por Intel. OpenCV significa Open Computer Vision (Visión Artificial Abierta). Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en una gran cantidad de aplicaciones, y hasta 2020 se la sigue mencionando como la biblioteca más popular de visión artificial. Detección de movimiento, reconocimiento de objetos, reconstrucción 3D a partir de imágenes, son sólo algunos ejemplos de aplicaciones de OpenCV.
3. Dlib HoG: El histograma de gradientes orientados (HOG) es un descriptor de características utilizado en la visión artificial y el procesamiento de imágenes con el fin de detectar objetos. Actualmente se encuentra implementado en diferentes librerías para la dirección de rostros, siendo un ejemplo la librería Dlib.

3.6. Métodos de reconocimiento

Para el problema de reconocimiento facial donde se pretende identificar individuos, existen numerosos mecanismos, de los cuales las redes neuronales han obtenido resultados de precisión muy altos. En la actualidad para mejorar la precisión en el reconocimiento se han establecido métodos para el reconocimiento, los cuales son modificaciones a las arquitecturas o modelos de redes neuronales existentes.

3.7. Facenet

Facenet [8] se basa en aprender una incrustación euclidiana (embedding) por imagen utilizando una red convolucional profunda. La red está entrenada de manera que las distancias L2 al cuadrado en el espacio del embedding se correspondan directamente con la similitud de rostros: los rostros de la misma persona tienen distancias pequeñas y los rostros de personas distintas tienen distancias grandes. Una vez que se ha producido el embedding, las tareas de verificación de rostros simplemente implican pasar por un umbral la distancia entre las dos incrustaciones; el reconocimiento se convierte en un problema de clasificación k-NN; y el agrupamiento se puede lograr utilizando técnicas estándar como k-means o agrupamiento aglomerativo.

3.8. VGGFace

Desarrollado en el Grupo de Geometría Visual (VGG) de la Universidad de Oxford, es la aplicación de la arquitectura muy profunda de ConvNet

VGG-16. Entrenada en una base de datos de 2,6 millones de imágenes de rostros y compuesta por 2622 identidades únicas, la base de datos utilizada se compone de hasta mil instancias de cada sujeto. El modelo está configurado para tomar una imagen RGB de 224 x 224 de tamaño fijo como entrada; como una forma de preprocesamiento, inicialmente normalizan en el centro todas las imágenes de entrenamiento.

3.9. DeepFace

DeepFace [9], es una biblioteca híbrida de reconocimiento facial que envuelve modelos de última generación: VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace y Dlib. DeepFace proporciona herramientas para el reconocimiento facial, verificación facial, detección facial, detección de puntos de referencia faciales, similitud, reconocimiento de edad y género. La solución es escalable y cuenta con un sistema de administración de roles de usuario que permite controlar fácilmente quién tiene acceso a los servicios de reconocimiento facial.

4. Materiales y métodos

Deep Learning se ha convertido en el modelo de referencia en muchos ámbitos, tales como la visión artificial o visión por computación. Por ejemplo, la aplicación cada vez más utilizada como el reconocimiento facial, es decir, la identificación computarizada de las personas presentes en una imagen o vídeo. Para lograr que el sistema sea capaz de lograr la identificación y validación de las personas que aparecen en una imagen, se requieren de ciertas etapas que a continuación se mencionan:

1. Detectar y capturar los rostros en la imagen.
2. Utilizar una red neuronal capaz de mapear las características del rostro humano en una representación numérica. Embedding o Encoding.
3. Medir la similitud entre la representación numérica y las representaciones de referencia disponibles en una base de datos.
4. Determinar si son similares para considerar que pertenecen a la misma persona y conceder el acceso. Ver Figura 3.

En la Figura 4, se describen los procesos seguido para el módulo de acceso, considerando dos subprocesos, el registro y la autenticación de usuarios. Para el registro es necesario almacenar información del usuario, tal como el nombre, la contraseña y la imagen. Posteriormente se captura la imagen del rostro en una representación vectorial que permitirá posteriormente su identificación. La detección del rostro tiene como objetivo localizar la región del mismo en la imagen de entrada, y la extracción del vector de representación del rostro, denominado Embedding, el cual es almacenado. Durante el proceso de inicio de sesión a través de reconocimiento facial, se captura el rostro generándose

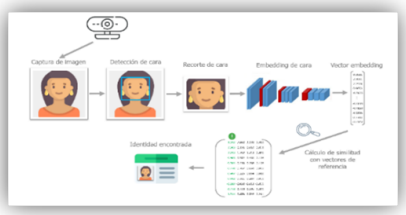


Fig. 3. Cascada en redes convolucionales profundas multitarea de tres etapas.

Tabla 1. Resultados accuracy, tiempo total en proceso de reconocimiento por imagen, en búsqueda de similitud.

ID	Detección	Reconocimiento	Accuracy	Tiempo(s)
1	MTCNN	DeepFace	0,6767	113,01
2	MTCNN	Facenet	0,9001	79,86

imágenes que serán ingresadas para que el sistema extraiga el Embedding de cada rostro, este, es comparado con los Embedding almacenados, con el fin de encontrar la similitud. El objetivo de obtener una representación numérica de las caras (embeddings) es poder cuantificar similitudes entre ellas. Dos formas de calcular esta similitud es utilizando la distancia euclídea o la distancia coseno entre embeddings. Cuanto menor es la distancia, mayor la similitud de las caras.

5. Resultados

Las pruebas se realizaron con el conjunto de datos propio del módulo fue instalado en un entorno local con las siguientes características:

1. Procesador, AMD A10-8700P Radeon R6, 10 Compute Cores 4C+6G 1.80 GHz.
2. Memoria, 12.0 GB.
3. Disco Duro, 1 TB.
4. Sistema Operativo, Windows 11.

Los resultados parciales muestran que la información obtenida son notorias las diferencias entre algunos métodos seleccionados. La Tabla 1 muestra un accuracy por debajo de 0.67 en el método identificado ID 1, la cual pertenece al modelo de reconocimiento Deepface, aunque el resultado con el de detección MTCNN es mejor cercano a una.

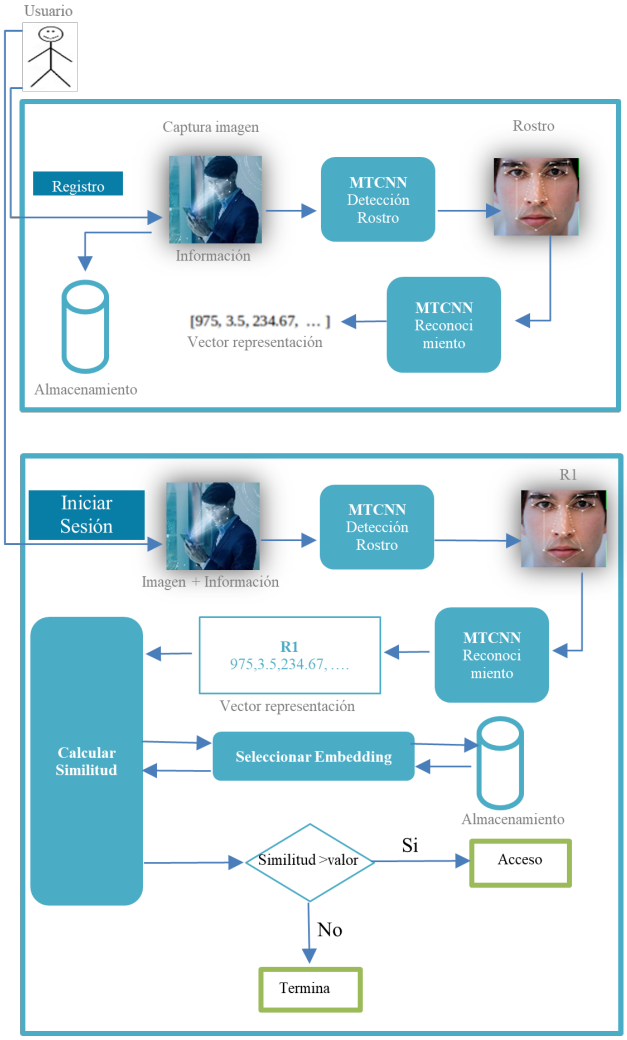


Fig. 4. Modelo de procesos.

6. Conclusión

Se desarrolló un módulo para el control de acceso al sistema de acceso basado en técnicas de detección y reconocimiento facial, utilizando modelos de aprendizaje profundo que permiten identificar al usuario a través del cálculo de similitud entre vectores con el propósito de detectar y reconocer rostros, la detección es a través de una cámara web mediante el algoritmo que se ha implementado durante el desarrollo del aplicativo.

Se analizó el desempeño del módulo elaborado realizando pruebas en casos reales,

mejorando la mejor exactitud de las pruebas en el reconocimiento de rostros, disminuyendo reconocimientos fallidos.

Usando la identificación de identidades a través del cálculo de similitud entre los vectores de características principales (embeddings), no se requiere volver a reentrenar a la red neuronal convolucional, desde el momento que se realizan nuevos registros, solo se almacena el vector que es utilizado para identificaciones posteriores.

Referencias

1. Muñoz, Edison. Desarrollo de un sistema de control de acceso de personal empleando reconocimiento facial respaldado con técnicas de aprendizaje profundo. Tesis (Título de Ingeniero en Electrónica, Automatización y Control). Ecuador: Universidad de las Fuerzas Armadas - Innovación para la Excelencia, 189 pp.
2. Ildefonso, Silva. Reconocimiento facial basado en redes neuronales convolucionales. Tesis (Grado en Ingeniería de las Tecnologías de Telecomunicación). España: Universidad de Sevilla, 2018, 77pp.
3. Cayllahua, Nestor y SUÁREZ, Juan. Redes neuronales de aprendizaje profundo para el reconocimiento facial y control de accesode estudiantes a un laboratorio. Tesis (Título de Ingeniero Electrónico).Lima: Universidad Ricardo Palma, 2019, 68pp.
4. Meza, Alain y Ramos, Maria. Sistema Inteligente Basado en Redes Neuronales para mejorar la identificación de rostros de delincuentes en el distrito de Laredo. Escuela academico - profesional de informatica. Tesis (Título de Ingeniero de Sistemas). Trujillo:Universidad Cesar Vallejo, 2018, 113 pp.
5. Justin Pinkney (2022). MTCNN Face Detection (https://github.com/matlab-deep-learning/ mtcnn-face-detection/releases/tag/ v1.2.4), GitHub. Retrieved November 12, 2022.
6. Deng, J., Guo, J., Ververas, E., Kotsia, I., y Zafeiriou, S. (2020). Retinaface: Single-shot multilevel face localisation in the wild. En 2020 ieee/cvf conference on computer vision and pattern recognition (cvpr) (p. 5202-5211). doi: 10.1109/CVPR42600.2020.00525
7. Deepinsight. (2021). Scrfd an efficient high accuracy face detection. https://github.com/deepinsight/insightface/tree/master/detection/scrfd.
8. Schroff, F., Kalenichenko, D., y Philbin, J. (2015, junio). FaceNet: A unified embedding for face recognition and clustering. IEEE. https://doi.org/10.1109 % 2Fcvpr.2015 .7298682. doi: 10.1109/cvpr.2015.7298682
9. Serengil, S. I. (2022). Deepface. https://github.com/serengil/deepface.

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación