ISSN 1870-4069

Implementación de una red neuronal convolucional para el reconocimiento de acciones humanas

Mitchell Ángel Gómez Ortega

Universidad Politecnica del Valle De México,
División de Ingeniería Mecatrónica y Mecánica Automotriz,
México

mitchell.gomez@outlook.com

Resumen. En esta investigación se desarrollan técnicas de deep learning para la implementación de una red neuronal convolucional para el reconocimiento de acciones humanas. Enunciando los principios que rigen a las redes neuronales convolucionales, sus problemas de implementación en sus respectivos tiempos al igual que la evolución tecnológica que han tenido a lo largo de las ultimas décadas. Ademas, considerando situaciones en las cuales el deep learning tiene una alta eficiencia y efectividad en la resolución de problemas en comparación de como lo hace una persona a través de la visión. Uno de los principales problemas en la implementación de deep learning para reconocimiento de acciones humanas esta relacionado a la cantidad de información que se requiere para realizar el entrenamiento en conjunto con el hardware requerido para su procesamiento e incremento de tiempo mediante las GPU's. Hoy en día gracias a las tecnologías de la información y comunicación y al acceso a gran cantidad de información por medio del Internet, este problema se reduce sin dejar de considerar que en ocasiones los datos son variantes tanto en resolución como en dimensiones y si son ocupados para realizar entrenamientos a redes convolucionales puede ocasionar un funcionamiento no deseado durante su implementación. Se propone una arquitectura para la clasificación de 5 acciones humanas tales como (Aplaudir, boxear, caminar, correr y empujar) por medio de la creación de una base de datos actualizada, el uso de googLeNet como red pre-entrenada para la extracción de características y la creación de una red para la clasificación de acciones humanas mediante espacios temporales y memorias de largo y corto plazo bidireccionales, que ayudan a reducir los tiempos de entrenamiento mejorando la precisión por cada clasificación realizada. De esta forma se previene el sobreajuste mediante el uso de algoritmos de entrenamiento y regularización que estabilizan la red y mejoran su precisión durante el proceso de entrenamiento. Finalmente, se realiza una concatenación de googLeNet y la red con memoria a largo y corto plazo para realizar la clasificación de videos de forma secuencial donde se presenta los resultados experimentales de cada una de las acciones con datos que no fueron ingresados para el entrenamiento de la red.

Palabras clave: GoogleNet, HAR, deep learning, Nvidia, RTX 3070, videos, secuencias, acciones humanas.

Mitchell Ángel Gómez Ortega

Implementation of a Convolutional Neural Network for the Recognition of Human Actions

Abstract. In this research, deep learning techniques are developed for the implementation of a convolutional neural network for the recognition of human actions. Enunciating the principles that govern convolutional neural networks, their implementation problems in their respective times as well as the technological evolution they have had in recent decades. Also, considering situations where deep learning has high efficiency and effectiveness in solving problems compared to how a person does it through vision. One of the main problems in the implementation of deep learning for the recognition of human actions is related to the amount of information that is required to perform the training together with the hardware required for its processing and time increase through the GPUs. Today, thanks to information and communication technologies and access to a large amount of information through the Internet, this problem is reduced without forgetting that sometimes the data are variants both in resolution and in dimensions and if they are used to form convolutional networks they can cause unwanted damage. behavior during execution. An architecture is proposed for the classification of 5 human actions such as (clapping, boxing, walking, running and pushing) through the creation of an updated database, the use of googLeNet as a pre-trained network for the extraction of characteristics and the creation of a network for the classification of human actions through temporary spaces and bidirectional long- and short-term memories, which help reduce training times by improving the accuracy of each classification made. In this way, overfitting is avoided by using training and regularization algorithms that stabilize the network and improve its accuracy during the training process. Finally, a concatenation of googLeNet and the network with long and short term memory is performed to sequentially classify the videos, where the experimental results of each of the actions with data that were not entered for training are presented net.

Keywords: GoogleNet, HAR, Deep Learning, Nvidia, RTX 3070, videos, footage, human actions.

1. Introducción

En los últimos años la inteligencia artificial ha revolucionado la forma en que el ser humano desarrolla actividades mientras hace su vida cotidiana y laboral mas sencilla. Muchas de las aplicaciones han sido impulsadas por la innovación tecnológica que se aplica en ordenadores, el internet, el Big Data, etc.

Para desarrollar aplicaciones con inteligencia artificial se requiere implementar técnicas Deep Learning ya que sigue siendo una extensión de las Redes Neuronales Artificiales (ANN) y es una técnica de Machine Learning que emplean las redes neuronales profundas.

Para que las las redes neuronales profundas tuvieran el éxito y la importancia que actualmente tienen, pasaron alrededor de 30 años debido a que no se encontró una

regla de aprendizaje adecuada para la red neuronal multicapa que hacia que durante el entrenamiento la información almacenada en la red neuronal fuera considerada inútil.

En 1986 se resolvió el problema del entrenamiento de la red neuronal multicapa cuando se introdujo el algoritmo de retropropagación (Back Propagation). Sin embargo, al momento de resolver problemas prácticos no cumplió con las expectativas en los diversos intentos de superar sus limitaciones en donde se realizo el incremento de capas ocultas y nodos de las capas ocultas que carecieron de éxito. Lo que ocasiono que se decidiera que la red neuronal no tenia posibilidad de mejora y fuera olvidada durante un largo periodo de tiempo.

A mediados de la década de 2000, cuando se introdujo el aprendizaje profundo las redes neuronales profundas tuvieron una nueva oportunidad aunque tardo tiempo en producir rendimiento suficiente debido a las dificultades para entrenar la red neuronal profunda. En conjunto con las tecnologías actuales en Deep Learning arrojan resultados sorprendentes en rendimiento que supera a otras técnicas de Machine Learning así como a otras redes neuronales, y prevalece en los estudios de Inteligencia Artificial [1].

2. Trabajos relacionados

Dentro de las aplicaciones de la inteligencia artificial con mayor demanda es la clasificación de imágenes y vídeos que se realiza con la implementación de Redes Neuronales Convolucionales (CNN) y algunas de sus aplicaciones es el reconocimiento de acciones humanas (HAR) [2, 3, 4, 5], el uso de la lógica difusa para mejorar el entrenamiento de la CNN en el reconocimiento de acciones humanas [6], la creación de base de datos (*datasets*) con un gran numero de vídeos en combinación con el uso de GPU's y multicores que doten a las CNN de una gran velocidad durante entrenamiento logrando un incremento en el desempeño ocupando el procesamiento en paralelo, y aumentando la velocidad de 10 hasta 12 veces con respecto al procesamiento serial.

[9], clasificar acciones humanas mediante técnicas de esqueletización para el modelo del cuerpo humano en 2D para la obtención de secuencias espaciales [7, 8, 10, 12]. Alternativamente se han empleado acelerómetros como fuente de información para la clasificación de acciones humanas [13, 14], utilizar arquitecturas de CNN propias mejorando la extracción de características para incrementar el nivel de precisión al momento de hacer el entrenamiento y pruebas ocupando algoritmos de Machine Learning [15, 16].

Tomando como punto clave la eficiencia debido a las características de los datos de entrada, en donde representan la información de movimiento y de apariencia, pueden incrementar la precisión ocupando una CNN 3D que tiene como tarea procesar toda la secuencia de vídeo como entra. No obstante, en comparación con la percepción humana con respecto a las acciones de otro, confirman, que en esta tarea específica, las características de movimiento son cruciales.

Esto puede significar que utilizar todo el vídeo puede generar redundancia y ruido en el método de aprendizaje. Por lo tanto, si se logra mitigar la redundancia y ruido se generan resultados superiores que sugieren que las características de movimiento pueden ser más importantes para la tarea de identificar comportamientos agresivos [17, 18, 19, 20].

3. Metodología y materiales

Para realizar una clasificación de acciones humanas mediante secuencias de vídeos, es necesario partir de una red pre-entrenada que permita agilizar y utilizar los pesos pre establecidos para definir las categorías a clasificar y permita partir de los patrones aprendidos para generar nuevo conocimiento, a este proceso se le conoce como *Transfer Learning*.

La razón por la cual se ocupa *Transfer Learning* es para reducir la complejidad en comparación con la creación de una CNN desde cero, las redes pre-entrenadas como: *AlexNet*, *ResNet50* y *GoogLeNet* fueron entrenadas con millones de imágenes que tienen como objetivo clasificar 1000 objetos con sus diferencias entre ellas y ocupando multicores para su aprendizaje y regularización con tiempos de entrenamiento mayores a una semana, las cuales requirieron diversas pruebas para poder llegar a una convergencia ideal.

Esto resulta una gran ventaja debido a que la creación de una CNN desde cero requiere que el programador posea amplia experiencia del tema a resolver, en caso de que el programador carezca de dicha experiencia al diseñar y entrenar la CNN desde cero existe una alta probabilidad de que el modelo no funcione de forma adecuada e inclusive existirían casos donde no se pueda prevenir el *Overfitting* ocasionando que la red en lugar de aprender de las características trate de memorizar todos los datos de entrenamiento, lo cual se vera reflejado que no pueda clasificar y reconocer datos de un escenario real.

3.1. Arquitectura HARNet

GoogLeNet es una red Directed Acrylic Graph (DAG) que se define con capas y conexiones entre las capas y tienen una arquitectura más compleja donde las capas pueden tener entradas o salidas a múltiples capas. Estas arquitecturas permiten entrenar Deep Networks.

Para que GoogLeNet pueda clasificar acciones humanas mediante secuencias de vídeos, es necesario realizar una modificación a la arquitectura de GoogLeNet debido a que solamente esta diseñada para clasificar imágenes.

Por lo tanto, se tiene que implementar una combinación de redes neuronales convolucionales y redes de memoria a largo y corto plazo (*Long Short-Term Memory LSTM*).

Las redes LSTM son diseñadas y utilizadas para aplicaciones en las que la entrada es una secuencia ordenada Figura 1 donde la información anterior en la secuencia puede ser importante, como lo es la clasificación de acciones humanas.

También son un tipo de redes recurrentes que son redes que reutilizan la salida de un paso anterior como entrada para el siguiente paso. Como toda las redes neuronales, el nodo realiza un cálculo utilizando las entradas y devuelve un valor de salida. en redes recurrentes, esta salida se utiliza junto con el siguiente elemento como entradas para el siguiente paso, y así sucesivamente.

El valor de entrada, la salida anterior y el estado interno se utiliza en el calculo de los nodos, los resultados del cálculo se utilizan no solo para proporcionar un valor de salida, sino también para actualizar el estado.

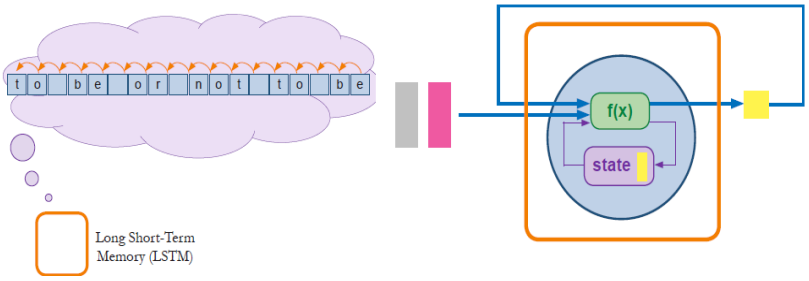


Fig. 1. Principio de funcionamiento de las redes LSTM.

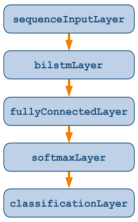


Fig. 2. Arquitectura de una red LSTM.

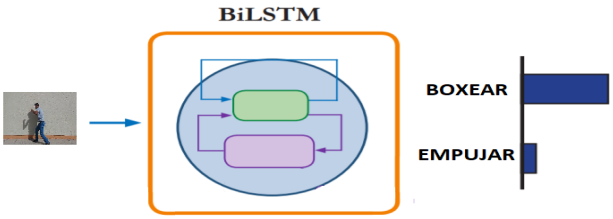


Fig. 3. Principio de la red BiLSTM para acciones humanas.

La arquitectura para clasificar secuencias se almacena en MATLAB como un vector de columnas de capas. Todas las LSTM comienzan con un capa de entrada, siguen con algunas capas LSTM o BiLSTM y terminan con las mismas capas de salida que una CNN como se muestra en la Figura 2.

Las redes LSTM incluyen la capa Bidireccional de memoria a largo y corto plazo (BiLSTM). En muchas situaciones, las BiLSTM pueden lograr una mayor precisión que las LSTM debido que al comienzo de la secuencia tiene el contexto final de la secuencia. Una capa LSTM solo tiene información sobre la datos anteriores en la secuencia.

En los primeros datos, la red no tiene información previa, por lo que la puntuación de predicción solo suele ser 0.5. Más tarde en la secuencia, la red obtiene suficiente información previa para hacer una predicción segura.

Las BiLSTM confía en su predicción a lo largo de la secuencia mientras procesan la secuencia hacia adelante y hacia atrás, por lo que el primer elemento de la secuencia tiene información sobre el resto de la secuencia como se muestra en la Figura 3.

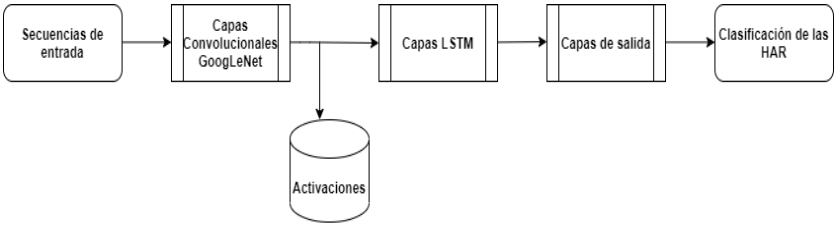


Fig. 4. Arquitectura para la clasificación de acciones humanas.

La unión de GoogLeNet y las capas BiLSTM se representa mediante el diagrama mostrado en la Figura 4, donde se utiliza las capas convolucionales de la red pre-entrenada para la extracción de características de las secuencias de vídeo y el arreglo de red BiLSTM para realizar una clasificación de cada secuencia ingresada por medio del entrenamiento y regularización de la red.

3.2. Categorías y dataset

La base de datos que se creo para el desarrollo de este trabajo fue realizada por 37 personas que se divide en 5 tipos de acciones humanas (Aplaudir, Boxear, Caminar, Correr y Empujar) que se le asigno el nombre de Human Action Recognition (HAR), la cantidad de videos por categoría se muestra en la Figura 5 en donde se seleccionaron diferentes escenarios como (Interiores, Exteriores, Exteriores con diferente ropa y Exteriores con zoom).

La base de datos contiene un total de 3690 vídeos que fueron grabados en ambientes urbanos tratando de conservar la homogeneidad con una velocidad de fotograma de 30 fps. La dimensión de los vídeos es de 320x240 píxeles y con duración exacta de 10 segundos por vídeo en el formato MP4. No obstante, se puede dividir en conjuntos de vídeos para entrenamiento, validación y test sin alterar la estructura de la base de datos. La cantidad exacta de vídeos por categoría se muestra en la Tabla 1.

A diferencia de otras dataset publicas como puede ser KTH Action dataset que esta hecha en escala de grises y con resoluciones muy bajas que ya no satisfacen resoluciones minimas de los dispositivos actuales.

La dataset HAR tiene ventajas y beneficios considerables es que la duración exacta de cada vídeo evita que la red tienda al sobre entrenamiento y baje la precisión debido a que evita la secuencias largas que generan redundancia en el entrenamiento y reduce el coste computacional.

3.3. Configuración de HARNet en MATLAB

La arquitectura mostrada en la Figura 4 y la base de datos se pueden utilizar y configurar en MATLAB mediante el Toolbox de Deep Learning. Para ello es necesario instalar la red pre-entrenada GoogLeNet por medio del Add-Ons de MATLAB como se muestra en la Figura 6. Una vez instalada, es posible cargarla directamente en el Workspace.

Para realizar las modificaciones de GoogLeNet para la clasificación de acciones humanas es necesario leer los vídeos correspondientes a la base de datos HAR.

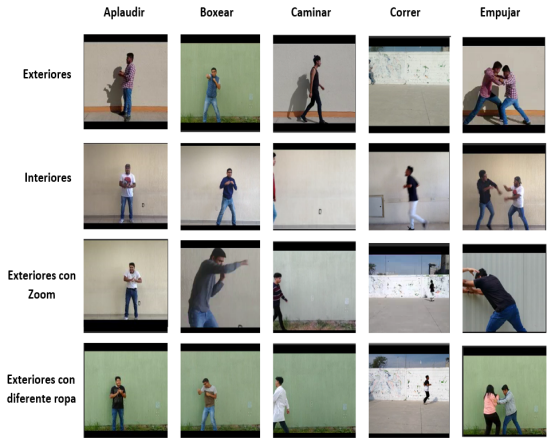


Fig. 5. Ejemplos de Dataset HAR. Se muestran ejemplos de los videos de la base de datos correspondientes a las 5 categorías y los diferentes escenarios en los cuales fueron grabados.

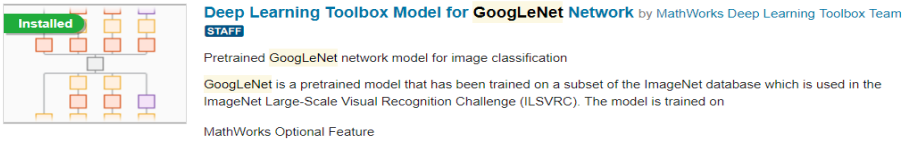


Fig. 6. Red GoogLeNet para MATLAB.

MATLAB no cuenta con una función nativa. Por lo que es necesario almacenar la etiqueta correspondiente al video y (H=Tamaño, W=Ancho, C=Número de canales y S=Número de frames del vídeo) que se almacenan en un vector de 1x4.

Una vez que se adquieren los datos de cada vídeo, se utiliza GoogLeNet como extractor de características para obtener las activaciones de cada vídeo al momento de ingresarlos a la red mediante la conversión de vídeos a secuencias de vectores característicos, en donde dichos vectores son la salida de la función de activación de la capa *pool57x7_s1*.

Se tiene que considerar que al momento de ingresar los vídeos se deben ajustar de acuerdo a los requerimientos de GoogLeNet. Habitualmente el proceso es muy tardado y es proporcional al tamaño de los vídeos de entrada, dado que depende directamente de las características de la computadora, si se llega a ocupara computadoras sin GPU's el proceso de horas pasaría a terminar de ejecutarse en semanas.

Una vez finalizado el proceso, para visualizar cada secuencia se crea una matriz de DxS donde: D=número de características que corresponde al tamaño de la salida de la capa de agrupación y S=número de fotogramas del vídeo.

Adquiriendo las secuencias de vídeo de la base de datos se preparan los datos de entrada en dos secciones para entrenamiento y validación ver en el cuadro 3.2. Correspondientes a la partición aleatoria de las secuencias de vídeo totales al 90 % para el entrenamiento y el 10 % para la validación.

Tabla 1. DataSet HAR.

Categoría	Cantidad de vídeos
Caminar	740
Correr	730
Aplaudir	740
Empujar	740
Boxear	740

Tabla 2. Conjunto de secuencias para el entrenamiento y validación.

Concepto	Cantidad de secuencias
Entrenamiento	3321
Validación	369

De acuerdo con lo mencionado anteriormente, la base de datos tiene la ventaja de reducir el sobre entrenamiento debido a que no existe redundancia por el tiempo de duración de los vídeos ayudando a la red a mejorar la precisión, sin embargo, es posible visualizar las secuencias totales de los datos de entrenamiento mediante histogramas como se muestra en la Figura 7.

En caso de que las secuencias de vídeo fueran variables y excedentes, es posible mejorar la precisión del clasificador eliminando las secuencias que exceden un valor promedio del total de secuencias ya que son mínimas junto con sus respectivas etiquetas.

Una vez procesados los datos para el entrenamiento y validación se requiere una red LSTM para la clasificación de los vectores característicos con las secuencias de vídeo que se procesaron mediante GoogLeNet. La aquitectura de la red LSTM es simple ya que no se requiere pre-procesar los datos de entrada su diseño final se muestra en la Tabla 3.

3.4. Resultados experimentales

Los resultados experimentales que se obtuvieron al entrenar e implementar HARNet en entornos reales con datos desconocidos. La arquitectura de HARNet se diseño en el capitulo anterior con el Toolbox de Deep Learning de MATLAB 2020. El entrenamiento se realizó ocupando una computadora con las siguientes características:

– Procesador AMD Ryzen 5 5600x.
– Tarjeta Gráfica Nvidia RTX3070.
– 64Gb de RAM.
– 1Tb de SSD.

En el proceso de entrenamiento, el conjunto de las 369 secuencias de vídeo fueron separadas para la validación y se realizaron pruebas conforme avanzaba el entrenamiento por cada época, clasificando cada secuencia de vídeo dentro del conjunto de validación.

Tabla 3. Arquitectura de HARNet.

Numero de capa	Nombre	Tipo	Activaciones
1	Secuencias de entrada	Sequence Input	1024
2	BiLSTM	BiLSTM	4000
3	Dropout 50 %	Dropout	400
4	Fc	Fully Connected	5
5	Softmax	Softmax	5
6	HAR Clasification	Classification Output	5

Tabla 4. Opciones de entrenamiento.

Solver	Adam
MinibatchSize	16
LearningRate	0.0001
GradientThreshold	2
Epochs	30

La precisión generada durante la clasificación ocasionaba ajustes en los pesos para reducir la función de perdida e incrementar la precisión de HARNet.

El entrenamiento se realizó en un tiempo 5 horas y 31 minutos en donde se ocuparon las configuraciones que se muestra en la Tabla 4. Se realizaron diferentes pruebas incrementando el numero de épocas con valores de 40, 80, y 100, sin embargo, el entrenamiento obtenía el mínimo error a partir de la época 30. Obteniendo una precisión de validación durante el entrenamiento de 94.31 %.

Para comprender el resultado del entrenamiento se genera una matriz de confusión, que es una tabla que se utiliza para describir el desempeño de un modelo de clasificación de un conjunto de datos de prueba para los que se conocen los valores verdaderos.

En la Figura 8a se muestra la matriz de confusión del entrenamiento en la Figura 8b se muestra la matriz de confusión para el conjunto de secuencias de validación que se considera la precisión real de HARNet generada durante el entrenamiento, donde se clasificaron de forma correcta 348 secuencias de vídeo de un total de 369, ocasionando errores de clasificación en la acción de boxear, correr y empujar.

Se puede inferir que dichos errores corresponden a los generados durante el entrenamiento y por los datos ingresados a HARNet.

El uso de las capa BiLSTM ayudo de forma eficiente en analizar las secuencias de videos de forma rapida, reduciendo los tiempos de entrenamiento y aumentando la precisión aprendiendo de forma adecuada de cada video dentro de la dataset HAR.

4. Conclusiones

La dataset HAR fue creada para entrenar a HARNet lo que incremento la eficiencia debido a su homogeneidad, duración exacta, FPS, la prevención de la redundancia dentro de cada video y la actualización de resoluciones en comparación con bases de datos existentes, creadas con resoluciones 160x120 pixeles en escala de grises que no son compatibles con los dispositivos actuales.

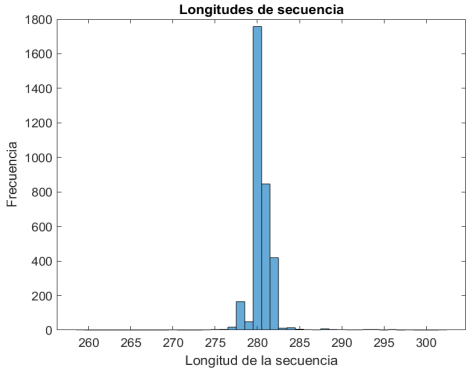
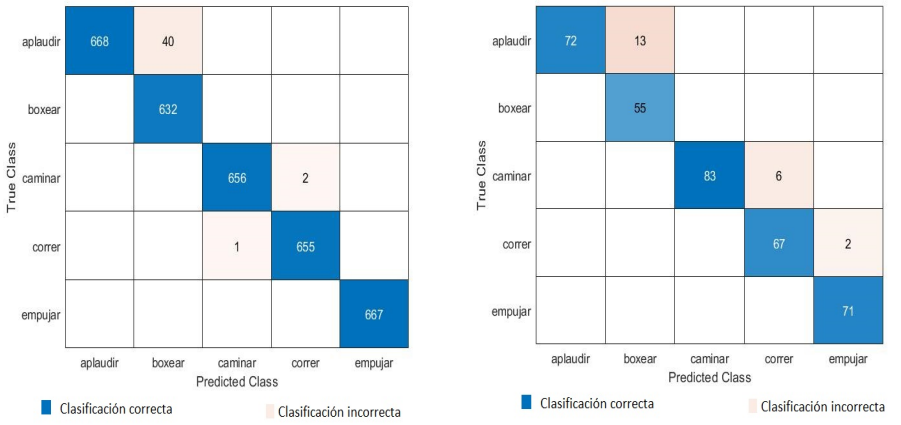


Fig. 7. Longitudes de secuencia para el entrenamiento.



(a) Matriz de confusión del entrenamiento de HARNet. **(b)** Matriz de Confusión de la validación de HARNet.

Fig. 8. Comparativa entre la matríz de confusión del entrenamiento y de validación.

Para realizar el entrenamiento por deep learning, tecnológicamente ya se cuenta con hardware de mayores capacidades como lo son las GPU's que ayuda directamente a reducir los tiempos de entrenamiento de datos del orden de miles hasta millones.

La arquitectura propuesta para HARNet desarrollada en MATLAB 2020a con el toolbox de Deep Learning incluye el uso de la red pre-entrenada googLeNet que funciona como un extractor de características para realizar la conversión de videos a secuencias de video y una red BiLSTM para el entrenamiento y clasificación, que reduce la complejidad, el costo computacional y la memoria ocupada para guardar los pesos creados durante el entrenamiento.

La configuración de las opciones de entrenamiento se realizo la selección de *Adam* como solver principal, los *minibatches* para evitar el sobreajuste, la *validación* y los *parámetros de regularización* de gradiente que generaron precisiones del *98.71 %* para entrenamiento y *94.31 %* para validación.

Mediante los entrenamientos que se realizaron a HARNet se determino que tenia una convergencia adecuada con forme se incremento el números de épocas y que la configuración de las opciones de entrenamiento prevenían el sobreajuste con cada época del entrenamiento.

En las pruebas realizadas a HARNet con datos desconocidos y ambientes no homogéneos, se clasificaron de forma correcta obteniendo niveles bajos de error, dichas pruebas demuestran la robustez de HARNet para la clasificación de las 5 acciones humanas.

La aplicación de esta red se puede trasladar sistemas de vigilancia remotos o embebidos para realizar el monitoreo de comportamientos, acciones agresivas, prevención de delitos entre otras mas, dado que el comportamiento de HARNet en entornos no homogéneos da la posibilidad de incrementar el numero de acciones humanas para su clasificación, sus resoluciones y su posible detección en tiempo real con detectores *You Only Look Once* (*YOLO*) mediante la implementación en sistemas que se encuentren embebidos en cámaras de seguridad con la finalidad de detectar acciones agresivas que pongan en riesgo la salud e integridad en un entorno especifico tales como escuelas, lugares cerrados, centros comerciales etc.

No obstante, incrementando el numero de videos de HAR y ocupando el procesamiento en paralelo con GPU's dotara a HARNet de mayor robustas y fiabilidad en conjunto con las herramientas proporcionadas por MATLAB en versiones futuras.

Referencias

1. Phil, K.: MATLAB deep learning with machine learning, neural networks and artificial intelligence (2017)
2. Valle, E. A., Starostenko, O.: Recognition of human walking/running action based on neural networks. In: 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 239–244 (2013) doi: 10.1109/ICEEE.2013.6676005
3. Ijjina, E. P., Mohan, C. K.: Human action recognition based on MOCAP information using convolution neural networks. In: 13th International Conference on Machine Learning and Applications, pp. 159–164 (2014) doi: 10.1109/ICMLA.2014.30
4. Ijjina, E. P., Mohan, C. K.: Human action recognition based on recognition of linear patterns in action bank features using convolutional neural networks. In: 13th International Conference on Machine Learning and Applications, pp. 178–182 (2014) doi: 10.1109/ICMLA.2014.33
5. Ijjina, E. P., Mohan, C. K.: One-Shot periodic activity recognition using convolutional neural networks. In: 13th International Conference on Machine Learning and Applications, pp. 388–391 (2014) doi: 10.1109/ICMLA.2014.69
6. Ijjina, E. P., Mohan, C. K.: Human action recognition based on motion capture information using fuzzy convolution neural networks. In: Eighth International Conference on Advances in Pattern Recognition (ICAPR), pp. 1–6 (2015) doi: 10.1109/ICAPR.2015.7050706
7. Rajeswar, M. S., Sankar, A. R., Balasubramaniam, V. N., Sudheer, C. D.: Sacling up the training of deep CNNs for human action recognition. In: IEEE International Parallel and Distributed Processing Symposium WorkShop, pp. 1172–1177 (2015) doi: 10.1109/IPDPSW.2015.93
8. Du, Y., Fu, Y., Wang, L.: Skeleton based action recognition with convolutional neural network. In: Proceedings of 3rd IAPR Asian Conference on Pattern Recognition, pp. 579–583 (2015) doi: 10.1109/ACPR.2015.7486569

9. Sun, L., Jia, K., Yeung, D. Y., Shi, B. E.: Human action recognition using factorized spatio-temporal convolutional networks. In: Proceedings of the IEE International Conference on Computer Vision, pp. 4597–4605 (2015)
10. Huang, C. D., Wang, C. Y., Wang, J. C.: Human action recognition system for elderly and children using three stream ConvNet. In: International Conference on Orange Technologies (ICOT), pp. 5–9 (2015) doi: 10.1109/ICOT.2015.7498476
11. Mahasseni, B., Todorovic, S.: Regularizing long short term memory with 3D human-skeleton sequences for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3054–3062 (2016)
12. Liu, L., Hu, F., Zhao, J.: Action recognition based on features fusion 3D convolutional neural networks. In: 9th International Synposium on Computational Intelligence and Design, vol. 1, pp. 178–181 (2016) doi: 10.1109/ISCID.2016.1048
13. Lee, S. M., Yoon, S. M., Cho, H.: Human activity recognition from accelerometer data using convolutional neural network. In: IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 131–134 (2017) doi: 10.1109/BIGCOMP.2017.7881728
14. Gammulle, H., Denman, S., Sridharan, S., Fookes, C.: Two stream LSTM : A deep fusion framework for human action recognition. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 177–186 (2017) doi: 10.1109/WACV.2017.27
15. Sargano, A. B., Wang, X., Angelov, P., Habib, Z.: Human action recognition using transfer learning with deep representations. In: International Joint Conference on Neural Networks, pp. 463–469 (2017) doi: 10.1109/IJCNN.2017.7965890
16. Li, J., Wang, T., Zhou, Y., Wang, Z., Snoussi, H.: Using Gabor filter in 3D convolutional neural networks for human action recognition. In: 36th Chinese Control Conference, pp. 11139–11144 (2017) doi: 10.23919/ChiCC.2017.8029134
17. Liu, M., Yuan, J.: Recognizing human actions as the evolution of pose estimation maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1159–1168 (2018)
18. Zhou, D., Feng, X., Yi, P., Yang, X., Zhang, Q., Wei, X., Yang, D.: 3D human motion synthesis based on convolutional neural network. IEEE Access, vol. 7, pp. 66325–66335 (2019) doi: 10.1109/ACCESS.2019.2917609
19. Kamel, A., Sheng, B., Yang, P., Li, P., Shen, R., Feng, D. D.: Deep convolutional neural networks for human action recognition using depth maps and postures. IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 9, pp. 1806–1819 (2019) doi: 10.1109/TSMC.2018.2850149
20. Wang, P., Yang, Y., Li, W., Zhang, L., Wang, M., Zhang, X., Zhu, M.: Research on human action recognition based on convolutional neural network. In: 28th Wireless and Optical Communication Conference (WOCC), pp. 1–5 (2019) doi: 10.1109/WOCC.2019.8770575