Detección de acciones a partir de información de profundidad mediante redes neuronales convolucionales

Sergio de López Diz
Escuela Politécnica Superior
Universidad de Alcalá
Email: s.lopezd@edu.uah.es

Cristina Losada Gutiérrez
Escuela Politécnica Superior
Universidad de Alcalá
Email: cristina.losada@uah.es

David Fuentes Jiménez
Escuela Politécnica Superior
Universidad de Alcalá
Email: d.fuentes@edu.uah.es

Marta Marrón Romera
Escuela Politécnica Superior
Universidad de Alcalá
Email: marta.marron@uah.es

***Abstract*—El reconocimiento de acciones humanas es una tarea fundamental en la visión artificial y ha cobrado gran importancia en los últimos años debido a sus múltiples aplicaciones en diferentes ámbitos, como el estudio del comportamiento humano, la seguridad o la vídeo-vigilancia. En este contexto, se propone un sistema de detección de acciones humanas basado en redes neuronales convolucionales 3D (3D-CNN) utilizando únicamente la información de profundidad proporcionada por un sensor RGB-D. El uso de este tipo de información permite reconocer las acciones realizadas por las personas, protegiendo la privacidad de las mismas, al no permitir reconocer su identidad. Las 3D-CNN permiten realizar de forma automática la extracción de características y clasificación de acciones a partir de la información espacial y temporal de las secuencias de información de profundidad. La evaluación de la propuesta se ha efectuado mediante pruebas experimentales. Para ello se ha empleado la base de datos *"NTU RGB+D Action Recognition Dataset"*, que incluye secuencias de vídeo de color y profundidad, con múltiples personas realizando acciones desde distintos puntos de vista en un entorno interior. Para las pruebas experimentales únicamente se ha empleado la información de profundidad, tanto para el entrenamiento como para la evaluación de la red, obteniéndose resultados que han permitido validar la propuesta.**

I. Introducción

En el contexto de la visión artificial, el reconocimiento de acciones ha cobrado una gran importancia en los últimos años, debido principalmente, a sus múltiples aplicaciones en el estudio del comportamiento humano, la seguridad o la vídeo-vigilancia, por lo que ha atraído la atención de muchos investigadores en los últimos años [1], [2], [3].

Gran parte de los trabajos se basan en uso de cámaras de color [4], [5], sin embargo, en los últimos años han aparecido sensores RGB-D que, además de una imagen de color, proporcionan un mapa de profundidad [6], [7], este hecho ha permitido que surjan un gran número de trabajos que emplean esta tecnología para el reconocimiento de acciones [8], [9], [10]. Estas propuestas proporcionan buenos resultados en condiciones controladas, sin embargo, presentan problemas en escenarios con un alto grado de oclusiones. Además, el hecho de emplear cámaras de color (RGB o RGB-D) implica la existencia de información que permite la identificación de los usuarios, por lo que pueden aparecer problemas relacionados con la privacidad. Por otro lado, los sensores de profundidad [11], [7] permiten obtener información de distancia de cada punto de la escena a la cámara mediante la medida indirecta del tiempo de vuelo de una señal infrarroja modulada. El uso de este tipo de sensores permite preservar la privacidad de las personas, al no ser posible reconocer su identidad con la información proporcionada. Otra ventaja a considerar es que no requieren fuentes de iluminación adicionales (la propia cámara incluye una fuente de iluminación IR).

Por otra parte, cabe destacar que gracias a la mejora experimentada en la tecnología de procesamiento en los últimos años, se han desarrollado con mucha fuerza los trabajos basados en redes neuronales profundas (DNNs) especialmente para aplicaciones de clasificación [12]. En el caso del reconocimiento de acciones, se emplean frecuentemente las redes neuronales recurrentes (RNN) que incluyen la componente temporal para la extracción de características, tanto con arquitecturas *Long Short Term Memory* (LSTMs) [13], [14] como basadas en capas convolucionales 3D [15], [16] utilizando la información de color (RGB) y la de profundidad (*Depth*).

En este contexto, el objetivo del presente trabajo es la detección de acciones, realizando tanto la extracción como la clasificación de características en una única etapa mediante el uso de redes neuronales convolucionales 3D, utilizando únicamente la información de profundidad proporcionada por un sensor de profundidad, basado en tiempo de vuelo (ToF).

Los trabajos basados en *Deep Learning* extraen características comenzando por un nivel de abstracción bajo y aumentando la misma a medida que se avanza. Este tipo de sistemas disponen de dos fases bien diferenciadas, la fase de extracción de características en la cual mediante el uso principalmente de filtros convolucionales entrenados y capas de filtrado como el *Max Pooling* se encargan de aprender y extraer las principales características concernientes al problema de interés, para posteriormente pasar a la fase de predicción en la cual se utilizan dichas características para clasificar o predecir una salida.

En este trabajo, se describe la implementación de un sistema de detección de acciones cuya entrada son vídeos con infor-

mación de profundidad, que se procesan e introducen en una CNN compuesta por una primera fase de extracción de características tridimensionales, que emplea capas Convolucionales 3D y capas de *Max Pooling* para filtrar la información y extraer características espaciales y temporales. Para finalmente emplear dos capas *fully connected* o totalmente conectadas que se encargan de clasificar las acciones consideradas en el entrenamiento de la red.

Tanto el entrenamiento, como la validación y el test, se han realizado con la base de datos *"NTU RGB+D Action Recognition Dataset"* [17], [18], puesta a disposición de la comunidad científica por el ROSE Lab de la Nanyang Technological University de Singapore. Se ha elegido esta base de datos debido a que proporciona un gran número de vídeos, tanto con información RGB como de profundiad, que incluyen numerosas personas realizando diferentes acciones, lo que permite entrenar y validar el sistema propuesto en este trabajo.

En el apartado II se describe la arquitectura de la red neuronal implementada. A continuación, en el apartado III, se expone el método de entrenamiento utilizado. Posteriormente, el apartado IV recoge los principales resultados experimentales obtenidos y finalmente, en el apartado V se incluyen las principales conclusiones del trabajo, así como los posibles líneas de trabajo futuro.

II. Arquitectura de la red

Para la consecución del objetivo marcado, el reconocimiento de acciones humanas en el contexto de la video-vigilancia y la seguridad, se propone el empleo de una red neuronal convolucional 3D (3D-CNN), cuya arquitectura se basa en el proyecto desarrollado por [19]. La propuesta inicial permite la detección de acciones utilizando la base de datos UCF-101 (contiene únicamente vídeos de acciones en RGB), por lo que ha sido necesario la modificación de ésta para permitir el correcto funcionamiento del sistema ante vídeos que incluyen información de profundidad. A continuación, se incluye una breve explicación sobre la operación convolucional 3D y posteriormente se describe la arquitectura completa de la red neuronal utilizada.

A. Convolución 3D

En contraposición a las redes neuronales convolucionales 2D, en las cuáles se realizan las operaciones únicamente sobre la dimensión espacial de las imágenes de entrada, en las 3D-CNN se extraen las características necesarias mediante la aplicación de los filtros convolucionales 3D sobre las dimensiones espacial y temporal de los vídeos de entrada, ya que es necesario conocer el contexto y cambio temporal para poder reconocer correctamente las acciones consideradas. En las capas con filtros de convolución 3D, por tanto, es necesario la aplicación de parches o *kernels* volúmetricos, es decir, tridimensionales que incluyan la componente temporal de las secuencias de imágenes de entrada. En la figura 1 se muestra un ejemplo de la operación de convolución 3D y el aspecto del kernel que se aplica.

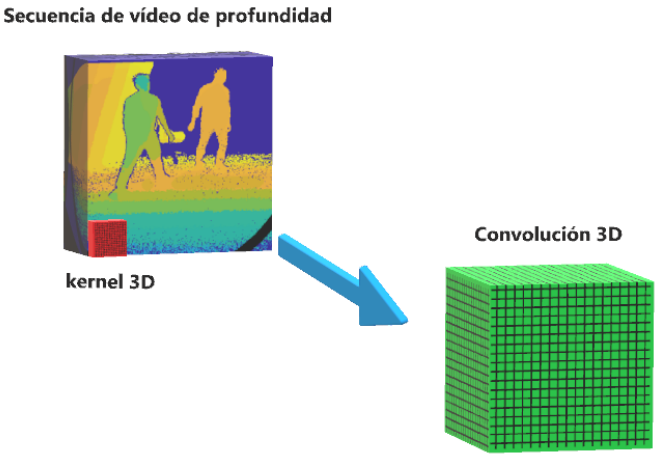

Fig. 1. Ejemplo de operación de convolución 3D sobre una secuencia de vídeo de profundidad en mapa de color, aplicando un kernel (rojo) y resultado de la operación (verde).

La arquitectura completa de la red se muestra en la Figura 2. La entrada a la misma es una secuencia de imágenes de profundidad de dimensiones 64x64 píxeles y con un número de *frames* fijo. En concreto se trata de un fragmento de vídeo de duración 1 segundo (30 *frames*), mostrando la ejecución de una acción determinada. El empleo de secuencias con un número de *frames* reducido, permite una mayor velocidad en el procesado de los vídeos, sin embargo, puede ocurrir que la acción no se muestre completa. En contraposición, el uso de secuencias más largas incrementa la carga computacional, reduciendo la velocidad en su procesado y puede aparecer solapamiento entre diferentes acciones. En este trabajo, se ha realizado un ajuste experimental del número de *frames* para cada secuencia, con el objetivo de alcanzar un equilibro entre el tiempo de cómputo y la precisión del algoritmo, determinando una longitud de las secuencias de 1 segundo (30 *frames*).

La secuencia de entrada se procesa aplicando operaciones de convolución 3D a través de las capas *"Conv3D 1" y "Conv3D 2"* con 32 filtros cada una de ellas. Posteriormente, se aplica una reducción de dimensionalidad del tensor mediante la capa *"Max Pooling 1"* y se vuelve a introducir en filtros convolucionales mediante las capas *"Conv3D 3" y "Conv3D 4"* (64 filtros cada una). A continuación se reduce de nuevo la dimensionalidad con la capa *"Max Pooling 2"* y se introduce el tensor de salida en una capa de *"Flatten"* que permite la adaptación de las capas finales densas *"Dense 1" y "Dense 2"* con 256 y 9 neuronas respectivamente. Finalmente, mediante la aplicación de una capa *"Softmax"* empleada para "comprimir" el vector de salida con valores reales aleatorios en valores en el rango entre [0,1] (ecuación 1), se obtiene una probabilidad por cada acción previamente entrenada. En la expresión 2 se muestra la fórmula aplicada para la obtención de las probabilidades por cada acción.

$$S(a) = [a_1 a_2 ... a_9] \rightarrow [S_1 S_2 ... S_9] \quad (1)$$

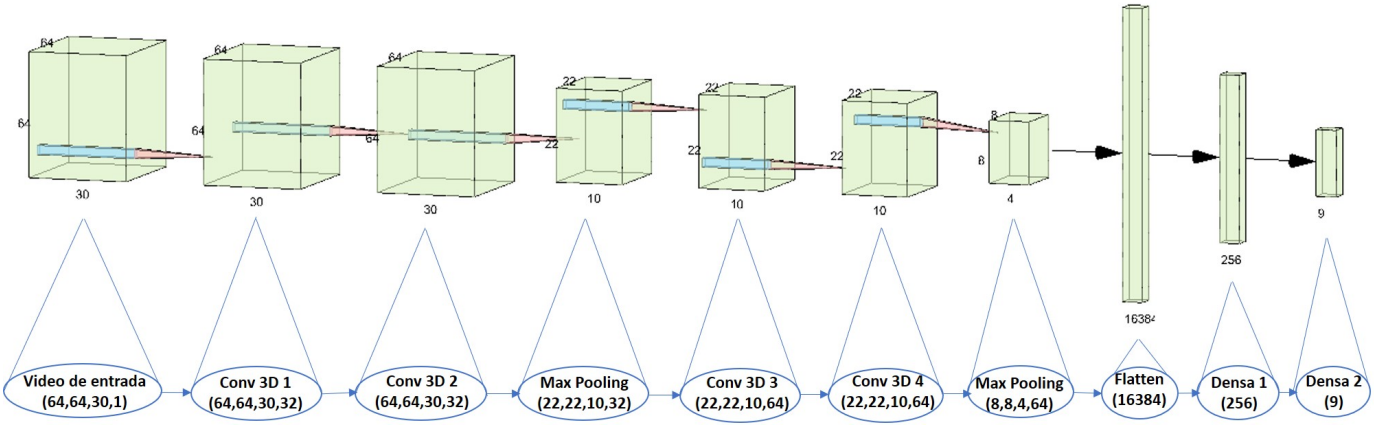

Fig. 2. Arquitectura de la red neuronal convolucional 3D (3D-CNN).

$$S_j = \frac{e^{a_j}}{\sum_{k=1}^{9} e^{a_k}} \qquad j \in 1..9 \tag{2}$$

Tanto las capas convolucionales 3D como las de *Pooling*, se realizan con un parche de dimensiones (3,3,3). La dimensión de las imágenes (64,64,30), no se modifica al aplicar los filtros convolucionales ya que se emplea el *padding* para evitar la reducción de dimensionalidad. A la salida de cada capa, se incluye una función de activación de tipo ReLU (*"Rectified Linear Unit"*). Se ha elegido este tipo de función debido a su alta eficiencia y a que proporciona la no linealidad necesaria para la resolución del reconocimiento de acciones. Además, se emplea la técnica de regulación de *Dropout* que consiste en ignorar nodos de manera aleatoria con el objetivo de evitar las interdependencias entre nodos, es decir, los nodos no aprenden funciones que se basan en valores de entrada de otro nodo, por tanto, el *Dropout* permite a la red aprender una relación más sólida.

En la Tabla I se muestra un resumen de las diferentes capas que conforman la red, así como los tamaños de salida y los parámetros fundamentales.

Capas red neuronal convolucional 3D		
Capa	**Tamaño de salida**	**Parámetros**
Entrada	$64 \times 64 \times 30 \times 1$	-
Conv3D 1	$64 \times 64 \times 30 \times 32$	kernel=(3, 3, 3) / strides=(1, 1, 1)
Activación	ReLU	
Conv3D 2	$64 \times 64 \times 30 \times 32$	kernel=(3, 3, 3) / strides=(1, 1, 1)
Activación	ReLU	
MaxPooling	$22 \times 22 \times 10 \times 32$	size=(3, 3, 3)
Dropout	0.25	
Conv3D 3	$22 \times 22 \times 10 \times 64$	kernel=(3, 3, 3) / strides=(1, 1, 1)
Activación	ReLU	
Conv3D 4	$22 \times 22 \times 10 \times 64$	kernel=(3, 3, 3) / strides=(1, 1, 1)
Activación	ReLU	
MaxPooling	$8 \times 8 \times 4 \times 64$	size=(3, 3, 3)
Dropout	0.25	
Flatten	16384	-
Densa 1	256	-
Activación	ReLU	
Dropout	0.5	
Densa 2	9	-
Activación	SoftMax	

Tabla I
ARQUITECTURA DE LA RED Y TAMAÑO DE LOS TENSORES DE CADA CAPA.

III. ENTRENAMIENTO

Como ya se ha comentado en la Introducción, se ha empleado la base de datos *NTU* [17], [18] tanto para el entrenamiento como para la evaluación de la red, ya que proporciona una gran cantidad de vídeos de acciones con información de profundidad. La base de datos *"NTU RGB+D Action Recognition"* está compuesta por 56.880 muestras de vídeo de una o varias personas ejecutando una determinada acción.Por cada secuencia se incluyen: vídeos RGB, mapas de profundidad, esqueletos en 3D e imagen de infrarrojos. Se han obtenido a partir de la grabación simultánea de 3 cámaras *"Microsoft Kinect v.2"* para proveer a la base de datos de diferentes puntos de vista. La resolución de los vídeos RGB es de 1920x1080 píxeles, mientras que los mapas de profundidad y los vídeos de infrarrojos son tienen una resolución de 512x424 píxeles. Por otra parte, los datos de esqueletos en 3D contienen las localizaciones en 3 dimensiones de las principales 25 articulaciones del cuerpo para cada frame. La base de datos contiene 60 clases de acciones que se pueden agrupar en 3 grandes grupos: acciones diarias, acciones mutuas y condiciones médicas. En el presente trabajo se han seleccionado 9 acciones, que se pueden dar en el contexto de la vídeo-vigilancia y la seguridad. Las acciones escogidas junto con el número de fragmentos empleados tanto para el entrenamiento y validación como para el test se muestran en la tabla II.

En total, se han empleado 8532 secuencias de mapas de profundidad, con distintos individuos realizando las acciones descritas en la tabla II, desde diferentes puntos de vista. En la figura 3 se muestran algunos ejemplos de las imágenes de profundidad utilizadas. En esta figura se ha utilizado una escala de color para representarlos diferentes valores de

profundidad medidos. Además, como se puede observar, en la información proporcionada se ha eliminado el fondo de las escenas.

De las secuencias disponibles, se ha utilizado el 80% para entrenamiento (6822), que a su vez de han dividido de forma que: el 80% se emplea para el entrenamiento y el 20% para validar. El 20% restante del total (1710) se emplea para la etapa de test y la extracción de resultados. El empleo de muestras con una alta variabilidad, es decir, un número considerable de individuos realizando acciones desde diferentes puntos de vista, permite aumentar el nivel de generalización de la red y, por tanto, la robustez del sistema.

Acciones utilizadas de la base de datos NTU				
Acción	**Cód.**	**Entren.**	**Validación**	**Test**
Propinar un puñetazo	A01	607	151	190
Levantarse	A02	607	151	190
Caerse	A03	607	151	190
Sentarse	A04	607	151	190
Separarse de una persona	A05	607	151	190
Propinar una patada	A06	607	151	190
Empujar	A07	607	151	190
Andar hacia una persona	A08	607	151	190
Lanzar un objeto	A09	607	151	190
Total	-	5463	1359	1710

Tabla II
DATOS UTILIZADOS EN LA ETAPA DE ENTRENAMIENTO DE LA RED.

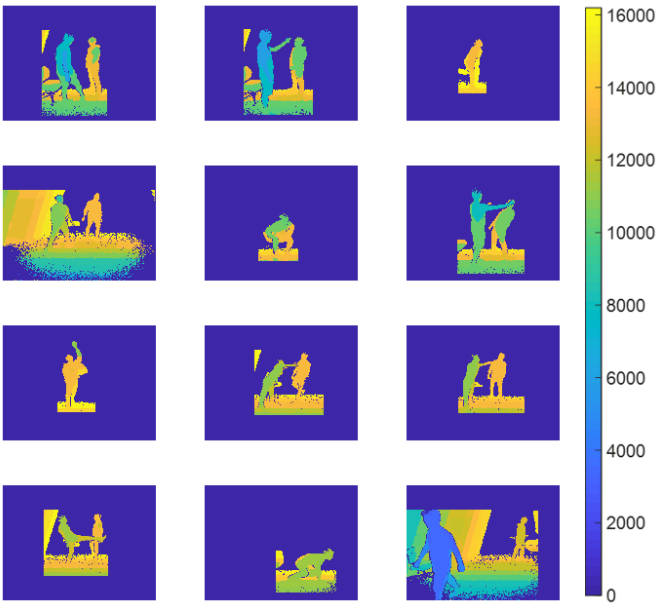

Fig. 3. Ejemplos de escenas de profundidad pertenecientes a diferentes secuencas de vídeo de la base de datos NTU, con varios individuos realizando una serie de acciones desde diferentes puntos de vista.

Puesto que el entrenamiento de la red neuronal se ha realizado con fragmentos de vídeo con información de profundidad de duración 1 segundo y con una resolución de 64x64 píxeles, ha sido necesario su procesado, tanto para la extracción de los 30 frames de los citados vídeos para la conversión en un tensor de dimensión (512,424,30,1) como el re-escalado de las imágenes para finalmente obtener un tensor de (64,64,30,1) que puede utilizarse como entrada a la red. Tanto el procesado anteriormente descrito, como la generación de los *batches* a introducir a la red para su entrenamiento, se lleva a cabo mediante una función generador que realiza una preparación de los datos de manera concurrente y en paralelo al proceso de entrenamiento de la red neuronal, asegurando la aleatoriedad en la selección de los vídeos destinados al entrenamiento. El tamaño de *batch* elegido es 32, seleccionado de forma experimental para optimizar la precisión en la clasificación de las acciones. En la figura 4, en la que se muestra la precisión obtenida para diferentes valores del tamaño de *batch*, puede observarse que el mejor resultado se obtiene en el caso de utilizar una tamaño de *batch* de 32.

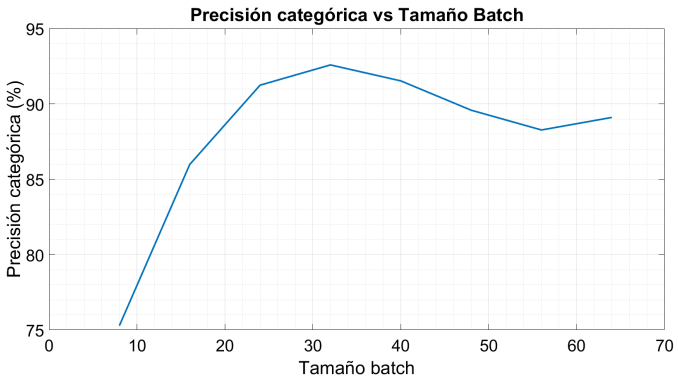

Fig. 4. Precisión obtenida en función del tamaño de *batch* seleccionado.

El número de épocas seleccionado es de 50, dado que tal como se puede observar en la curva de entrenamiento permite a la CNN alcanzar el regimen oscilatorio final que marca la precisión máxima a la que puede llegar. Tal y cómo se observa en la Figura 5 y 6 el entrenamiento no muestra rasgos de sobreentrenamiento dado que la precisión categórica de validación tiene un valor cercano a la de entrenamiento, lo cual permite deducir que se ha realizado una correcta generalización del entrenamiento por parte de la red neuronal. El algoritmo de optimización empleado es *Adam* [20], que ha sido elegido por sus capacidades adaptativas en cuanto al ratio de entrenamiento o *learning rate*. La gran ventaja de Adam es el hecho de que parte de un ratio de entrenamiento inicial marcado por el usuario para posteriormente emplear los primeros y segundos momentos del gradiente para adaptar el ratio del entrenamiento a la situación estipulada por Adam en la función de pérdidas. Esto brinda una velocidad y robustez superior respecto a otras alternativas, convirtiendo a Adam en una buena opción para el problema aquí propuesto.

Por último, la función de pérdidas empleada para obtener la diferencia entre los datos reales etiquetados y la salida predicha por la red durante el proceso de entrenamiento es la denominada *categorical crossentropy* (CCE)definida según la ecuación 3.

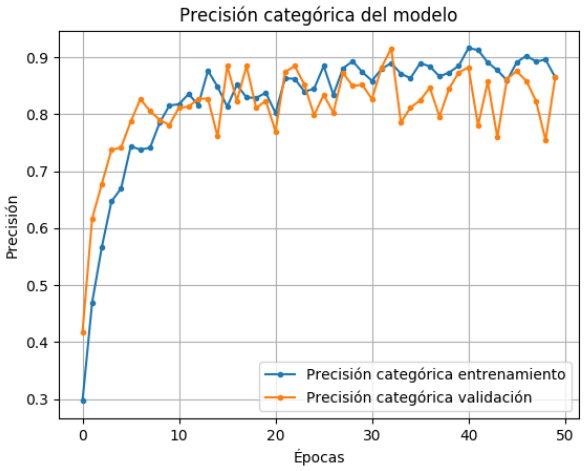

Fig. 5. Precisión categórica obtenida en cada época durante el entrenamiento.

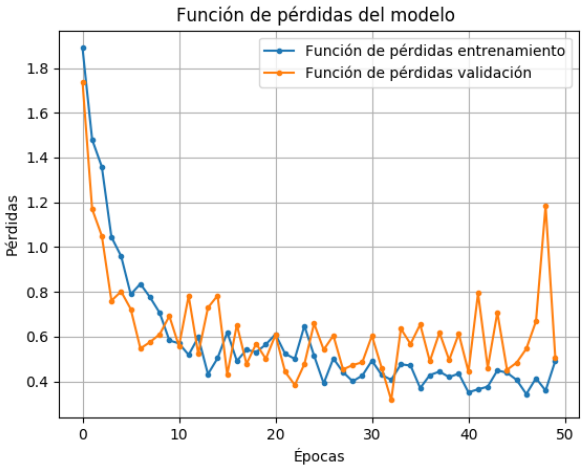

Fig. 6. Función de pérdidas en cada época durante el entrenamiento.

$$CCE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right) \tag{3}$$

La función de pérdidas CCE se aplica entre la salida etiquetada, cuyo formato es un vector de 9 elementos, uno por cada clase de acción, de tipo *one hot*, es decir, el elemento correspondiente a la acción de entrada toma el valor 1 y los demás el valor 0, y el vector de salida de la red proporcionada por la función de activación *SoftMax* que incluye una probabilidad por cada acción.

IV. Resultados experimentales

La evaluación del sistema de reconocimiento de acciones implementado se ha llevado a cabo, nuevamente, a través de la base de datos NTU, con los vídeos destinados para test.

La obtención de la precisión del sistema de reconocimiento de acciones se ha realizado mediante la comparación de la posición del valor "1" de los vectores de salida etiquetados en formato *one hot* y la posición con mayor probabilidad de los vectores de salida de la red neuronal, siempre y cuando dicha probabilidad sea superior a 0.5 (50%). El procedimiento descrito anteriormente, se resumen en el algoritmo 1:

```
Inicialización: Vector de salida real;
Vector de salida predicha;
Aciertos = 0;
for Número de vídeos de test do
    PosVectorReal = MaxPos(Vector de salida real);
    PosVectorRed = MaxPos(Vector de salida
     predicha);
    if PosVectorReal == PosVectorRed then
        if MaxValor(Vector de salida predicha) ≥ 0.5
         then
            Aciertos+=1;
        end
    end
end
Precisión = Aciertos/Número de vídeos de test;
```

Algorithm 1: Obtención de precisión categórica

Tras la aplicación del algoritmo descrito, se obtiene un valor de precisión del 92.8%.

Además, en la figura 7 se muestra la matriz de confusión obtenida para las 9 acciones consideradas.

En esta figura, se representan las acciones por su código de acción definido en la Tabla II. En las filas de la matriz se representa la clase real, es decir, los vídeos de acción que se encuentran etiquetados, y en las columnas, se representa la clase predicha, la predicción que proporciona la red neuronal como salida ante un determinado vídeo de entrada.

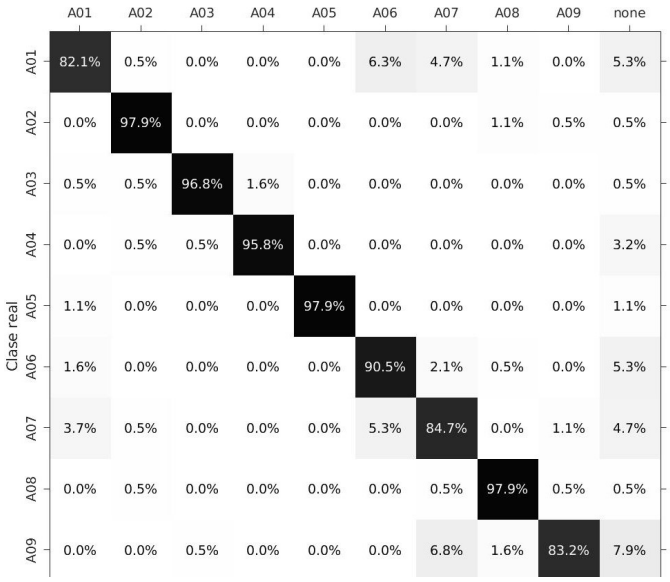

Fig. 7. Matriz de confusión del sistema de clasificación de acciones con información de profundidad basado en 3DCNN.

Analizando la matriz de confusión, se observa que la mayor parte de las clases presentan precisiones elevadas (por

encima del 90%). Las acciones más problemáticas son las de *"propinar un puñetazo"* (A01),*"empujar"* (A07) y la de *"lanzar un objeto"*, entre las que existe cierta confusión, debido principalmente a la dificultad que presenta la base de datos utilizada, al incluir cambios importantes de punto de vista. Así como a la complejidad de la tarea, al utilizar únicamente información de profundidad. Sin embargo, a pesar de este hecho, se consigue un valor de precisión global en la clasificación elevado (92.8%), tal como se ha comentado previamente.

Por último, se puede extraer que el empleo de un generador garantiza la correcta aleatoriedad tanto en los vídeos destinados al propio entrenamiento como para la validación de los pesos calculados al finalizar una época, por lo que tiene como consecuencia un mayor porcentaje de acierto al clasificar las acciones, ya que se consigue una mayor generalización y, por tanto, aumentar la robustez del sistema.

V. Conclusiones y trabajos futuros

En este trabajo se presenta un sistema para reconocimiento de acciones empleando únicamente información de profundidad. La propuesta se basa en la implementación y el entrenamiento de una Red Neuronal Convolucional 3D (3D-CNN) que desarrolla las operaciones de convolución sobre secuencias de información de profundidad a partir de un parche volumétrico previamente definido.

El uso de información de profundidad permite determinar las acciones realizadas por diferentes personas en la escena, preservando su privacidad, al no ser posible reconocer la identidad de cada individuo, por lo que puede ser utilizado incluso en entornos en los que existan requisitos relacionadas con la privacidad, por ejemplo hospitales, centros de mayores, etc.

La propuesta presentada ha sido evaluada experimentalmente de forma exhaustiva, empleando una base de datos pública (disponible para investigación). Los resultados obtenidos han permitido validar la propuesta al tener una precisión global en la clasificación de acciones superior al 92%, a pesar de la complejidad de la base de datos.

Se trata de un trabajo abierto, con múltiples líneas de trabajo futuro entre las que destacan la optimización de las técnicas de entrenamiento que permitan el incremento de la precisión en la clasificación de acciones y la implementación de un sistema en un entorno realista, que permita la validación de la propuesta con escenas de video-vigilancia.

Agradecimientos

Este trabajo ha sido posible gracias a la financiación del Ministerio de Economía y Competitividad a través del proyecto HEIMDAL-UAH (TIN2016-75982-C2- 1-R), por la Universidad de Alcalá mediante los proyectos JANO (CCGP2017/EXP-025) y ACERCA (CCG2018/EXP-029) y las "Becas de Colaboración" del Ministerio de Educación.

Referencias

[1] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[2] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer vision and image understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[3] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4405–4425, 2017.

[4] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1234–1241.

[5] J. Zhang, Y. Han, J. Tang, Q. Hu, and J. Jiang, "Semi-supervised image-to-video adaptation for video action recognition," *IEEE transactions on cybernetics*, vol. 47, no. 4, pp. 960–973, 2017.

[6] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[7] J. Sell and P. O'Connor, "The Xbox one system on a chip and Kinect sensor," *Micro, IEEE*, vol. 34, no. 2, pp. 44–53, Mar 2014.

[8] N. Ashraf, C. Sun, and H. Foroosh, "View invariant action recognition using projective depth," *Computer Vision and Image Understanding*, vol. 123, pp. 41–52, 2014.

[9] A.-A. Liu, W.-Z. Nie, Y.-T. Su, L. Ma, T. Hao, and Z.-X. Yang, "Coupled hidden conditional random fields for rgb-d human action recognition," *Signal Processing*, vol. 112, pp. 74 – 82, 2015, signal Processing and Learning Methods for 3D Semantic Analysis. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168414004022

[10] P. Khaire, P. Kumar, and J. Imran, "Combining cnn streams of rgb-d and skeletal data for human activity recognition," *Pattern Recognition Letters*, vol. 115, pp. 107 – 116, 2018, multimodal Fusion for Pattern Recognition. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865518301636

[11] R. Lange and P. Seitz, "Solid-state time-of-flight range camera," *Quantum Electronics, IEEE Journal of*, vol. 37, no. 3, pp. 390–397, Mar 2001.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[13] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[14] S. Das, M. Koperski, F. Bremond, and G. Francesca, "A fusion of appearance based cnns and temporal evolution of skeleton with LSTM for daily living action recognition," *CoRR*, vol. abs/1802.00421, 2018. [Online]. Available: http://arxiv.org/abs/1802.00421

[15] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan 2013.

[16] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, and S. J. Maybank, "Asymmetric 3d convolutional neural networks for action recognition," *Pattern Recognition*, vol. 85, pp. 1 – 12, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320318302632

[17] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[18] "NTU RGB+D Action Recognition dataset," disponible online: http://rose1.ntu.edu.sg/datasets/actionrecognition.asp (Último acceso 24/01/2019).

[19] dipakkr, "3d-cnn-action-recognition," https://github.com/dipakkr/3d-cnn-action-recognition, 2018.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980