elektron ámbar

Original Article

Revista elektron, Vol. 5, No. 1, pp. 37-44 (2021)
https://doi.org/10.37537/rev.elektron.5.1.130.2021

CNN–LSTM con Mecanismo de Atención Suave para el Reconocimiento de Acciones Humanas en Videos

CNN–LSTM with Soft Attention Mechanism for Human Action Recognition in Videos

Carlos Ismael Orozco*[1], María Elena Buemi†[2] and Julio Jacobo-Berlles†[3]

*Departamento de Informática, Facultad de Ciencias Exactas, Universidad Nacional de Salta.
Avda. Bolivia 5150. Salta, Argentina.
[1] iorozco@exa.unsa.edu.ar

†Departamento de Computación. Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires
Intendente Güiraldes, Pabellón 1 2160, 1428 Buenos Aires, Argentina.
[2] mebuemi@dc.uba.ar
[3] jacobo@dc.uba.ar

Recibido: 31/03/21; Aceptado: 11/06/21

***Resumen*—El reconocimiento de acciones en videos es actualmente un tema de interés en el área de la visión por computador, debido a potenciales aplicaciones como: indexación multimedia, vigilancia en espacios públicos, entre otras. Los mecanismos de atención se han convertido en un concepto muy importante dentro del enfoque de aprendizaje profundo, su operación intenta imitar la capacidad visual de las personas que les permite enfocar su atención en partes relevantes de una escena para extraer información importante. En este artículo proponemos un mecanismo de atención suave adaptado para degradar la arquitectura CNN–LSTM. Primero, una red neuronal convolucional VGG16 extrae las características del video de entrada. Para llevar a cabo las fases de entrenamiento y prueba, usamos los conjuntos de datos HMDB-51 y UCF-101. Evaluamos el desempeño de nuestro sistema usando la precisión como métrica de evaluación, obteniendo $40{,}7\,\%$ (enfoque base), $51{,}2\,\%$ (con atención) para HMDB-51 y $75{,}8\,\%$ (enfoque base), $87{,}2\,\%$ (con atención) para UCF-101.**

Palabras clave: reconocimiento de acciones; redes neuronales convolucionales; redes neuronales lstm; mecanismo de atención.

***Abstract*— Action recognition in videos is currently a topic of interest in the area of computer vision, due to potential applications such as: multimedia indexing, surveillance in public spaces, among others. Attention mechanisms have become a very important concept within deep learning approach, their operation tries to imitate the visual capacity of people that allows them to focus their attention on relevant parts of a scene to extract important information. In this paper we propose a soft attention mechanism adapted to a base CNN–LSTM architecture. First, a VGG16 convolutional neural network extracts the features from the input video. Then an LSTM classifies the video into a particular class. To carry out the training and testing phases, we used the HMDB-51 and UCF-101 datasets. We evaluate the performance of our system using accuracy as an evaluation metric, obtaining $40{,}7\,\%$ (base approach), $51{,}2\,\%$ (with attention) for HMDB-51 and $75{,}8\,\%$ (base approach), $87{,}2\,\%$ (with attention) for UCF-101.**

Keywords: action recognition, convolutional neural network, long short-term memory, attention mechanism.

I. Introducción

Dada una secuencia de observaciones X y una etiqueta particular Y, los sistemas de reconocimiento de acciones humanas (HAR) en general modelan la probabilidad a posteriori $P(Y|X)$ para aprender una relación que vincula las entradas X a su correspondiente etiqueta de clase Y, es decir, dada una lista de posibles acciones y un video en el que se muestra a un actor llevando a cabo una de ellas, el objetivo del problema es reconocer qué acción está siendo ejecutada. Para un ser humano reconocer una determinada acción empleando únicamente su campo visual es un problema simple, sin embargo, implementar una solución automática es un problema complejo debido a los numerosos factores implicados, como la gran diversidad existente entre las personas tanto en su apariencia (constitución física, ropas, ...) como en el estilo de ejecución de la acción; el escenario donde se realiza, a menudo se trata de entornos concurridos, afectados por sombras, cambio de iluminación y oclusiones; además, intervienen otros factores como el ángulo de visión y la distancia del sujeto respecto a la cámara. Las acciones humanas llevan asociadas una componente espacial y una temporal, ambas altamente aleatorias, por lo que la realización de una misma acción nunca es idéntica. Para una guía completa de los desafíos actuales del problema, debe leerse el trabajo de Jegham et al. [1].

HAR es un tema de gran interés en el campo de reconocimiento de patrones y visión por computadora ya que la identificación automática de la acción ejecutada en un video puede ser una herramienta valiosa para muchas aplicaciones:

- Un ejemplo común es el análisis de videos de vigilancia [2]. Muchos sistemas de seguridad se basan en los datos capturados por varias cámaras. Cuando el número de cámaras es grande, puede ser difícil, o incluso imposible, detectar manualmente eventos importantes en los videos.

- Una aplicación relacionada al inciso anterior es el uso de técnicas de comprensión de videos para el cuidado de ancianos y niños en predios cerrados como las casas y los hospitales inteligentes.
- El monitoreo y el reconocimiento automático de actividades diarias puede ser de gran ayuda en la asistencia de los residentes, así como en la obtención de informes acerca de sus capacidades funcionales y su salud [3].
- Otra aplicación relacionada es el resumen automático de vídeos, que intenta obtener vídeos cortos a partir de las escenas importantes del video original. La búsqueda basada en contenido en bases de datos de videos. La habilidad de obtener de manera automática descripciones textuales de un video dado evita la necesidad de realizar anotaciones manuales, y puede ser crucial para el desarrollo de bases de datos más útiles e informativas.
- La interacción humano-computadora [4] es otro campo de aplicación que se beneficia de las mejoras en las técnicas de reconocimiento de acciones. Por ejemplo, dichas técnicas pueden ser usadas para proveer interfaces para personas con movilidad reducida, facilitando su interacción con computadoras y con otras personas.
- Otro ejemplo es el desarrollo de videojuegos [5] que permiten que el usuario interactúe con la consola/computadora sin la necesidad de usar un dispositivo físico.
- La biometría basada en comportamiento ha recibido también mucha atención en los últimos años. A diferencia de las herramientas biométricas clásicas (como las huellas digitales), las técnicas basadas en comportamiento obtienen datos para identificación sin interferir con la actividad de la persona. Un ejemplo típico es la identificación a partir del modo de andar de las personas.
- Una aplicación relacionada es el desarrollo de herramientas que guíen de manera automática a pacientes en rehabilitación con problemas motrices.

En resumen, nuevos desarrollos en métodos de reconocimiento de acciones en videos, pueden ser de gran interés para una amplia gama de aplicaciones.

El objetivo de este trabajo es implementar un sistema de reconocimiento de acciones en video. Para ello proponemos: (1) trabajar sobre una arquitectura CNN–LSTM, esto es, una red neuronal convolucional extrae las características del video, mientras que una red neuronal LSTM clasifica el video en una categoría determinada. (2) incluir mecanismo de atención sobre esta propuesta base. El trabajo está organizado de la siguiente manera: en la sección II se presenta un resumen del estado del arte del problema en cuestión; en la sección III, se describe la estructura general base del sistema; en la sección IV se presenta el mecanismo de atención. En la sección V se explican las bases de datos utilizadas, el método de evaluación, los experimentos realizados y los resultados obtenidos. Finalmente, en la sección VI se presentan las conclusiones y el trabajo futuro.

II. Estado del Arte

Los métodos utilizados en la literatura para abordar el problema de reconocimiento de acciones humanas puede ser clasificado en tres grandes grupos:

II-A. Enfoques clásicos

Los enfoques clásicos consisten en extraer descriptores, tanto para la apariencia como para la información de movimiento de los fotogramas de video.

- Métodos basados en Puntos de Interés Espacio-Temporal (STIP), captura puntos del video en el dominio espacio-temporal. Un punto de interés puede ser detectado de forma robusta mediante un detector basado en STIP, por ejemplo, un punto de esquina o un punto aislado donde la intensidad es máxima o mínima o el punto final de una línea o el punto de una curva donde la curvatura es máxima. Laptev et al. [6] generaliza el detector de bordes de Harris [7] a un detector 3D-Harris. STIP son invariantes a la translación y a escala, sin embargo no son invariantes a la rotación.
- Métodos basados en trayectorias utilizan la ruta de seguimiento de puntos característicos para representar las acciones. Wang et al. [8] proponen un enfoque de trayectorias densa. Primero, se muestrea nubes de puntos característicos de cada fotograma del video, luego, la información de desplazamiento se calcula rastreando dichos puntos característicos entre los fotogramas empleando un enfoque de flujo óptico. Las trayectorias resultantes se utilizan para representar los videos. En esta línea de investigación, se comprobó que corregir el movimiento de la cámara, es decir, hacer coincidir los puntos característicos entre fotogramas utilizando descriptores *funciones robustas aceleradas* (SURF [9]) y también combinarlos con otros descriptores locales *histograma de gradientes orientados* (HOG [10]), *histograma de flujo óptico* (HOF [11]) e *histograma de límites de movimiento* (MBH [12]) han logrado muy buenos resultados [13] en entornos controlados.

II-B. Enfoques profundos

En los últimos años, la aplicación del aprendizaje profundo en problemas de visión por computadora ha obtenido resultados muy destacados. El éxito clave de las redes neuronales artificiales reside en su capacidad para aproximarse a cualquier función continua con suficientes neuronas [14]. Las redes neuronales profundas deben contener una cantidad suficiente de capas y neuronas para aprender mapeos significativos de entradas y salidas de manera efectiva. Aprenden representaciones jerárquicas a partir de datos sin procesar con un nivel creciente de abstracción antes de la etapa de clasificación.

Las Redes Neuronales Convolucionales (CNNs) fueron diseñadas para manejar datos espaciales, a pesar de su éxito en la tarea de clasificación de imágenes, tienen el problema de ignorar la estructura temporal en el caso de datos espacio-temporales como videos. Particularment, las CNN-3D son una extensión de las CNNs en el dominio del tiempo que no solo puede capturar características

espaciales en un fotograma, sino también capturar la evolución temporal entre fotogramas consecutivos. Ji et al. [15] proponen un enfoque 3D-CNN, extracción de las características de los datos en dos dimensiones: espacial y temporal, capturando así información de movimiento en las transmisiones de video. Esta arquitectura genera múltiples mapas de características procesando información de los fotogramas consecutivos del video realizando operaciones de convolución y submuestreo por separado en cada canal. La representación de la característica final se obtiene combinando todos los canales. Los resultados experimentales muestran que los modelos propuestos superan significativamente la arquitectura 2D-CNN y otros métodos clásicos. Cabe mencionar que la cantidad de fotogramas consecutivos que se toma es fijo para cada operación de convolución, por lo que se considera un problema a la hora de generalizar dicho mapas de características.

Las Redes Neuronales Recurrentes (RNNs) se utilizan para aprender dinámicas temporales complejas. Ellas han sido exploradas exitosamente en muchas tareas como por ejemplo procesamiento de texto y reconocimiento de voz. Como las acciones humanas están hechas de secuencias complejas de movimientos motores que pueden verse como dinámica temporal, las RNN representan una solución adecuada. Las Redes Neuronales de Corta y Larga memoria (LSTM) [16] son una arquitectura RNN específica que tienen la capacidad de usar celdas de memoria para almacenar, modificar y acceder al estado interno para descubrir dependencias temporales de largo alcance.

- Ye et al. [17] implementan una arquitectura híbrida donde las características espacio temporales se extraen de los videos utilizando redes convolucionales 3D, y luego se aplica una red neuronal LSTM para incrustar la información secuencial en la representación de características finales del video.
- Zhang et al. [18], que utilizó el vector de movimiento en la secuencia de video en lugar de la secuencia de flujo óptico para mejorar la velocidad de cálculo y realizar el reconocimiento de acciones humanas en tiempo real.
- Sharma et al. [19] proponen una red neuronal LSTM con un mecanismo de atención que permite a cada fotograma del video enfocarse en una región que es más distintiva para la tarea en cuestión. Aprender tales pesos forma parte del entrenamiento del modelo.

II-C. Enfoques doble-flujo

Goodale et al. [20] describe la hipótesis de dos corrientes, donde la corteza visual humana contiene dos vías: la corriente ventral (que realiza el reconocimiento de objetos) y la corriente dorsal (que reconoce el movimiento).

- Simonyan et al. [21] presentan una red de dos flujos que contienen una red espacial y temporal mediante la explotación del conjunto de datos ImageNet para el preentrenamiento y el cálculo del flujo óptico para capturar explícitamente información de movimiento.
- Feichtenhofer et al. [22] implementan una red de dos flujos con arquitectura ResNet [23] y conexiones adicionales entre flujos [24]. Los enfoques adicionales de dos flujos incluyen Redes de segmentos temporales [25], Acción Transformaciones [26] y Fusión convolucional [27].

III. Enfoque CNN-LSTM Base

Sea $v = \{x_1, x_2, \ldots, x_n\}$ un video compuesto por una secuencia de frames x_i con $i = 1, \ldots, n$, la Figura 1 muestra un sistema HAR base formado por: Fase de entrada: el video es normalizado en un total de 40 frames. Fase CNN: Una VGG16 pre-entrenada extrae las características del video obteniendo features de tamaño 40×25088. Fase LSTM junto con una capa densa con un nodo por cada clase para la clasificación final. En el siguiente repositorio se puede encontrar el código para la arquitectura propuesta[1].

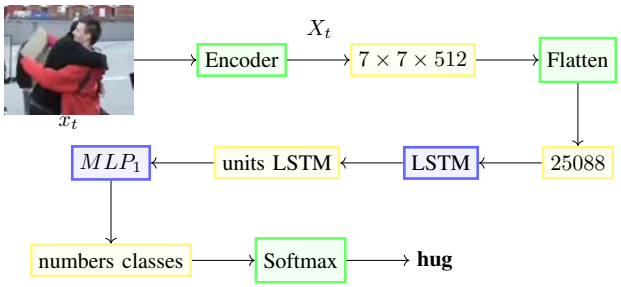

Figura 1: Arquitectura base CNN–LSTM propuesta por [28].

Los bloques de color que constituyen la arquitectura base CNN–LSTM, mostrados en la Figura 1, son:

- **Encoder**: Usamos la arquitectura convolucional VGG16 propuesta por [21]. Para cada $x_t \in v$ codificamos el fotograma en un cuboide X_t de tamaño $7 \times 7 \times 512$ resultante de la capa de submuestreo VGG16.
- **LSTM**: Propuesto por [16] tiene como comportamiento natural, recordar información durante largos períodos de tiempo. Las entradas para un tiempo específico t son: fotograma x_t, estado anterior h_{t-1} y memoria anterior c_{t-1}. Mientras que las salidas son: estado actual h_t y memoria actual c_t.
- **MLP_1**: El perceptrón multicapa está formado por tres o más capas: una entrada, otra salida y el resto de capas intermedias llamadas capas ocultas. Los detalles de la etapa de clasificación se presentan en la Tabla II (primera fila).
- : Indica la dimensión de salida.

Cabe señalar que los pesos asociados con el LSTM, como el MLP, serán parte del entrenamiento de arquitectura.

III-A. Extracción de Características

Las CNNs están compuestas por un conjunto ordenado de capas. Cada una de ellas, a su vez, está constituida por unidades de procesamiento que operan sobre la salida de la capa anterior. Las capas más utilizadas son: (a) capas convolucionales, que tienen k filtros (o kernels) dedicados a producir k feature maps. (b) capa de submuestreo: Cada feature map se submuestrea por lo general mediante una operación de max-pooling, un proceso que reduce progresivamente el tamaño espacial de la representación y la

[1] https://gitlab.com/ciorozco/actionrecognition-cnn-lstm

cantidad de parámetros a entrenar. (c) Capa densa: capa con neuronas totalmente conectadas. El propósito es usar estas características es clasificar la imagen de entrada en una de varias clases según el conjunto de datos de entrenamiento.

La arquitectura convolucional VGG16 propuesta por [21] obtuvo muy buenos resultados en las tareas de clasificación y ubicación en el desafío de reconocimiento visual a gran escala ImageNet (ILSVRC-2014). El Cuadro I muestra las capas que componen esta arquitectura, la primera columna indica el número de capa y el tipo de operación (por ejemplo: 2 × Conv: Convolución, Max Pool: Max Polling, FC: Fully Connected). La segunda columna indica la cantidad de Feature maps. Size el tamaño de los features de salida de la capa. Kernel y Stride parámetros de la arquitectura.

Cuadro I: Implementación de VGG utilizando el modelo pre-entrenado [29].

Capa		Feature map	Size	Kernel
Entrada	Imagen	1	$224 \times 224 \times 3$	-
1	$2 \times$ Conv	64	$224 \times 224 \times 64$	3×3
	Max pool	64	$112 \times 112 \times 64$	3×3
3	$2 \times$ Conv	128	$112 \times 112 \times 128$	3×3
	Max pool	128	$56 \times 56 \times 128$	3×3
5	$2 \times$ Conv	256	$56 \times 56 \times 256$	3×3
	Max pool	256	$28 \times 28 \times 256$	3×3
7	$3 \times$ Conv	512	$28 \times 28 \times 512$	3×3
	Max pool	512	$14 \times 14 \times 512$	3×3
10	$3 \times$ Conv	512	$14 \times 14 \times 512$	3×3
	Max pool	512	$7 \times 7 \times 512$	3×3
13	FC	-	25088	-
14	FC	-	4096	-
15	FC	-	4096	-
Salida	FC	-	1000	-

Por cada $x_i \in v$ codificamos el frame a un cuboide de forma X_i de $7 \times 7 \times 512$ resultado de la capa de submuestreo de la VGG16.

III-B. Clasificación

Las redes LSTM propuesta por [16] tiene como comportamiento natural recordar información por largos períodos de tiempo. La Figura 2 muestra la estructura general de una unidad LSTM. Las entradas para un tiempo específico t son: fotograma x_t, salida previa h_{t-1} y memoria previa c_{t-1}. Mientras que las salidas son: salida actual h_t y memoria actual c_t.

El LSTM tiene la capacidad de agregar o quitar información a esta celda de memoria usando puertas (o gates). Las puertas permiten que el sistema deje pasar información opcionalmente, actualice la celda de memoria o deje salir la información.

El primer paso en el LSTM es decidir qué información se mantendrá en la celda de memoria. Esta decisión es tomada por la puerta de olvido, que en el momento t mira la salida del bloque de memoria en el momento $t-1$, h_t, en la secuencia de entrada en el momento t, x_t, y en el estado anterior de la celda de memoria, c_{t-1}. La ecuación 1 muestra cómo la puerta de olvido calcula su valor.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (1)$$

El siguiente paso es decidir qué información se agregará a la celda de memoria. Esto se hace en dos pasos diferentes; el

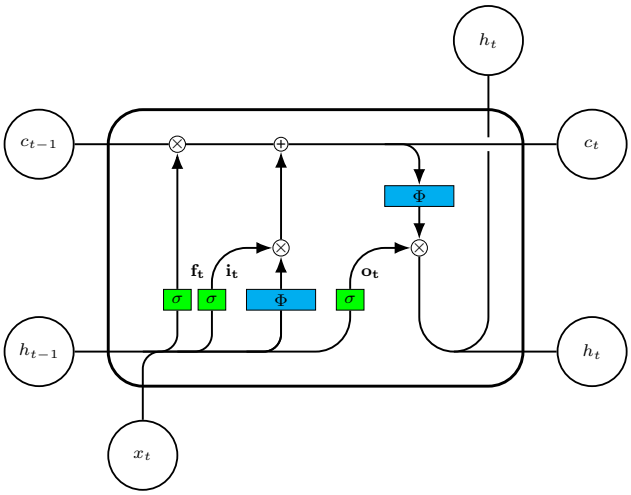

Figura 2: Unidad LSTM.

primer paso mira la entrada a la red neuronal y la salida del bloque LSTM en el tiempo $t-1$ para calcular el vector que actualizará la celda de memoria, mientras que el segundo paso es calcular la puerta de entrada, que es muy similar a la puerta del olvido, pero esta vez nos dice qué cantidad de esta nueva información se dejará entrar en la celda de memoria. Las ecuaciones 2 y 3 muestran estos cálculos.

$$z_t = \phi(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (2)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (3)$$

Una vez hemos calculado todos estos valores tenemos todos los elementos necesarios para actualizar la celda de memoria, así que procedemos a hacerlo. Primero, el valor antiguo de la celda de memoria se multiplica por la puerta de olvido para olvidar toda la información que la puerta de olvido decidió olvidar. Luego, otorgamos acceso a la nueva información en la celda agregando esta nueva información escalada por cuánto decidió la puerta de entrada actualizar la celda de memoria. Ambas cosas se hacen en un solo paso como se muestra en la ecuación 4.

$$c_t = f_t \otimes c_{t-1} \oplus i_t \otimes \phi(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (4)$$

Finalmente, necesitamos decidir qué información vamos a generar. La salida del bloque LSTM es el valor de la celda de memoria con ligeras modificaciones. Primero ejecutamos una activación sigmoidea, llamada puerta de salida, similar a las puertas de olvido o entrada, que decidirá qué partes de la celda de memoria vamos a generar. Luego, colocamos los valores de la celda de memoria a través de un tanh para que la salida se limite a un rango entre -1 y 1 y lo multiplicamos por el valor de la puerta de salida que acabamos de calcular, de modo que solo generemos las partes que decidimos. Estos pasos se realizan como en las ecuaciones 5 y 6.

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \otimes \phi(c_t) \quad (6)$$

Donde: σ es la función sigmoidal, ϕ es la tangente hiperbólica, $\otimes$ representa por el producto con los valores de los gates y los pesos de la matriz denotada por W_{ij}.

IV. Enfoque CNN–LSTM con Atención

Siguiendo la arquitectura base vista en la Sección III incluimos un mecanismo de atención propuesta por [19]. La Figura 3 muestra el esquema general de la arquitectura. Para generar un mapa de atención:

- Comprimimos el cuboide x_t a una forma vectorial realizando un promedio por mapa de característica para alimentar un mlp_4.
- Tomamos el vector de contexto h_{t-1} para alimentar un mlp_2.
- El vector de ponderación se construye a partir de la salida del mlp_3, de esta manera, redefinimos el vector a un tamaño de $F \times F$ que representa la probabiliad sobre todos los píxeles de cada uno de los mapas de características. Cuanto mayor sea el valor del píxel en el mapa de atención, más importante será esa región de la imagen para la decisión en la etapa de clasificación.

La Tabla II muestra la configuración de los MLPs.

Cuadro II: Configuración de los MLPs

MLP	Capa	Parametro
$\mathbf{MLP_1}$	Dropout	0.5
	FC	#classes (neurons)
$\mathbf{MLP_2}$, $\mathbf{MLP_3}$ and $\mathbf{MLP_4}$	FC	128 (neurons)
	Dropout	0.5
$\mathbf{MLP_h}$ and $\mathbf{MLP_c}$	FC	256 (neurons)
	Dropout	0.5

Para la inicialización de h_0 y c_0 Xu et al. [30], comprime toda la información del video v logrando una convergencia mas rápida, esto se calcula como:

$$h_0 = mlp_h(\frac{1}{T}\sum_{t=1}^{T}(\frac{1}{F^2}\sum_{i=1}^{F^2}X_{t,i})) \tag{7}$$

$$c_0 = mlp_c(\frac{1}{T}\sum_{t=1}^{T}(\frac{1}{F^2}\sum_{i=1}^{F^2}X_{t,i})) \tag{8}$$

donde $T = 40$ cantidad de frames de los videos y $F = 7$ dimensión del feature map de la VGG16. Todos los fotogramas $x_i \in v$ a través de la VGG16, produciendo T cuboides. Para comprimir esta información, primero tomamos un promedio sobre el número de cuboidess y luego sobre todos los valores de píxeles en cada mapa de características. El vector resultante alimentará un mlp_h para obtener el estado inicial h_0 y un mlp_c para obtener la memoria inicial c_0. La tabla II muestra la configuración de los MLP para la inicialización.

V. Experimentos y Resultados

V-A. Metricas de evaluación

Para evaluar el rendimiento del sistema vamos a emplear las siguientes métricas de evaluación compatibles con clasificaciones multiclase:

Exactitud (Accuracy): es definida como las muestras clasificadas correctamente divididas por el número total de muestras. Esto es:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Precisión (Precision) es definida como las muestras que son clasificadas correctamente para la clase i sobre el total de muestras clasificadas como la clase i.

$$Precision_i = \frac{TP}{TP + FP} \tag{10}$$

Exhaustividad (Recall): es definida como la porción de muestras de clase i que son clasificados correctamente

$$Recall_i = \frac{TP}{TP + FN} \tag{11}$$

donde TP: Verdaderos Positivos, TN: Verdaderos Negativos, FP: Falsos Positivos y FN: Falsos Negativos.

V-B. Base de datos

- HMDB-51 Human Motion dataset [31] es un conjunto de datos de categorías de acción de videos recopilados de diferentes fuentes, incluidas películas, base de datos de archivo Prelinger, videos de YouTube y Google. Las acciones se agrupan en cinco tipos: acciones faciales generales, acciones faciales con manipulación de objetos, movimiento corporal general, movimiento corporal con interacción de objetos, movimientos corporales para interacción humana. El dataset proporciona información para realizar 3 divisiones, cada una de las cuales consta de 5100 videos, 3570 para entrenamiento y 1530 para test, es decir, una proporción de 70/30 por clase.
- UCF-101 dataset propuesto por [32] Estas 101 categorías se pueden clasificar en 5 tipos (interacción humano-objeto, solo movimiento corporal, interacción humano-humana, tocar instrumentos musicales y deportes). La duración total de estos videos es de más de 27 horas. Todos los videos se recopilan de YouTube y tienen una velocidad de 25 FPS con una resolución de 320×240. El dataset también proporciona información para realizar 3 divisiones, los videos de una clase se dividen en 25 grupos y cada división de test tiene 7 grupos los 18 grupos restantes se utilizan para el entrenamiento.

V-C. Resultados

Nuestro sistema fue implementado en Python usando la librería Tesnsorflow [33] sobre una computadora Intel CORE i7-6700HQ con 16GB de memoria DDR3 y sistema operativo Ubuntu 16,04. Los experimentos se llevaron a cabo sobre una GPU NVIDIA Titan Xp montada en un servidor con características similares.

Los parámetros de la red se optimizan minimizando la función de pérdida de entropía cruzada utilizando el descenso de gradiente estocástico con la regla de actualización RMSProp [34].

El Cuadro III muestra los resultados obtenidos por nuestro sistema aplicando un k-fold cross validation con $k = 4$, usando 3 folds para entrenamiento y 1 fold para test. La

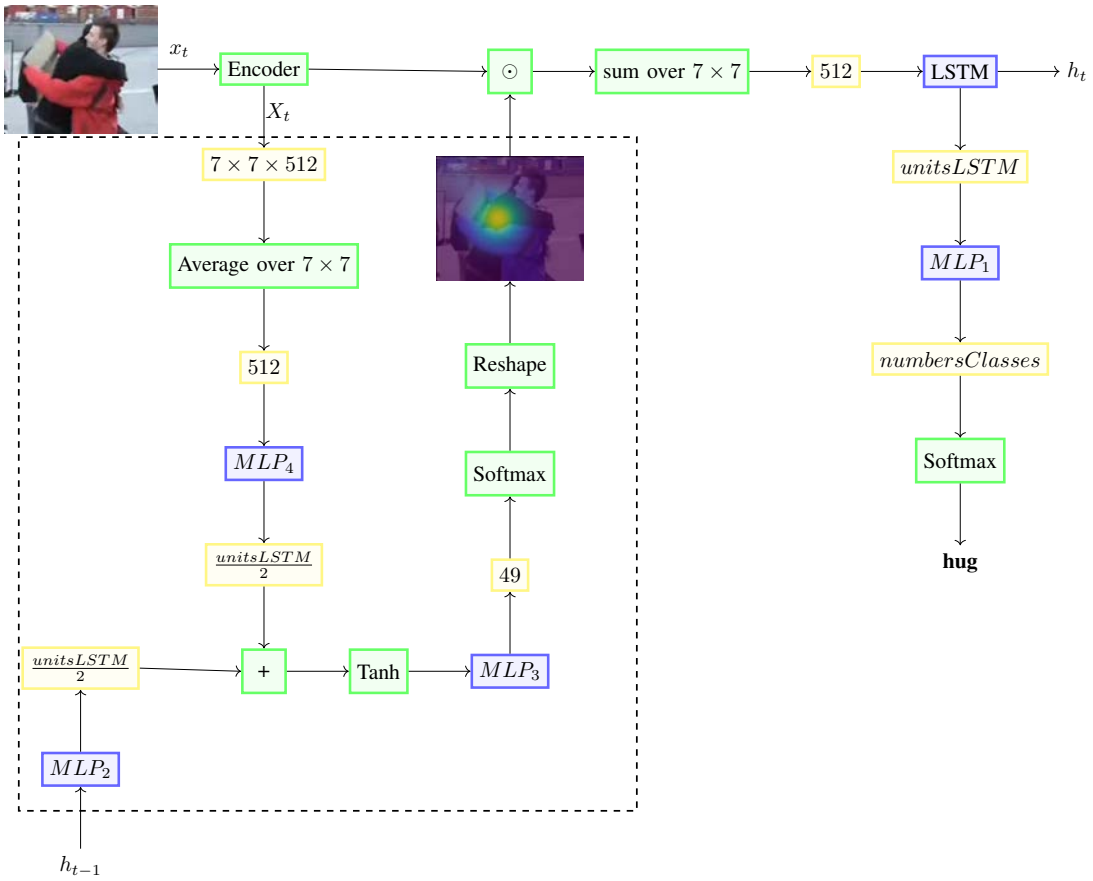

Figura 3: Arquitectura con mecanismo de atención.

precision promedio ($\overline{Precision}$) y recall promedio $\overline{Recall}$ final del algoritmo es el promedio de las k iteraciones.

Cuadro III: Resultados para las métricas $\overline{Precision}$ (precisión promedio) y $\overline{Recall}$ (*recall* promedio).

		$\overline{\text{Precision}}$	$\overline{\text{Recall}}$
HMDB-51	Enfoque base	39,56 %	41,67 %
	Enfoque con atención	47,14 %	48,21 %
UCF-101	Enfoque base	71,94 %	72,02 %
	Enfoque con atención	87,33 %	89,97 %

En el Cuadro III puede apreciarse un aumento en $\overline{Precision}$ y en $\overline{Recall}$ al aplicar atención, para ambas bases de datos.

Durante la experimentación, se observó también que la discriminación se tornó difícil para algunos pares de clases consistentes en acciones similares o en acciones que presentaban fondos similares. Por ejemplo: en la base HMDB-51, las clases *drink* y *eat* tendían a confundirse más. Lo mismo ocurría con las clases *smile* y *smoke*, también de la base HMDB-1. En la base UCF-101 las clases *brushing teeth* y *apply lipstick* tendieron a confundirse. Lo mismo sucedió con *field hockey penalty* y *golf swing*.

El Cuadro IV resume la Exactitud (ACC) obtenidos por nuestro sistema, aplicando el protocolo de evaluación original para cada una de las bases de datos HMDB-51 y UCF-101 respectivamente, junto a los resultados obtenidos con otros enfoques citados en la bibliografía. La columna *Tipo* describe el enfoque general siguiendo la clasificación descripta en la sección II (donde EC: Enfoque Clásico, AP: Aprendizaje Profundo y DF: Redes Doble Flujo).

Cuadro IV: Resultados obtenidos empleando la configuración de evaluación de las bases de datos y reporte de otros enfoques propuestos en la bibliografía.

Artículo reportado	**Tipo**	**ACC (%)**	
		HMDB-51	**UCF-101**
Kuehne et al. [31]	EC	23,0 %	-
Jiang et al. [35]	EC	40,7 %	-
Gaidon et al. [36]	EC	41,3 %	-
Sharma et al. [19]	AP	41,3 %	-
Wang et al. [13]	EC	46,6 %	-
Ye et al. [17]	AP	55,2 %	85,4 %
Zhang et al. [18]	AP	-	86,4 %
Meng et al. [37]	AP	53,1 %	87,1 %
Li et al. [38]	AP	54,3 %	86,7 %
Simoyan et al. [21]	DF	59,4 %	88,0 %
Feichtenhofer et al. [22]	DF	58,5 %	91,4 %
Nuestro enfoque base		40,7 %	75,8 %
Nuestro enfoque con atención		51,2 %	87,2 %

Nuestra propuesta CNN–LSTM base muestra un rendimiento similar a los propuestos por [31], [35], [36]. A diferencia de los enfoques clásicos donde la extracción de características se tiene que diseñar con cuidado, empleamos las características pre-definidas de una CNN (VGG16) donde los hiperparámetros son ajustados para una tarea de clasificación de imágenes (1000 clases en total). La implementación del mecanismo de atención adaptado a la arquitectura base muestra un mejor rendimiento en ambas bases de datos. Dentro de los métodos de

aprendizaje profundo (AP), nuestra propuesta obtiene un mejor rendimiento con respecto al enfoque con atención propuesto por Sharma [19], donde por ejemplo usan una extractor de características GoogleNet. Como así también los enfoques propuestos por [13], [18], [37]. Finalmente, podemos ver que nuestro resultado obtenido es competitivo con enfoques doble flujo (DF) donde intervienen al menos dos arquitecturas profundas, en este caso implica mayor costo computacional para el procesamiento como asi tambien tiempo de entrenamiento.

La Figura 4 muestra ejemplos de la salida de nuestro sistema para los conjuntos de datos HMDB-51 (arriba) y UCF-101 (abajo) respectivamente. Cada ejemplo viene acompañado de la siguiente información:

- Etiqueta (Label): acción real etiquetada para el video.
- Predicción (Prediction): respuesta de nuestro sistema correspondiente a la clase con puntuación más alta, es decir la clase más probable.
- Superposición del mapa de atención. La región en amarillo son hacia donde mira el sistema y el brillo indica la ponderación.

VI. Conclusiones y Trabajo Futuro

En este trabajo implementamos un sistema de reconocimiento de acciones de video, utilizando una red neuronal CNN–LSTM. Primero, un VGG 16 extrae las características del video. Luego, una red neuronal LSTM clasifica la escena en la clase a la que pertenece. Incluimos un mecanismo de atención adaptada para la arquitectura base. La arquitectura se implementó en Python usando la librería Tensorflow, se entrenó y se probó usando las bases de datos HMDB-51 [31] y UCF-101 [32] se realizó en una GPU NVIDIA Titan Xp.

Evaluamos el rendimiento de la arquitectura siguiendo las métricas de evaluación estandár para las bases de datos empleadas. obtenemos $40{,}7\,\%$ (base) y $51{,}2\,\%$ (con atención) para HMDB-51, $75{,}8\,\%$ (base) y $87{,}2\,\%$ (con atención) para UCF-101. Queremos destacar la mejora del resultado final de la arquitectura base con respecto a la utilización del mecanismo de atención, resultados competitivos con los de la literatura teniendo en cuenta la simplicidad de la arquitectura. El aporte que se muestra en este artículo consiste en mostrar una solución que utiliza pocos recursos y obtiene buenos resultados comparables con otras propuestas que consumen más recursos.

Como trabajo futuro:

- Vamos profundizar sobre las métricas de evaluación para complementar la evaluación de rendimiento de nuestra propuesta.
- Se consideraran el uso de otras bases de datos, como Hollywood2 [39] y UCF-50 [40] para hacer que el sistema sea más robusto y profundizar sobre técnicas para evitar el sobreajuste.
- Proponer el uso de otras redes neuronales convolucionales para la extracción de características, por ejemplo ResNet [23]. Profundizar sobre los mecanismo de atención [38], [41].
- Otra línea de investigación es aplicar nuevos enfoques Transformer [42] para el problema en cuestión.

Agradecimientos

Los autores agradecen a NVIDIA por la donación de una GPU TITAN Xp para el Departamento de Informática - Facultad de Ciencias Exactas - Universidad Nacional de Salta, Argentina.

Referencias

[1] I. Jegham, A. B. Khalifa, I. Alouani, and M. A. Mahjoub, "Vision-based human action recognition: An overview and real world challenges," *Forensic Science International: Digital Investigation*, vol. 32, p. 200901, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S174228761930283X

[2] M. A. Khan, K. Javed, S. A. Khan, T. Saba, U. Habib, J. A. Khan, and A. A. Abbasi, "Human action recognition using fusion of multiview and deep features: an application to video surveillance," *Multimedia tools and applications*, pp. 1–27, 2020.

[3] J. Bao, M. Ye, and Y. Dou, "Mobile phone-based internet of things human action recognition for e-health," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*. IEEE, 2016, pp. 957–962.

[4] N. Jaouedi, N. Boujnah, O. Htiwich, and M. S. Bouhlel, "Human action recognition to human behavior analysis," in *2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*. IEEE, 2016, pp. 263–266.

[5] V. Bloom, D. Makris, and V. Argyriou, "G3d: A gaming action dataset and real time action recognition evaluation framework," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2012, pp. 7–12.

[6] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[7] C. G. Harris, M. Stephens *et al.*, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[8] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *CVPR 2011*, June 2011, pp. 3169–3176.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[11] J. Perš, V. Sulić, M. Kristan, M. Perše, K. Polanec, and S. Kovačič, "Histograms of optical flow for efficient representation of body motion," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1369–1376, 2010.

[12] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC 2009 - British Machine Vision Conference*, A. Cavallaro, S. Prince, and D. Alexander, Eds. London, United Kingdom: BMVA Press, Sep. 2009, pp. 124.1–124.11. [Online]. Available: https://hal.inria.fr/inria-00439769

[13] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013.

[14] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[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, 2013.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[17] Y. Ye and Y. Tian, "Embedding sequential information into spatiotemporal features for action recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 1110–1118.

[18] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with deeply transferred motion vector cnns," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2326–2339, 2018.

[19] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *CoRR*, vol. abs/1511.04119, 2015. [Online]. Available: http://arxiv.org/abs/1511.04119

Figura 4: Ejemplos de salida de nuestra arquitectura. HMDB-51 (superior) y UCF-101 (inferior), junto con la clase de mayor puntuación (score).

[20] M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends in neurosciences*, vol. 15, no. 1, pp. 20–25, 1992.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[22] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[24] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal residual networks for video action recognition," *CoRR*, vol. abs/1611.02155, 2016. [Online]. Available: http://arxiv.org/abs/1611.02155

[25] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, and B. Li, "Hierarchical attention network for action recognition in videos," *arXiv preprint arXiv:1607.06416*, 2016.

[26] X. Wang, A. Farhadi, and A. Gupta, "Actions˜ transformations," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 2658–2667.

[27] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[28] C. I. Orozco, M. E. Buemi, and J. J. Berlles, "Cnn-lstm architecture for action recognition in videos," in *I Simposio Argentino de Imágenes y Visión (SAIV 2019)-JAIIO 48 (Salta)*, 2019.

[29] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[30] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[31] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[32] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012. [Online]. Available: http://arxiv.org/abs/1212.0402

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[34] Y. Dauphin, H. de Vries, and Y. Bengio, "Rmsprop and equilibrated adaptive learning rates for non-convex optimization," in *NIPS*, 2015.

[35] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo, "Trajectory-based modeling of human actions with motion reference points," in *European Conference on Computer Vision*. Springer, 2012, pp. 425–438.

[36] A. Gaidon, Z. Harchaoui, and C. Schmid, "Activity representation with motion hierarchies," *International journal of computer vision*, vol. 107, no. 3, pp. 219–238, 2014.

[37] L. Meng, B. Zhao, B. Chang, G. Huang, F. Tung, and L. Sigal, "Where and when to look? spatio-temporal attention for action recognition in videos," *CoRR*, vol. abs/1810.04511, 2018. [Online]. Available: http://arxiv.org/abs/1810.04511

[38] X. Li, M. Xie, Y. Zhang, G. Ding, and W. Tong, "Dual attention convolutional network for action recognition," *IET Image Processing*, vol. 14, no. 6, pp. 1059–1065, 2020.

[39] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 2929–2936.

[40] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Mach. Vision Appl.*, vol. 24, no. 5, pp. 971–981, Jul. 2013. [Online]. Available: http://dx.doi.org/10.1007/s00138-012-0450-4

[41] H. Yang, C. Yuan, L. Zhang, Y. Sun, W. Hu, and S. J. Maybank, "Sta-cnn: convolutional spatial-temporal attention learning for action recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 5783–5793, 2020.

[42] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 244–253.