



Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Reconocimiento de acciones humanas en videos utilizando redes neuronales profundas

Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área de Ciencias de la Computación

Lic. Carlos Ismael Orozco

Director de tesis:	Dr. Julio César A. Jacobo Berlles
Directora asistente:	Dra. María Elena Buemi
Consejera de estudios:	Dra. Marta E. Mejail
Lugar de Trabajo:	Departamento de Computación - FCEyN - UBA Grupo de Procesamiento de Imágenes y Visión por Computadora
Fecha de defensa:	24/09/2025

Buenos Aires, 2025

Reconocimiento de Acciones Humanas en Videos Utilizando Redes Neuronales Profundas

Resumen. El Reconocimiento de Acciones Humanas en videos (HAR) consiste en identificar y clasificar diferentes acciones que involucran una o más personas. Este campo continúa en desarrollo ya que los resultados aún no alcanzan niveles satisfactorios para muchas de sus aplicaciones tales como vigilancia, interacción hombre-máquina, asistencia sanitaria, indexación de videos, entre otras. La importancia de HAR radica en su aplicabilidad a diversos entornos donde la interpretación del comportamiento humano a partir de videos es clave.

El objetivo de esta tesis fue desarrollar enfoques de aprendizaje profundo aplicados a HAR. La propuesta integró el uso de Redes Convolucionales (CNN) para extraer características espaciales y de Redes Recurrentes (LSTM) para modelar la dinámica temporal de las acciones.

Se incorporó un mecanismo de atención que enfocó las regiones más relevantes de cada fotograma, mejorando el rendimiento al destacar características esenciales y reducir la influencia de los fotogramas poco representativos. La combinación de CNN, LSTM y atención ofreció una solución robusta para los desafíos de variabilidad en contextos y entornos.

Además, se implementó un módulo de selección de fotogramas que optimizó el proceso de reconocimiento al identificar y utilizar los fotogramas más representativos. Este no solo mejoró la precisión del modelo, sino que también redujo los tiempos de entrenamiento al procesar menos fotogramas sin comprometer la calidad del reconocimiento.

Para evaluar el rendimiento de las propuestas, se utilizaron bases de datos públicas especializadas en este tipo de problemas. Los resultados obtenidos fueron competitivos en comparación con modelos más complejos reportados en la literatura.

Palabras clave: Reconocimiento de Acciones Humanas, Redes Neuronales Convolucionales, Redes LSTM, Mecanismos de Atención.

Human Action Recognition in Videos Using Deep Neural Networks

Abstract. Human Action Recognition (HAR) in videos consists of identifying and classifying different actions involving one or more individuals. This field remains under active development, as current results still fall short of satisfactory performance levels for many real-world applications, such as surveillance, human-computer interaction, healthcare assistance, video indexing, among others. The importance of HAR lies in its applicability to diverse settings where understanding human behavior from video data is essential.

The objective of this thesis was to develop deep learning approaches applied to HAR. The proposed method integrated Convolutional Neural Networks (CNN) for extracting spatial features and Recurrent Neural Networks (LSTM) for modeling the temporal dynamics of actions.

An attention mechanism was incorporated to focus on the most relevant regions of each video frame, improving performance by highlighting essential features and reducing the influence of less representative frames. The combination of CNN, LSTM, and attention provided a robust solution to the challenges posed by variability in contexts and environments.

Additionally, a frame selection module was implemented to optimize the recognition process by identifying and utilizing the most representative frames. This not only enhanced the model's accuracy but also reduced training time by processing fewer frames without compromising recognition quality.

To evaluate the performance of the proposed methods, publicly available datasets specialized in this domain were used. The results obtained were competitive when compared to more complex models reported in the literature.

Keywords: Human action recognition, CNN, LSTM, Attention mechanisms.

Dedicatoria

A mi padre Juan, a quien extraño cada día de mi vida y cuya memoria me
guía y acompaña siempre.

A mi hija Julieta, para que crea en sus sueños y trabaje con dedicación para
lograrlos. ¡Te amo!

Agradecimientos

Con la escritura de esta tesis concluye una etapa significativa en mi vida. Ha sido un recorrido extenso, marcado por múltiples desafíos, momentos de incertidumbre, y también por grandes aprendizajes y oportunidades de crecimiento, tanto en lo académico como en lo personal. Este trabajo representa no solo el cierre de un proceso formativo, sino también la consolidación de años de esfuerzo, dedicación y compromiso con la investigación y el desarrollo profesional. En este momento, deseo tomarme un tiempo para expresar mi agradecimiento a quienes me acompañaron, alentaron y sostuvieron en distintas etapas del camino.

A mi familia, por su presencia constante y por el respaldo incondicional en cada uno de mis proyectos. Su apoyo silencioso pero firme fue un pilar fundamental para avanzar en los momentos más exigentes. A mis padres, Nélide y Juan, por transmitirme, con su ejemplo, el valor del esfuerzo, la honestidad y la perseverancia. Sus enseñanzas han sido una guía permanente en mi formación y en la manera de afrontar los desafíos. A mi hija Julietita, que es, sin dudas, la persona más importante de mi vida. Su sonrisa, su energía y su forma de ver el mundo me inspiran todos los días a ser mejor. A Eli, Sole y Emi, por estar siempre, por su cariño constante y por acompañarme con palabras de aliento y gestos sinceros.

A Cristian, por confiar en mí incluso cuando yo mismo dudaba, por estar presente en momentos clave y por acompañarme de forma constante

a lo largo de estos años. Su presencia y orientación marcaron una diferencia en muchas etapas de este recorrido, y le estoy profundamente agradecido.

A mis directores, María Elena y Julio, por su compromiso, paciencia y orientación permanente. Les agradezco sinceramente su generosidad, su tiempo y su confianza. Han sabido acompañarme no solo desde lo académico, sino también desde lo humano. Esta tesis no habría sido posible sin su guía continua y su disposición para abrir caminos, ofrecer ideas y ayudarme a superar los obstáculos.

Al grupo de Procesamiento de Imágenes y Visión por Computadora: Marta, Daniel, Francisco (Patxi), Sebastián, Norberto, Mailén, Saúl, Florencia y Pablo. Gracias por recibirme con los brazos abiertos y por hacerme sentir parte del equipo desde el primer día. El clima de colaboración, el intercambio de ideas, las discusiones técnicas y los momentos de distensión compartidos fueron una parte muy valiosa de esta experiencia. Me llevo el afecto, la calidez y la camaradería de un grupo humano que supo equilibrar la exigencia con la cercanía. Me hicieron sentir como en casa y los considero amigos de la vida.

A todas estas personas, y a quienes quizás no mencioné pero también dejaron su huella en este proceso, ¡gracias! Esta tesis también es el resultado del acompañamiento, la dedicación y la generosidad de cada uno. Lleva en sus páginas una parte de todo lo vivido, aprendido y compartido a lo largo de estos años.

Índice general

1	Introducción	13
1.1	Descripción del Problema	13
1.2	Aportes	15
1.3	Publicaciones	16
1.4	Organización de la Tesis	18
2	Marco de Trabajo	21
2.1	Estado del Arte	21
2.1.1	Enfoques Clásicos	22
2.1.2	Enfoques Basados en Redes Profundas	26
2.1.3	Enfoques Basados en Mecanismos de Atención	31
2.2	Métricas de evaluación	35
2.2.1	Exactitud (Accuracy)	35
2.2.2	Matriz de Confusión	37
2.2.3	Validación Cruzada	38
2.3	Bases de datos	39
2.3.1	KTH	39
2.3.2	UCF-11	40
2.3.3	HMDB-51	41
2.3.4	UCF-101	43
3	Redes Neuronales CNN-LSTM	47
3.1	Camino Hacia el Modelo CNN-LSTM	47
3.1.1	Redes Neuronales Tradicionales	47

ÍNDICE GENERAL

3.1.2	Redes Neuronales Convolucionales	51
3.1.3	Redes Neuronales Recurrentes	59
3.2	Enfoque CNN-LSTM	67
3.3	Experimentos y Resultados	70
3.3.1	Entornos Controlados: KTH y UCF-11	70
3.3.2	Entornos No Controlados: HMDB-51 y UCF-101	73
3.4	Resumen	78
4	Mecanismo de Atención Sobre Redes LSTM	80
4.1	Enfoque CNN-LSTM + Att	80
4.2	Enfoque CNN-LSTM + Att + FS	82
4.3	Experimentos y Resultados	86
4.4	Resumen	91
5	Conclusiones y Trabajo a Futuro	93
5.1	Conclusiones	93
5.2	Trabajo a Futuro	95
	Bibliografía	98

Índice de figuras

2.1	Estructura general de un sistema HAR	21
2.2	Clasificación de métodos HAR	22
2.3	Convolución 2D vs Convolución 3D	28
2.4	Datos de referencia vs Predicción	36
2.5	Ejemplo de K -fold cross validation con $K = 5$	38
3.1	Ejemplo de un MLP.	48
3.2	Funciones de activación	49
3.3	Arquitectura LeNet-5	52
3.4	Arquitectura AlexNet	52
3.5	Arquitectura VGG16	53
3.6	Arquitectura ResNet	54
3.7	Capa de convolución: Conectividad local	56
3.8	Capa de agrupación: Tipos de agrupaciones	58
3.9	Ejemplo de una RNN.	59
3.10	Unidad oculta RNN desplegada a profundidad 3	61
3.11	Unidad LSTM.	62
3.12	Bloque LSTM.	63
3.13	Bloque LSTM: Puerta de olvido	64
3.14	Bloque LSTM: Puerta de entrada	65
3.15	Bloque LSTM: Actualización de memoria	66
3.16	Bloque LSTM: Salida	67
3.17	Arquitectura base	68
3.18	Curva ROC: Variación entre exactitud y nro de unidades LSTM	71

ÍNDICE DE FIGURAS

3.19 Ejemplos de salida de la arquitectura para KTH	72
3.20 Ejemplos de salida de la arquitectura para UCF-11	73
3.21 Ejemplos de salida del modelo para la base HMDB-51	76
3.22 Ejemplos de salida del modelo para la base UCF-101	77
4.1 Arquitectura con mecanismo de atención.	81
4.2 Ejemplos de medida de disimilitud	85
4.3 Ejemplos de salida de la arquitectura para HMDB-51	89
4.4 Ejemplos de salida de la arquitectura para UCF-101	90

Introducción

1.1 Descripción del Problema

Reconocer acciones humanas a partir de información visual constituye una tarea natural para las personas. Sin embargo, su automatización mediante sistemas computacionales representa un desafío considerable debido a diversos factores que inciden en la calidad y consistencia de los datos visuales disponibles. Entre estos factores se encuentran la variabilidad en la apariencia de los sujetos (por ejemplo, diferencias en constitución física, vestimenta o estatura), así como las condiciones del entorno, que pueden incluir escenas concurridas, oclusiones parciales, presencia de sombras y variaciones en la iluminación. Asimismo, parámetros técnicos como el ángulo de visión y la distancia del sujeto con respecto a la cámara afectan significativamente la percepción visual de la acción [2].

Adicionalmente, las acciones humanas exhiben una elevada variabilidad tanto espacial como temporal, lo cual implica que una misma acción puede manifestarse de múltiples formas en distintas ejecuciones. Esta variabilidad está determinada por diferencias individuales en el estilo de movimiento, la velocidad de ejecución y las transiciones posturales. En consecuencia, modelar computacionalmente este tipo de comportamiento requiere estrategias robustas frente a dichas fluctuaciones, al tiempo que conserven la capacidad de generalización en diferentes contextos y sujetos [52].

1.1 Descripción del Problema

Matemáticamente, dada una secuencia de observaciones X y una etiqueta particular Y , los sistemas de reconocimiento de acciones humanas (HAR) tienden a modelar la probabilidad a posteriori $P(Y|X)$ para aprender una función que asocie las observaciones X con sus etiquetas de clase Y . En este contexto, X representa una secuencia temporal de fotogramas que capturan la evolución de la acción en el video, e Y corresponde a la clase de acción asociada a dicha secuencia, es decir, dada una lista de posibles acciones y un video en el que se observa a un actor en movimiento, el objetivo del problema es determinar cuál de las acciones está siendo ejecutada.

La importancia de los sistemas HAR para videos en el área de reconocimiento de patrones y visión por computadora radica en que constituyen herramientas valiosas para múltiples aplicaciones como:

- El análisis de videos de vigilancia [63]. 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, que observadores humanos detecten eventos importantes en los videos.
- La comprensión automática de videos resulta especialmente útil en el cuidado de personas mayores y niños, así como en el desarrollo de herramientas para la rehabilitación de pacientes con dificultades motrices, tanto en viviendas como en centros de salud inteligentes. Estas tecnologías permiten el monitoreo y reconocimiento automático de actividades diarias, facilitando la asistencia a los residentes, la guía de pacientes en rehabilitación y la obtención de datos valiosos sobre sus capacidades funcionales e indicadores de salud [11].
- El resumen automático de videos se enfoca en generar versiones cortas a partir de las escenas más relevantes del video original, facilitando así el acceso rápido a la información esencial para la toma de decisiones o análisis posteriores. Esta técnica es fundamental para la búsqueda basada en contenido dentro de grandes bases de datos de videos. Al incorporar descripciones textuales automáticas de un

1.2 Aportes

video, se elimina la necesidad de anotaciones manuales, lo que resulta importante para complementar las bases de datos.

- La interacción humano-computadora [60] 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 dispositivos y con otras personas.
- Otra aplicación es el desarrollo de videojuegos [19] que permiten que el usuario interactúe con la consola o la computadora sin la necesidad de usar un dispositivo físico.
- En el campo de la biometría basada en comportamiento, a diferencia de las herramientas biométricas clásicas (como las huellas digitales), estas técnicas permiten obtener datos para la identificación de las personas sin interferir con su actividad. Un ejemplo típico es la identificación a partir del modo de andar de las personas.

1.2 Aportes

Las principales contribuciones de esta tesis son:

- En el capítulo 3, se propone un enfoque basado en CNN-LSTM para el reconocimiento de acciones en videos. La arquitectura utiliza una CNN para extraer características espaciales relevantes de los fotogramas, seguida de una red LSTM para capturar la dinámica temporal de las secuencias. Esta configuración inicial establece una base sólida para futuras mejoras desarrolladas en los capítulos siguientes.
- En el capítulo 4 se integra un mecanismo de atención en la arquitectura base CNN-LSTM, que permite al modelo enfocarse en las regiones más relevantes de cada fotograma. Este enfoque mejora la capacidad del modelo para capturar detalles críticos en la secuencia de video,

1.3 Publicaciones

lo que ha resultado en un rendimiento significativamente superior en comparación con la arquitectura sin atención.

- Asimismo, se incorpora un módulo de selección de fotogramas a la arquitectura CNN-LSTM con atención, que permite al modelo identificar y priorizar los fotogramas más representativos de la secuencia de video. Este módulo contribuye a reducir el ruido y a enfocar el análisis en los momentos clave, lo que mejora aún más el rendimiento del modelo en tareas de reconocimiento de acciones.

1.3 Publicaciones

El desarrollo de esta tesis ha dado lugar a varios trabajos. Algunos de ellos ya han sido publicados, mientras que otros han sido enviados o lo serán en un futuro cercano.

Trabajos publicados en revistas nacionales con arbitraje:

- Carlos Ismael Orozco. María Elena Buemi. Julio Jacobo Berlles. CNN-LSTM with soft attention mechanism for human action recognition in videos. *Elektron: Ciencia y tecnología en la electrónica de hoy*. Vol. 5, No. 1, pp. 37-44. ISSN 2525-0159. DOI: <https://doi.org/10.37537/rev.elektron.5.1.130.2021>. Año 2021.
- Carlos Ismael Orozco. Eduardo Xamena. María Elena Buemi. Julio Jacobo Berlles. Human action recognition in videos using a robust CNN LSTM approach. *Ciencia y Tecnología*, 21-34. ISSN 2344-9217. DOI: <https://doi.org/10.18682/cyt.vi0.3288>. Año 2020.

Trabajos publicados en conferencias y congresos nacionales e internacionales con arbitraje:

- Carlos Ismael Orozco. María Elena Buemi. Julio Jacobo Berlles. BiLSTM with CNN Features For HAR in Videos. 2do Simposio Argentino de Imágenes y Video. 50 JAIIO. Proceedings en prensa. Año 2021.

1.3 Publicaciones

- Carlos Ismael Orozco. María Elena Buemi. Julio Jacobo Berlles. Spatial attention adapted to a LSTM architecture with frame selection for HAR. 38th International Conference on Machine Learning (ICML). Proceedings en prensa. Año 2021.
- Carlos Ismael Orozco. María Elena Buemi. Julio Jacobo Berlles. Towards an Attention Mechanism LSTM Framework for Human Action Recognition in Videos. 2020 IEEE Congreso Bienal de Argentina (ARGENCON). pp. 1-6, doi: 10.1109/ARGENCON49523.2020.9505440. Año 2020.
- Carlos Ismael Orozco. María Elena Buemi. Julio Jacobo Berlles. Action recognition in videos: LSTM network parameter setting. 3er Congreso Internacional de Ciencias de la Computación y Sistemas de Información CICC SI 2019. Mendoza. Argentina. ISBN 978-987-45683-7-3. Año 2019.
- Carlos Ismael Orozco, María Elena Buemi and Julio Jacobo Berlles. CNN-LSTM architecture for action recognition in videos. 1er Simposio Argentino de Imágenes y Video. 48 JAIIO. ISSN: 2683-8990. Salta. Argentina. Año 2019.
- Carlos Ismael Orozco, María Elena Buemi. Julio Jacobo Berlles. Video to text study using an encoder decoder networks approach. 37th International Conference of the Chilean Computer Science Society (SCCC), Santiago, Chile, 2018, pp. 1-5. doi: 10.1109/SCCC.2018.8705254. (*Best Paper*).

Trabajos relacionados publicados en conferencias internacionales con arbitraje:

- Carlos Ismael Orozco, Florencia Iglesias, María Elena Buemi and Julio Jacobo Berlles. Real-time gender recognition from face images using deep convolutional neural network. 7th Latin American Conference on Networked and Electronic Media (LACNEM 2017), Valparaíso, Chile, 2017, pp. 7-11. doi: 10.1049/ic.2017.0027.

1.4 Organización de la Tesis

- Carlos Ismael Orozco, María Elena Buemi and Julio Jacobo Berlles, New deep convolutional neural network architecture for pedestrian detection. 8th International Conference of Pattern Recognition Systems (ICPRS 2017), Madrid, España, 2017, pp. 1-6. doi: 10.1049/cp.2017.0156.
- Carlos Ismael Orozco, María Elena Buemi and Julio Jacobo Berlles. A study on pedestrian detection using a deep convolutional neural network. 7th International Conference on Pattern Recognition Systems (ICPRS 2016), Talca, Chile, 2016, pp. 1-4. doi: 10.1049/ic.2016.0039.

Otras participaciones relevantes:

- Carlos Ismael Orozco. Ponencia: Aprendizaje profundo para la clasificación de videos. 1er Congreso Peruano-Argentino de Ingeniería en Sistemas e Informática (COPAI SI 2020).

En evaluación:

- Carlos Ismael Orozco, María Elena Buemi and Julio Jacobo Berlles. Generate natural language descriptions of video using long short-term memory approach. Journal IEEE Latin America Transaction 2019.

1.4 Organización de la Tesis

Esta tesis se estructura en varios capítulos fundamentales que exploran y profundizan en el reconocimiento de acciones humanas en videos mediante el uso de arquitecturas de aprendizaje profundo. A continuación, se detalla la organización del resto de esta tesis:

- El capítulo 2 establece el contexto teórico y práctico del reconocimiento de acciones en videos. Se examina el estado del arte en este campo, abordando desde los enfoques clásicos hasta los más recientes basados en redes neuronales profundas. Además, se discuten las métricas estándar utilizadas para evaluar el rendimiento de los modelos, como la exactitud, la matriz de confusión y la validación cruzada.

1.4 Organización de la Tesis

También se realiza un análisis detallado de diversas bases de datos ampliamente utilizadas en la investigación de reconocimiento de acciones, incluyendo KTH, UCF-11, HMDB-51 y UCF-101.

- El capítulo 3 se centra en el estudio de las redes neuronales y su aplicación al reconocimiento de acciones en videos. Se proporciona un marco teórico completo sobre redes neuronales convolucionales (CNN) y redes de memoria a largo plazo (LSTM). Se detalla cómo estas arquitecturas son utilizadas para extraer características espaciales y modelar la dinámica temporal de las acciones humanas en secuencias de video. Además, se presenta la propuesta de un enfoque integrado CNN-LSTM, diseñado específicamente para mejorar la precisión en la clasificación de actividades complejas y variadas.
- El capítulo 4 introduce y explora en profundidad los mecanismos de atención en el contexto del reconocimiento de acciones en videos. Se analiza cómo estos mecanismos permiten centrarse en las partes clave de la secuencia de video, mejorando significativamente la capacidad del modelo para discernir acciones en entornos complejos y dinámicos. Al enfocar la atención en los aspectos más relevantes, el sistema no solo se vuelve más preciso, sino también más robusto ante variaciones y ruidos en los datos.

Adicionalmente, se implementa un módulo de selección de fotogramas, el cual desempeña un papel crucial en la optimización del proceso en cuestión. Este módulo identifica y utiliza los fotogramas más representativos dentro de cada secuencia, lo que no solo mejora la exactitud, sino que también contribuye a una mayor eficiencia en términos de procesamiento y tiempos de entrenamiento.

La combinación de los mecanismos de atención con este módulo de selección de fotogramas en la arquitectura CNN-LSTM presentada en esta tesis refuerza tanto la robustez como la eficacia del modelo final. Este enfoque integrado demuestra cómo la atención selectiva y la reducción inteligente de la cantidad de datos procesados pueden trabajar en conjunto para ofrecer un rendimiento superior en tareas

1.4 Organización de la Tesis

de reconocimiento de acciones, haciendo que el modelo sea más adaptable y eficiente en escenarios sin restricciones.

- El capítulo 5 presenta las conclusiones alcanzadas a lo largo de la investigación. Se discuten los resultados obtenidos en la implementación de la arquitectura propuesta y se reflexiona sobre los desafíos superados y las limitaciones encontradas. Además, se proponen diversas líneas de investigación futuras para continuar avanzando en este campo, incluyendo la exploración de nuevos conjuntos de datos, la mejora de técnicas de modelado temporal y la aplicación de enfoques más avanzados de aprendizaje profundo.

Marco de Trabajo

2.1 Estado del Arte

En esta sección revisamos enfoques del estado del arte para el reconocimiento de acciones en video, incluyendo métodos clásicos basados en descriptores estadísticos y de apariencia, modelos profundos contruidos con redes neuronales de múltiples capas, y propuestas más recientes que incorporan mecanismos de atención para mejorar la interpretación de las secuencias espacio-temporales.

La Figura 2.1 muestra un esquema general de cualquier sistema de Reconocimiento de Acciones Humanas (HAR) típico formada por una etapa de representación del video y una etapa de clasificación.

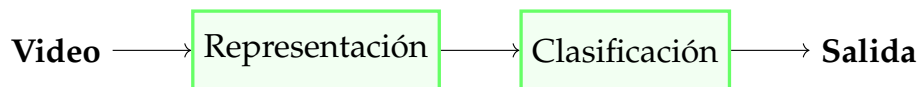


Figura 2.1: Estructura general de un sistema HAR

Para obtener una representación del video, un mecanismo de extracción de características calcula información numérica a partir de los fotogramas del video. Es deseable que la representación de la información posea las siguientes características:

1. Ser discriminativa: las representaciones de acciones de la misma clase llevan información similar, mientras que las representaciones de acciones de diferentes clases llevan información distintiva.

2.1 Estado del Arte

2. Ser eficiente: la representación de la acción debe ser computacionalmente factible, permitiendo su extracción en un tiempo razonable sin comprometer la calidad de la información.
3. Poseer baja dimensionalidad: esto implica un bajo costo de clasificación y ahorro de características.

Los sistemas HAR pueden ser clasificados en dos grandes grupos: enfoques clásicos y enfoques de aprendizaje profundo. A diferencia del método de representación clásico, en el que las características (features) se diseñan manualmente, el método de representación de aprendizaje profundo aprende las características automáticamente a partir del video, dicho aprendizaje forma parte del entrenamiento del sistema. La Figura 2.2 muestra los sistemas propuestos en la bibliografía clasificados por enfoque.

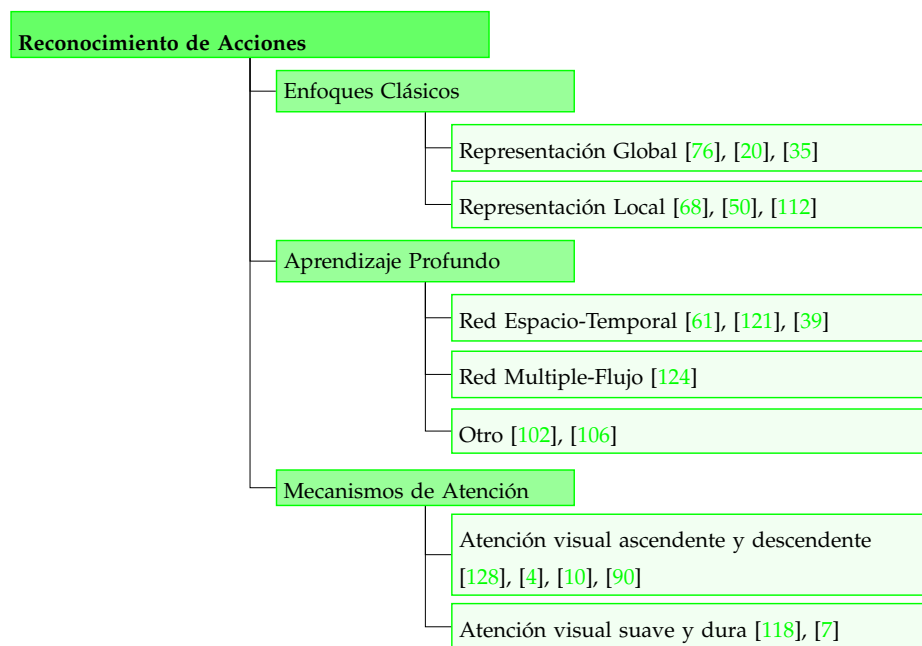


Figura 2.2: Clasificación de los métodos utilizados en la bibliografía para abordar el problema del reconocimiento de acciones humanas.

2.1.1 Enfoques Clásicos

Durante al menos tres décadas, el reconocimiento de acciones humanas se abordó principalmente a través del diseño y la extracción de características

2.1 Estado del Arte

artesanales (hand-crafted features). Estos enfoques consisten en extraer descriptores, tanto para la apariencia como para la información de movimiento de los fotogramas de video.

Representaciones Globales

Las representaciones globales, también llamadas holísticas, buscan describir una acción considerando el video como un todo, extrayendo características generales sin enfocarse en regiones específicas. Ejemplos clásicos incluyen el uso de silueta [76] y plantillas como MEI/MHI (Motion Energy Image/Motion-History Image) [20].

Una contribución destacada en esta línea es la propuesta de Gorelick et al. [44], donde las acciones son modeladas como formas espacio-temporales tridimensionales. En este enfoque, se segmentan las siluetas del sujeto en movimiento en cada fotograma, y se combinan a lo largo del tiempo para formar un volumen 3D. Sobre esta forma se aplican descriptores globales que permiten identificar la acción. El método mostró un rendimiento competitivo para el dataset Weizmann [18], gracias a su capacidad para capturar la dinámica estructural completa de cada acción.

No obstante, los métodos globales presentan limitaciones importantes. Según Dollar et al. [35], este tipo de enfoque puede ser demasiado rígido para adaptarse a variaciones en el punto de vista, a la apariencia del sujeto o a la presencia de oclusiones. Además, técnicas como la de Gorelick requieren una segmentación precisa del cuerpo en movimiento, lo cual es difícil de lograr en entornos reales con ruido, fondos complejos o múltiples sujetos en movimiento.

Representaciones Locales

Las representaciones locales se orientan a detectar patrones distintivos en regiones específicas del video, mediante la identificación de puntos de interés en el dominio espacio-temporal. Estos puntos permiten capturar simultáneamente, información espacial y temporal al localizar regiones donde se producen variaciones significativas en la intensidad o en la

2.1 Estado del Arte

forma a lo largo del tiempo. En este contexto, una imagen estática $I(x, y)$ se generaliza como una secuencia $I(x, y, t)$, incorporando la dimensión temporal para reflejar la evolución del contenido visual. Los puntos de interés pueden corresponder a esquinas, regiones con intensidad extrema o curvas con alta curvatura, y se detectan a través de técnicas diseñadas para operar sobre volúmenes espacio-temporales.

Una de las primeras aproximaciones en esta línea fue la que propusieron Schüldt et al. [100], quienes introdujeron el uso de cuboides espacio-temporales extraídos de secuencias de video para entrenar clasificadores SVM. Adicionalmente presentaron el conjunto de datos KTH [100] el que aún hoy es muy utilizado en evaluaciones comparativas y sentó las bases para posteriores desarrollos más sofisticados. Debido a su simplicidad metodológica, junto con una ejecución eficiente, permitió obtener resultados sólidos, aunque con un margen limitado para capturar variaciones complejas en el movimiento.

Poco después, Laptev et al. [68] generalizaron el detector de bordes de Harris [50] a un detector 3D-Harris. Este enfoque permitió identificar regiones con grandes variaciones espaciales y temporales, lo que facilitó la detección de eventos relevantes en los videos.

Además, surgieron métodos basados en trayectorias que analizan el movimiento a lo largo del tiempo siguiendo puntos característicos entre fotogramas. Uno de estos enfoques es el de las trayectorias densas, propuesto por Wang et al. [112]. En este método, se muestrean puntos característicos en cada fotograma del video y luego se calcula su desplazamiento mediante el seguimiento entre fotogramas usando flujo óptico. Las trayectorias generadas a partir de este proceso sirven como representación del contenido del video. Dentro de esta línea de investigación, se comprobó que corregir el movimiento de la cámara mejora los resultados. Esto se logra haciendo coincidir los puntos característicos entre fotogramas mediante descriptores robustos como SURF [12]. Además, mediante la combinación de estas trayectorias con otros descriptores locales, como el Histograma de Gradientes Orientados (HOG [32]), el Histograma de Flujo Óptico (HOF [92]) y el Histograma de Bordes del Movimiento

2.1 Estado del Arte

(MBH [116]), se obtuvieron resultados aún mejores [113].

En otra variante, Gaidon et al. [41] emplearon una técnica de agrupación con el objetivo de construir una jerarquía sobre las trayectorias y así representar diferentes niveles de movimiento en el cálculo de las características de la acción humana. Otro enfoque local es el basado en Local Binary Patterns (LBP), originalmente concebido para el análisis de texturas en imágenes estáticas, esto fue tomado por Zhao et al. [127] quienes adaptaron esta técnica al reconocimiento de acciones humanas en video mediante la extensión del operador LBP al dominio espacio-temporal, conocido como LBP-TOP (Three Orthogonal Planes). Esta variante extrae patrones binarios a partir de tres proyecciones del volumen de video: los planos XY, XT e YT, capturando simultáneamente la información espacial y temporal. El método fue evaluado en el dataset Weizmann [18], logrando buenos resultados en escenarios con fondos estáticos y acciones claramente definidas, pero disminuye ante variaciones de iluminación, el fondo o el punto de vista, debido a la sensibilidad del descriptor a condiciones no controladas. Pese a eso, se destaca por su bajo costo computacional y su fácil implementación, lo que lo hace adecuado para aplicaciones en tiempo real o en entornos con recursos acotados.

A pesar de los avances logrados por los enfoques clásicos, sus limitaciones se vuelven evidentes ante escenarios complejos y condiciones no controladas. Muchos métodos dependen en gran medida de una segmentación precisa del cuerpo humano o del supuesto de fondos estáticos, lo que reduce su aplicabilidad en contextos reales con múltiples sujetos, movimientos de cámara o variaciones en la iluminación [35, 44]. Además, el diseño manual de características suele requerir un conocimiento profundo del dominio y no siempre generaliza bien a nuevas acciones o bases de datos [52]. Estas restricciones motivaron la búsqueda de estrategias más robustas y adaptativas, capaces de aprender automáticamente representaciones discriminativas a partir de los datos. En este contexto, los enfoques basados en aprendizaje profundo han emergido como una alternativa poderosa para abordar las complejidades del reconocimiento de acciones humanas en video.

2.1.2 Enfoques Basados en Redes Profundas

A diferencia de los métodos tradicionales basados en características diseñadas manualmente (hand-crafted features), que dependen en gran medida del conocimiento experto para extraer descriptores específicos, el aprendizaje profundo ha revolucionado los sistemas de visión al permitir que estos aprendan representaciones directamente a partir de datos sin procesar. Las redes neuronales convolucionales (CNNs), en particular, están compuestas por múltiples capas jerárquicas que aprenden automáticamente características progresivamente más abstractas y discriminativas, optimizando la tarea de clasificación de manera end-to-end. Este paradigma reduce la dependencia de la ingeniería manual y mejora la capacidad de generalización a escenarios complejos y variados, fundamentándose en la capacidad universal de aproximación de las redes neuronales profundas [55].

Redes Espacio-Temporales

Las Redes Neuronales Convolucionales (CNNs) originalmente fueron inspiradas por la estructura del sistema visual y en particular por los modelos propuestos por Hubel y Wisel [57]. Las CNNs se construyen mediante redes neuronales interconectadas de múltiples capas, en las que se extraen características jerárquicas de bajo, intermedio y alto nivel estables para la clasificación. Además, son capaces de aprender estructuras complejas siendo invariantes a traslaciones y rotaciones, así como robustas a pequeñas deformaciones [86].

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 2D, que tienen k filtros (o kernels) dedicados a producir k mapas de características espaciales de las imágenes; la Figura 2.3(a) muestra un ejemplo de convolución. (b) capa de submuestreo: cada mapa de característica generalmente se submuestra mediante una operación de max-pooling, un proceso que reduce progresivamente el

2.1 Estado del Arte

tamaño espacial de la representación y la 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 llegada de las CNNs se remonta a la década de 1980, cuando LeCun presentó una red neuronal convolucional conocida como LeNet [69] para clasificar dígitos manuscritos de la base MNIST. Gana relevancia con el advenimiento de hardware más potente, conjuntos de datos más grandes (por ejemplo, ImageNet) [34] y nuevas ideas, que en consecuencia mejoraron la arquitectura de red [108]. La idea original de las CNN profundas continuó siendo desarrollada por Krizhevsky y sus colegas, quienes diseñaron una CNN revolucionaria, conocida como AlexNet, pionera de las CNN profundas modernas, con múltiples capas convolucionales y de agrupación máxima que proporcionan un aprendizaje de características más profundo. Se han logrado éxitos posteriores desde 2014, cuando VGG [104], GoogLeNet (es decir, la red Inception) [108], ResNet [51] y Xception [30] se introdujeron en el ImageNet Large-Scale Visual Recognition Challenge (ILSVRC).

Las 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. Las CNN-3D son una extensión de CNN en el dominio del tiempo que no sólo capturan características espaciales en un fotograma, sino también tiene en cuenta la evolución temporal entre fotogramas consecutivos. La Figura 2.3 ilustra la diferencia entre la operación de convolución bidimensional 2D y 3D, destacando cómo se aplican sobre los datos espaciales y espacio temporales respectivamente.

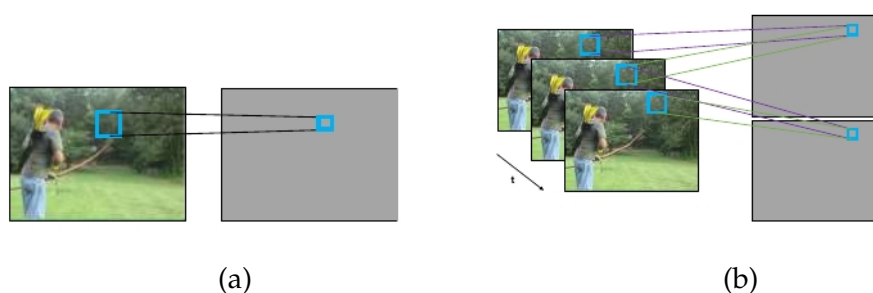


Figura 2.3: Convolución 2D vs Convolución 3D

Entre los primeros enfoques basados en redes convolucionales tridimensionales se encuentra el trabajo de Ji et al. [61], quienes propusieron una arquitectura 3D-CNN para capturar características tanto espaciales como temporales en secuencias de video. Su modelo realiza convoluciones y submuestreo por separado en canales generados a partir de marcos de video adyacentes, combinando luego dicha información para obtener una representación global. Esta propuesta demostró una mejora significativa respecto a arquitecturas 2D tradicionales, aunque su evaluación se limitó a conjuntos como KTH, lo cual restringía su capacidad de generalización.

Posteriormente, Ye et al. [121] presentaron una arquitectura híbrida que combina redes convolucionales 3D para la extracción de características espacio-temporales con redes LSTM, permitiendo incorporar dependencias temporales de más largo alcance. Además, integraron trayectorias densas como una representación manual complementaria, logrando buenos resultados en datasets como HMDB-51, aunque a costa de una mayor complejidad computacional.

Feichtenhofer et al. [39], por su parte, introdujeron mejoras en la fusión de la información espacial y temporal al trasladar esta operación desde la capa de clasificación final hacia etapas intermedias de la red, lo que incrementó la precisión del reconocimiento de acciones, manteniendo una arquitectura eficiente.

Uno de los avances más significativos fue el modelo C3D propuesto por Tran et al. [109], que utiliza convoluciones tridimensionales puras sobre clips de 16 fotogramas para aprender directamente representaciones espacio-temporales. C3D fue entrenado sobre el masivo dataset Sports-1M

2.1 Estado del Arte

y evaluado exitosamente en UCF-101, estableciendo un nuevo estándar sin recurrir a componentes recurrentes. Sin embargo, su rigidez en cuanto a la duración fija de los clips y la alta demanda computacional limitaron su aplicabilidad en contextos más dinámicos.

Con el objetivo de mejorar la eficiencia sin sacrificar rendimiento, Carreira y Zisserman introdujeron el modelo I3D [25], que extiende arquitecturas 2D preentrenadas (como Inception-V1) inflando sus filtros a 3D. Esta técnica de transferencia permitió reducir los tiempos de entrenamiento y mejorar los resultados sobre datasets desafiantes como Kinetics, HMDB-51 y UCF-101. No obstante, la estructura inflada conserva la necesidad de secuencias con longitud constante y sigue siendo intensiva en recursos.

Finalmente, una propuesta intermedia entre precisión y eficiencia fue R(2+1)D [110], que descompone la convolución 3D en dos operaciones sucesivas: una convolución 2D espacial seguida de una convolución 1D temporal. Esta separación facilita la optimización del entrenamiento y reduce la complejidad, logrando un rendimiento competitivo frente a I3D en benchmarks como Sports-1M y Kinetics.

A pesar de los avances logrados por estos enfoques, aún persisten diversas limitaciones que motivan la necesidad de nuevas investigaciones. En primer lugar, las CNN-3D requieren definir una dimensión temporal fija (X, Y, T), con T constante, lo que restringe su capacidad para adaptarse a secuencias de distinta duración, dificultando su aplicación. Además, el uso de convoluciones tridimensionales incrementa considerablemente la complejidad computacional, tanto en términos de parámetros como de requerimientos de procesamiento, lo que hace que estos modelos sean más costosos de entrenar y ejecutar, especialmente en conjuntos de datos grandes o en aplicaciones en tiempo real [48].

Las Redes Neuronales Recurrentes (RNNs) han sido utilizadas para modelar dinámicas temporales complejas en secuencias, siendo especialmente exitosas en tareas como generación de texto y reconocimiento de voz. En el campo del reconocimiento de acciones humanas, que involucra secuencias de movimientos con patrones temporales ricos y variados, las

2.1 Estado del Arte

RNNs y en particular las Redes Neuronales de Memoria a Corto y Largo Plazo (LSTM) [54] se han consolidado como una solución adecuada.

Uno de los primeros trabajos en aplicar LSTM al reconocimiento de acciones fue el de Baccouche et al. [8], quienes utilizaron un modelo basado en LSTM para procesar características hechas a mano extraídas con descriptores clásicos como SIFT y BoVW. Aunque mostraron que las LSTM podían capturar la dinámica temporal de las acciones, su dependencia de características manuales limitaba el potencial del modelo para aprender características más discriminativas directamente desde los datos crudos.

Posteriormente, Yue et al. [123] propusieron apilar capas LSTM sobre redes convolucionales 2D, entrenando conjuntamente las capas convolucionales y recurrentes directamente a partir de secuencias de cuadros RGB y flujo óptico. Esto permitió un modelado más efectivo de la información espacial y temporal en video, logrando avances significativos en datasets como UCF-101.

En paralelo, Donahue et al. [36] presentaron la arquitectura Long-term Recurrent Convolutional Networks (LRCN), que combina CNNs para extracción de características con LSTM para modelar dependencias temporales en videos y secuencias visuales, estableciendo una base para modelos end-to-end capaces de aprender representaciones robustas.

Wang et al. [114] introdujeron una estructura de atención jerárquica que integraba información de movimiento a corto plazo entre fotogramas y estructuras temporales a largo plazo entre segmentos, mejorando la capacidad para capturar transiciones temporales complejas en videos con acciones prolongadas o compuestas, evaluado en datasets como ActivityNet.

Zhang et al. [124] propusieron usar vectores de movimiento en lugar de flujo óptico, mejorando la velocidad de cálculo y permitiendo aplicaciones en tiempo real sin sacrificar significativamente la precisión, una característica importante para sistemas con restricciones computacionales.

Finalmente, Carreira et al. [25] realizaron una comparación amplia de distintas arquitecturas, incluyendo LSTM acopladas a CNN, redes 3D convolucionales y modelos híbridos, destacando que las LSTM consideran

2.1 Estado del Arte

explícitamente el video como una secuencia ordenada de cuadros, permitiendo modelar los cambios dinámicos a lo largo del tiempo de forma directa, lo que resulta esencial para capturar la variabilidad temporal de las acciones humanas.

No obstante, a pesar de sus ventajas, estos modelos también presentan limitaciones que motivan el desarrollo de arquitecturas más especializadas, las cuales se abordarán en las siguientes secciones.

2.1.3 Enfoques Basados en Mecanismos de Atención

El procesamiento visual humano se caracteriza por la focalización selectiva de la atención en regiones relevantes, facilitando la interpretación eficiente de escenas complejas sin analizar exhaustivamente todo el contenido. Este fenómeno ha motivado el desarrollo de mecanismos de atención en visión por computadora y aprendizaje profundo, que han demostrado ser eficaces en tareas como reconocimiento de objetos, generación automática de descripciones, reconocimiento de acciones y respuesta visual a preguntas VQA [82].

Atención Ascendente y Atención Descendente

Los mecanismos de atención visual se clasifican principalmente según el origen o la fuente que guía la atención. La atención ascendente (bottom-up) se basa en características visuales intrínsecas de bajo nivel, como color, intensidad y orientación, que capturan la atención de manera automática y sin requerir conocimiento previo o contexto semántico. Los métodos clásicos emplean descriptores manuales para construir mapas de atención [1, 58, 81], con algoritmos representativos como el contraste basado en histograma (HC) y el contraste basado en región (RC) [27]. Anderson et al. [4] integraron esta atención con una arquitectura Faster R-CNN [96], detectando regiones de interés mediante cuadros delimitadores, mejorando tareas como VQA al alinear las regiones con entidades semánticas relevantes. Sin embargo, estos enfoques presentan limitaciones al no adaptarse dinámicamente a tareas específicas, pues

2.1 Estado del Arte

dependen exclusivamente de señales visuales de bajo nivel. Zhou et al. [128] propusieron superar esta limitación usando promedio global (global average pooling) [75] para identificar automáticamente características discriminativas en redes convolucionales, optimizando la atención para categorías específicas.

En contraste, la atención descendente (top-down) se fundamenta en información de alto nivel, como el contexto o el objetivo de la tarea, que guía el modelo para enfocar regiones relevantes. Modelos pioneros incluyen el de Mnih et al. [90], que usa un agente para explorar dinámicamente entornos visuales, y Ba et al. [7], que aplicó redes recurrentes con aprendizaje por refuerzo para reconocer múltiples objetos. Bahdan et al. [10] introdujeron la atención suave para traducción automática, entrenable de extremo a extremo sin refuerzo, facilitando su integración en modelos secuenciales. Xu et al. [118] compararon atención suave y dura en generación de subtítulos de imágenes, mostrando que la atención suave es más estable, mientras que la dura puede ser más precisa pero requiere aprendizaje por refuerzo. Estas bases inspiraron trabajos posteriores como Sharma et al. [102] para reconocimiento de acciones en video y Yao et al. [120] en generación de descripciones de video.

Más recientemente, la introducción de arquitecturas basadas en Transformers ha revolucionado el uso de mecanismos de atención en visión por computadora. Dosovitskiy et al. [37] presentaron el Vision Transformer (ViT), que adapta el mecanismo de atención global propio de los transformers al dominio visual, reemplazando las convoluciones clásicas. El ViT muestra resultados superiores en clasificación de imágenes cuando se entrena con datasets de gran escala, ya que tiene capacidad para modelar relaciones de largo alcance. Sin embargo, su desempeño depende del tamaño y calidad del dataset, y puede ser menos eficiente en escenarios con datos limitados debido a la ausencia de los sesgos inductivos que caracterizan a las redes convolucionales.

Para el análisis de video, Arnab et al. [5] propusieron ViViT (Video Vision Transformer), que extiende el enfoque transformer para capturar dependencias espaciales y temporales, mejorando el reconocimiento de

2.1 Estado del Arte

acciones. Sin embargo, este modelo requiere gran capacidad computacional para su entrenamiento.

Carion et al. [24] desarrollaron DETR, un modelo de detección de objetos de extremo a extremo basado en transformers, que elimina la necesidad de heurísticas tradicionales como las cajas de ancla (anchor boxes). DETR mejora la interpretabilidad del proceso de detección y reduce su complejidad estructural. Sin embargo, su entrenamiento resulta más lento y requiere un volumen considerable de datos para lograr una convergencia adecuada.

Bertasius et al. [15] propusieron mecanismos de atención espacio temporal eficientes para video, que balancean precisión y costos computacionales al separar atención espacial y temporal. Por su parte, Liu et al. [83] presentaron el Video Swin Transformer, que utiliza ventanas deslizantes jerárquicas para escalar transformers a secuencias largas de video, mejorando el reconocimiento de acciones con mayor eficiencia computacional. No obstante, tanto este como otros modelos recientes mantienen una alta demanda computacional y requieren grandes conjuntos de datos para su entrenamiento óptimo.

Girdhar et al. [42] integraron la atención para enfocarse en actores y contexto en el reconocimiento de acciones, mostrando mejora en precisión en conjuntos de datos complejos, aunque sin explorar mecanismos de atención dura ni métodos de refuerzo.

En cuanto a mejoras en eficiencia, Choromanski et al. [31] propusieron Performers, una variante de los transformers que reemplaza el mecanismo de atención tradicional por una versión más eficiente basada en aproximaciones con kernels. Esta técnica reduce significativamente la complejidad computacional de la atención, facilitando su escalabilidad a secuencias largas, aunque aún presenta algunas limitaciones teóricas en ciertos contextos.

Atención Suave y Atención Dura

Desde un punto de vista computacional, los mecanismos de atención se dividen en suave y dura. La suave asigna pesos continuos y diferenciables

2.1 Estado del Arte

a todas las regiones o elementos, permitiendo el entrenamiento mediante retropropagación estándar, como en los trabajos de Bahdan et al. [10] y Xu et al. [118]. En tanto, la dura realiza selecciones discretas y no diferenciables, requiriendo técnicas de aprendizaje por refuerzo, lo que incrementa la complejidad del entrenamiento.

El modelo de generación de subtítulos de imágenes de Xu et al. [118] es un ejemplo seminal que implementa ambos tipos, mejorando la calidad y eficiencia en la generación de texto visual.

Estos mecanismos han sido ampliamente aplicados a tareas de subtitulado de imágenes, como en You et al. [122], que incorporan atención semántica basada en atributos, y Chen et al. [26], que desarrollan atención por canal en mapas convolucionales. Anderson et al. [4] combinan atención ascendente y descendente para mejorar notablemente la generación de descripciones textuales.

La atención también es esencial en la detección de objetos, con modelos como los de Caicedo et al. [23] que usan Aprendizaje por Refuerzo(RL) para localización, Hara et al. [49] que proponen redes de atención para localización, y Li et al. [70] que integran contextos globales y locales en Fast R-CNN [43]. A su vez, Wang et al. [111] proponen las redes de atención residual, escalables a cientos de capas.

La atención ha demostrado su valor en tareas complejas como la reidentificación de personas. Liu et al. [78] presentan redes de atención comparativas, Rahimpour et al. [94] diseñan modelos CNN con atención para arquitecturas de triplete, y LiWey et al. [72] junto con Si et al. [103] exploran mecanismos de atención múltiple y dual, mejorando así la precisión y adaptabilidad en tareas avanzadas de visión por computadora.

Si bien los avances en mecanismos de atención han potenciado notablemente el reconocimiento de acciones, muchas soluciones actuales enfrentan limitaciones prácticas vinculadas al alto costo computacional, la necesidad de grandes volúmenes de datos y la falta de adaptabilidad en entornos más restrictivos. En este contexto, nuestro enfoque se posiciona como una alternativa eficiente y eficaz que busca capitalizar los beneficios del aprendizaje profundo con atención, sin incurrir en

2.2 Métricas de evaluación

los altos requerimientos de los modelos basados en Transformers. En particular, proponemos una arquitectura CNN-LSTM complementada con un mecanismo de atención que permite focalizar dinámicamente en regiones y momentos temporales relevantes. A esto se suma un módulo de selección de fotogramas representativos, que reduce la redundancia temporal y mejora la eficiencia del procesamiento sin sacrificar precisión. Esta estrategia híbrida apunta a lograr un equilibrio entre rendimiento, interpretabilidad y viabilidad computacional, especialmente útil en escenarios donde los recursos son limitados o los datos disponibles son acotados.

2.2 Métricas de evaluación

Para evaluar el rendimiento de las diferentes propuestas, las bases de datos para el reconocimiento de acciones humanas en videos suelen emplear una métrica de evaluación de exactitud (accuracy) junto a una validación cruzada como se describe en el trabajo de Bishop [16]. Esta métrica ha sido ampliamente aplicada en tareas de reconocimiento de acciones humanas [115]. A continuación, revisaremos estos conceptos en detalle.

2.2.1 Exactitud (Accuracy)

Se define como la proporción entre el número de muestras que son clasificadas correctamente y el total de muestras analizadas. La Figura 2.4 muestra los principales componentes involucrados en el cálculo de la exactitud en un problema binario.

2.2 Métricas de evaluación

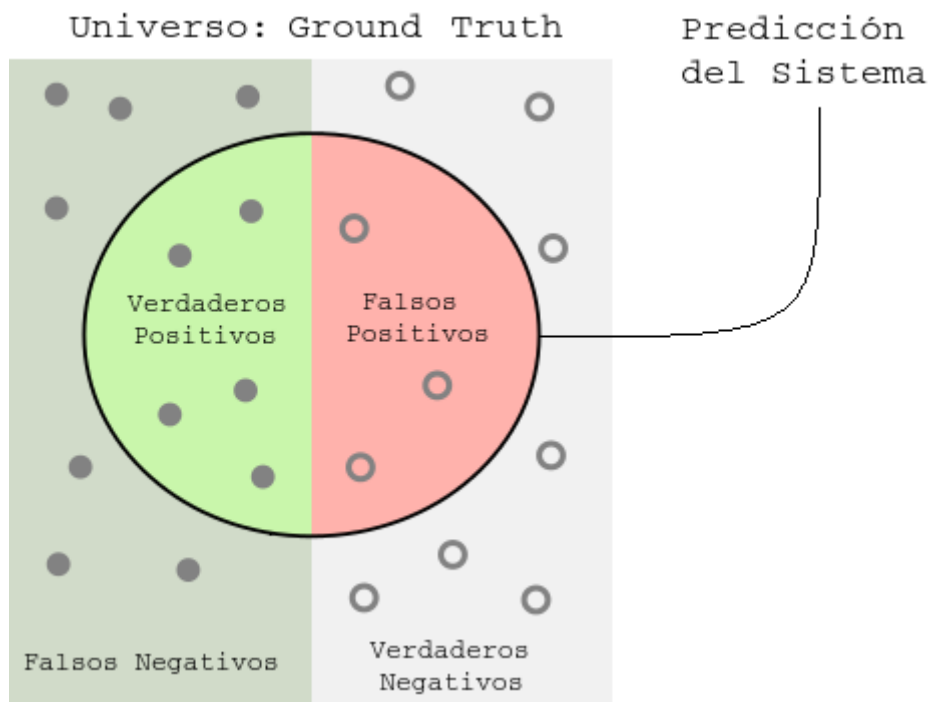


Figura 2.4: Ejemplo de clasificación de un problema binario: Datos de referencia vs predicción.

- (a) Las muestras están etiquetadas de acuerdo con su clase real, conocida como *Ground Truth* (GT). En el ejemplo, se representan con ● para la Clase 1 y ○ para la Clase 2.
- (b) Un modelo de clasificación intenta predecir la clase a la que pertenece cada muestra. En la figura, la predicción se representa mediante un círculo: todas las muestras ubicadas dentro del círculo se asignan a la Clase 1 (●), mientras que las que quedan fuera se asignan a la Clase 2 (○).

Pueden existir diferencias entre lo predicho por el modelo y la clase real. A continuación, se describen los posibles casos de clasificación:

- Verdaderos Positivos (TP): muestras de clase 1 correctamente clasificadas como clase 1.
- Verdaderos Negativos (TN): muestras de clase 2 correctamente clasificadas como clase 2.

2.2 Métricas de evaluación

- Falsos Positivos (FP): muestras de clase 2 incorrectamente clasificadas como clase 1.
- Falsos Negativos (FN): muestras de clase 1 incorrectamente clasificadas como clase 2.

Finalmente, la exactitud se obtiene mediante:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

La métrica está acotada en el intervalo $[0, 1]$, donde valores más cercanos al 1 nos indican mejor desempeño del sistema.

2.2.2 Matriz de Confusión

Cuando el problema no es binario, la exactitud se obtiene a partir de la matriz de confusión. La Tabla 2.1 muestra un ejemplo general de una matriz de confusión para n -clases. La matriz es cuadrada, de dimensión $n \times n$, donde n representa el número de clases. Las filas hacen referencia a las etiquetas verdaderas de las muestras, mientras que las columnas representan las clases predichas por el sistema.

	Predicción del Sistema			
	$Clase_1$	$Clase_2$	$\dots$	$Clase_n$
$Clase_1$	$matriz_{11}$	$matriz_{12}$	$\dots$	$matriz_{1n}$
$Clase_2$	$matriz_{21}$	$matriz_{22}$	$\dots$	$matriz_{2n}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$Clase_n$	$matriz_{n1}$	$matriz_{n2}$	$\dots$	$matriz_{nn}$

Cuadro 2.1: Matriz de confusión para n clases.

La matriz de confusión se construye de la siguiente manera:

- Inicialización: $\forall i, j = 1 \dots n : matriz_{ij} = 0$.
- Luego, $\forall$ muestra $\in$ Test:

$g \leftarrow \text{ClaseReal}(\text{muestra})$

2.2 Métricas de evaluación

$p \leftarrow \text{Predecir}(\text{muestra})$

$\text{matriz}_{gp} \leftarrow \text{matriz}_{gp} + 1$

Los elementos en la diagonal representan las muestras que fueron clasificadas correctamente, es decir, aquellas para las que la predicción coincide con la etiqueta verdadera. La exactitud (*accuracy*) para un problema de clasificación multiclase se calcula como la proporción entre la suma de las muestras correctamente clasificadas y el total de muestras evaluadas, es decir, la suma de todos los elementos de la matriz, expresada mediante la fórmula:

$$\text{accuracy} = \frac{\sum_{i=1}^n \text{matriz}_{ii}}{\sum_{i=1}^n \sum_{j=1}^n \text{matriz}_{ij}} \quad (2.2)$$

2.2.3 Validación Cruzada

La Validación Cruzada se utiliza para garantizar la independencia de los datos en los entrenamientos y en los testeos. En la validación cruzada de K iteraciones (K -fold cross-validation) el conjunto de datos se divide en K subconjuntos. Uno de los subconjuntos se utiliza como conjunto de prueba y el resto ($K-1$) como conjunto de entrenamiento. El proceso de validación cruzada se repite durante K iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Por último, se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. La Figura 2.5 muestra un ejemplo de partición de los datos con $K = 5$.

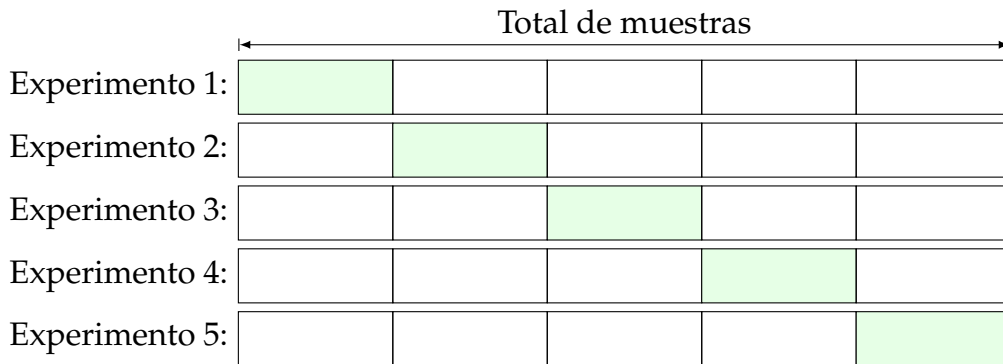


Figura 2.5: Ejemplo de K -fold cross validation con $K = 5$, el área gris representa el subconjunto de prueba.

2.3 Bases de datos

El reconocimiento de acciones en videos ha experimentado avances significativos gracias a la creciente disponibilidad de bases de datos especializadas. Estas no sólo facilitan la evaluación del rendimiento de los modelos propuestos, sino que también permiten comparaciones directas con los resultados reportados en la literatura.

Dado que la tesis no se centra en un dominio de aplicación específico, se seleccionaron los conjuntos de datos KTH y UCF-11 para las pruebas preliminares, por su número reducido de clases y menor complejidad. Para evaluaciones más exigentes, se emplearon las bases HMDB-51 y UCF-101, las cuales presentan una mayor diversidad y complejidad. Estos conjuntos son ampliamente reconocidos como benchmarks estándar en el campo del reconocimiento de acciones humanas en video, y ofrecen una serie de desafíos para medir el desempeño de los modelos propuestos.

A continuación, se describen en detalle las características principales de estos datasets.

2.3.1 KTH

KTH [100] es un conjunto clásico y que es muy utilizado en tareas de reconocimiento de acciones en video. Incluye secuencias de video correspondientes a seis clases de acciones humanas simples (Walking, Jogging, Running, Boxing, Hand waving y Hand clapping), cada una etiquetada y capturada en entornos controlados con fondo uniforme y sin oclusiones significativas. El Cuadro 2.2 presenta un resumen de las clases de acciones mencionadas en este conjunto KTH, con información de cantidad de videos por clase y una descripción general de la variabilidad contextual.

Nº	Clase (Nombre en Castellano)	# Videos	Variabilidad en Contextos y Entornos
1	Walking (Caminar)	100	Exteriores/interiores, diferentes sujetos
2	Jogging (Trotar)	100	Cambios de fondo y velocidad moderada
3	Running (Correr)	100	Fondo estático, diferentes velocidades
4	Boxing (Boxeo)	100	Simulación de golpes al aire, sin oponentes

2.3 Bases de datos

5	Hand waving (Saludo con la mano)	100	Diferentes manos, posiciones del cuerpo
6	Hand clapping (Aplauso)	100	Variaciones entre sujetos, ritmo de palmadas

Cuadro 2.2: Descripción del dataset KTH con la cantidad de videos, y la variabilidad en contextos y entornos por clase.

2.3.2 UCF-11

UCF-11 [79] es un conjunto de datos ampliamente utilizado para el reconocimiento de acciones humanas en video. Contiene videos obtenidos de la plataforma YouTube cubriendo 11 clases de acciones humanas, principalmente de tipo deportivo o recreativo, con mucha variabilidad de fondo, iluminación, calidad de cámara y condiciones de grabación. El Cuadro 2.3 presenta un resumen de las clases del conjunto de datos UCF-11, incluyendo la cantidad de videos por categoría y su variabilidad contextual.

N°	Clase (Nombre en Castellano)	# Videos	Variabilidad en Contextos y Entornos
1	Basketball (Baloncesto)	100	Canchas cubiertas, al aire libre, equipos
2	Diving (Clavados)	100	Piscinas, plataformas, alturas
3	Horse Riding (Montar a Caballo)	100	Terrenos, competencias, entrenamientos
4	Soccer Juggling (Malabarismo con Balón)	100	Canchas, entrenamientos, competencias
5	Swing (Columpio)	100	Parques, patios, columpios
6	Tennis Swing (Swing Tennis)	100	Canchas de tenis, competencias
7	Trampoline Jumping (Saltos en Trampolín)	100	Gimnasios, parques, trampolines
8	Volleyball Spiking (Remate de Voleibol)	100	Canchas de voleibol, competencias
9	Walking (Caminar)	100	Calles urbanas, parques, interiores
10	Golf Swing (Swing Golf)	100	Campos de golf, prácticas, competencias
11	Skiing (Esquí)	100	Estaciones, pistas, condiciones climáticas

Cuadro 2.3: Descripción del dataset UCF-11 con la cantidad de videos, y la variabilidad en contextos y entornos por clase.

2.3 Bases de datos

2.3.3 HMDB-51

HMDB-51 Human Motion dataset [66] es un conjunto de datos de categorías de acción en videos de la vida real recopilados de diversas fuentes, como películas, videos de YouTube, Google y otros. 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 organizada en 3 divisiones. Cada una de las cuales consta de 5,100 videos, 3,570 para entrenamiento y 1,530 para test, es decir, una proporción de 70/30 por clase. El Cuadro 2.4 muestra un resumen de la cantidad de videos acompañado de una breve descripción de cada clase.

Nº	Clase (Nombre en Castellano)	# Videos	Variabilidad en Contextos y Entornos
1	brush.hair (Cepillarse el pelo)	120	Hogar, exteriores, diferentes ángulos
2	cartwheel (Voltereta lateral)	101	Gimnasio, parques, diferentes superficies
3	catch (Atrapar)	100	Interior, exterior, diferentes tipos de objetos
4	chew (Masticar)	121	Diferentes tipos de comida, interiores, exteriores
5	clap (Applaudir)	115	Eventos, solitario, en grupo
6	climb (Escalar)	116	Rocas, paredes, árboles
7	climb_stairs (Subir escaleras)	104	Diferentes tipos de escaleras, interiores, exteriores
8	dive (Zambullirse)	94	Piscinas, océanos, diferentes alturas
9	draw_sword (Sacar espada)	105	Escenarios históricos, prácticas, competiciones
10	dribble (Regateo de baloncesto)	95	Canchas de baloncesto, calles, diferentes balones
11	drink (Beber)	117	Diferentes tipos de bebidas, interiores, exteriores
12	eat (Comer)	110	Diferentes tipos de comida, interiores, exteriores
13	fall_floor (Caer al suelo)	107	Diferentes superficies, interiores, exteriores
14	fencing (Esgrimir)	103	Competencias, prácticas, diferentes lugares
15	flic_flac (Mortal atrás con manos)	102	Gimnasios, exteriores, diferentes superficies
16	golf (Jugar golf)	101	Campos de golf, prácticas, competiciones
17	handstand (Hacer el pino)	100	Gimnasios, exteriores, diferentes superficies
18	hit (Golpear)	101	Diferentes objetos, interiores, exteriores

2.3 Bases de datos

19	hug (Abrazar)	105	Diferentes entornos sociales, eventos
20	jump (Saltar)	111	Diferentes superficies, interiores, exteriores
21	kick.ball (Patear un balón)	101	Canchas, parques, diferentes tipos de balones
22	kick (Patear)	99	Diferentes superficies, interiores, exteriores
23	kiss (Besar)	111	Diferentes entornos sociales, eventos
24	laugh (Reír)	102	Diferentes entornos sociales, interiores, exteriores
25	pick (Recoger)	105	Diferentes objetos, interiores, exteriores
26	pour (Verter)	104	Diferentes tipos de recipientes, interiores, exteriores
27	pullup (Hacer dominadas)	103	Gimnasios, parques, diferentes barras
28	punch (Dar puñetazo)	106	Diferentes entornos de pelea, interiores, exteriores
29	push (Empujar)	105	Diferentes objetos, interiores, exteriores
30	pushup (Flexión de brazos)	102	Diferentes superficies, interiores, exteriores
31	ride.bike (Montar en bicicleta)	100	Diferentes terrenos, parques, calles
32	ride.horse (Montar a caballo)	104	Diferentes terrenos, competiciones, prácticas
33	run (Correr)	104	Diferentes superficies, parques, calles
34	shake.hands (Estrechar manos)	103	Diferentes entornos sociales, eventos
35	shoot.ball (Disparar un balón)	112	Canchas, parques, diferentes tipos de balones
36	shoot.bow (Disparar con arco)	108	Prácticas, competiciones, diferentes entornos
37	shoot.gun (Disparar un arma)	111	Diferentes entornos, prácticas, competiciones
38	sit (Sentarse)	106	Diferentes superficies, interiores, exteriores
39	situp (Hacer abdominales)	100	Gimnasios, hogares, diferentes superficies
40	smile (Sonreír)	107	Diferentes entornos sociales, interiores, exteriores
41	smoke (Fumar)	108	Diferentes entornos sociales, interiores, exteriores
42	somersault (Hacer una voltereta)	99	Gimnasios, parques, diferentes superficies
43	stand (Estar de pie)	106	Diferentes entornos, interiores, exteriores
44	swing.baseball (Batear en béisbol)	110	Canchas de béisbol, prácticas, competiciones
45	sword.exercise (Ejercicio con espada)	105	Prácticas, competiciones, diferentes entornos
46	sword (Espada)	105	Prácticas, competiciones, diferentes entornos
47	talk (Hablar)	109	Diferentes entornos sociales, interiores, exteriores
48	throw (Lanzar)	104	Diferentes superficies, interiores, exteriores
49	turn (Girar)	105	Diferentes superficies, interiores, exteriores

2.3 Bases de datos

50	walk (Caminar)	104	Diferentes superficies, interiores, exteriores
51	wave (Saludar con la mano)	109	Diferentes entornos sociales, eventos

Cuadro 2.4: Descripción del dataset HMDB-51 con la cantidad de videos, y la variabilidad en contextos y entornos por clase.

2.3.4 UCF-101

El dataset UCF-101 propuesto por [107] consta de 101 categorías que 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 recopilieron 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 restantes se utilizan para el entrenamiento. El Cuadro 2.5 muestra un resumen de la cantidad de videos acompañada de una breve descripción de cada clase.

Nº	Clase (Nombre en Castellano)	# Videos	Variabilidad en Contextos y Entornos
1	ApplyEyeMakeup (Aplicar maquillaje en los ojos)	145	Hogar, salones de belleza, diferentes estilos de maquillaje
2	ApplyLipstick (Aplicar lápiz labial)	114	Hogar, salones de belleza, diferentes colores y estilos
3	Archery (Tiro con arco)	145	Campos de tiro, competiciones, prácticas al aire libre
4	BabyCrawling (Gateo de bebé)	132	Hogar, guarderías, diferentes tipos de superficies
5	BalanceBeam (Barra de equilibrio)	108	Gimnasios, competiciones, diferentes niveles de habilidad
6	BandMarching (Marcha de banda)	155	Desfiles, competiciones, diferentes bandas y músicas
7	BaseballPitch (Lanzamiento de béisbol)	150	Campos de béisbol, entrenamientos, diferentes técnicas de lanzamiento
8	Basketball (Baloncesto)	134	Canchas de baloncesto, competiciones, prácticas
9	BenchPress (Press de banca)	166	Gimnasios, competiciones, diferentes pesos y técnicas
10	Biking (Ciclismo)	150	Diferentes terrenos, parques, calles
11	Billiards (Billar)	160	Salones de billar, competiciones, diferentes tipos de mesas
12	BlowDryHair (Secar el cabello)	146	Hogar, salones de belleza, diferentes tipos de secadores
13	BlowingCandles (Soplar velas)	100	Hogar, fiestas, diferentes tipos de pasteles

2.3 Bases de datos

14	BodyWeightSquats (Sentadillas con peso corporal)	112	Gimnasios, hogares, diferentes técnicas de entrenamiento
15	Bowling (Bolos)	157	Boleras, competiciones, diferentes tipos de bolos
16	BoxingPunchingBag (Boxeo contra saco de boxeo)	166	Gimnasios, entrenamientos, diferentes técnicas de boxeo
17	BreastStroke (Estilo braza en natación)	100	Piscinas, competiciones, diferentes estilos de natación
18	BrushingTeeth (Cepillarse los dientes)	100	Hogar, baños, diferentes tipos de cepillos
19	CleanAndJerk (Levantamiento olímpico)	143	Gimnasios, competiciones, diferentes pesos y técnicas
20	CliffDiving (Zambullirse desde el acantilado)	142	Acantilados, océanos, diferentes alturas y técnicas
21	CricketBowling (Lanzamiento de cricket)	150	Campos de cricket, competiciones, diferentes técnicas de lanzamiento
22	CricketShot (Golpe de cricket)	151	Campos de cricket, competiciones, diferentes estilos de golpe
23	CuttingInKitchen (Cortar en la cocina)	134	Hogar, cocinas, diferentes tipos de alimentos y cuchillos
24	Diving (Zambullirse)	120	Piscinas, competiciones, diferentes tipos de clavados
25	Drumming (Tocar la batería)	138	Escenarios, competiciones, diferentes estilos de música y tambores
26	Fencing (Esgrima)	147	Salones de esgrima, competiciones, diferentes estilos y armas
27	FieldHockeyPenalty (Penal de hockey sobre césped)	118	Campos de hockey, competiciones, diferentes técnicas de penal
28	FloorGymnastics (Gimnasia de piso)	200	Gimnasios, competiciones, diferentes rutinas y niveles de habilidad
29	FrisbeeCatch (Atrapar frisbee)	142	Parques, competiciones, diferentes estilos de lanzamiento y frisbees
30	FrontCrawl (Estilo libre en natación)	100	Piscinas, competiciones, diferentes estilos de natación
31	GolfSwing (Swing de golf)	151	Campos de golf, prácticas, diferentes técnicas de swing
32	Haircut (Corte de cabello)	100	Salones de belleza, hogar, diferentes estilos de cortes
33	HammerThrow (Lanzamiento de martillo)	138	Estadios, competiciones, diferentes técnicas de lanzamiento
34	Hammering (Martillar)	100	Hogar, construcción, diferentes tipos de martillos y materiales
35	HandstandPushups (Flexiones de mano)	140	Gimnasios, entrenamientos, diferentes niveles de habilidad
36	HeadMassage (Masaje de cabeza)	100	Spa, salones de masaje, diferentes técnicas y métodos
37	HighJump (Salto de altura)	106	Pistas de atletismo, competiciones, diferentes técnicas de salto
38	HorseRace (Carrera de caballos)	110	Hipódromos, competiciones, diferentes tipos de carreras y caballos
39	HulaHoop (Hula hoop)	100	Parques, competiciones, diferentes estilos de hula hoops
40	IceDancing (Danza sobre hielo)	100	Pistas de hielo, competiciones, diferentes estilos de danza
41	JavelinThrow (Lanzamiento de jabalina)	100	Estadios, competiciones, diferentes técnicas de lanzamiento
42	JugglingBalls (Malabares con pelotas)	119	Parques, circos, diferentes números y estilos de malabarismo
43	JumpingJack (Saltos jumping jack)	112	Gimnasios, hogares, diferentes ritmos y técnicas de salto
44	JumpRope (Saltar la cuerda)	160	Gimnasios, parques, diferentes estilos y ritmos de saltos

2.3 Bases de datos

45	Kayaking (Kayak)	150	Ríos, lagos, océanos, diferentes tipos de aguas y técnicas de remo
46	Knitting (Tejer)	100	Hogar, talleres, diferentes tipos de lanas y técnicas de tejido
47	LongJump (Salto de longitud)	132	Pistas de atletismo, competiciones, diferentes técnicas de salto
48	Lunges (Zancadas)	100	Gimnasios, hogares, diferentes técnicas de ejercicio y ritmos
49	MilitaryParade (Desfile militar)	100	Desfiles, eventos, diferentes tipos de tropas y ceremonias
50	Mixing (Mezclar)	100	Cocinas, hogares, diferentes ingredientes y métodos de mezcla
51	MoppingFloor (Trapear el piso)	100	Hogar, limpieza, diferentes tipos de pisos y equipos de trapeado
52	Motorcycling (Motociclismo)	122	Calles, carreteras, diferentes tipos de motocicletas y terrenos
53	NailPolishing (Pintarse las uñas)	100	Hogar, salones de belleza, diferentes colores y estilos de esmalte
54	Nunchucks (Nunchaku)	142	Gimnasios, prácticas, diferentes estilos de artes marciales
55	ParallelBars (Barras paralelas)	100	Gimnasios, competiciones, diferentes niveles de habilidad y ejercicios
56	PizzaTossing (Lanzamiento de pizza)	100	Cocinas, restaurantes, diferentes tipos de pizzas y técnicas de lanzamiento
57	PlayingCello (Tocar el violoncello)	100	Escenarios, estudios, diferentes estilos de música y técnicas de violoncello
58	PlayingDaf (Tocar daf)	112	Escenarios, festivales, diferentes estilos de música y técnicas de daf
59	PlayingDhol (Tocar dhol)	100	Escenarios, festivales, diferentes estilos de música y técnicas de dhol
60	PlayingFlute (Tocar flauta)	105	Escenarios, estudios, diferentes estilos de música y técnicas de flauta
61	PlayingGuitar (Tocar guitarra)	100	Escenarios, estudios, diferentes estilos de música y técnicas de guitarra
62	PlayingPiano (Tocar piano)	105	Escenarios, estudios, diferentes estilos de música y técnicas de piano
63	PlayingSitar (Tocar sitar)	117	Escenarios, festivales, diferentes estilos de música y técnicas de sitar
64	PlayingTabla (Tocar tabla)	100	Escenarios, festivales, diferentes estilos de música y técnicas de tabla
65	PlayingViolin (Tocar violín)	100	Escenarios, estudios, diferentes estilos de música y técnicas de violín
66	PoleVault (Salto con pértiga)	112	Pistas de atletismo, competiciones, diferentes técnicas de salto
67	PommelHorse (Caballo con arcos)	100	Gimnasios, competiciones, diferentes niveles de habilidad y técnicas
68	PullUps (Dominadas)	150	Gimnasios, parques, diferentes niveles de habilidad y técnicas
69	BasketballDunk	48	Canchas interiores, exteriores, competiciones, juegos informales
70	PushUps (Flexiones de brazos)	130	Gimnasios, hogares, diferentes estilos y ritmos de flexiones
71	Rafting (Rafting)	124	Ríos, rápidos, diferentes tipos de balsas y técnicas de remo
72	RockClimbingIndoor (Escalada en roca indoor)	100	Gimnasios, competiciones, diferentes niveles de habilidad y rutas de escalada
73	RopeClimbing (Escalada de cuerda)	100	Gimnasios, competiciones, diferentes alturas y técnicas de escalada
74	Rowing (Remo)	100	Ríos, lagos, océanos, diferentes tipos de botes y técnicas de remo
75	SalsaSpin (Giro de salsa)	100	Escenarios, clubes, diferentes estilos de baile y giros de salsa

2.3 Bases de datos

76	ShavingBeard (Afeitarse la barba)	100	Hogar, baños, diferentes tipos de afeitadoras y técnicas
77	Shotput (Lanzamiento de bala)	134	Estadios, competiciones, diferentes técnicas de lanzamiento
78	SkateBoarding (Patineta)	120	Calles, parques, diferentes trucos y estilos de patineta
79	Skiing (Esquí)	136	Montañas, estaciones de esquí, diferentes tipos de nieve y técnicas de esquí
80	Skijet (Motos de agua)	100	Lagos, océanos, diferentes velocidades y maniobras de motos de agua
81	SkyDiving (Paracaidismo)	103	Cielos, competiciones, diferentes altitudes y técnicas de salto
82	SoccerJuggling (Malabares con balón de fútbol)	100	Parques, canchas, diferentes estilos y números de malabares
83	SoccerPenalty (Penal de fútbol)	134	Canchas de fútbol, competiciones, diferentes técnicas de penal
84	StillRings (Anillas)	122	Gimnasios, competiciones, diferentes niveles de habilidad y ejercicios
85	SumoWrestling (Sumo)	100	Rings de sumo, competiciones, diferentes estilos de lucha y técnicas
86	Surfing (Surf)	100	Océanos, playas, diferentes tipos de olas y técnicas de surf
87	SwingingOnBars (Columpios)	100	Parques, competiciones, diferentes estilos y alturas de columpios
88	TableTennisShot (Tiro de tenis de mesa)	110	Mesas de ping pong, competiciones, diferentes técnicas de tiro
89	TaiChi (Tai Chi)	100	Parques, clases, diferentes estilos y ritmos de Tai Chi
90	TennisSwing (Swing de tenis)	100	Canchas de tenis, competiciones, diferentes técnicas de swing
91	ThrowDiscus (Lanzamiento de disco)	100	Estadios, competiciones, diferentes técnicas de lanzamiento
92	TrampolineJumping (Saltos en trampolín)	119	Gimnasios, parques, diferentes tipos de saltos y trampolines
93	Typing (Escribir en teclado)	100	Oficinas, hogares, diferentes velocidades y técnicas de escritura
94	UnevenBars (Barras asimétricas)	100	Gimnasios, competiciones, diferentes niveles de habilidad y ejercicios
95	VolleyballSpiking (Remate de voleibol)	133	Canchas de voleibol, competiciones, diferentes estilos de remate
96	WalkingWithDog (Caminar con perro)	100	Parques, calles, diferentes tipos de perros y estilos de caminar
97	WallPushups (Flexiones de pared)	116	Gimnasios, hogares, diferentes estilos y ritmos de flexiones
98	WritingOnBoard (Escribir en pizarra)	100	Escuelas, oficinas, diferentes tipos de pizarras y estilos de escritura
99	YoYo (Yo-Yo)	137	Parques, competiciones, diferentes estilos y trucos de yo-yo
100	Badminton	59	Canchas interiores, exteriores, torneos, juegos recreativos
101	Ballet	50	Teatros, estudios de danza, actuaciones, competencias

Cuadro 2.5: Descripción del dataset UCF-101 con la cantidad de videos, y la variabilidad en contextos y entornos por clase.

Redes Neuronales CNN-LSTM

3.1 Camino Hacia el Modelo CNN-LSTM

3.1.1 Redes Neuronales Tradicionales

El primer modelo artificial de una neurona fue propuesto por McCulloch et al. [88] en 1943, donde modelaron una red neuronal simple para describir el funcionamiento de las neuronas en el cerebro. En 1950, Hebb [6] destaca el hecho de que las conexiones neuronales se fortalecen cada vez que se utilizan. La estructura básica de una red neuronal está compuesta por un conjunto de nodos, cada uno con una función de activación que simboliza la neurona en el modelo biológico. Estos nodos están interconectados a través de conexiones ponderadas, las cuales representan la fuerza de las sinapsis entre las neuronas. La red se alimenta con una entrada, activando algunos o todos los nodos, y luego las conexiones ponderadas distribuyen estas activaciones por toda la red.

A lo largo de los años, se han propuesto diferentes tipos de redes neuronales, en las que se destacan el perceptrón simple [97], las redes de función de base radial [22], mapas de Kohonen [64], y la más utilizada es el Perceptrón Multicapa o Multilayer Perceptron (MLP) [17]. En este caso, Hornik et al. [56] demuestra que un MLP con una sola capa oculta que contiene un número suficiente de unidades no lineales puede aproximar cualquier función continua en un dominio acotado.

3.1 Camino Hacia el Modelo CNN-LSTM

Parte del éxito de las redes neuronales se debe a los esfuerzos para entender y mejorar algoritmos que permiten entrenar un modelo con el fin de realizar una tarea específica. Rumelhart et al. [98] proponen retropropagación para entrenar una red neuronal. La idea general consiste en encontrar la combinación de pesos que minimiza la función de error para un conjunto de entrenamiento determinado. Para ello, se emplea el método del descenso de gradiente, calculando la derivada parcial $\frac{\partial O}{\partial w_{ij}}$ de la función de costo O con respecto a cualquier peso en la red w_{ij} .

En una red neuronal MLP, las unidades están organizadas en capas y las conexiones avanzan desde cada nodo en una capa a todos los nodos de la siguiente capa. La Figura 3.1 (derecha) muestra un ejemplo de la arquitectura MLP. Los vectores de entrada alimentan a la capa de entrada y las activaciones resultantes se propagan a través de las conexiones ponderadas hacia la capa de salida. Por lo tanto, un MLP puede verse como una función que mapea un vector de entrada a un vector de salida.

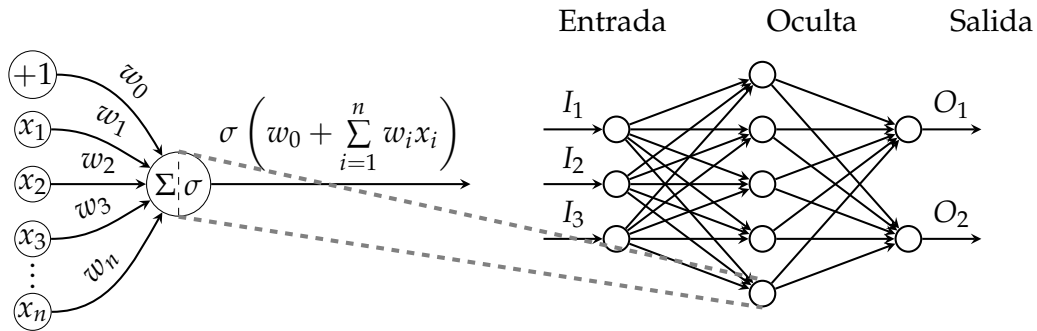


Figura 3.1: Ejemplo de un MLP.

La Figura 3.1 (izquierda) muestra la estructura de una neurona. El perceptrón consta de n entradas, un sesgo, una función de activación y una salida. Las entradas pueden ser la salida de una capa anterior o la entrada misma, si la unidad está en la primera capa oculta. El perceptrón pondera la información de entrada por la conexión propia de sus pesos. Mientras que el sesgo, que es una constante agregada a las entradas, puede verse como un complemento de la función de activación. En la Figura 3.2 se muestran las funciones de activación más utilizadas: Lineal, Sigmoidea y Relu respectivamente.

3.1 Camino Hacia el Modelo CNN-LSTM

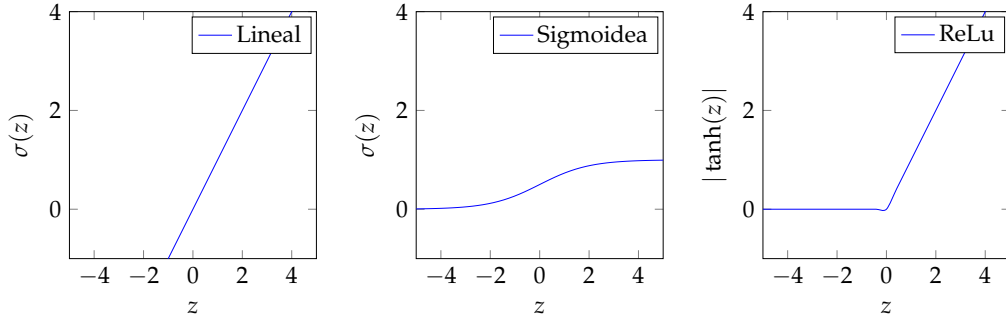


Figura 3.2: Funciones de activación

Sea un MLP con N unidades de entrada y un vector de entrada x de dimensión N , cada unidad en la primera capa oculta calcula una suma ponderada de las unidades de entrada. Nos referimos a esta suma en la unidad oculta h como la entrada de red a la unidad h , denotada por a_h . Entonces se aplica la función de activación θ_h calculando la salida de la unidad b_h . Si denotamos con w_{ih} el peso que conecta la unidad de entrada i con la unidad oculta h , entonces la entrada a_h a la unidad h se calcula como:

$$a_h = w_{0h} + \sum_{i=1}^N w_{ih}x_i \quad (3.1)$$

Aplicando la función de activación θ_h , se obtiene la salida de la unidad h , denotada por b_h :

$$b_h = \theta_h(a_h) \quad (3.2)$$

Una vez que se han calculado todas las activaciones en la primera capa oculta, el proceso se repite capa por capa, completando el pase directo hasta la capa de salida. Por ejemplo, para la unidad h en la capa oculta l , donde H_l es el número de unidades en la capa oculta l tenemos:

$$a'_h = w_{0h'} + \sum_{h'=1}^{H_{l-1}} w_{h'h}b_{h'} \quad (3.3)$$

Una de las funciones de activación más comunes es la función sigmoidea:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.4)$$

3.1 Camino Hacia el Modelo CNN-LSTM

De la ecuación se observa que la función sigmoidea es no lineal. Sabemos que cualquier combinación de operadores lineales es en sí mismo un operador lineal, lo que significa que cualquier MLP con funciones de activación lineales, sin importar cuántas capas ocultas tenga, es equivalente a otro MLP con una sola capa oculta. Por el contrario, las redes neuronales no lineales pueden ganar un poder considerable mediante el uso de capas ocultas sucesivas para representar los datos de entrada [14, 53], por lo que son capaces, por ejemplo, de encontrar límites de clasificación no lineales o modelar ecuaciones no lineales, lo que las hace mucho más robustas. Por lo general, tratamos de encontrar empíricamente el número óptimo de capas ocultas según la complejidad del problema que estamos resolviendo.

La performance de los modelos en problemas de clasificación depende, tanto del número de capas ocultas como de la función de activación. Para estos problemas de clasificación con K clases, usualmente definimos K unidades en la capa de salida y normalizamos la activación de salida usando la función softmax, que es una función que asigna un vector de números reales arbitrarios a un vector de valores reales en el rango de 0 a 1 que suman 1 [21]. De esta manera, se lee la salida de la red neuronal como las probabilidades de que la entrada pertenezca a una clase, como se muestra en la siguiente ecuación:

$$p(C_k|x) = y_k = \frac{e^{a_k}}{\sum_{k'=1}^K e^{a_{k'}}} \quad (3.5)$$

La red neuronal genera un vector de salida de tamaño K , donde cada elemento representa la probabilidad de pertenecer a una clase. Durante el entrenamiento, la salida esperada para cada entrada se representa mediante un vector “one-hot”, en el que todos los valores son 0 excepto en la posición correspondiente a la clase correcta, que se establece en 1. Por ejemplo, si $K = 6$ y la clase correcta es C_4 , la salida de destino estará representada por $(0, 0, 0, 1, 0, 0)$.

Con esta configuración, usar un MLP para una tarea de clasificación es bastante simple. Se alimenta un vector de entrada, luego se propaga la activación hasta la capa de salida; finalmente, se elige la clase correspondiente a la unidad de salida más activa.

3.1 Camino Hacia el Modelo CNN-LSTM

Todas las ecuaciones en MLP son diferenciables, por lo que se pueden entrenar para minimizar cualquier función objetivo diferenciable utilizando el descenso de gradiente a través del algoritmo de retropropagación mencionado anteriormente. La idea básica es encontrar la derivada de la función objetivo con respecto a cada uno de los pesos y sesgos de la red, y ajustar los pesos en la dirección opuesta de esta derivada. La propagación hacia atrás, a veces denominada paso hacia atrás de la red, puede verse como una aplicación repetida de la regla de la cadena para derivadas parciales. Los MLP son operadores diferenciables, por lo que solo necesitamos una función objetivo diferenciable, a menudo denominada función de costo. Hay muchas funciones de costos diferentes, una de las opciones más populares para los problemas de clasificación es el error de entropía cruzada:

$$L = - \sum_{(x,y) \in S} \sum_{y=1}^Y y_k \ln y'_k \quad (3.6)$$

donde S es el conjunto de entrenamiento, y es la salida objetivo e y' es la salida de la red neuronal.

3.1.2 Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNN) originalmente fueron inspiradas en la organización del sistema visual biológico, particularmente en los modelos propuestos por Hubel et al. [57], donde identificaron la existencia de neuronas especializadas en la detección de patrones visuales locales, dando origen a la noción de conectividad local en el procesamiento visual.

La conceptualización inicial de modelos computacionales basados en esta conectividad local y transformaciones organizadas jerárquicamente fue propuesta por Fukushima et al. [40] con el desarrollo del Neocognitron.

LeCun et al. [69] proponen la arquitectura LeNet-5, diseñada específicamente para el reconocimiento de dígitos manuscritos. La Figura 3.3 muestra un esquema general de la arquitectura, la imagen de entrada es de dimensión 32×32 , luego, dos conjuntos de capas

3.1 Camino Hacia el Modelo CNN-LSTM

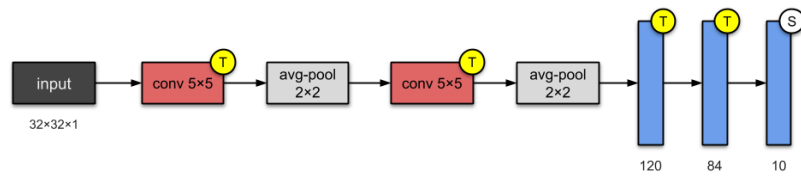


Figura 3.3: Arquitectura LeNet-5: La arquitectura LeNet-5 se compone de capas convolucionales con función de activación Tanh (T), encargadas de extraer características relevantes. Estas son seguidas por capas de submuestreo que reducen la dimensionalidad. Posteriormente, capas totalmente conectadas integran la información extraída, y finalmente, una capa de salida con función Softmax (S) realiza la clasificación.

convolucionales y capas de submuestreo seguidas de una convolucional y dos capas completamente conectadas y finalmente, un clasificador softmax. Este trabajo marcó un punto de inflexión en la aplicación práctica de estas redes en tareas de visión por computadora, ya que no solo logró un progreso sustancial en la comprensión de la conectividad local, sino que también introdujo principios fundamentales del aprendizaje profundo.

Serre et al. [101] propusieron modelos para el reconocimiento rápido de objetos, complementando la evolución de las CNN al considerar aspectos más complejos de la percepción visual. La Figura 3.4 muestra una representación gráfica del modelo.

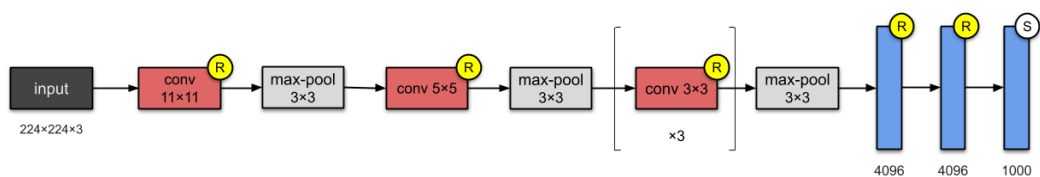


Figura 3.4: Arquitectura AlexNet: Ganó el desafío de ImageNet en 2012. Consta de cinco capas convolucionales seguidas de capas de *pooling* y utiliza la función de activación ReLU (R). Además, tres capas totalmente conectadas para la clasificación.

Simonyan et al. [104] proponen una arquitectura VGG (Visual Geometry Group), conocida por su enfoque simple y profundo, destacando en

3.1 Camino Hacia el Modelo CNN-LSTM

particular variantes como VGG16 y VGG19. Estos modelos se caracterizan por su profunda uniformidad, empleando bloques convolucionales repetitivos con filtros 3×3 píxeles seguidos de capas de submuestreo. Esta estructura, a pesar de su simplicidad, ha demostrado ser eficaz en la clasificación de imágenes, siendo validada en benchmarks como ImageNet, caracterizado por su gran cantidad de clases y variabilidad intra-clase. La Figura 3.5 muestra una representación gráfica del modelo.

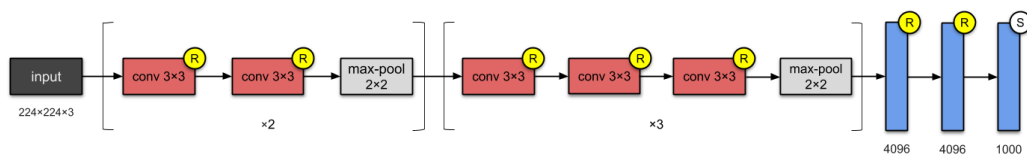


Figura 3.5: La arquitectura VGG16 es una red neuronal convolucional que consta de 16 capas, incluyendo 13 capas convolutivas y 3 capas totalmente conectadas

A medida que las redes neuronales se vuelven más profundas, la precisión tiende a saturarse y, en ocasiones, a degradarse. Este fenómeno se debe principalmente al desvanecimiento de los gradientes (*Vanishing gradients*), un problema bien conocido en el entrenamiento de las redes profundas y ocurre durante el proceso de retropropagación, cuando los gradientes de error se multiplican a través de las capas de la red. En redes profundas, los gradientes pueden volverse extremadamente pequeños a medida que se retropropagan hacia las capas iniciales, lo que impide que las primeras capas ajusten sus pesos de manera significativa. Este fenómeno se debe a la multiplicación repetida de pequeñas derivadas a lo largo de las capas, lo que causa que los gradientes disminuyan exponencialmente, hasta llegar a valores tan pequeños que pierden capacidad de actualización.

El resultado es que las capas más profundas de la red no reciben señales de error lo suficientemente fuertes como para realizar un aprendizaje efectivo, lo que lleva a una saturación de la precisión, ya que el modelo es incapaz de mejorar en tareas complejas debido a la falta de aprendizaje en las primeras capas. Este problema se agrava cuando la red sigue

3.1 Camino Hacia el Modelo CNN-LSTM

aumentando su profundidad, ya que la información útil se diluye al pasar por más capas.

Para abordar este desafío, He et al. [51] introducen la arquitectura ResNet, que incorpora conexiones residuales. Estas conexiones permiten que la información fluya sin restricciones a través de las capas, proporcionando un atajo para los gradientes y permitiendo que las actualizaciones de los pesos no se vean afectadas por el vanishing gradients. Gracias a estas conexiones, la red no solo puede aprender representaciones más profundas, sino que también facilita la propagación de los gradientes a través de las capas, lo que mejora significativamente el rendimiento de las redes profundas. Esta estructura permite que el modelo aprenda de manera más eficiente incluso en arquitecturas muy profundas.

La Figura 3.6 muestra una representación gráfica del modelo.

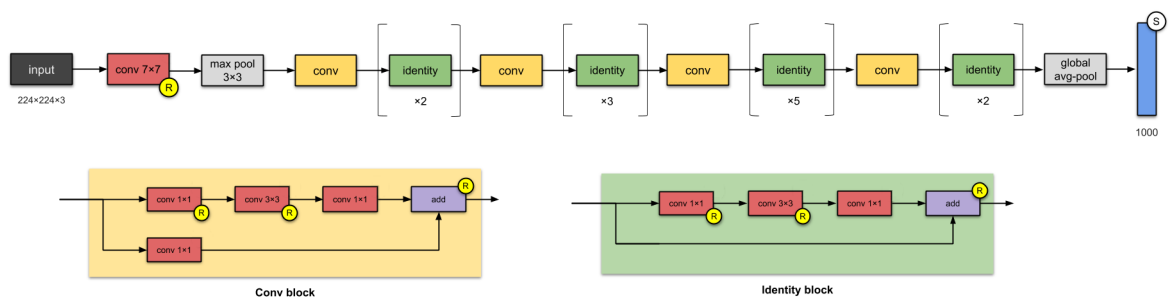


Figura 3.6: La arquitectura ResNet se distingue por su enfoque revolucionario para resolver la degradación del rendimiento en redes neuronales profundas. Introduce bloques residuales que contienen conexiones de salto, permitiendo que la información fluya sin obstáculos a través de las capas al agregar la entrada original a la salida.

Entendiendo las CNN

Una CNN está compuesta generalmente por múltiples capas de convolución y agrupación, seguidas comúnmente por un MLP. Las capas de convolución tienen la tarea de extraer características útiles de la entrada, generando así múltiples mapas de características. La capa de agrupación se utiliza para reducir el tamaño espacial de estos mapas de características.

3.1 Camino Hacia el Modelo CNN-LSTM

La adaptabilidad de las CNN radica en su capacidad para aprender automáticamente estos filtros y características relevantes durante el entrenamiento. Este enfoque dinámico permite a las CNN ajustarse a diversas tareas y conjuntos de datos, convirtiéndolas en una herramienta poderosa y versátil para aplicaciones de visión por computadora y procesamiento de imágenes.

Capa de Convolución

Una capa de convolución, al igual que una red neuronal tradicional, consta de unidades, pero con una disposición y conexión diferente. Las principales diferencias con las redes neuronales son:

- Disposición tridimensional de las unidades.
- Pesos compartidos.
- Conectividad local

Una imagen en colores está formada por tres canales (RGB) y cada canal se define mediante una matriz bidimensional. Por lo tanto, la entrada de la CNN es una matriz tridimensional. La salida de una capa de convolución es nuevamente una matriz tridimensional, con n mapas de características de dos dimensiones, es decir, uno por cada filtro.

El peso compartido significa que se utilizan los mismos pesos para varias unidades de salida. Gracias a esto, las CNNs obtienen la propiedad de invariancia frente a la traslación. Esto significa que se puede detectar una característica específica en cualquier parte de la entrada.

Que la conectividad sea local significa que no todas las unidades de entrada están conectadas con la unidad de salida. La Figura 3.7 muestra la conexión entre la entrada y dos unidades de salida del mapa de características. La entrada $Input[:, :, i]$ puede estar formada por los canales R, G y B de una imagen o por los mapas de características de una capa inmediatamente anterior. Como resultado, se muestra un mapa de características. Para los siguientes mapas de características, el comportamiento es similar. En el ejemplo, los pesos $W_{i,o}$ forman una matriz

3.1 Camino Hacia el Modelo CNN-LSTM

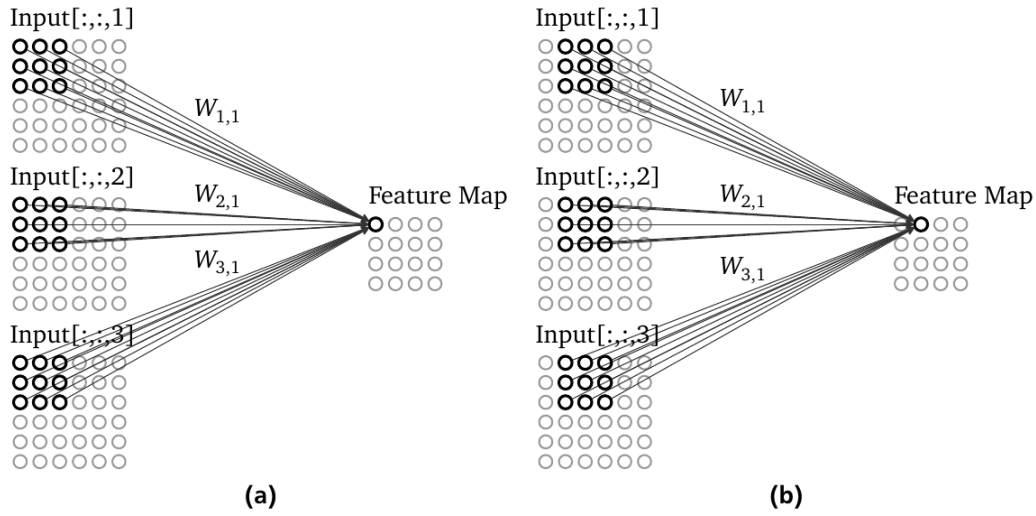


Figura 3.7: Las dos figuras muestran la conexión entre la entrada y dos unidades de salida de la capa de convolución. Además, la capa de convolución tiene solo un mapa de características para la salida. La conectividad local se muestra en las dos figuras, con un tamaño de núcleo de 3×3 . (a) y (b) comparten los mismos pesos para el cálculo del mapa de características. El canal de salida se llama mapa de características.

de 3×3 llamada *kernel*, la cual aplica la operación de convolución en la región marcada de la entrada. Los pesos $W_{:,o}$ se comparten para cada mapa de características, es decir, se utilizan los mismos pesos para el cálculo de todas las unidades en un mapa de características.

El tamaño espacial de los mapas de características es menor que el tamaño espacial de los canales de entrada. En el ejemplo, el tamaño espacial de entrada es 6×6 y el mapa de características solo 4×4 . Esto proviene del tamaño del kernel, porque el kernel solo se puede aplicar en 16 posiciones diferentes en el canal de entrada.

El número de unidades de salida depende del tamaño espacial de la entrada y del número de filtros. El número de filtros describe la cantidad de mapas de características que se deben generar. Pero el tamaño espacial del mapa de características y, por tanto, el número de unidades, depende tanto del tamaño espacial de entrada como del tamaño del kernel.

A través del uso compartido de pesos y la conectividad local, el número de pesos disminuye exponencialmente en comparación con una

3.1 Camino Hacia el Modelo CNN-LSTM

capa completamente conectada con un número similar de unidades. Sea K el ancho y alto de un kernel, I el número de canales de entrada y F el número de filtros, que es equivalente al número de mapas de características, entonces la cantidad de pesos de la capa es $K^2 \cdot I \cdot F$. En el ejemplo de la Figura 3.7, tenemos $K = 3$, $I = 3$ y $F = 1$, por lo tanto, el número de pesos es $3^2 \cdot 3 \cdot 1 = 27$. Mientras que en una red completamente conectada, el número de pesos sería $6 \cdot 6 \cdot 3 \cdot 4 \cdot 4 = 1728$.

Otro parámetro para una capa de convolución, además del número de filtros y el tamaño del kernel, es el paso. El paso es el tamaño del deslizamiento sobre el canal de entrada. En el ejemplo, tenemos un paso de $(1, 1)$, el kernel se desliza una unidad hacia la derecha o hacia abajo para la siguiente unidad de salida. Si aumentamos el paso a $(2, 2)$, el kernel se desliza dos unidades hacia la derecha o hacia abajo para calcular la siguiente unidad de salida. Esto tiene el efecto de que la dimensión espacial se reduce para cada mapa de características.

Finalmente, cabe destacar que matemáticamente, la operación de convolución que se aplica entre $Input[:, :, i] = V_i$ y los pesos $W_{i,o}$ se calcula como:

$$Z_o = \sigma \left(\sum_{i=0} V_i * W_{i,o} + b_o \right) \quad (3.7)$$

donde $*$ denota la convolución, σ la función de activación y b_o el sesgo compartido para el mapa de características. Como V_i y $W_{i,o}$ son matrices bidimensionales, la convolución discreta 2D para dos matrices I , K en el punto (i, j) se define como:

$$(K * I)_{i,j} = \sum_m \sum_n I_{i-m, j-n} K_{m,n} \quad (3.8)$$

En una capa de convolución, se realizan múltiples convoluciones y los resultados se suman, como se muestra en la Ecuación 3.8. Esta suma de convoluciones a través de los diferentes canales de entrada permite combinar características de manera efectiva, lo cual es esencial para el funcionamiento de una CNN.

3.1 Camino Hacia el Modelo CNN-LSTM

Capa de Agrupación

La capa de agrupación se suele aplicar después de algunas capas de convolución. Su objetivo principal es reducir el tamaño espacial de los mapas de características. Esto implica una reducción en la cantidad de parámetros y tiempo de cálculo, ya que la siguiente capa procesa una versión más compacta de los mapas de características. Con una operación de agrupación con paso (stride) 2×2 , la dimensión espacial de los mapas de características se reduce en un 75 %. La agrupación opera de manera independiente en cada mapa de características y no utiliza función de activación ni pesos entrenables.

Además, la capa de agrupación ayuda a hacer que la red sea invariante a pequeñas traslaciones en la entrada. Esto se debe a que una pequeña traslación generalmente no altera significativamente los valores de salida de la capa de agrupación.

Las capas de agrupación más comunes son la agrupación máxima y media. Ambas capas de agrupación toman como parámetros el tamaño del filtro y el paso. La agrupación máxima calcula el máximo, que se encuentra en el campo receptivo del filtro, y la agrupación de promedios calcula el promedio. En la Figura 3.8 se muestra un ejemplo de ambas agrupaciones. Los valores recomendados para el tamaño del filtro y el paso suelen ser de 2×2 . Otra opción que también muestra un gran rendimiento es un filtro de tamaño 3×3 y paso de 2×2 [65].

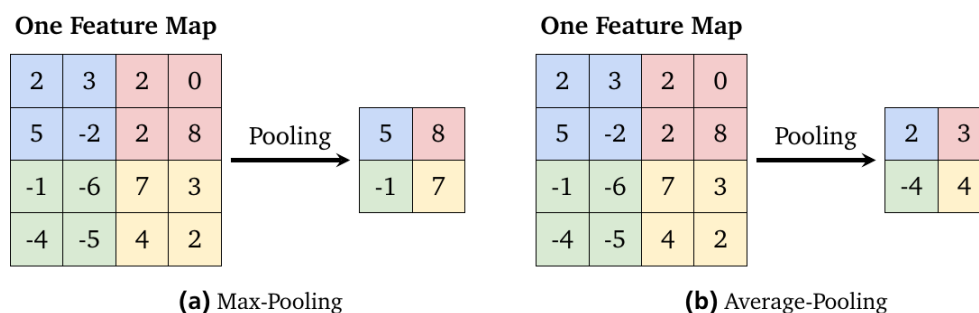


Figura 3.8: Ejemplo de agrupación máxima y agrupación media con un tamaño de filtro de 2×2 y una paso de 2×2 .

3.1.3 Redes Neuronales Recurrentes

Uno de los problemas de las redes neuronales no cíclicas es que la salida depende solo de la entrada, sin tener en cuenta ninguna entrada anterior o futura a la red neuronal. Por ejemplo, en un video, olvidan un fotograma apenas ven el siguiente.

Las redes neuronales recurrentes (RNN) son redes cuyas conexiones forman ciclos. Hay muchos tipos diferentes de redes neuronales recurrentes [38, 59, 67]. La Figura 3.9 muestra el esquema de una RNN simple, que contiene una capa oculta única y autoconectada. Un RNN puede, teóricamente, mapear todo el historial de entradas anteriores a cada salida. Esto significa que el estado interno de una red neuronal recurrente actúa como una memoria de entradas anteriores que se pueden utilizar al calcular la salida de la red.

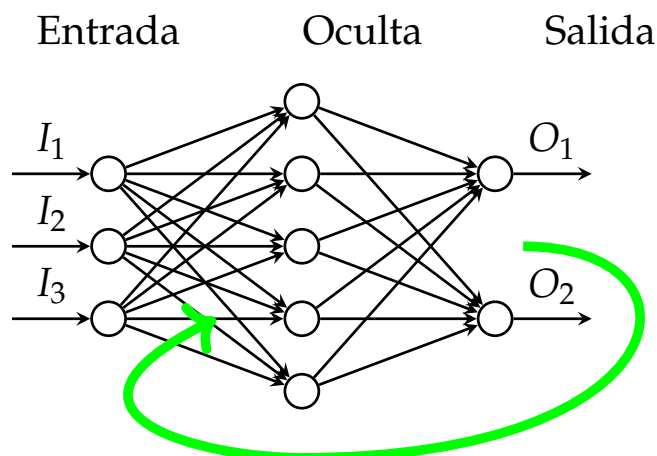


Figura 3.9: Ejemplo de una RNN.

El pase hacia adelante (forward) de un RNN es similar a un MLP, pero las entradas a una unidad oculta provienen no solo de la capa anterior, sino también de la misma capa en el paso de tiempo anterior. Si tenemos una secuencia de entrada x con N características y longitud T , y una RNN con H unidades ocultas y K unidades de salida, sea x_i^t el valor de la entrada i en el tiempo t , y la entrada de la unidad h en el tiempo t y b_j^t la salida de la unidad j en el tiempo t tenemos:

3.1 Camino Hacia el Modelo CNN-LSTM

$$a_h^t = \sum_{i=1}^I w_{ih} x_i^t + \sum_{h'=1}^H w_{h'h} b_{h'}^{t-1} \quad (3.9)$$

$$b_h^t = \theta_h(a_h^t) \quad (3.10)$$

Por lo tanto, se puede calcular una secuencia de activaciones ocultas comenzando desde el principio de la secuencia y aplicando las ecuaciones 3.9 y 3.10 de forma recursiva. Para hacer esto, necesitamos algunos valores iniciales para b_0 , que es el estado de la red antes de que la secuencia ingrese al RNN. La estrategia más común es simplemente establecer estos valores iniciales en 0.

El algoritmo más popular para entrenar RNN es la retropropagación en el tiempo (BPTT) [117]. Es una variante de retropropagación para RNN basada en el mismo concepto de una aplicación repetida de la regla de la cadena. La única diferencia es que ahora la función objetivo depende de la salida de la capa oculta no solo a través de su influencia en la capa de salida, sino también a través de su influencia en la capa oculta en el siguiente paso de tiempo.

En la práctica, BPTT despliega la RNN a lo largo de K pasos de tiempo, como se muestra en la Figura 3.10 (superior), donde f es la función de activación de un nodo recurrente, g es la función de activación de la salida, a_t es la entrada de la anterior. La capa en el tiempo t y x_t es la conexión recurrente en el tiempo t . La red desplegada contiene K instancias de cada unidad oculta recurrente. Luego, el entrenamiento procede de la misma manera que la retropropagación en un MLP, excepto que los patrones de entrenamiento deben visitarse en orden secuencial.

Aunque este tipo de RNN simples teóricamente tiene la capacidad de tener en cuenta cualquier dependencia prolongada, en la práctica, los RNN han estado plagados de un problema práctico importante que hace que su entrenamiento sea un problema muy difícil. Como hemos visto en la Figura 3.10 (abajo), desplegamos la red neuronal a cierta profundidad, típicamente en el rango de 20 a 100. Así, cuando calculamos el gradiente del error de salida total con respecto a las entradas anteriores y se propaga a

3.1 Camino Hacia el Modelo CNN-LSTM

través de la red desplegada, desaparece rápidamente a medida que pasa el tiempo entre las entradas relevantes, conocido como problema de vanishing gradients [14]. Por lo tanto, los RNN clásicos no logran aprender cuando los retrasos de tiempo exceden más de 5 a 10 pasos de tiempo discretos entre las entradas relevantes y las señales objetivo [99]. Este inconveniente de las RNN clásicas es lo que motivó la aparición de las RNN de memoria a corto y largo plazo (LSTM), que se tratan a continuación.

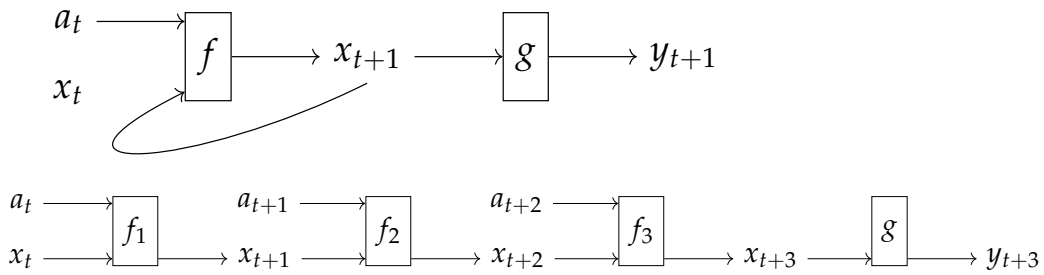


Figura 3.10: Unidad oculta RNN desplegada a profundidad 3

Redes Neuronales LSTM Se han realizado varios intentos para abordar el problema del vanishing gradients para las RNN, algunos de ellos se basaron en algoritmos de entrenamiento sin gradiente [13,67,91], mientras que otros se basaron en la compresión de secuencias jerárquicas [99]. La arquitectura LSTM, introducida por primera vez en [54], ha demostrado ser una solución exitosa y sigue siendo ampliamente utilizada para superar el vanishing gradients en tareas de secuencias temporales.

Las LSTM se diseñaron explícitamente para evitar el problema de vanishing gradients, de modo que sean capaces de aprender las dependencias a largo plazo. La arquitectura LSTM consta de un conjunto de bloques de memoria conectados de forma recurrente, como el que se muestra en la figura 3.11. Estos bloques pueden verse como una versión diferenciable de un chip de memoria electrónica; cada bloque tiene una celda de memoria y tres puertas llamadas puertas de entrada, salida y olvido. La puerta de entrada permite que la información fluya hacia el bloque, la puerta de salida permite que la información del bloque fluya hacia la red y la puerta de reinicio borra la memoria. Por lo tanto, serían similares a una versión continua de escritura, lectura y reinicio en el chip de

3.1 Camino Hacia el Modelo CNN-LSTM

memoria electrónica análoga.

Estas puertas permiten que las redes LSTM almacenen información durante largos períodos de tiempo. Si la puerta de entrada está cerrada, la activación no fluirá hacia la celda, lo que permite que la memoria LSTM mantenga los datos sin cambios para que puedan leerse mucho más tarde en la secuencia, activando la puerta de salida. Además, las puertas también están activas mientras se hace la propagación inversa del error. Por lo tanto, el valor de la celda de memoria puede permanecer sin cambios para cualquier número de pasos de tiempo, tanto en el tiempo de entrenamiento (si las puertas impiden que los gradientes de error se propaguen a la celda) como en el tiempo de prueba (si las puertas impiden que la nueva información entre en la celda de memoria). Por tanto, se evitará el problema del vanishing gradients.

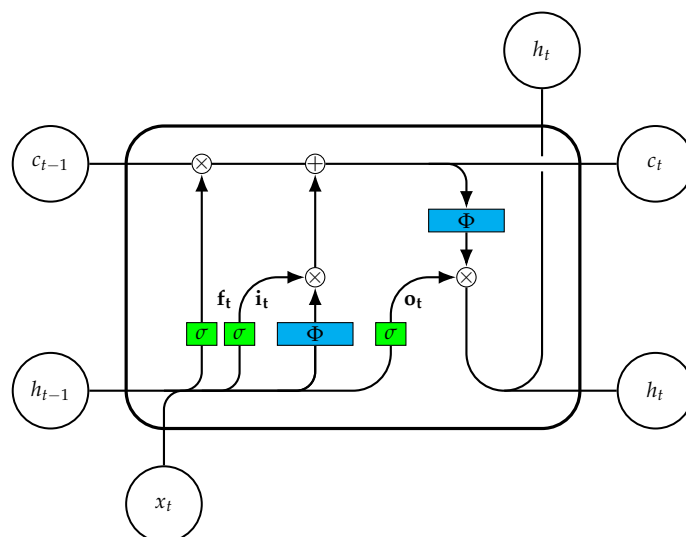


Figura 3.11: Unidad LSTM.

Entendiendo las LSTM

En esta subsección, el objetivo es comprender el paso hacia adelante de las LSTM observando cómo funciona cada parte que se muestra en la figura 3.11. En esta figura, x_t es la entrada de secuencia al bloque en el tiempo t , h_t es la salida del bloque en el tiempo t , c_t es el estado del tiempo de celda t . f_t , o_t y i_t son el valor de olvido, salida y entrada en el tiempo t respectivamente. Se denominan mirillas a las conexiones de la celda de memoria con las tres

3.1 Camino Hacia el Modelo CNN-LSTM

diferentes puertas, que no estaban presentes en las LSTM originales pero que se utilizan mucho en la actualidad, ya que han demostrado mejorar la formulación original [45].

El núcleo del LSTM es la memoria o el estado de la celda, como se muestra en la Figura 3.12. La salida del bloque LSTM consiste en el estado de la celda con algunas interacciones menores, por lo que es muy fácil que la información se mantenga o fluya desde la celda sin cambios, lo que permite que el LSTM tenga dependencias a largo plazo.

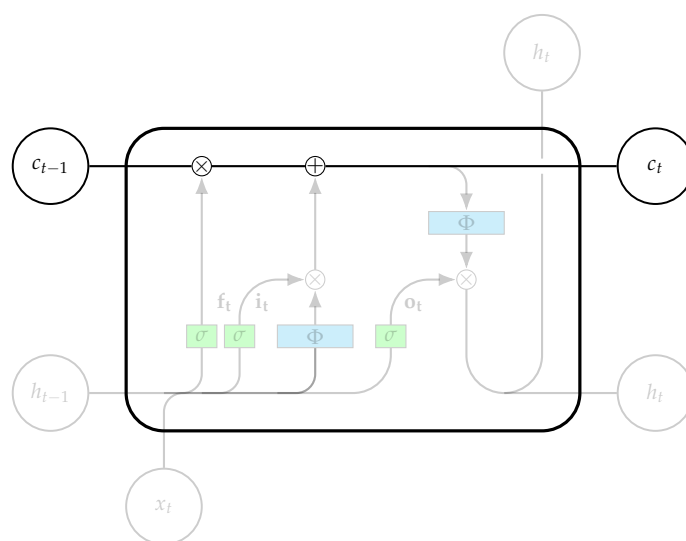


Figura 3.12: Bloque LSTM.

Las puertas controlan el flujo de información dentro de una celda LSTM, permitiendo regular qué datos se incorporan, cuáles se retienen en la memoria y cuáles se descartan. Estas puertas suelen estar compuestas por una activación sigmoidea seguida de una multiplicación elemento a elemento. La función sigmoidea produce valores en el rango de 0 a 1, donde un valor cercano a 0 bloquea el paso de la información, mientras que un valor cercano a 1 permite que la información fluya sin restricciones.

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 secuencia de entrada en el momento t (x_t), y en el estado anterior (h_{t-1}). La ecuación 3.11 muestra cómo la puerta de olvido calcula su valor, donde W_{xf} son los pesos entrenados correspondientes a la

3.1 Camino Hacia el Modelo CNN-LSTM

puerta de olvido y b_f a su sesgo. La figura 3.13 muestra qué parte del bloque de memoria corresponde a este cálculo.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (3.11)$$

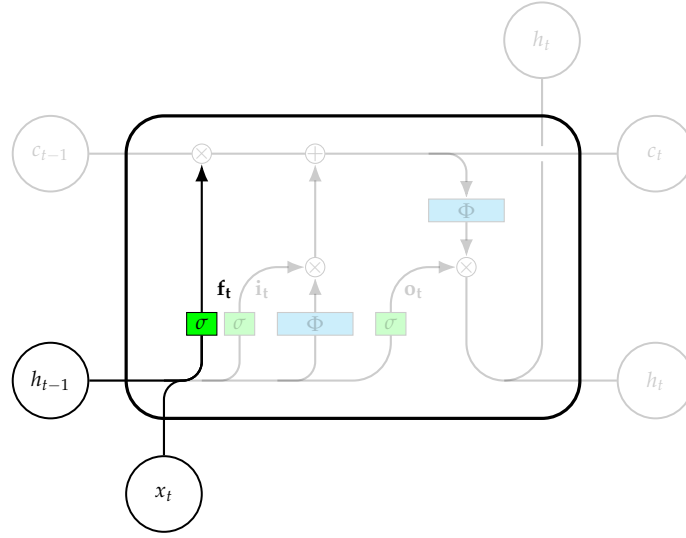


Figura 3.13: Bloque LSTM: Puerta de olvido

El siguiente paso es decidir qué información se agregará a la celda de memoria. Esto se hace en dos pasos diferentes; el 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 partes del bloque de memoria involucradas en esta tarea se muestran en la figura 3.14. Las ecuaciones 3.12 y 3.13 muestran estos cálculos, donde i_t es el valor calculado de la puerta de entrada y z_t es la nueva información que podría agregarse a la celda de memoria.

$$z_t = \phi(W_{xz}x_t + W_{hz}h_{t-1}) \quad (3.12)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (3.13)$$

3.1 Camino Hacia el Modelo CNN-LSTM

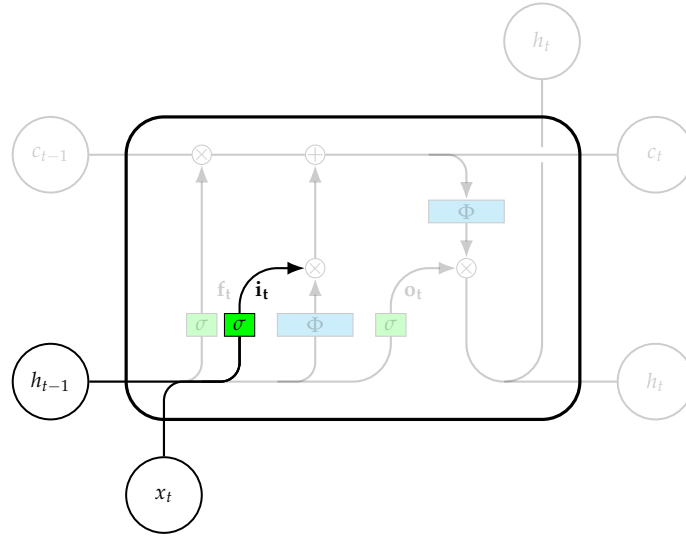


Figura 3.14: Bloque LSTM: Puerta de entrada

Una vez calculados todos estos valores, tenemos todos los recursos 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 3.14. La figura 3.15 muestra todas las partes involucradas en este paso.

$$c_t = f_t \otimes c_{t-1} \oplus i_t \otimes \phi(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3.14)$$

3.1 Camino Hacia el Modelo CNN-LSTM

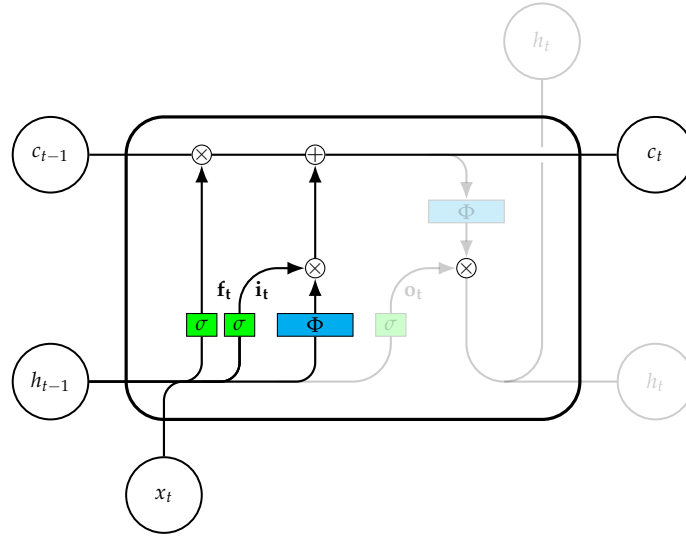


Figura 3.15: Bloque LSTM: Actualización de memoria

Finalmente, debemos decidir qué información 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 multipliquemos por el valor de la puerta de salida que acabamos de calcular, de modo que solo generemos las partes que decidimos. Estos pasos se describen en las ecuaciones 3.15 y 3.16. La figura 3.16 resalta las partes del bloque de memoria que están involucradas en este paso.

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (3.15)$$

$$h_t = o_t \otimes \phi(c_t) \quad (3.16)$$

3.2 Enfoque CNN-LSTM

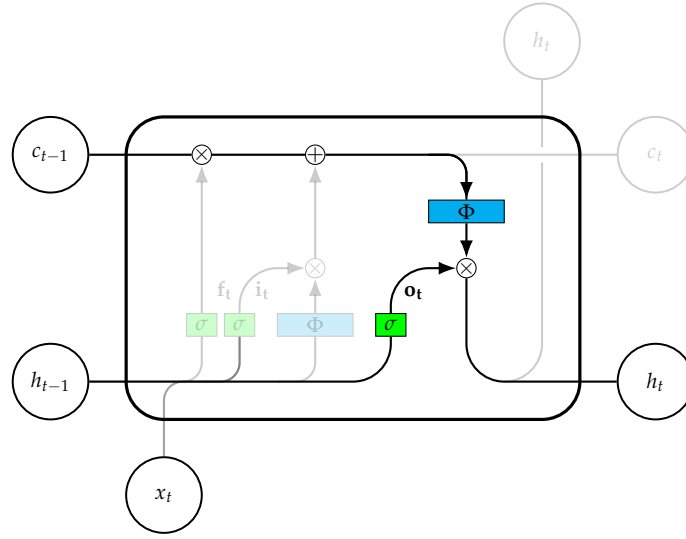


Figura 3.16: Bloque LSTM: Salida

Para entrenar una LSTM, el algoritmo más utilizado es la retropropagación en el tiempo [117] exactamente de la misma manera que se explicó en la sección anterior para los RNN; primero desplegamos la LSTM y todos sus pesos K pasos de tiempo a través del tiempo como se muestra en la figura 3.10 para RNN con la única diferencia de que por cada unidad oculta tenemos 4 pesos en lugar de uno (puertas de entrada, olvido y salida). Después de haber desplegado la LSTM, realizamos la retropropagación del error en orden secuencial.

3.2 Enfoque CNN-LSTM

Sea $v = \{X_1, X_2, \dots, X_n\}$ un video compuesto por una secuencia de fotogramas X_i , con $i = 1, \dots, n$. La Figura 3.17 muestra el sistema HAR base propuesto en este capítulo, el cual se compone de varias fases.

3.2 Enfoque CNN-LSTM

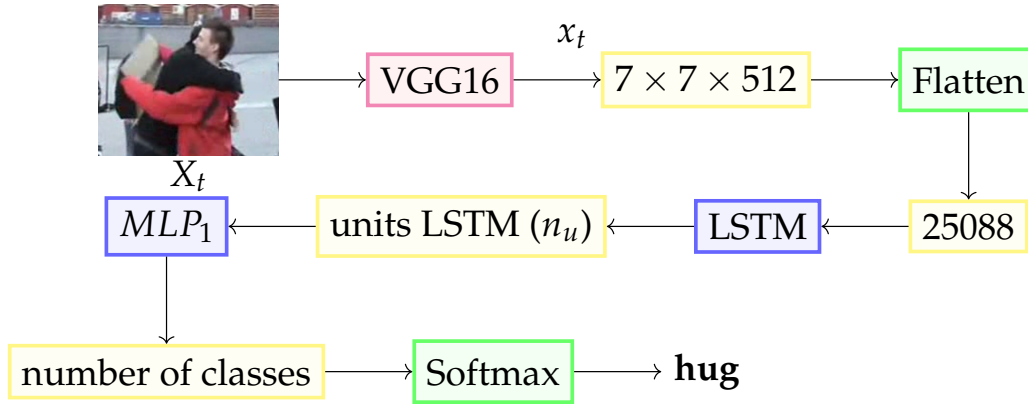


Figura 3.17: Arquitectura base

La fase de entrada normaliza cada video para contener un total de 128 fotogramas, lo que permite estandarizar la secuencia de entrada de todos los videos y facilitar el procesamiento por parte del modelo. Cabe destacar que la cantidad de fotogramas utilizada en los distintos artículos varía típicamente entre 64 y 256, siendo 128 una de las longitudes más empleadas [47, 84], ya que ofrece un equilibrio adecuado entre capturar suficiente información temporal y mantener la eficiencia computacional.

En la fase de extracción de características, se emplea una red neuronal convolucional preentrenada como extractor de características, omitiendo la capa de clasificación típica de la arquitectura. Específicamente, empleamos la arquitectura VGG16 propuesta por [104], la cual procesa cada fotograma $X_t \in v$ y genera un cuboide x_t de tamaño $7 \times 7 \times 512$ como salida de su capa de submuestreo. El Cuadro 3.1 muestra las capas que componen la red, la primera columna indica el número de capa y el tipo de operación (es decir: $2 \times \text{Conv}$: *Convolution*, Max Pool : *Max Pooling*, FC : *Fully Connected*). Mapas y Dimensión indican la cantidad de mapas de características y el tamaño de salida de la capa. Cada cuboide se transforma en un vector unidimensional mediante aplanamiento (*flattening*), representando el video como un conjunto de características de tamaño $128 \times 25,088$. Aquí 25,088 es la dimensión del vector de características de salida de la VGG16 tras la vectorización.

3.2 Enfoque CNN-LSTM

Capa		Mapas	Dimensión	Filtro
Entrada	Imagen	1	$224 \times 224 \times 3$	-
1	$2 \times \text{Conv}$	64	$224 \times 224 \times 64$	3×3
	Max pool	64	$112 \times 112 \times 64$	3×3
3	$2 \times \text{Conv}$	128	$112 \times 112 \times 128$	3×3
	Max pool	128	$56 \times 56 \times 128$	3×3
5	$2 \times \text{Conv}$	256	$56 \times 56 \times 256$	3×3
	Max pool	256	$28 \times 28 \times 256$	3×3
7	$3 \times \text{Conv}$	512	$28 \times 28 \times 512$	3×3
	Max pool	512	$14 \times 14 \times 512$	3×3
10	$3 \times \text{Conv}$	512	$14 \times 14 \times 512$	3×3
	Max pool	512	$7 \times 7 \times 512$	3×3
13	FC	-	25.088	-
14	FC	-	4.096	-
15	FC	-	4.096	-
Salida	FC	-	1.000	-

Cuadro 3.1: Descripción de la arquitectura VGG16 utilizada como extractor de características, omitiendo sus capas totalmente conectadas (FC) [29].

Para modelar la secuencialidad temporal de los fotogramas, empleamos una red LSTM que es capaz de mantener y actualizar una memoria interna a lo largo del tiempo, manteniendo una representación contextual de la secuencia temporal, lo cual resulta útil para clasificar acciones. En cada instante t , la LSTM recibe como entrada el vector de característica x_t , el estado anterior h_{t-1} y la memoria anterior c_{t-1} , y produce como salida el estado actual h_t y la memoria actual c_t , capturando así las características temporales del video.

Después de la LSTM, se utiliza un MLP para la clasificación final. La disposición específica del MLP se detalla en el Cuadro 3.2, donde se especifican los detalles de la configuración de las capas y las funciones de activación utilizadas.

3.3 Experimentos y Resultados

MLP	Capa	Parámetro
MLP ₁	Dropout	0.5
	FC	Número de clases (neuronas)

Cuadro 3.2: Configuración de los MLPs

Finalmente, la dimensión de salida de la arquitectura, representada por el bloque amarillo en la figura, indica el número total de clases de acciones reconocibles, con un nodo por cada clase.

Es fundamental destacar que los pesos asociados tanto con la LSTM como con el MLP se ajustarán durante el entrenamiento de la arquitectura completa, adaptándose específicamente a los datos y a la tarea de reconocimiento de acciones en videos. Para ello, se minimiza la función de pérdida de entropía cruzada mediante el descenso de gradiente estocástico, empleando el optimizador RMSProp [33]. Este optimizador fue elegido por su eficacia al manejar problemas con secuencias temporales, como los que se presentan al trabajar con LSTM, ya que ajusta dinámicamente la tasa de aprendizaje para cada parámetro en función de una media móvil de los cuadrados de los gradientes, ayudando a mitigar oscilaciones y mejorando la convergencia. Si bien algoritmos como Adam han demostrado un rendimiento competitivo y, a menudo, superior en tareas de visión por computador debido a su combinación de momentos adaptativos, en este caso se obtuvo una convergencia más estable con RMSProp, lo cual podría deberse al predominio de componentes recurrentes en esta arquitectura.

3.3 Experimentos y Resultados

3.3.1 Entornos Controlados: KTH y UCF-11

Como primer experimento, se busca determinar el número óptimo de unidades LSTM (n_u) a emplear en la arquitectura. Para ello, se analiza la variación del rendimiento en función de la cantidad de unidades LSTM configuradas. En la Figura 3.18 se presentan dos curvas ROC que muestran la relación entre la exactitud final y el número de unidades n_u para las

3.3 Experimentos y Resultados

bases de datos KTH (izquierda) y UCF-11 (derecha), respectivamente. Con $n_u = 320$ unidades LSTM, se alcanzó una exactitud del 93,86 % en KTH y del 91,94 % en UCF-11. El Cuadro 3.3 resume estos resultados, junto a otros reportados en la literatura. Dichos resultados superan en exactitud a los propuestos por [9, 28, 62, 77, 79, 102, 112], de acuerdo con las métricas de evaluación correspondientes.

Las Figuras 3.19 y 3.20 muestran ejemplos de resultados del enfoque LSTM para los datasets KTH y UCF-11 respectivamente. Cada imagen incluye un fotograma del video de entrada, la etiqueta de la clase y la predicción obtenida junto a la puntuación asociada.

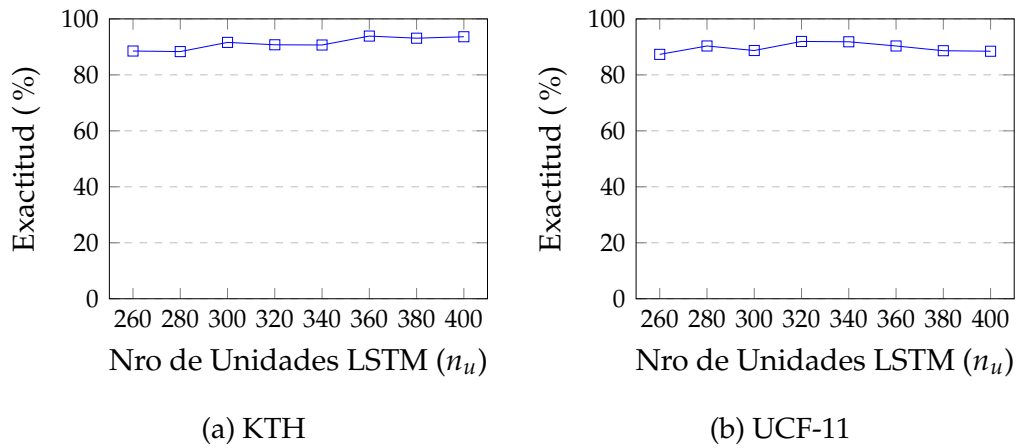


Figura 3.18: Curvas ROC que muestran la variación entre la exactitud final y el número de unidades n_u .

3.3 Experimentos y Resultados

Enfoque	Bibliografía	Base de datos	
		KTH	UCF-11
Liu	[79]	-	71,2 %
Liu	[77]	93,67 %	76,1 %
Wang	[112]	-	84,2 %
Sharma	[102]	-	85,0 %
Cho	[28]	-	88,0 %
Baccouche	[9]	91,04 %	-
Jones	[62]	93,20 %	-
Enfoque CNN-LSTM		93.86 %	91.94 %

Cuadro 3.3: Resultados de la clasificación de videos utilizando la base de datos KTH [100] y UCF-11 [79]. El cuadro resume los resultados obtenidos por la arquitectura junto con otros reportados en la literatura.



Figura 3.19: Ejemplos de salida de la arquitectura. KTH junto con la clase de mayor puntuación (score).

3.3 Experimentos y Resultados

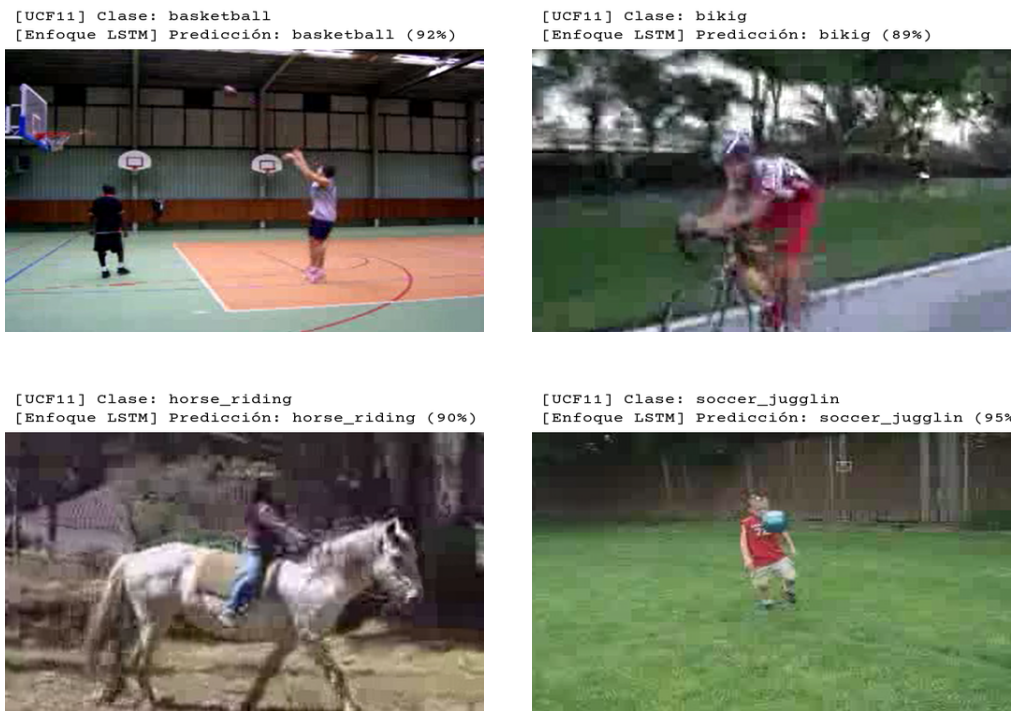


Figura 3.20: Ejemplos de salida de la arquitectura. UCF-11 junto con la clase de mayor puntuación (score).

3.3.2 Entornos No Controlados: HMDB-51 y UCF-101

Siguiendo las métricas de evaluación para cada una de las bases de datos, el Cuadro 3.4 resume los valores de exactitud alcanzados por el enfoque **CNN-LSTM**, junto con los resultados reportados por otros métodos en la literatura. El cuadro incluye la columna *Tipo*, que indica el enfoque general según la clasificación descrita en la sección 2.1, donde EC: Enfoque Clásico, EP: Enfoque basado en Red Profunda, y MA: Enfoque basado en Mecanismo de Atención.

Aunque los resultados no alcanzan un desempeño óptimo, para la base de datos HMDB-51, el modelo logra un desempeño comparable al de enfoques clásicos y a métodos basados en mecanismos de atención sencillos. Para la base de datos UCF-101, el modelo logra un desempeño aceptable, considerando su simplicidad y baja complejidad computacional. Esta característica convierte al modelo en una base sólida y eficiente, que servirá como punto de partida para las mejoras y extensiones que se

3.3 Experimentos y Resultados

abordarán en el capítulo siguiente.

Las Figuras 3.21 (HMDB-51) y 3.22 (UCF-101) muestran ejemplos de predicción del enfoque CNN-LSTM en distintos videos. En cada caso, se indica en la parte superior la base de datos, la clase de referencia, la clase predicha y el nivel de confianza de la predicción.

3.3 Experimentos y Resultados

Enfoque	Año	Tipo	Base de datos	
			HMDB-51	UCF-101
Kuehne et al. [66]	2011	EC	23,0 %	—
Wang et al. [113]	2013	EC	46,6 %	—
Gaidon et al. [41]	2014	EC	41,3 %	—
Simoyan et al. [105]	2014	EP (*)	59,4 %	88,0 %
Sharma et al. [102]	2015	MA	41,3 %	—
Tran et al. [109]	2015	EP	—	90,4 %
Wang et al. [114]	2016	EC	60,1 %	86,0 %
Ye et al. [121]	2016	EP	55,2 %	85,4 %
Feichtenhofer et al. [39]	2016	EP (*)	58,5 %	91,4 %
Qiu et al. [93]	2017	EP	—	88,6 %
Carreira et al. [25]	2017	EP (*)	80,2 %	97,9 %
Li et al. [74]	2018	EP	63,0 %	—
Meng et al. [89]	2018	EP	53,1 %	87,1 %
Zhang et al. [124]	2018	EP	—	86,4 %
Liu et al. [80]	2020	EP	69,1 %	94,3 %
Yang et al. [119]	2020	EP	71,5 %	93,8 %
Li et al. [73]	2020	EP	54,3 %	86,7 %
Majd et al. [85]	2020	EP	61,3 %	92,8 %
Liu et al. [84]	2021	EP	74,4 %	96,6 %
Li et al. [71]	2022	MA	83,1 %	98,8 %
Zhang et al. [126]	2023	EP	79,8 %	91,5 %
Enfoque CNN-LSTM		EP	40,7 %	75,8 %

Cuadro 3.4: Resultados de la clasificación de videos utilizando la base de datos HMDB-51 [66] y UCF-101 [107]. El cuadro resume los resultados obtenidos por el enfoque **CNN-LSTM** junto con otros reportados en la literatura.

3.3 Experimentos y Resultados

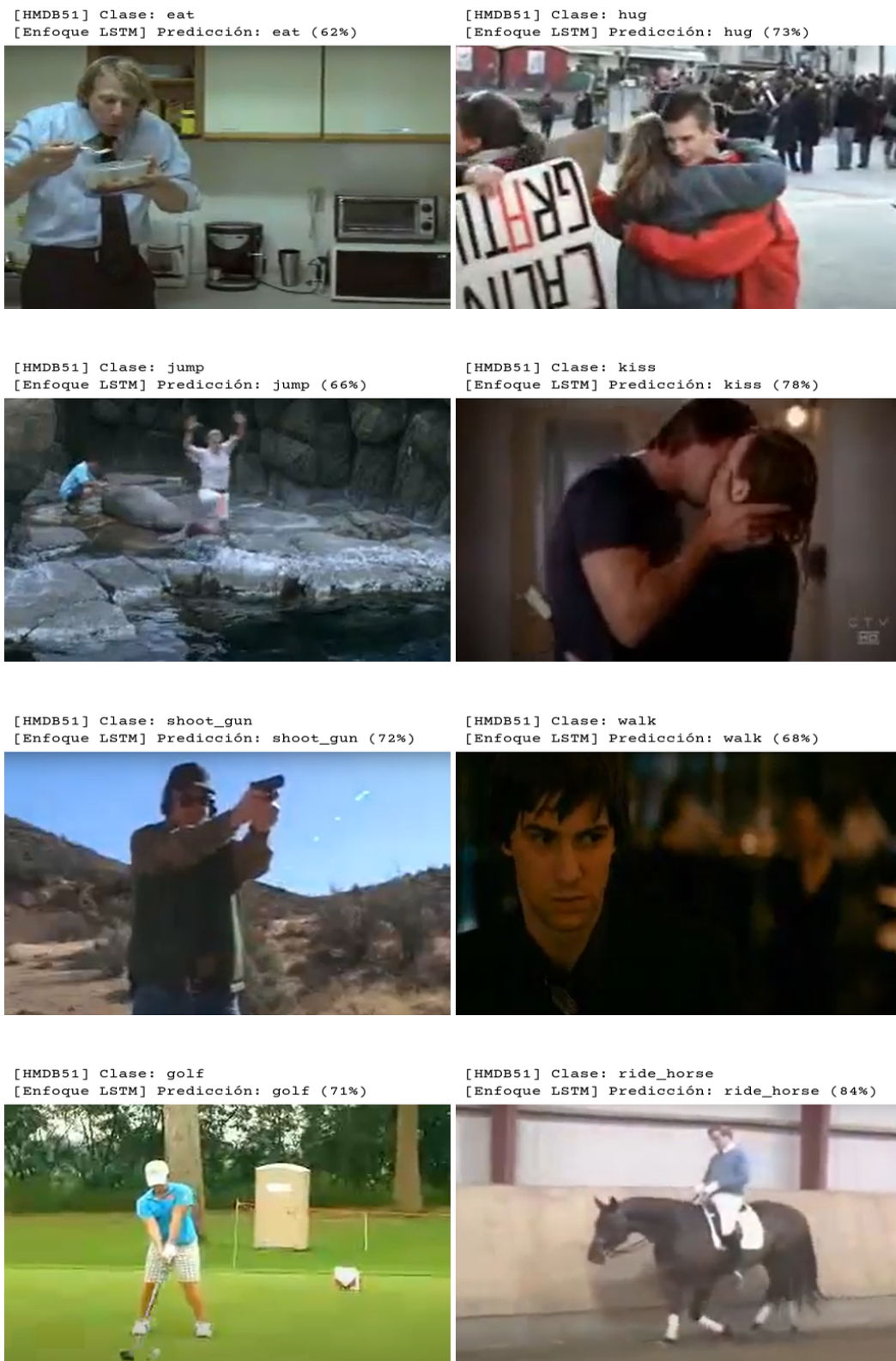


Figura 3.21: Ejemplos de salida del modelo para la base HMDB-51, mostrando la clase predicha con mayor puntuación (*score*).

3.3 Experimentos y Resultados

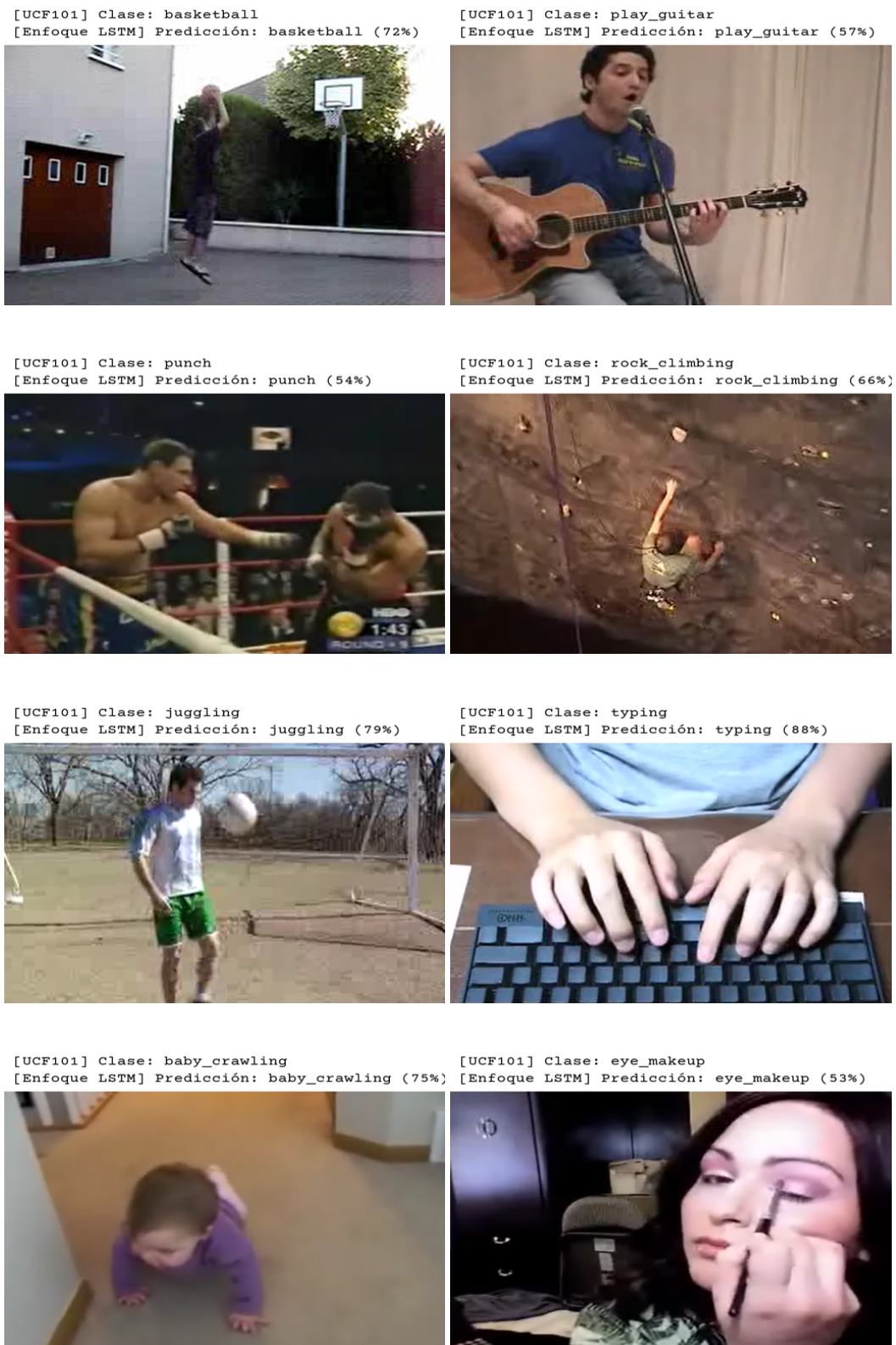


Figura 3.22: Ejemplos de salida del modelo para la base UCF-101, mostrando la clase predicha con mayor puntuación (*score*).

3.4 Resumen

Este capítulo presentó el diseño, implementación y evaluación de una arquitectura CNN-LSTM para el reconocimiento de acciones humanas en video. El enfoque combina redes convolucionales (CNN) para la extracción de características espaciales con redes LSTM, capaces de modelar dependencias temporales en secuencias de imágenes.

Como primer paso, el modelo fue validado sobre las bases de datos KTH y UCF-11, caracterizadas por sus condiciones controladas. En estos entornos, la arquitectura propuesta superó los resultados previamente reportados. En el caso de KTH, la arquitectura CNN-LSTM alcanzó una precisión del 93,86 %, superando a enfoques como el de Baccouche et al. [9] (91,04 %) y Jones et al. [62] (93,20 %), y posicionándose como uno de los mejores resultados dentro de esta base de datos. En UCF-11, el modelo obtuvo un desempeño del 91,94 %, superando significativamente a trabajos anteriores como los de Liu et al. [77,79], Wang et al. [112], Sharma et al. [102] y Cho et al. [28], cuyos resultados se ubicaron entre el 71,2 % y el 88,0 %. Estos valores reafirman la capacidad del enfoque propuesto para capturar de manera robusta patrones espacio-temporales en contextos controlados, incluso en comparación con alternativas más complejas o entrenadas con estrategias adicionales. Además, estos resultados proporcionan una línea base sólida frente a la cual se evalúan las extensiones posteriores propuestas en esta tesis.

En una segunda etapa, se evaluó el desempeño del modelo propuesto en bases de datos considerablemente más desafiantes: HMDB-51 y UCF-101. En este contexto, la arquitectura CNN-LSTM alcanzó una precisión del 40,7 % en HMDB-51 y del 75,8 % en UCF-101. Estos resultados se ubican por debajo de los enfoques del estado del arte recientes, que superan el 90 % en UCF-101 y alcanzan hasta el 83,1 % en HMDB-51 [71]. Sin embargo, es importante destacar que muchas de estas propuestas utilizan técnicas más complejas, tales como redes 3D convolucionales (I3D [25]), mecanismos de atención multimodal, preentrenamiento en grandes corpus (como Kinetics), o arquitecturas especializadas para aprendizaje espacio-temporal

3.4 Resumen

(SlowFast [39]).

La principal fortaleza del enfoque radica en su simplicidad estructural. A pesar de no incorporar mecanismos de atención, estrategias de preentrenamiento ni arquitecturas híbridas avanzadas, la CNN-LSTM logra capturar patrones espacio-temporales de forma eficaz. Esto la posiciona como una alternativa viable, especialmente en contextos donde los recursos computacionales son limitados o se requiere una solución interpretativa y extensible.

En conclusión, a pesar de que la arquitectura CNN-LSTM básica presenta margen de mejora en escenarios complejos, constituye una base sólida y eficiente sobre la cual pueden construirse futuras extensiones. Su desempeño estable y su bajo nivel de complejidad justifican su elección como punto de partida para investigaciones posteriores, tal como se abordará en el próximo capítulo.

Mecanismo de Atención Sobre Redes LSTM

4.1 Enfoque CNN-LSTM + Att

Siguiendo la arquitectura base descrita en la Sección 3.2 incluiremos un mecanismo de atención basada en la propuesta de Sharma et al. [102]. La Figura 4.1 muestra el esquema general de la arquitectura. Para generar un mapa de atención, se comprime el cuboide a una forma vectorial, que luego se combina con el vector de contexto para construir un vector de ponderación. A continuación, se detallan los pasos para construir el vector de atención:

- El cuboide x_t se comprime en una forma vectorial realizando un promedio sobre sus mapas de características, el cual se utiliza como entrada del MLP_4 .
- El vector de contexto h_{t-1} correspondiente al estado oculto de la LSTM en el tiempo anterior, se procesa mediante el MLP_2 , generando una representación intermedia que sintetiza la información temporal acumulada hasta el instante $t - 1$.
- La salida del MLP_3 , construida a partir de la combinación de las representaciones anteriores, se reorganiza como un mapa de atención bidimensional de tamaño $F \times F$. Este mapa indica, para cada píxel

4.1 Enfoque CNN-LSTM + Att

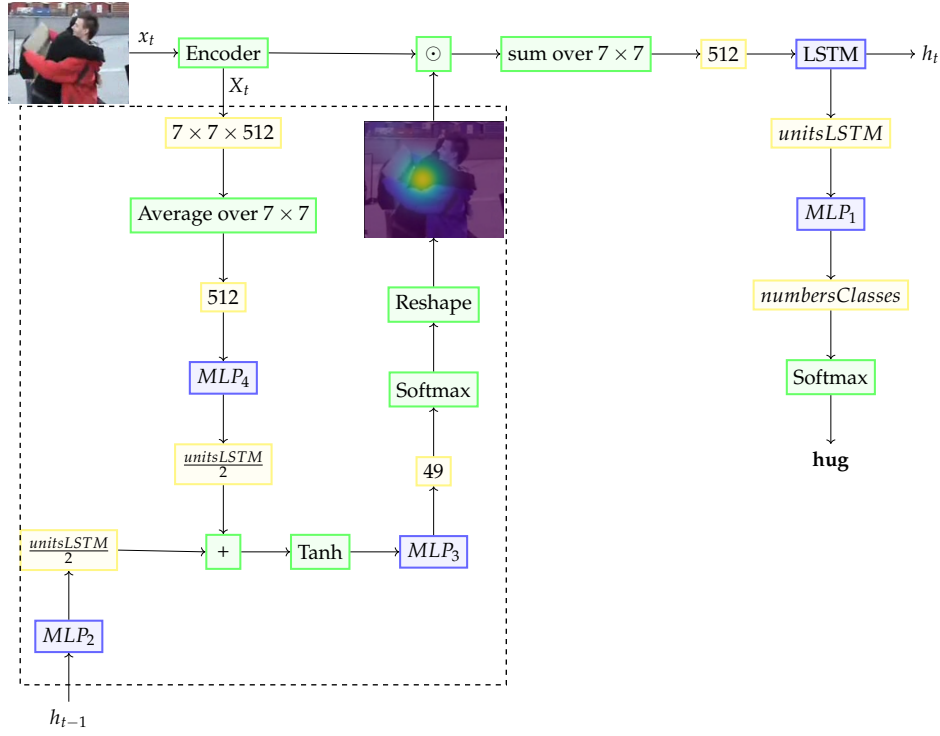


Figura 4.1: Arquitectura con mecanismo de atención.

del mapa de características, la probabilidad de ser relevante. Cuanto mayor es el valor asignado, mayor será la influencia de esa región de la imagen en la decisión final del clasificador.

El Cuadro 4.1 muestra la configuración de los MLPs.

MLP	Capa	Parámetro
MLP ₁	Dropout	0.5
	FC	#classes (neurons)
MLP ₂ , MLP ₃ and MLP ₄	FC	128 (neurons)
	Dropout	0.5
MLP _h and MLP _c	FC	256 (neurons)
	Dropout	0.5

Cuadro 4.1: Configuración de los MLPs

Para la inicialización de h_0 y c_0 Xu et al. [118], comprimen toda la información del video v logrando una convergencia más rápida, esto se calcula como:

$$h_0 = mlp_h\left(\frac{1}{T} \sum_{t=1}^T \left(\frac{1}{F^2} \sum_{i=1}^{F^2} X_{t,i}\right)\right) \quad (4.1)$$

$$c_0 = mlp_c\left(\frac{1}{T} \sum_{t=1}^T \left(\frac{1}{F^2} \sum_{i=1}^{F^2} X_{t,i}\right)\right) \quad (4.2)$$

donde $T = 128$ cantidad de fotogramas de los videos y $F = 7$ dimensión del *feature map* de la VGG16. Todos los fotogramas $x_i \in v$ son procesados a través de la VGG16, produciendo T cuboides. Para comprimir esta información, primero tomamos un promedio sobre el número de cuboides 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 .

4.2 Enfoque CNN-LSTM + Att + FS

Los videos en las bases de datos de esta tesis fueron capturados en entornos no controlados. En muchos casos, estos videos son recortes extraídos de diferentes plataformas como YouTube, Facebook, entre otros. Esta variedad de fuentes presenta ciertos desafíos, como cambios bruscos de escena y la inclusión de fragmentos que no pertenecen a la acción principal. Para abordar estos problemas y mejorar la calidad de la propuesta, complementaremos la arquitectura **CNN-LSTM + Att** descrita en la Sección 4.1 con un módulo de selección automática de fotogramas capaz de seleccionar un subconjunto de fotogramas de un video que sea más representativo de su contenido, maximizando la similitud dentro del subconjunto seleccionado. Para ello, utilizaremos como medida de disimilitud la distancia de Bhattacharyya entre fotogramas consecutivos.

Dada una imagen I , primero convertimos la imagen a escala de grises I_g . Luego, calculamos el histograma normalizado de la imagen en escala de grises. La disimilitud D entre dos fotogramas consecutivos I_1 e I_2 se define como:

$$D(I_1, I_2) = \sqrt{1 - \rho(H_1, H_2)} \quad (4.3)$$

4.2 Enfoque CNN-LSTM + Att + FS

donde H_1 y H_2 son los histogramas normalizados de I_1 e I_2 , y ρ es el coeficiente de Bhattacharyya:

$$\rho(H_1, H_2) = \sum_{i=1}^n \sqrt{H_1(i) \cdot H_2(i)} \quad (4.4)$$

Con el objetivo de seleccionar n fotogramas representativos, se propone un módulo que identifica el rango donde la disimilitud entre fotogramas consecutivos es mínima, lo que indica una alta similitud temporal. Este enfoque permite capturar transiciones sutiles y continuas en la escena, priorizando la coherencia visual. El procedimiento para la selección de fotogramas consta de los siguientes pasos:

1. Calcular la disimilitud entre fotogramas consecutivos: Para cada par de fotogramas consecutivos (I_t, I_{t+1}) en el video, calculamos la disimilitud $D(I_t, I_{t+1})$.
2. Identificar el rango con menor disimilitud: Dividimos el video en ventanas deslizantes de tamaño n y calculamos la suma de las disimilitudes dentro de cada ventana. Seleccionamos la ventana con la suma de disimilitudes más baja, ya que esto indica que los fotogramas dentro de esta ventana son los más similares entre sí.
3. Seleccionar fotogramas: Los n fotogramas dentro de la ventana identificada en el paso anterior son seleccionados como los fotogramas representativos del video.
4. Formalmente: Sea $V = \{I_1, I_2, \dots, I_T\}$ el conjunto de fotogramas del video con T fotogramas, y sea n el número de fotogramas a seleccionar. Definimos la suma de disimilitudes en una ventana de tamaño n que comienza en el fotograma k como:

$$S(k, n) = \sum_{i=k}^{k+n-2} D(I_i, I_{i+1})$$

El objetivo es encontrar el índice k^* que minimice $S(k, n)$:

$$k^* = \arg \min_{k \in \{1, \dots, T-n+1\}} S(k, n)$$

4.2 Enfoque CNN-LSTM + Att + FS

Los fotogramas seleccionados son entonces:

$$\{I_{k^*}, I_{k^*+1}, \dots, I_{k^*+n-1}\}$$

Aplicamos el módulo a varios videos. Se observa que minimizar la disimilitud al seleccionar los fotogramas produce un conjunto altamente representativo del contenido del video. Esto se traduce en una mejor compresión y resumen del video, capturando cambios sutiles y detalles importantes. La Figura 4.2 presenta dos ejemplos del comportamiento de la disimilitud entre fotogramas. Las dos primeras imágenes corresponden a la acción salto (jump), donde se observa un caso con alta disimilitud (arriba) y otro con alta similitud (abajo). Las dos imágenes siguientes muestran la acción golpear (hit), siguiendo el mismo criterio comparativo. Cabe destacar que a mayor valor de la medida, mayor es la diferencia entre los fotogramas evaluados.

4.2 Enfoque CNN-LSTM + Att + FS

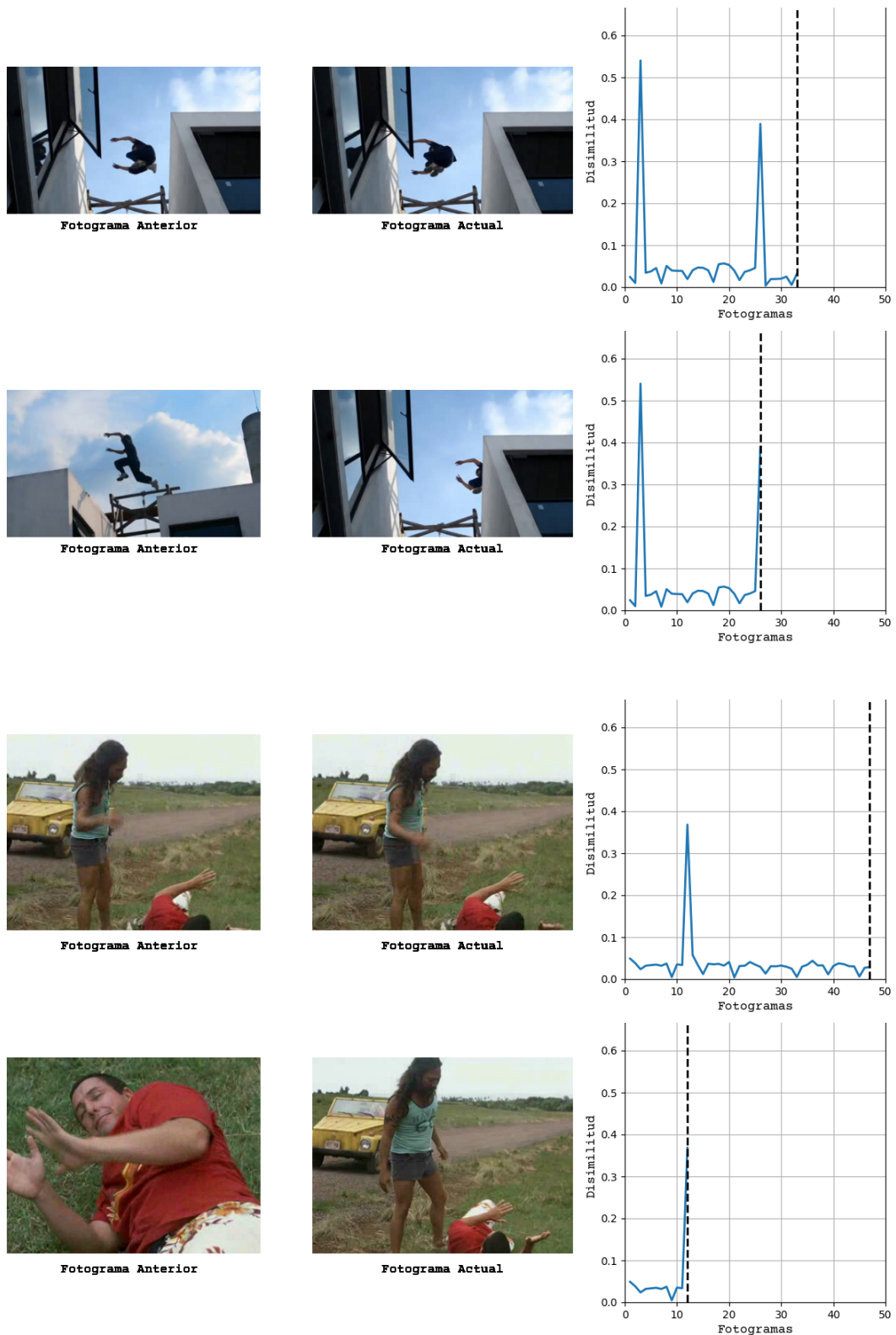


Figura 4.2: Ejemplos de medida de disimilitud mediante la distancia de Bhattacharyya para las acciones *jump* y *hit*. A mayor valor de la métrica, mayor es la diferencia entre los fotogramas.

4.3 Experimentos y Resultados

En el Cuadro 4.2 se presentan los resultados obtenidos por los enfoques **CNN-LSTM + Att** y **CNN-LSTM + Att + FS**. El análisis comparativo aplicado a los conjuntos de datos HMDB-51 y UCF-101 evidencia una mejora en la precisión del modelo al incorporar mecanismos de atención y selección de fotogramas bajo las condiciones experimentales específicas utilizadas en este trabajo.

Inicialmente, el modelo **CNN-LSTM** alcanzó una exactitud del 40,7 % en HMDB-51 y 75,8 % en UCF-101, lo cual sirvió como punto de partida. Con la incorporación del mecanismo de atención, el enfoque **CNN-LSTM + Att** mejoró estos valores hasta 51,2 % y 87,2 % respectivamente, lo que indica que focalizarse en regiones relevantes del espacio de características puede contribuir significativamente al rendimiento, al menos dentro del marco experimental definido.

Posteriormente, la inclusión del módulo de selección de fotogramas en **CNN-LSTM + Att + FS** permitió alcanzar una precisión del 57,3 % en HMDB-51 y 90,4 % en UCF-101. Esta mejora se logró reduciendo la secuencia de entrada de 128 a 64 fotogramas, lo que implicó una reducción aproximada del 50 % en el tiempo de entrenamiento, sin afectar negativamente el desempeño bajo estas condiciones. La elección de 64 fotogramas responde a un compromiso entre eficiencia computacional y conservación de la información temporal, alineándose con prácticas comunes en la literatura.

En comparación con métodos más sofisticados, como los de Feichtenhofer et al. [39] o Simonyan et al. [105], marcados como $EP^{(*)}$ en el Cuadro 4.2, que emplean redes de doble flujo (Two-Stream) y arquitecturas profundas de alto consumo computacional, el enfoque aquí propuesto es competitivo en términos de precisión dentro del contexto experimental descrito. Esto sugiere que arquitecturas relativamente simples, pero estratégicamente reforzadas mediante mecanismos de atención y selección de fotogramas, pueden lograr resultados comparables con una mayor eficiencia y menor costo de entrenamiento, aunque la

4.3 Experimentos y Resultados

superioridad en rendimiento absoluto depende del conjunto de datos y configuración específica.

Las Figuras 4.3 (HMDB-51) y 4.4 (UCF-101) muestran ejemplos de la salida del enfoque **CNN-LSTM + Att**. Cada ejemplo se presenta con la siguiente información:

- Base de Datos y Etiquetas: video perteneciente a la base de datos y su etiquetada de referencia para el video.
- Predicción: respuesta del enfoque CNN-LSTM + Att correspondiente a la clase con puntuación más alta, es decir la clase más probable.
- Superposición del mapa de atención. La región en verde y amarillo indica la zona de enfoca del sistema, y el brillo indica la ponderación.

4.3 Experimentos y Resultados

Enfoque	Año	Tipo	Base de datos	
			HMDB-51	UCF-101
Kuehne et al. [66]	2011	EC	23,0 %	—
Wang et al. [113]	2013	EC	46,6 %	—
Gaidon et al. [41]	2014	EC	41,3 %	—
Simoyan et al. [105]	2014	EP (*)	59,4 %	88,0 %
Sharma et al. [102]	2015	MA	41,3 %	—
Tran et al. [109]	2015	EP	—	90,4 %
Wang et al. [114]	2016	EC	60,1 %	86,0 %
Ye et al. [121]	2016	EP	55,2 %	85,4 %
Feichtenhofer et al. [39]	2016	EP (*)	58,5 %	91,4 %
Qiu et al. [93]	2017	EP	—	88,6 %
Carreira et al. [25]	2017	EP (*)	80,2 %	97,9 %
Li et al. [74]	2018	EP	63,0 %	—
Meng et al. [89]	2018	EP	53,1 %	87,1 %
Zhang et al. [124]	2018	EP	—	86,4 %
Liu et al. [80]	2020	EP	69,1 %	94,3 %
Yang et al. [119]	2020	EP	71,5 %	93,8 %
Li et al. [73]	2020	EP	54,3 %	86,7 %
Majd et al. [85]	2020	EP	61,3 %	92,8 %
Liu et al. [84]	2021	EP	74,4 %	96,6 %
Li et al. [71]	2022	MA	83,1 %	98,8 %
Zhang et al. [126]	2023	EP	79,8 %	91,5 %
Enfoque CNN-LSTM			40,7 %	75,8 %
Enfoque CNN-LSTM + Att			51,2 %	87,2 %
Enfoque CNN-LSTM + Att + FS			57,3 %	90,4 %

Cuadro 4.2: Resultados de la clasificación de videos utilizando la base de datos HMDB-51 [66] y UCF-101 [107]. El cuadro resume los resultados obtenidos por el enfoque **CNN-LSTM**, **CNN-LSTM + Att** y **CNN-LSTM + Att + FS** junto con otros reportados en la literatura.

4.3 Experimentos y Resultados

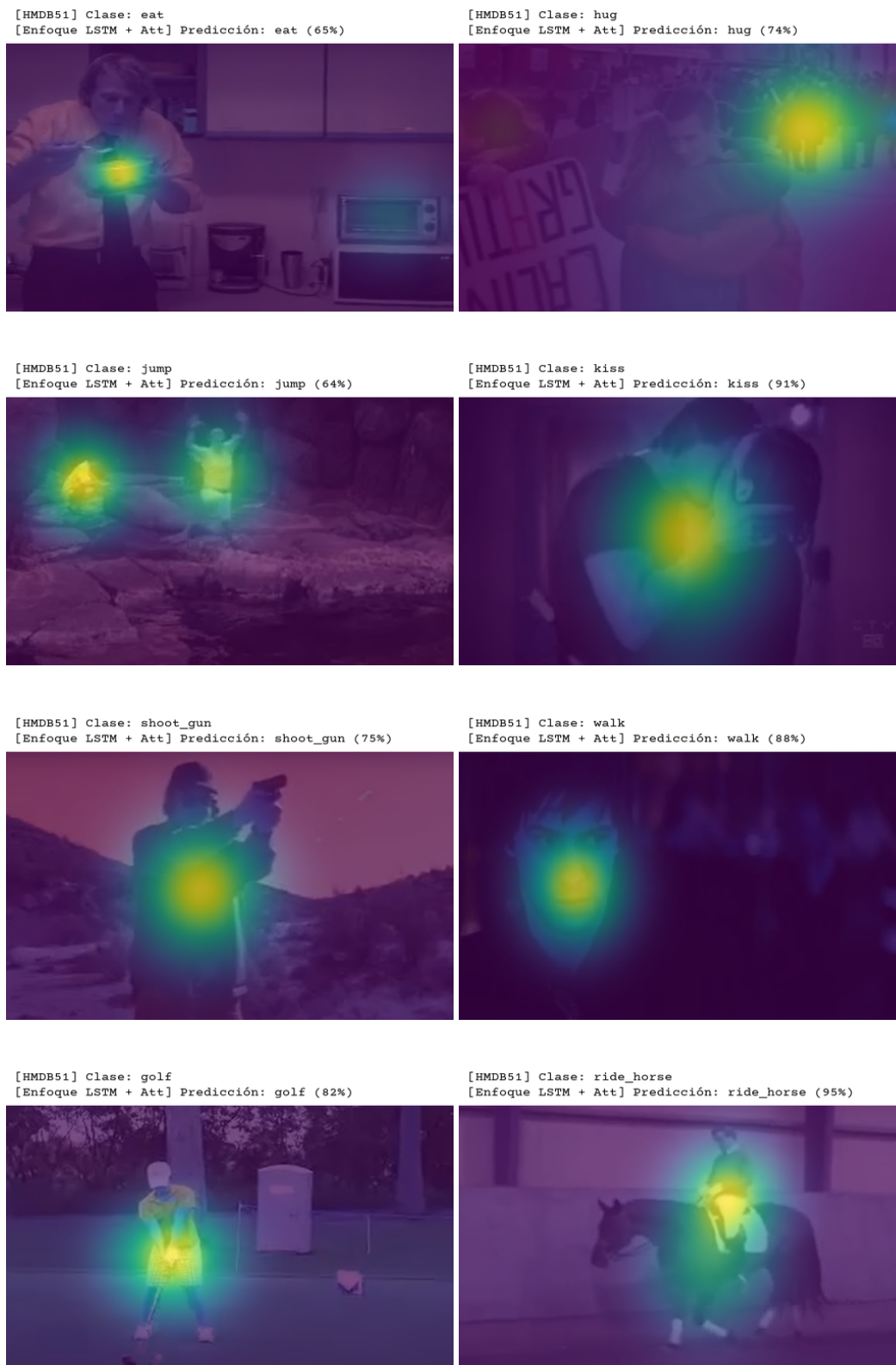


Figura 4.3: Ejemplos de salida de la arquitectura. HMDB-51 junto con la clase de mayor puntuación (score).

4.3 Experimentos y Resultados



Figura 4.4: Ejemplos de salida de la arquitectura. UCF-101 junto con la clase de mayor puntuación (score).

4.4 Resumen

En este capítulo, se desarrolló una arquitectura combinada de CNN-LSTM, complementada con un mecanismo de atención y un módulo de selección de fotogramas, para abordar el problema del reconocimiento de acciones humanas en videos. Los resultados obtenidos evidencian una mejora significativa respecto a la propuesta inicial y muestran un desempeño competitivo en comparación con otros enfoques reportados en la literatura, considerando las condiciones experimentales utilizadas. La arquitectura propuesta alcanzó una precisión del 57,3 % en el dataset HMDB-51 y del 90,4 % en el dataset UCF-101, superando ampliamente los resultados previos de 40,7 % y 75,8 %, respectivamente.

Estos resultados destacan la efectividad del enfoque CNN-LSTM complementado con mecanismos de atención en el procesamiento y reconocimiento de secuencias de video. La capacidad de la CNN para extraer características espaciales, combinada con la habilidad de la LSTM para capturar dependencias temporales, permite un análisis más profundo y completo de las acciones humanas. El mecanismo de atención, por su parte, refuerza la importancia de ciertas regiones del video, mejorando la precisión del modelo al centrar su atención en las áreas más relevantes de la secuencia temporal.

En comparación con ciertos métodos reportados en la literatura, el enfoque propuesto presenta resultados competitivos. En el caso del dataset HMDB-51, se obtuvieron mejoras respecto a técnicas como DT+HOG/HOF/MBH [74], SDT Tree ATEP [85] y Soft-Attention [80]. Para el dataset UCF-101, la arquitectura alcanzó mejores resultados que métodos como DT+HOG/HOF/MBH [84] y HD C3D+LSTM [89]. Estas comparaciones se realizan bajo condiciones experimentales específicas y sugieren que el enfoque es robusto y eficaz en los datasets evaluados.

La principal ventaja de este enfoque radica en la incorporación de un mecanismo de atención que complementa la propuesta inicial basada en CNN-LSTM. Esta adición no solo mejora los resultados en términos de precisión, sino que también permite al modelo focalizarse en las regiones

4.4 Resumen

más informativas de la secuencia de video, optimizando así el proceso de identificación de acciones relevantes. En conjunto, estos factores repercuten positivamente en el desempeño general del sistema y posicionan a la arquitectura como una solución eficiente y adaptable para aplicaciones reales de reconocimiento de acciones humanas en videos.

Conclusiones y Trabajo a Futuro

5.1 Conclusiones

En esta tesis se ha presentado un recorrido sistemático en el diseño y evaluación de arquitecturas para el reconocimiento de acciones humanas en videos, partiendo de modelos base hasta alcanzar variantes más robustas, eficientes y precisas. A lo largo del trabajo, se realizaron aportes metodológicos en tres niveles principales, los cuales se reflejan tanto en los resultados obtenidos como en las distintas publicaciones desarrolladas durante el doctorado.

En primer lugar, se diseñó una arquitectura inicial basada en la combinación CNN-LSTM, que sirvió como base para todos los avances posteriores. Esta configuración demostró ser especialmente eficaz en bases de datos controladas como KTH y UCF-11, superando resultados previos y posicionándose como una alternativa simple pero potente para entornos con condiciones estables. Este primer modelo sentó las bases para una serie de publicaciones iniciales enfocadas en el uso de LSTM para el reconocimiento de acciones, el ajuste de sus parámetros y la robustez del enfoque frente a alternativas más complejas.

En una segunda etapa, se avanzó hacia la incorporación de un mecanismo de atención visual suave, con el objetivo de mejorar la precisión del modelo y su capacidad de focalizarse en regiones relevantes de los fotogramas. Esta mejora tuvo un impacto directo en el rendimiento,

5.1 Conclusiones

especialmente en bases de datos más complejas como HMDB-51 y UCF-101. Los resultados mostraron incrementos significativos en la precisión respecto al modelo base, confirmando la utilidad del mecanismo de atención. Esta línea de trabajo derivó en publicaciones donde se analizó la integración de la atención con LSTM y su efecto en la mejora del reconocimiento.

Posteriormente, se introdujo un módulo adicional de selección automática de fotogramas clave, que redujo a la mitad la cantidad de imágenes procesadas por video sin afectar negativamente el rendimiento. Esta estrategia permitió disminuir notablemente el tiempo de entrenamiento, aportando una mejora clara en términos de eficiencia. La combinación del mecanismo de atención con esta etapa de preselección de fotogramas consolidó el modelo final propuesto, y fue documentada en trabajos centrados en la atención espacial adaptada y la eficiencia computacional del sistema.

Estos desarrollos no solo condujeron a resultados competitivos en comparación con métodos del estado del arte, sino que lo hicieron utilizando una arquitectura considerablemente más sencilla. Si bien modelos como I3D o SlowFast logran mejores valores absolutos, lo hacen a costa de una complejidad estructural mucho mayor, con requerimientos computacionales significativos. En contraste, el modelo propuesto ofrece una solución más liviana y escalable, sin sacrificar precisión en escenarios con alta variabilidad visual y condiciones no controladas. Esto lo convierte en una alternativa viable para aplicaciones reales en contextos donde la disponibilidad de hardware es limitada o donde se requiere interpretabilidad.

Los distintos capítulos de esta tesis reflejan la evolución progresiva de la propuesta: desde un enfoque base robusto y bien justificado teóricamente, hasta su enriquecimiento mediante técnicas actuales como los mecanismos de atención y la reducción de datos. Cada etapa del trabajo fue acompañada por publicaciones científicas que validan y difunden los aportes realizados, tanto a nivel nacional como internacional. Las publicaciones cubren desde estudios exploratorios y pruebas de arquitectura, hasta el diseño de modelos optimizados con atención y selección de fotogramas, consolidando

una línea de investigación sostenida y reconocible en el área.

En resumen, se ha logrado una contribución significativa al área, combinando fundamentos teóricos sólidos, una experimentación rigurosa y un impacto científico demostrado a través de múltiples publicaciones. Este recorrido constituye una base firme para continuar desarrollando soluciones innovadoras y aplicables en un campo en constante evolución.

5.2 Trabajo a Futuro

A partir de los avances desarrollados en esta tesis, se abren múltiples líneas de trabajo futuro que pueden extender y fortalecer la investigación en reconocimiento de acciones humanas en videos:

- **Incorporación de redes bidireccionales (BiLSTM):** Una mejora natural al modelo LSTM consiste en utilizar redes BiLSTM, que capturan dependencias temporales en ambas direcciones (pasado y futuro). Esta característica es particularmente beneficiosa para la comprensión de acciones que dependen de contexto global a lo largo de la secuencia. Diversos estudios han demostrado la efectividad de las BiLSTM en tareas de reconocimiento de acciones y secuencias temporales [46, 125]. Se propone una integración más profunda y sistemática de BiLSTM para evaluar su impacto en la precisión y robustez del modelo.
- **Exploración de arquitecturas convolucionales más profundas:** En lugar de la CNN básica utilizada para la extracción de características, podrían integrarse arquitecturas más avanzadas como ResNet [51], Inception o EfficientNet. Estas redes han demostrado ser más eficientes y precisas en múltiples tareas de visión por computadora, y su incorporación podría potenciar el rendimiento del modelo general sin alterar la estructura LSTM posterior.
- **Evaluación en bases de datos más desafiantes y no controladas:** Se propone ampliar los experimentos hacia conjuntos de datos

5.2 Trabajo a Futuro

más complejos como Hollywood2 [87], UCF-50 [95], ActivityNet, Kinetics [25], que presentan mayor variabilidad en las condiciones de grabación, duración y tipos de acción. Esto permitirá evaluar la capacidad de generalización del modelo en contextos más realistas.

- **Análisis de eficiencia computacional y rendimiento en tiempo real:** Dado que la arquitectura propuesta ha demostrado ser más liviana que otras soluciones del estado del arte, sería interesante explorar su uso en entornos con restricciones computacionales (dispositivos móviles, cámaras inteligentes) o en aplicaciones en tiempo real. Para ello, podrían realizarse mediciones sistemáticas de velocidad de inferencia, consumo de recursos y escalabilidad.
- **Ampliación de los mecanismos de atención:** Actualmente, la atención visual y la selección de fotogramas han mostrado mejoras claras. Se propone profundizar en otros tipos de atención, como atención temporal jerárquica, atención multimodal (combinando información visual con audio o texto), o incluso mecanismos autoatentivos derivados de Transformers [3,73,119].
- **Integración de modelos tipo Transformer:** Las arquitecturas Transformer han revolucionado el procesamiento secuencial, y su aplicación a videos ha comenzado a dar resultados prometedores. Se propone investigar modelos híbridos que combinen la base CNN para extracción espacial con capas Transformer para el modelado temporal, o directamente arquitecturas completas tipo TimeSformer [42] o Video Swin Transformer [126].
- **Explicabilidad del modelo y visualización de la atención:** Otra línea interesante es el desarrollo de técnicas de interpretación del modelo, especialmente enfocadas en la visualización de las regiones de atención o los fotogramas seleccionados. Esto puede facilitar la validación del sistema, mejorar su transparencia y abrir posibilidades de uso en contextos críticos como vigilancia, salud o deporte.
- **Uso de datos sintéticos y aumentación avanzada:** La creación

5.2 Trabajo a Futuro

de conjuntos de datos sintéticos generados por modelos como GANs o simuladores puede ayudar a ampliar la capacidad de entrenamiento en escenarios poco representados. También se propone investigar técnicas de data augmentation específicas para secuencias de video, como el recorte espacial-temporal adaptativo o la mezcla de secuencias.

- **Fusión de múltiples fuentes de información:** Finalmente, se podría extender el modelo incorporando información adicional como pose humana, flujo óptico, profundidad o audio. La fusión multimodal puede permitir capturar detalles más ricos del comportamiento humano y aumentar la robustez frente a ruido o ambigüedad visual.

Bibliografía

- [1] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süssstrunk. Salient region detection and segmentation. In *International conference on computer vision systems*, pages 66–75. Springer, 2008.
- [2] Jake K. Aggarwal and Michael S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):1–43, 2011.
- [3] Saghir Alfasly, Charles K Chui, Qingtang Jiang, Jian Lu, and Chen Xu. An effective video transformer with synchronized spatiotemporal and spatial self-attention for action recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 35(2):2496–2509, 2022.
- [4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.
- [5] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Patrick Luc, Joao Carreira, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.
- [6] Fred Attneave and MB. The organization of behavior; a neuropsychological theory, 1950.

- [7] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.
- [8] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *International Conference on Artificial Neural Networks*, pages 154–159. Springer, 2010.
- [9] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential Deep Learning for Human Action Recognition. In B. Lepri A.A. Salah, editor, *2nd International Workshop on Human Behavior Understanding (HBU)*, pages 29–39, Amsterdam, Netherlands, November 2011. Springer.
- [10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
- [11] Jiao Bao, Mao Ye, and Yumin Dou. Mobile phone-based internet of things human action recognition for e-health. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pages 957–962. IEEE, 2016.
- [12] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.
- [13] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.
- [14] Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5):1–41, 2007.
- [15] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Space-time attention for video transformers. *arXiv preprint arXiv:2102.05095*, 2021.

BIBLIOGRAFÍA

- [16] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [17] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [18] Meirav Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1402, 2005.
- [19] Victoria Bloom, Dimitrios Makris, and Vasileios 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*, pages 7–12. IEEE, 2012.
- [20] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.
- [21] John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.
- [22] David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.
- [23] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2488–2496, 2015.
- [24] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *European Conference on Computer Vision (ECCV)*, pages 213–229, 2020.

- [25] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017.
- [26] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667, 2017.
- [27] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):569–582, 2014.
- [28] Jungchan Cho, Minsik Lee, Hyung Jin Chang, and Songhwai Oh. Robust action recognition using local motion and group sparsity. *Pattern Recognition*, 47(5):1813 – 1825, 2014.
- [29] François Chollet et al. Keras. <https://keras.io>, 2015.
- [30] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.
- [31] Krzysztof Choromanski, Victor Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Jakob Hawkins, James Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.
- [32] 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)*, volume 1, pages 886–893 vol. 1, 2005.
- [33] Yann Dauphin, Harm de Vries, and Yoshua Bengio. Rmsprop and equilibrated adaptive learning rates for non-convex optimization. In *NIPS*, 2015.

- [34] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [35] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005.
- [36] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, 2015.
- [37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [38] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
- [39] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.
- [40] Kunihiro Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.
- [41] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Activity representation with motion hierarchies. *International journal of computer vision*, 107(3):219–238, 2014.

- [42] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Video action transformer network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253, 2019.
- [43] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [44] Lior Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(12):2247–2253, 2007.
- [45] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer, 2012.
- [46] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2017.
- [47] Nitesh Bharadwaj Gundavarapu, Luke Friedman, Raghav Goyal, Chaitra Hegde, Eirikur Agustsson, Sagar M. Waghmare, Mikhail Sirotenko, Ming-Hsuan Yang, Tobias Weyand, Boqing Gong, and Leonid Sigal. Extending video masked autoencoders to 128 frames, 2024.
- [48] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555, 2018.
- [49] Kota Hara, Ming-Yu Liu, Oncel Tuzel, and Amir-massoud Farahmand. Attentional network for visual object detection. *arXiv preprint arXiv:1702.01478*, 2017.
- [50] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, pages 10–5244. Citeseer, 1988.

- [51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [52] Saliya Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and Vision Computing*, 60:4–21, 2017.
- [53] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
- [54] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
- [55] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.
- [56] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [57] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.
- [58] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.
- [59] Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

- [60] Neziha Jaouedi, Nouredine Boujnah, Oumayma Htiwich, and Med Salim Bouhlef. Human action recognition to human behavior analysis. In *2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pages 263–266. IEEE, 2016.
- [61] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.
- [62] Simon Jones, Ling Shao, Jianguo Zhang, and Yan Liu. Relevance feedback for real-world human action retrieval. *Pattern Recognition Letters*, 33(4):446 – 452, 2012. Intelligent Multimedia Interactivity.
- [63] Muhammad Attique Khan, Kashif Javed, Sajid Ali Khan, Tanzila Saba, Usman Habib, Junaid Ali Khan, and Aaqif Afzaal Abbasi. Human action recognition using fusion of multiview and deep features: an application to video surveillance. *Multimedia tools and applications*, pages 1–27, 2020.
- [64] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [66] 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.
- [67] Kevin J Lang, Alex H Waibel, and Geoffrey E Hinton. A time-delay neural network architecture for isolated word recognition. *Neural networks*, 3(1):23–43, 1990.
- [68] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005.

BIBLIOGRAFÍA

- [69] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [70] Jianan Li, Yunchao Wei, Xiaodan Liang, Jian Dong, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Attentive contexts for object detection. *IEEE Transactions on Multimedia*, 19(5):944–954, 2016.
- [71] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552*, 2022.
- [72] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [73] Xiaoqiang Li, Miao Xie, Yin Zhang, Guangtai Ding, and Weiqin Tong. Dual attention convolutional network for action recognition. *IET Image Processing*, 14(6):1059–1065, 2020.
- [74] Zhenyang Li, Kirill Gavriluk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018.
- [75] MC Lin. Qiang; shuicheng yan. network in network. *arXiv preprint arXiv:1312.4400*, 2014.
- [76] Zhe Lin, Zhuolin Jiang, and Larry S Davis. Recognizing actions by shape-motion prototype trees. In *2009 IEEE 12th international conference on computer vision*, pages 444–451. IEEE, 2009.
- [77] D. Liu, M. Shyu, and G. Zhao. Spatial-temporal motion information integration for action detection and recognition in non-static background. In *2013 IEEE 14th International Conference on Information Reuse Integration (IRI)*, pages 626–633, Aug 2013.

BIBLIOGRAFÍA

- [78] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing*, 26(7):3492–3506, 2017.
- [79] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos in the wild. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pages 1996 – 2003, 07 2009.
- [80] Shaocan Liu, Xin Ma, Hanbo Wu, and Yibin Li. An end to end framework with adaptive spatio-temporal attention module for human action recognition. *IEEE Access*, 8:47220–47231, 2020.
- [81] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, 2010.
- [82] Xinyi Liu, Mariofanna Milanova, and Mini Review. Visual attention in deep learning: a review volume 4 issue 3-2018 visual attention in deep learning: a review 155 citation: Liu x, milanova m. visual attention in deep learning: a review. int rob auto. *International Journal of Robotics and Automation*, 4, 05 2018.
- [83] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2022.
- [84] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer, 2021.
- [85] Mahshid Majd and Reza Safabakhsh. Correlational convolutional lstm for human action recognition. *Neurocomputing*, 396:224–229, 2020.

- [86] Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203, 2016.
- [87] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936, June 2009.
- [88] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [89] Lili Meng, Bo Zhao, Bo Chang, Gao Huang, Frederick Tung, and Leonid Sigal. Where and when to look? spatio-temporal attention for action recognition in videos. *CoRR*, abs/1810.04511, 2018.
- [90] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.
- [91] Michael C Mozer. Induction of multiscale temporal structure. In *Advances in neural information processing systems*, pages 275–282. Citeseer, 1992.
- [92] Janez Perš, Vildana Sulić, Matej Kristan, Matej Perše, Klemen Polanec, and Stanislav Kovačič. Histograms of optical flow for efficient representation of body motion. *Pattern Recognition Letters*, 31(11):1369–1376, 2010.
- [93] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.
- [94] Alireza Rahimpour, Liu Liu, Ali Taalimi, Yang Song, and Hairong Qi. Person re-identification using visual attention. *CoRR*, abs/1707.07336, 2017.

BIBLIOGRAFÍA

- [95] Kishore K. Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Mach. Vision Appl.*, 24(5):971–981, July 2013.
- [96] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.
- [97] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [98] David E Rumelhart, Geoffrey E Hinton, James L McClelland, et al. A general framework for parallel distributed processing. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(45-76):26, 1986.
- [99] Jürgen Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.
- [100] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36 Vol.3, Aug 2004.
- [101] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. *IEEE transactions on pattern analysis and machine intelligence*, 29(3):411–426, 2007.
- [102] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *CoRR*, abs/1511.04119, 2015.
- [103] Jianlou Si, Honggang Zhang, Chun-Guang Li, Jason Kuen, Xiangfei Kong, Alex C Kot, and Gang Wang. Dual attention matching network for context-aware feature sequence based person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5363–5372, 2018.

- [104] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [105] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
- [106] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. Spatio-temporal attention-based lstm networks for 3d action recognition and detection. *IEEE Transactions on Image Processing*, 27(7):3459–3471, 2018.
- [107] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.
- [108] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [109] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [110] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018.
- [111] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE*

BIBLIOGRAFÍA

- conference on computer vision and pattern recognition*, pages 3156–3164, 2017.
- [112] H. Wang, A. Klaser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR 2011*, pages 3169–3176, June 2011.
- [113] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.
- [114] Heng Wang, Dan Oneata, Jakob Verbeek, and Cordelia Schmid. A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, 119(3):219–238, 2016.
- [115] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.
- [116] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In A. Cavallaro, S. Prince, and D. Alexander, editors, *BMVC 2009 - British Machine Vision Conference*, pages 124.1–124.11, London, United Kingdom, September 2009. BMVA Press.
- [117] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [118] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.
- [119] Hao Yang, Chunfeng Yuan, Li Zhang, Yunda Sun, Weiming Hu, and Stephen J Maybank. Sta-cnn: convolutional spatial-temporal attention learning for action recognition. *IEEE Transactions on Image Processing*, 29:5783–5793, 2020.

- [120] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.
- [121] 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)*, pages 1110–1118, 2016.
- [122] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.
- [123] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.
- [124] 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*, 27(5):2326–2339, 2018.
- [125] Jun Zhang, Quan Liu, and Yonggang Wu. Bidirectional lstm networks for human action recognition. In *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5. IEEE, 2018.
- [126] Wei Zhang, Ming Li, Junjie Chen, and Yafei Wang. Multimodal transformer for video retrieval. *IEEE Transactions on Multimedia*, 25:1234–1247, 2023.
- [127] L. Zhao, J. Han, and S. Xie. A local binary pattern based method for human action recognition in video sequences. *Procedia Engineering*, 29:1759–1764, 2012.

BIBLIOGRAFÍA

- [128] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.