UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

MÁSTER UNIVERSITARIO EN INTELIGENCIA ARTIFICIAL,
RECONOCIMIENTO DE FORMAS E IMAGEN DIGITAL


# Clasificación de vídeos mediante Redes Neuronales Artificiales

Trabajo Final de Máster

Autor:
Javier Jorge Cano

Dirigido por:
Dr. Roberto Paredes Palacios

13 de julio de 2015

*A mi familia y a mi pequeña alegría, que me siguen acompañando en el camino, por muy duro que sea.*

**Abstract**

Nowadays, the research on computer vision and machine learning is in its best moment. The computational capacity and communications currently available in any device, have risen new challenges. Among them, the task of human or object recognition on images and video are impulsed by the best universities and technological companies. Concretely, human activity recognition in videos has a direct application in many environments: security systems, interaction analysis, illness identification, etc.

For this reason, this project proposes a prospective study about the task of THUMOS competition on computer vision. In this task, it is required to classify videos by activity, among a set of 101 activities, belonging to 5 different kinds: Human-Human interaction, Human-Object interaction, sports, body-motion, and playing musical instruments.

This thesis proposes, applied to this task for the first time, a model based on artificial neural networks that uses improved Dense Trajectories as a feature extraction technique. This thesis will analize the current state-of-the-art, and it will perform experiments in order to obtain the best model for this task, and afterwards, these experiments will be compared with the results provided by the approaches on the top ten of the THUMOS classification.

*Keywords*: computer vision, machine learning, artificial neural networks, human activity recognition, improved dense trajectories

**Resumen**

Actualmente, la investigación en el campo de la visión por computador y el aprendizaje automático se encuentra en su mejor momento. La capacidad de cómputo y de comunicación disponible hoy en dia en cualquier dispositivo ha despertado nuevos desafios. Entre ellos, las tareas de reconocimiento de personas o elementos dentro de imágenes o vídeos, se encuentran impulsadas por las mejores universidades y empresas tecnológicas. Concretamente, el reconocimiento de la actividad llevada a cabo por personas dentro de los vídeos, comprende una tarea que tiene aplicabilidad directa en numerosos entornos: sistemas de seguridad, análisis de la interacción, identificación de enfermedades, etc.

Por ello, en este proyecto se propone un estudio prospectivo sobre la tarea planteada en la competición de visión por computador THUMOS. En esta tarea, se requiere la clasificación de vídeos por actividad, de entre un conjunto de 101 actividades, pertenecientes a 5 diferentes grupos: interacción humano-humano, interacción humano-objeto, deportes, movimientos corporales y personas tocando diversos instrumentos.

En este trabajo se plantea un modelo basado en redes neuronales artificiales, que se aplica por primera vez a esta tarea, utilizando la técnica del estado del arte *improved Dense Trajectories* para la extracción de características. Se analizará, además, el estado de la cuestión hasta el momento, y se llevará a cabo la experimentación con el objetivo de obtener el mejor modelo, para posteriormente comparar los resultados con los obtenidos en las aproximaciones que conforman el *top-ten* de la clasificación.

*Palabras clave*: visión por computador, aprendizaje automático, redes neuronales artificiales, reconocimiento de la actividad humana, improved dense trajectories

# Índice general

# Índice de figuras

# Índice de tablas

# Capítulo 1
# Introducción

## 1.1. Motivación

Actualmente, se producen y consumen multitud de contenidos multimedia *online*. De todos estos medios, los vídeos, de cualquier tipo y temática, son los que más interés despiertan. Desde vídeos educativos hasta virales que reciben millones de visitas, las plataformas de vídeo *online* almacenan enormes cantidades de estos contenidos. La plataforma de referencia en el sector del vídeo *online*, *Youtube*, cuenta con más de mil millones de usuarios, los cuales suben más de 300 horas de vídeo por minuto, generando un continuo incremento en los beneficios de la empresa año tras año[1]. Las oportunidades de negocio son numerosas: publicidad, marketing, televisión a la carta o *e-learning*, entre otras.

La omnipresencia de estos contenidos en *Internet* ha potenciado nuevas oportunidades de investigación. Campos como la inteligencia artificial, la visión por computador o el aprendizaje automático han intensificado sus esfuerzos, reflejando la necesidad del desarrollo de nuevos algoritmos y procedimientos. Estos planteamientos deben enfrentarse a volúmenes de datos elevados y contenidos no restringidos y diversos, ya sean los presentes en las plataformas de contenido *online* o los percibidos por un sistema de visión por computador. La detección de objetos o el reconocimiento facial son algunos ejemplos de tareas habituales en este sector.

Concretamente, el reconocimiento de las acciones realizadas por personas, ya sea interaccionando con otras o con objetos, ha tomado gran interés en los últimos años. Las posibilidades para este tipo de técnica cubren un

[1] https://www.youtube.com/yt/press/es/statistics.html - Consulta: 10 de julio de 2015

amplio abanico de campos, desde sistemas de vigilancia hasta el análisis del usuario durante la interacción hombre-máquina. Llevar a cabo esta tarea, de forma eficiente y cercana a un entorno lo más realista posible, es uno de los retos candentes de la investigación en visión por computador y aprendizaje automático.

Con el objetivo de estudiar esta problemática, en este trabajo final de Máster se plantea una aproximación para la tarea de identificación de la actividad desarrollada por humanos dentro de un vídeo, interaccionando con otros humanos u objetos. El presente trabajo se estructura como sigue: En primer lugar se plantearán los objetivos a alcanzar y el estado de la cuestión hasta el momento. A continuación, en el segundo capítulo, se plantearán las técnicas y los modelos a utilizar para dar respuesta a la tarea propuesta, para posteriormente llevar a cabo la experimentación en el capítulo tercero. En el apartado de la experimentación se analizará el proceso proseguido para la obtención de los parámetros del modelo que mejor resultado han ofrecido, comparando los resultados obtenidos con las conclusiones planteadas en la literatura. Finalmente, el cuarto y último capítulo del trabajo finalizará con las conclusiones alcanzadas y los planteamientos para futuros trabajos.

## 1.2. Objetivos

Se plantea como objetivo principal de este proyecto, estudiar empíricamente la aplicación de redes neuronales artificiales para la tarea de clasificación de vídeos según la actividad que las personas desarrollan en él. Este objetivo principal se concreta en los siguientes objetivos secundarios:

- Analizar el conjunto de datos UCF-101 sobre el que se evaluará la técnica.
- Llevar a cabo la extracción de características locales de los vídeos.
- Entrenar un sistema de aprendizaje automático basado en Redes Neuronales Artificiales para la clasificación de las características locales y utilizar un esquema de fusión para la clasificación final.
- Evaluar la precisión del sistema.
- Realizar una comparativa con los resultados obtenidos para el mismo conjunto de datos por otros trabajos.

## 1.3. Estado del arte

El objetivo del reconocimiento de la actividad humana es analizar automáticamente la actividad que se está desarrollando dentro de un vídeo por una o varias personas. Esta tarea cubre desde el análisis de una acción sencilla ejecutada por una persona, i.e: levantar el brazo, hasta la detección de diversas acciones que cambian en el tiempo y que son realizadas por varias personas, i.e: un equipo de fútbol que pasa de defender a atacar.

Siguiendo la revisión planteada en [1] sobre la detección de actividades llevadas a cabo por humanos, podemos distinguir diferentes tipos de acciones y metodologías. Dada la taxonomía planteada en esta revisión, en relación a los tipos de acciones, se contemplan cuatro tipos diferentes:

- Gestos: Los gestos son movimientos atómicos llevados a cabo por el cuerpo humano, por ejemplo "levantar la mano" o "flexionar la pierna".
- Acciones: Las acciones son movimientos llevados a cabo por una sola persona, compuestos de gestos, como "caminar" o "nadar".
- Interacciones: Contemplan las actividades humanas que involucran a dos o más personas y/o objetos, como "amigos charlando" o "personas compartiendo una bebida".
- Actividades de grupos: Se compone de uno o varios conjuntos de personas y/o objetos llevando a cabo una actividad, por ejemplo "dos equipos jugando a fútbol" o "una banda de música tocando".

Por otro lado, en este trabajo se distingue entre diferentes metodologías. En primer lugar, aproximaciones de una sola capa, es decir, utilizando solamente la secuencia de imágenes que componen el vídeo para la detección y el reconocimiento, y en segundo lugar, una aproximación jerárquica, donde se representan las acciones de alto nivel con la composición de acciones más sencillas, i.e: correr-lanzar pelota-batear conformarían la acción de alto nivel *baseball*. Este último conjunto de técnicas están orientadas a la detección de estas acciones, que combinan otras más sencillas, siendo necesarias técnicas de predicción estructurada para detectar la secuencialidad de los eventos. Se pueden encontrar algunos ejemplos de estas técnicas en [33] y [17], donde se emplean modelos probabilisticos como los Modelos Ocultos de *Markov* (*Hidden Markov Models* - HMM) o las gramáticas incontextuales.

En cuanto a las aproximaciones basadas en una capa, se plantean dos subtipos: las secuenciales y las espacio-temporales. De nuevo, las secuenciales

interpretan el vídeo como una secuencia de observaciones, i.e: una secuencia ordenada de vectores. En estos ámbitos se emplean técnicas que den cuenta de esta secuencialidad, como los HMM [52] o el Alineamiento Temporal Dinámico (*Dynamic Time Warping* - DTW) [10]. Este tipo de técnicas consideran la concatenación de acciones para posibilitar la detección de actividades complejas y no periódicas, como la detección de los protocolos de interacción en una reunión, i.e: presentación, exposición, etc, quedando fuera de los objetivos de la tarea de detección de la actividad humana de este trabajo.

En la aproximación espacio-temporal se engloban las técnicas que guardan una relación más estrecha con el objetivo del proyecto, y las que más éxito están obteniendo en los últimos años. Dentro de esta categoría se distinguen tres grupos: las técnicas que trabajan directamente con los volúmenes del vídeo, es decir, cuboides, las que utilizan las trayectorias y las que llevan a cabo la extracción de características locales de las trayectorias y/o los volúmenes. Se muestran en la Figura 1.1 ejemplos de las técnicas basadas en la extracción de las trayectorias y los volúmenes.

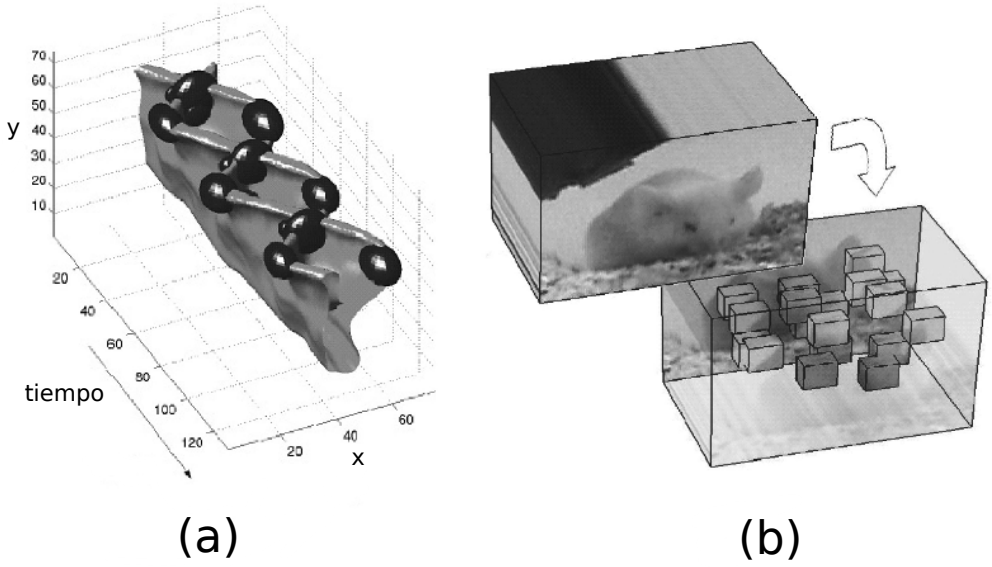


**Figura 1.1:** Ejemplos de técnicas espacio-temporales: (a) Basada en la trayectoria y (b) Basada en volúmenes del vídeo [1].

El trabajo con los volúmenes del vídeo o la trayectoria directa y exclusivamente no han demostrado obtener mejores resultados para el reconocimiento de la acción que las características locales, más eficientes y sencillas, ya que

tratan de extraer la información relevante de estos fenomenos visuales, tanto de forma combinada como individualmente. Este último tipo de característica conforma el estado de arte en la detección de la acción. Se puede consultar [1] para obtener más información de todas estas técnicas.

Para concluir este punto de la revisión del estado del arte, la última aproximación dentro del grupo de una sola capa y espacio-temporales son las características locales espacio-temporales. En este caso se extrae, de la secuencia de imágenes, volúmenes tridimensionales o trayectorias, de los cuales se extraerán puntos de interés, características locales o descriptores que representen el contenido y la actividad a la que pertenecen. Las principales ventajas de esta técnica son su robustez ante el ruido y las variaciones presentes en los entornos realistas, i.e.: cambios de luz, entornos dinámicos, etc. Sin embargo, no presentan un buen funcionamiento para reconocer tareas complejas compuestas por diferentes eventos, ya que se pierde la secuencialidad. Este tipo de técnicas se están usando ampliamente para la tarea que comprende este trabajo, y por ello serán analizadas a continuación con mayor profundidad.

### 1.3.1. Características locales espacio-temporales

Las características locales espacio-temporales o *local space-time features*, han demostrado ser las más adecuadas para la tarea del reconocimiento de la acción, siendo las más habituales entre los participantes de la competición "THUMOS *large scale action recognition challenge*" [21, 22, 14]. Organizado por el grupo *Center for Research in Computer Vision* (CRCV)[2], este evento es uno de los más influyentes dentro del campo de la visión por computador, por su orientación a las tareas de detección realistas y no restringidas en contenido, contemplando un gran número de actividades a reconocer.

Muchas de las técnicas relacionadas con la extracción de características locales provienen de los trabajos previos en visión por computador como la detección de bordes o esquinas sobre imágenes. Estas técnicas resultan útiles para la obtención de puntos de interés, es decir, zonas de la imagen con información de cierta relevancia. Algunos ejemplos de la aplicación de estas técnicas podemos encontrarlos en [25], donde se extiende la técnica de detección de esquinas mediante detectores de Harris [15] para la detección de puntos de interés a lo largo del vídeo, o en [5], donde se utilizan filtros de Gabor [13] que dan diferentes respuestas en el dominio temporal, para

[2]http://crcv.ucf.edu/ - Consulta: 10 de julio de 2015

obtener información de la distribución global de los puntos de interés. Otros planteamientos se han adaptado con el objetivo de extender descriptores que han mostrado un buen funcionamiento con imágenes, como por ejemplo los descriptores SIFT [30], i.e: 3D-SIFT [41]. Otro tipo de características han sido desarrolladas directamtente para la detección de acciones en vídeos, como los descriptores STIP[25], basados en la detección de esquinas espacio-temporales, generando un volumen de descriptores basado en los puntos de interés detectados.

Alternativamente, otro conjunto de técnicas ha enfocado el problema evitando tratar los puntos de interés como puntos tridimensionales, separando el espacio y el tiempo, con el objetivo de darle un tratamiento diferente a cada dimensión. En [44] utilizan de nuevo los descriptores SIFT, concatenándolos a través de las imágenes, para trazar trayectorias y extraer información sobre ellas. En otra aproximación, se ha empleado el muestreo denso (*dense sampling*) en posiciones regulares de la imagen a diferentes escalas, en lugar de detectar puntos de interés. Un estudio comparativo sobre las mencionadas técnicas puede encontrarse en [49].

En el estudio planteado en [49] se observa como la combinación de los histogramas de gradientes orientados (*Histograms of Oriented Gradients* - HOG) [9], junto con los histogramas del flujo óptico (*Histograms of Optical Flow* - HOF) [6], extraídos mediante *dense sampling*, ofrece buenos resultados para la tarea del reconocimiento de la acción en entornos no controlados. Mientras HOG captura la información estática y modela la apariencia, las características HOF miden el movimiento aparente entre dos imágenes, obteniendo información local sobre los elementos dinámicos. La Figura 1.2 ilustra el resultado de la computación de estos descriptores.

Combinando *dense sampling*, además de las características HOG y HOF, se plantea en [48] la técnica *Dense trajectories*. En este trabajo, sobre la imagen muestreada mediante *dense sampling*, se extraen las trayectorias de los píxeles mediante la computación del flujo óptico denso [11], en lugar de llevar a cabo la extracción de los puntos de interés con, por ejemplo, SIFT. Los puntos obtenidos a lo largo de $L$ imágenes se concatenan formando trayectorias, descartando las que presenten irregularidades como grandes desplazamientos. En la Figura 1.3 se pueden observar los puntos y las trayectorias obtenidas mediante SIFT y *Dense Trajectories*. Además, se incluye otro tipo de descriptor que trata de eliminar el ruido de fondo introducido por el flujo óptico, denominado histograma de los bordes del movimiento (*Motion Boundary Histograms* o MBH).

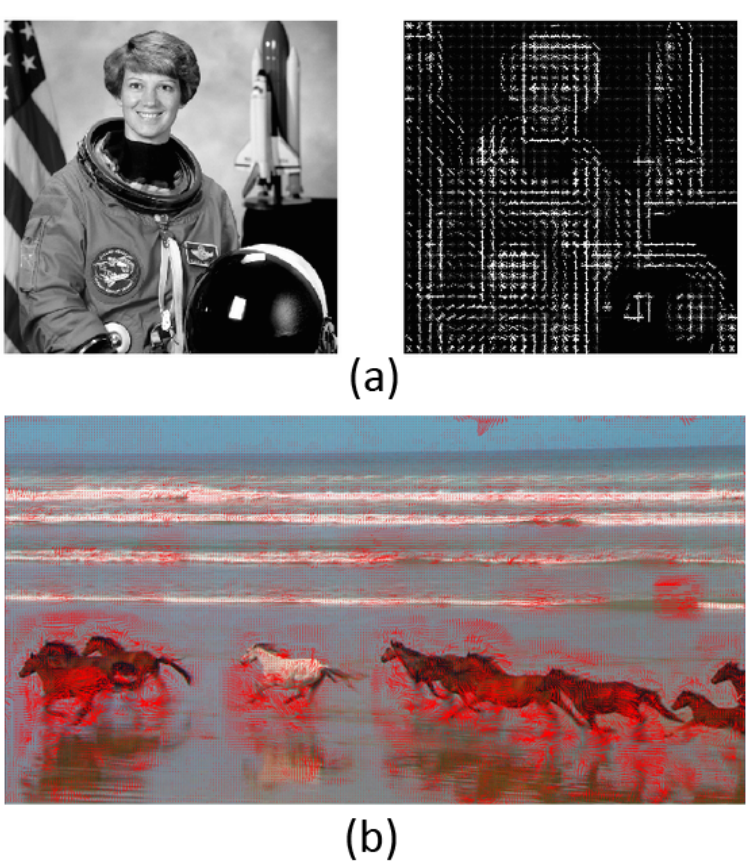


**Figura 1.2:** Ejemplos de HOG y HOF: (a) Obtención de los gradientes, que proporcionan información sobre la forma del contenido y (b) Obtención del flujo óptico que resalta el movimiento que transcurre en la secuencia.

Posteriormente, se planteó una extensión de estos descriptores en [47] acuñada como *improved Dense Trajectories* (iDT), convirtiéndose en los descriptores que conforman el estado del arte, superando cualquier otro planteamiento como puede verse en los resultados publicados recientemente en la edición del 2015 de la competición THUMOS [14]. En este trabajo, se extienden los descriptores *Dense trajectories* mediante la estimación del movimiento de la cámara, para poder eliminar el movimiento global del fondo, utilizando el algoritmo RANSAC planteado en [12], pudiéndose combinar con un detector de personas [36], que evita que se introduzca ruido ya sea por el fondo o por la detección de elementos que no sean personas.


**Figura 1.3:** Visualización de SIFT y *Dense trajectories*, los puntos rojos son las posiciones actuales de la trayectoria. Se puede observar como *Dense Trajectories* captura de forma más precisa que SIFT los patrones de movimiento [48].

Con esto se completa la revisión de las características locales espacio-temporales. A continuación, se analizarán diferentes técnicas de codificación de estas características con el objetivo de aplicar técnicas de reconocimiento de formas y aprendizaje automático.

### Codificación

Entre las técnicas actuales de codificación de las características, se puede encontrar la representación de *bag-of-Visual-Words*(BoVW). Inspirado en la técnica *bag-of-words* desarrollada para los documentos y utilizada en primera instancia, en este ámbito, para la detección de texturas [7], esta técnica se ha utilizado y se sigue utilizando actualmente, por su sencillez y efectividad.

En esta aproximación, los descriptores se agrupan mediante *clustering*, proporcionando los centroides que darán lugar a un vocabulario visual de una longitud fija. Con este vocabulario visual, tras la extracción de características, un vídeo queda representado por el histograma de frecuencias sobre el vocabulario, es decir, contando cuantas características son las más cercanas para cada término. La Figura 1.4 ilustra el proceso completo, desde el entrenamiento a la clasificación.

La siguiente propuesta surgió para evitar los problemas de los que adolece la representación BoVW, como qué tipo de histograma utilizar o cómo llevar a cabo la cuantificación, además de la escalabilidad. Basándose en el principio de Fisher *Kernel* propuesto en [19], donde se combinan los beneficios de

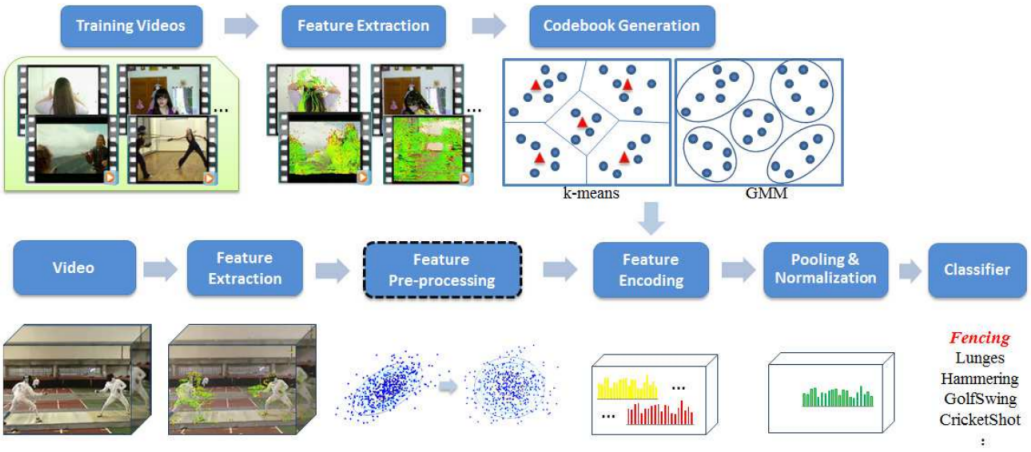


**Figura 1.4:** *Pipeline* de la generación y utilización de BoVW, desde el entrenamiento a la clasificación. [35].

los modelos generativos y discriminativos, se propone trasladar esta técnica al terreno de las imágenes mediante los denominados *Fisher Vectors*(FV) [39].

En líneas generales, esta técnica consiste en caracterizar una muestra por su desviación con respecto a un modelo generativo. Esta desviación se mide mediante el gradiente de la log-verosimilitud de la muestra con respecto a los parámetros del modelo. Instanciado para el reconocimiento de imágenes, se interpreta como muestras las características locales extraídas y, como modelo generativo, un modelo de mixtura de Gaussianas (*Gaussian Mixture Model* - GMM).

Con esta técnica, el modelo se hace independiente del número de características y toma en consideración mayor información que el modelo BoVW, como por ejemplo estadísticos como la desviación del vocabulario visual y su covarianza. Esta técnica, junto con los descriptores *iDT* ha desbancado a la representación mediante BoVW, obteniendo los mejores resultados en las tareas de clasificación de las acciones, convirtiéndose en los descriptores más utilizados. La precursora de esta técnica, los descriptores VLAD [20], también siguen utilizándose solos o combinados con otros. Estas características se encuentran a medio camino entre BOVW y FV, obteniendo información de la suma de desviaciones con respecto a los centroides de la representación BOWV.

### 1.3.2. Clasificación

Los descriptores mencionados en el apartado anterior proporcionan una representación que puede ser aprendida mediante cualquier técnica de clasificación. En las ediciones de THUMOS se ha utilizado mayoritariamente *Support Vector Machines* (SVM) [46], exclusivamente o combinado con otro tipo de técnicas.

Una de las técnicas de clasificación que más éxito tiene dentro del campo de las imágenes y del reconocimiento de vídeo, y que han marcado tendencia, son las redes neuronales convolucionales (*Convolutional Neural Networks* - CNN) [27]. Este tipo de arquitecturas presentan ideas interesantes para la invarianza a los cambios y las distorsiones, frecuentes en entornos realistas como los vídeos que comprenden las tareas de THUMOS, como por ejemplo el submuestreo temporal y espacial o convoluciones sobre la imagen que actúan de campos receptivos locales.

Tras establecer los mejores resultados en el reconocimiento de imágenes en [23], se han propuesto ampliaciones de estos modelos para la tarea de reconocimiento de acciones, basadas tanto en técnicas 2D sobre las imágenes de los vídeos [42] como en técnicas 3D considerando cuboides de tres dimensiones [45]. Esta última técnica ha conseguido igualar los resultados obtenidos con las técnicas vistas previamente y superarlos si se combina con otras, por ejemplo, las características iDT, como presentan en [45].

Sin embargo, este tipo de técnicas requiere un gran despliegue de recursos computacionales: *clusters* de CPU, GPU de última generación y una programación orientada al paralelismo sobre estos recursos *hardware*. Por estos requisitos, el desarrollo de estos sistemas no está al alcance de todos los investigadores.

En otros trabajos recientes, también se han utilizado otras fuentes de información con el objetivo de enriquecer el modelo, utilizando varias técnicas de las mostradas y otras que no están relacionadas con la visión por computador. En [50], ganador de la recientemente finalizada THUMOS 2015 [14], se utilizaron tanto descriptores iDT, como técnicas de redes convolucionales, incluyendo además otras características relacionadas con el audio, utilizando reconocimiento automático del habla y extrayendo características como los coeficientes cepstrales de Mel (*Mel-frequency cepstral coefficients* - MFCCs), utilizados ampliamente para tareas relacionadas con el reconocimiento del habla.

Con este último apartado, concluye el resumen de las principales líneas de investigación o desarrollos en el campo del reconocimiento de la actividad. Se trata de una línea de trabajo viva en la que se realizan avances continuos, que tienen aplicación directa en las herramientas comerciales. Se presentará en el siguiente punto la aproximación propuesta en este trabajo.

# Capítulo 2
# Metodología

Se planteará un modelo para la tarea del reconocimiento de vídeo basada en la extracción de características con técnicas que son utilizadas actualmente por las aproximaciones que mejores resultados han proporcionado, así como una nueva técnica para clasificar los vídeos mediante estas características de bajo nivel. A continuación se detalla la estructura y propiedades del modelo propuesto.

## 2.1. Modelo de aprendizaje

Se utilizará el esquema clásico de reconocimiento de formas, dividido en diferentes etapas. La Figura 2.1 muestra el esquema genérico de las aplicaciones de este tipo. En la primera etapa se lleva a cabo el proproceso del vídeo de entrada, para convertirlo a otro tipo de representacion. En la etapa de extracción de características, se computa el vector de características a partir de los datos preprocesados que contendrá la información considerada relevante. Por último, en la etapa de clasificación, en primer lugar, se aprenderan los modelos que den solución al problema utilizando los vectores de características extraídos, y en segundo lugar, se podrá llevar a cabo la clasificación con los modelos aprendidos en el entrenamiento para vectores no vistos previamente por el modelo. Se planteará a continuación cómo se han adaptado cada una de las fases para llevar a cabo la tarea propuesta.

### 2.1.1. Extracción de características y preprocesado

La técnica escogida para extraer las características de los vídeos ha sido la técnica planteada en [47] denominada *improved Dense Trajectories*. Se ha

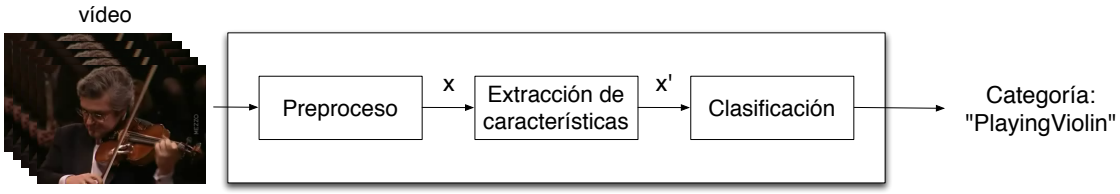


**Figura 2.1:** Esquema del sistema de reconocimiento de formas.

utilizado esta técnica por su ubicuidad en las competiciones y tareas relacionadas con el reconocimiento de acciones, siendo las características "*hand-crafted*" o diseñadas que mejor resultado han dado hasta la fecha.

Planteadas para proporcionar más y mejores trayectorias que las ofrecidas por SIFT [30], mediante *dense sampling* y algorítmos para calcular el flujo óptico de forma eficiente, estas características extraen información local a la trayectoria calculada, siguiendo el desplazamiento del elemento en movimiento dentro del vídeo. Esto proporciona un seguimiento natural, en lugar del rígido planteamiento de los cuboides tridimensionales.

Como los vídeos ya han sido captados por una cámara, y la parte del preprocesado se lleva a cabo a lo largo de esta técnica, las etapas de preprocesado y extracción de los descriptores convergen en una sola. Esta técnica comprende las siguientes fases:

### *Dense trajectories*

Mediante el muestreo con una rejilla regular, a través del algorítmo de flujo óptico planteado en [11] sobre las imágenes del vídeo, denominados habitualmente *frames*, y mediante el seguimiento de la evolución del flujo óptico, se obtiene una trayectoria. Para evitar los cambios bruscos, la trayectoria se limita a una longitud de $L$ puntos. Por otro lado, se detectan las zonas homogéneas, como paredes o suelos, para no realizar el seguimiento de la trayectoria, evitando obtener información de elementos estáticos en la vídeo, centrándose en la información dinámica. Es, en estas zonas, donde con mayor probabilidad tiene lugar la acción. La trayectoria obtenida, además de utilizarse para el resto de pasos, también conforma la primera característica, codificándose mediante un vector normalizado de desplazamientos. La Figura 2.2 ilustra el proceso de detección de las trayectorias sobre un vídeo.

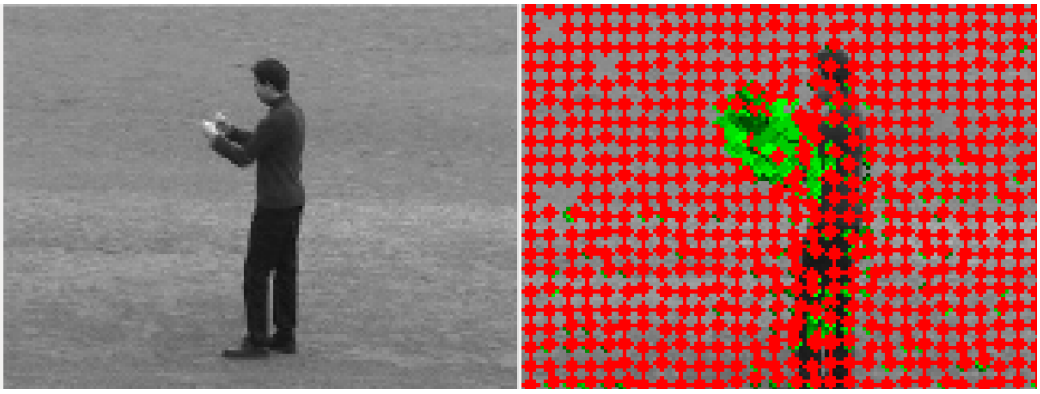

**Figura 2.2:** Ejemplo del muestreo mediante *Dense sampling*, en el vídeo el hombre aparece moviendo los brazos simulando estar golpeando. Los puntos rojos determinan el estado actual de la trayectoria y las líneas verdes el seguimiento de las mismas.

### *Descriptores alineados a la trayectoria*

La trayectoria computada previamente generará un volumen tridimensional sobre el que se extraerán las características siguientes:

- Histograma de Gradientes Orientados (*Histogram of Oriented Gradients* - HOG) [9]: Estas características están centradas en la información estática de la imagen, detectando contornos y formas.

- Histograma de Flujo Óptico (*Histogram of Optical Flow* - HOF) [6]: Este tipo de descriptores capturan la información local relacionada con el movimiento. Esta técnica ha demostrado ser efectiva para capturar las partes donde ocurren eventos dinámicos.

- Histograma de los bordes del movimiento (*Motion Boundary Histograms* - MBH) [48]: Este último tipo de descriptor es robusto frente a los movimientos de cámara habituales en los vídeos reales, como el *zoom* o la rotación. El cálculo de esta característica se lleva a cabo sepradamente en el eje $X$ y en el eje $Y$.

Los histogramas cuentan con 8 *bins*, salvo HOF, que cuenta con uno adicional denominado *zero-bin* [26], normalizados todos ellos mediante la normalización $L_2$. Se puede observar un ejemplo gráfico de la extracción de estas características en la Figura 2.3.

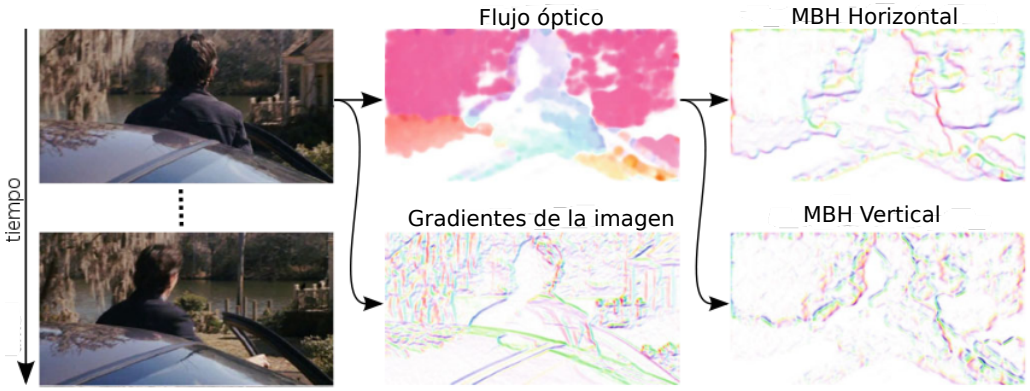


**Figura 2.3:** Ilustración de la información capturada por HOG, HOF y MBH. Debido al movimiento de la cámara (desplazándose de izquierda a derecha) el flujo óptico muestra movimiento constante de fondo. La mayoría de este ruido es eliminado mediante MBH, obteniendo la información del movimiento de la persona relativo al fondo. [48].

En cuanto a la complejidad temporal, el paso que mayor coste requiere es el cálculo del flujo óptico denso, utilizado para obtener tanto la trayectoria como los descriptores HOF y MBH, ocupando un 50 % del tiempo de cómputo. La implementación eficiente de este proceso mediante el algoritmo planteado en [11] alivia parte del coste, obteniéndose un proceso razonablemente abordable. Para un mayor análisis sobre la complejidad, se puede consultar [48], donde llevan a cabo una evaluación exhaustiva.

Finalmente, en la Figura 2.4 se muestra un esquema con el funcionamiento completo de la técnica. Desde los puntos *sampleados* de forma densa mediante diferentes escalas espaciales, se obtienen las trayectorias mediante el cálculo del flujo óptico durante $L$ *frames*, para finalmente computar los descriptores alrededor de la misma. Esta técnica cuenta con varios parámetros configurables como la longitud de la trayectoria, el desplazamiento entre píxeles, o las dimensiones del volumen obtenido que se describen en el Apéndice A, junto con el formato y dimensiones de las características extraídas.

### 2.1.2. Clasificación

Para esta tarea se utilizará el modelo basado en características locales planteado en [34], pero utilizando redes neuronales artificiales en lugar del vecino más cercano para la clasificación. Las redes neuronales artificiales se

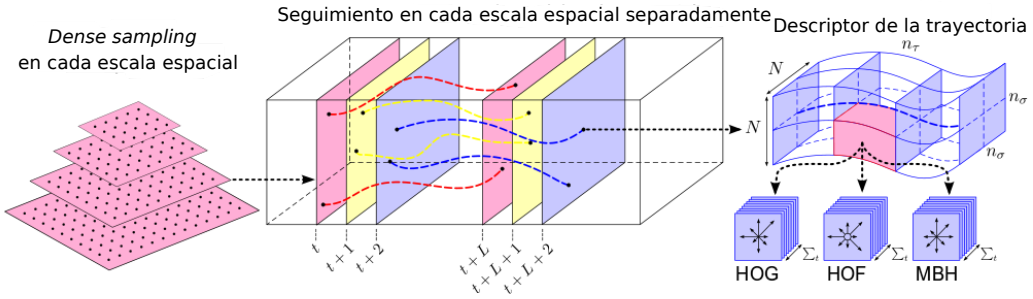


**Figura 2.4:** Ilustración del proceso completo de extracción de los descriptores *Dense trajectories*, *sampleado* a diferentes escalas (izquierda), se generan las trayectorias (centro) de las que se extraerán los descriptores (derecha). [48]

pueden interpretar como modelos que representan un grafo dirigido y ponderado, compuesto por neuronas que realizan el papel de nodos, y conexiones que realizan el papel de aristas ponderadas, mediante una relación entrada-salida entre neuronas[38].

Basadas en la dirección de las aristas, se pueden distinguir entre redes neuronales *feed-forward*, que representan estructuras acíclicas, y las redes recurrentes donde existen conexiones en ambas direcciones entre neuronas. Los denominados perceptrones multi-capa son las arquitecturas más comunes, donde las neuronas son organizadas por capas con conexiones unidireccionales entre ellas. Estas capas suelen dividirse entre capas de entrada, una o más capas ocultas, y una capa de salida.

El entrenamiento sobre este modelo se lleva a cabo de forma supervisada, mediante la actualización iterativa de los pesos con los que son ponderadas las conexiones que unen las neuronas. El algoritmo *Backpropagation* [38] proporciona una manera sencilla y eficiente de entrenar este tipo de modelos con el objetivo de resolver una tarea concreta. Este algoritmo se basa en la derivación mediante descenso por gradiente para la optimización de una función objetivo, definida como una medida de diferencia entre la salida al presentarse una entrada concreta y la salida realmente esperada.

Dentro del paradigma de clasificación, la estructura habitual suele contemplar $d$ neuronas en la capa de entrada, que coinciden con la dimensionalidad del problema, una o más capas ocultas compuestas por un número arbitrario de neuronas, y una capa de salida que comprende $C$ neuronas, donde $C$ es el número de clases contempladas para la tarea. Varias de estas

capas y neuronas pueden utilizar funciones de activación no líneales como la función *Sigmoid*[38] o la más recientemente planteada función *Rectified Linear*[8], que proporcionan que las redes neuronales puedan aproximar funciones no líneales. En cuanto a la salida final, el valor de activación refleja una aproximación de la probabilidad *a posteriori* de que la muestra presentada pertenezca a la clase que representa la neurona de salida.

Actualmente, este tipo de modelos han resurigido con fuerza por sus buenos resultados en diversas tareas como el reconocimiento del habla [29] o de objetos [23], gracias a los avances teóricos y computacionales recientes. Los esfuerzos en esta dirección tratan de obtener modelos más eficientes y precisos, tratando de capturar el conocimiento almacenado en las ingentes cantidades de datos con las que se cuenta actualmente para ciertas tareas.

El modelo planteado está basado en el entrenamiento de una red neuronal para la clasificación de las características locales de una muestra. En este caso, por cada vídeo, se extraen $m$ características locales que se utilizarán para entrenar una red neuronal discriminativa. La red entrenada clasificará cada característica local en su categoría correspondiente y posteriormente, mediante un esquema de fusión sobre la salida de la clasificación, se le asignará a cada vídeo la categoría predicha. Se puede ver un esquema del funcionamiento en la Figura 2.5.

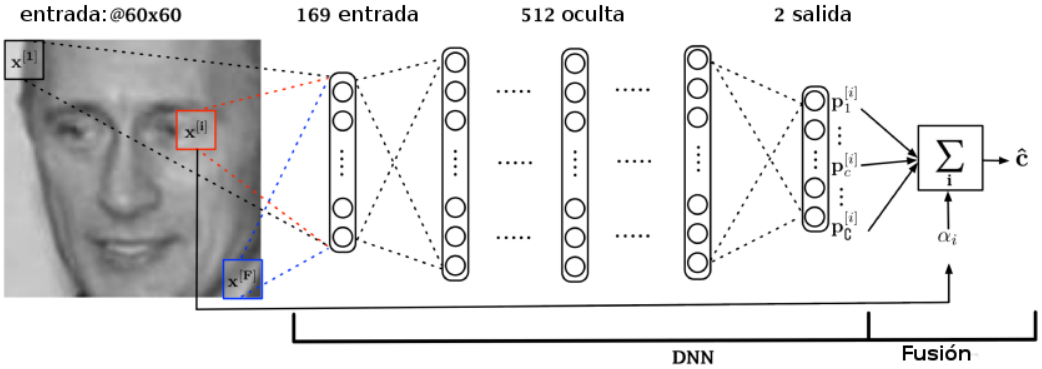


**Figura 2.5:** Modelo de Red Neuronal de características locales para el reconocimiento de imagen, donde se ilustra la parte discriminativa a nivel de las características y la fusión final.

Formalmente, como se muestra en [34], la extracción de características locales puede verse como sigue. Siendo $V = v_1, v_2, ..., v_n$ un conjunto de en-

trenamiento de $n$ vídeos, pertenecientes a $c \in 1, ..., C$ diferentes categorías, de cada vídeo $v$, se extraen $F$ características locales. Se asume que cada característica contendrá información parcial sobre el vídeo, pero relevante para determinar la clase a la que pertenece, siendo $c_i \in 1, ..., C$ la clase de la característica.

De acuerdo con esto, la probabilidad *a posteriori* de que $v$ pertenezca a $c$ se calcula mediante el modelo completo conformado por todas las características locales de $v$:

$$p(c|v) = \sum_{c_1=1}^{C} ... \sum_{c_F=1}^{C} p(c, c_1, ..., c_F|v) \tag{2.1}$$

Mediante las propiedades de la probabilidad conjunta, podemos descomponer en dos submodelos el término condicionado. El primer modelo, para la probabilidad condicionada para las clases de las características, es decir, las clases a nivel local, y el segundo modelo que calcula la probabilidad de la clase a nivel global, dada la información de clase local y el dato de entrada.

$$p(c, c_1, ..., c_F|v) = p(c_1, ..., c_F|v)p(c|v, c_1, ..., c_F) \tag{2.2}$$

Con el objetivo de obtener un modelo práctico para la probabilidad $p(c|x)$, se simplificará el primer modelo asumiendo independencia estadística entre las clases de las características locales condicionadas por $v$, aplicando la siguiente aproximación.

$$p(c_1, ..., c_F|v) \approx \prod_{i=1}^{F} p(c_i|v^{[i]}) \tag{2.3}$$

Donde $v^{[i]}$ representa las partes relevantes de $v$ para predecir $c_i$, es decir, la característica local $i$-ésima. La siguiente asunción es la independencia, dadas las clases de las características, del dato de entrada, simplificando el segundo modelo.

$$p(c|v, c_1, ..., c_F) \approx p(c|c_1, ..., c_F) \tag{2.4}$$

Para esta probabilidad *a posteriori*, se propone la aproximación siguiente, poco realista pero razonable si la clasificación de una característica local se

puede clasificar se puede llevar a cabo de forma precisa, independientemente del resto, por ejemplo, conteniendo información claramente discriminativa. La asunción propuesta es que la probabilidad *a posteriori* de la clase dadas las clases de los descriptores se aproxime mediante el voto de cada una de las clases de las características locales, ponderado por un peso de "confianza".

$$p(c|c_1, ..., c_F) \approx \sum_{i=1}^{F} \alpha_i \delta(c_i, c) \tag{2.5}$$

Donde $\delta(.,.)$ es la función *delta* de *Kronecker*, definida como:

$$\delta(c_i, c) = \begin{cases} 1, & \text{if } c_i = c, \\ 0, & \text{if } c_i \neq c. \end{cases} \tag{2.6}$$

El parámetro $\alpha$ define la importancia de la característica local en el cómputo de la probabilidad, sujeto a $0 \leq \alpha_i \leq 1, i = 1, ..., F$ y $\sum_i \alpha_i = 1$, para considerarlo una ponderación sobre las características. Este factor codifica el poder discriminativo de una característica local. En el caso más sencillo, todas son equiprobables, es decir:

$$\alpha_1 = \alpha_2 = ... = \alpha_F = \frac{1}{F} \tag{2.7}$$

Este modelo, mediante las asunciones propuestas, no toma en consideración ni la posición dentro del vídeo ni la posible relación entre las locales, considerando que la información contenida en los descriptores es suficientemente discriminativa. Siguiendo el desarrollo del artículo, el cálculo de la probabilidad *a posteriori* quedaría:

$$p(c|v) \approx \sum_{i=1}^{F} \alpha_i p(c|v^{[i]}) \tag{2.8}$$

Que es la suma ponderada de las probabilidades *a posteriori* sobre todas las características locales. Las probabilidades *a posteriori* serán estimadas utilizando una red neuronal artificial, que será entrenada de forma supervisada. La clasificación proporcionada por la red será la entrada para la siguiente fase del proceso, en el que se fusionarán los resultados de la clasificación para dar lugar a la clasificación final

Una vez entrenada la red, como se ha comentado previamente, cada neurona proporciona una respuesta a la entrada presentada. Esta respuesta se

puede interpretar como un vector $s \in \mathbb{R}^C$, y mediante la función de salida *SoftMax*[38], como una probabilidad. La regla de decisión para clasificar un vídeo se definirá escogiendo la clase según uno de estos dos criterios, relacionados con el vector de salida proporcionado por la red:

- La clase más votada, es decir, la categoría que proporciona el máximo para un vector de salida dado, representa un voto, sumando los votos y escogiendo el máximo, se obtiene la clase predicha.
- La clase con mayor suma de probabilidades *a posteriori*, es decir, sumando todos los vectores de salida y determinando como clase ganadora la que mayor suma ha obtenido.

La elección de este modelo viene motivada por la idea de aportar un enfoque diferente y poder realizar un estudio prospectivo sobre la tarea dentro de los plazos propuestos para este trabajo, mediante un modelo que no había sido utilizado previamente en el ámbito del reconocimiento de vídeo, llevando a cabo la clasificación basándose solamente en características de bajo nivel, como son los descriptores iDT. Esto facilita el reducir el número de etapas del proceso, en comparación con las técnicas actuales, delegando la codificación de las características al mecanismo de aprendizaje de la red. Otras aproximaciones han hecho uso de redes neuronales pero con un modelo completamente diferente, con la idea de que la red aprenda a discriminar las posiciones relevantes del vídeo [4].

Dado que existen implementaciones eficientes y sencillas para las diferentes partes del modelo, esta propuesta no necesita un entorno de altas prestaciones para su ejecución, pero, sin embargo, puede aprovechar de forma muy ventajosa los avances en el campo de la parelelización, debido a que los modelos de redes neuronales y su entrenamiento comprenden operaciones matriciales bien conocidas que admiten este tipo de optimizaciones. Como desventaja, este tipo de modelos pierden tanto la información relativa al orden como a la posición, ya que aprenden a clasificar partes del vídeo independientemente de estos factores. Por otro lado, las redes neuronales tienen diversos parámetros de ajuste, los cuales pueden conducir a un sobreaprendizaje del problema, necesitando ser evaluados con cautela mediante un conjunto de validación. En ocasiones, dar con el conjunto de parámetros óptimo es extremadamente díficil, necesitando realizar numerosas pruebas y experimentos para obtener buenos resultados.

En la Figura2.6, se ilustra la instanciación del modelo genérico dentro del paradigma de reconocimiento de formas para esta tarea de reconocimiento de

la actividad humana en vídeos no restringidos. Resumiendo el proceso, la primera etapa preprocesará y extraerá las características relevantes de los vídeos mediante *improved Dense trajectories*. La segunda etapa recibirá los descriptores locales extraídos para entrenar un modelo discriminativo mediante una red neuronal artificial. La salida de la red se someterá a un proceso de fusión sobre todas las características locales de un vídeo para proporcionar la clase final de la muestra presentada.

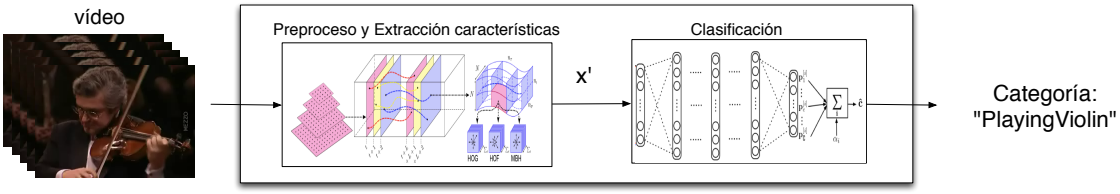


**Figura 2.6:** Esquema del sistema de reconocimiento de formas planteado para el reconocimiento de la actividad

Capítulo 3

# Experimentación y resultados

## 3.1. Descripción de la tarea

La tarea a la que pretende dar solución este trabajo es la planteada en la competición THUMOS en la edición del 2013 [51], sobre clasificación de acciones. Se ha escogido la tarea planteada en esta edición por las limitaciones temporales, ya que en la edición de 2014[18] se amplió el conjunto de datos y se cambió el modelo de evaluación, aumentando la complejidad y por tanto el tiempo requerido. La edición del 2015 estaba comenzando a plantearse cuando se consideró el desarrollo de este proyecto.

En esta tarea se plantea la clasificación de vídeos reales obtenidos de *Youtube* en un número establecido de categorías de acciones. El objetivo es llevar a cabo la clasificación de vídeos grabados en un entorno no controlado, con cambios en cualquier característica, como iluminación o posición del elemento de interés. Para ello, se utiliza un conjunto de datos, propuesto por los organizadores, que se ha convertido en uno de los bancos de pruebas más utilizados para la evaluación de sistemas de reconocimiento de acciones, el *dataset* UCF-101.

### 3.1.1. UCF-101

El *corpus* UCF-101[43] nace con el objetivo de recopilar un conjunto de vídeos subidos por los usuarios a *Youtube*, para conformar un *dataset* variado y realista. Este conjunto de datos lideró en su momento el *ranking* como el conjunto con mayor número de vídeos, más de 13.000 frente a los menos de 7.000 que ofrecía UCF50 [37], la versión previa de este conjunto, y mayor número de categorías, planteando 101 categorías frente a las 51 de

HMDB51 [24], otro conjunto ampliamente utilizado compuesto por escenas de películas.

La principal ventaja de este conjunto de datos fue que proporcionaba multitud de vídeos caseros cuyo entorno no está restringido y donde las condiciones no son perfectas, ofreciendo un entorno realista, a diferencia de KTH [40] o Weizman [3], donde las acciones se llevaban a cabo mediante actores y en situaciones poco realistas.

Las 101 acciones que comprende el *dataset* se subdividen en 5 tipos:

- *Human-Object interaction*: Con 20 categorías distintas, i.e: "*Apply Eye Makeup*" o "*Brushing Teeth*".
- *Body-Motion only*: Con 16 categorías distintas, i.e: "*Baby Crawling*" o "*Swing*"
- *Human-Human interaction*: Con 5 categorías distintas, i.e: "*Band Marching*" o "*Head Massage*"
- *Playing Musical instruments*: Con 10 categorías distintas, i.e: "*Playing Piano*" o "*Playing Violin*"
- *Sports*: Con 50 categorías distintas, i.e: "*Archery*" o "*Bowling*"

En la Figura 3.1 se muestra, con una imagen en cada caso, las categorías que conforman el *corpus*. Los vídeos de una categoría estan divididos en 25 grupos, de 4-7 vídeos cada uno. Los vídeos dentro de un grupo contienen características semejantes, por ejemplo, el fondo o las personas que participan en él.

Un resumen de las características de este conjunto de datos puede verse en la Tabla 3.1. Todos los vídeos han sido descargados de la plataforma *Youtube* y se ha normalizado tanto el número de imágenes por segundo (*frame rate* o fps) como la resolución, a 25 fps y 320 x 240 píxeles, respectivamente. El audio se ha capturado para las 51 nuevas acciones incluidas que no estaban presentes en UCF50.

El número de vídeos para cada clase puede verse en la Figura 3.2, así como su duración. La media de vídeos por clase es de 132, siendo 100 y 167 el mínimo y el máximo número de vídeos. Entre las clases que menos vídeos tienen, encontramos "*PlayingViolin*", "*PullUps*", "*Skijet*" y "*TaiChi*", con

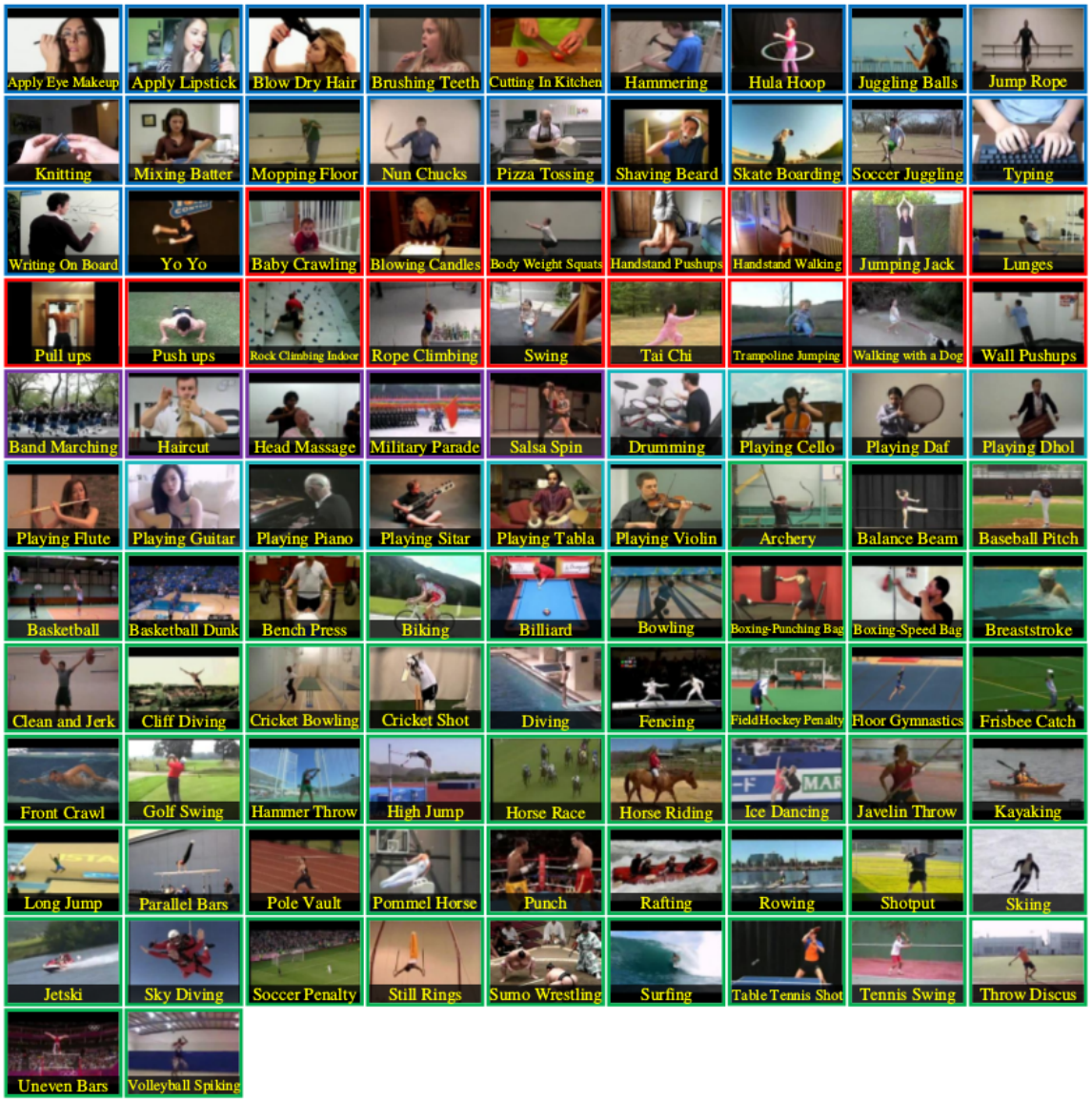


**Figura 3.1:** Ejemplos de fotogramas de cada una de las 101 clases de actividades presentes en el conjunto de datos. Los marcos de las imágenes representan a qué grupo de acciones pertenecen: azul para *Human-Object interaction*, rojo para *Body-Motion*, morado para *Human-Human interaction*, turquesa para *Playing Musical Instruments* y verde para *Sports*. [43].

100 vídeos, y entre las que más "*CricketShot*", "*TennisSwing*", con 167 y 166 respectivamente.

| Acciones | 101 |
|---|---|
| Clips | 13.320 |
| Grupos por acción | 25 |
| Clips por grupo | 4-7 |
| Duración media del clip | 7.21 seg |
| Duración total | 1.600 mins |
| Duración mínima | 1.06 seg |
| Duración máxima | 71.04 seg |
| Resolución y *Frame Rate* | 320x240@25fps |

**Tabla 3.1:** Características del corpus UCF101

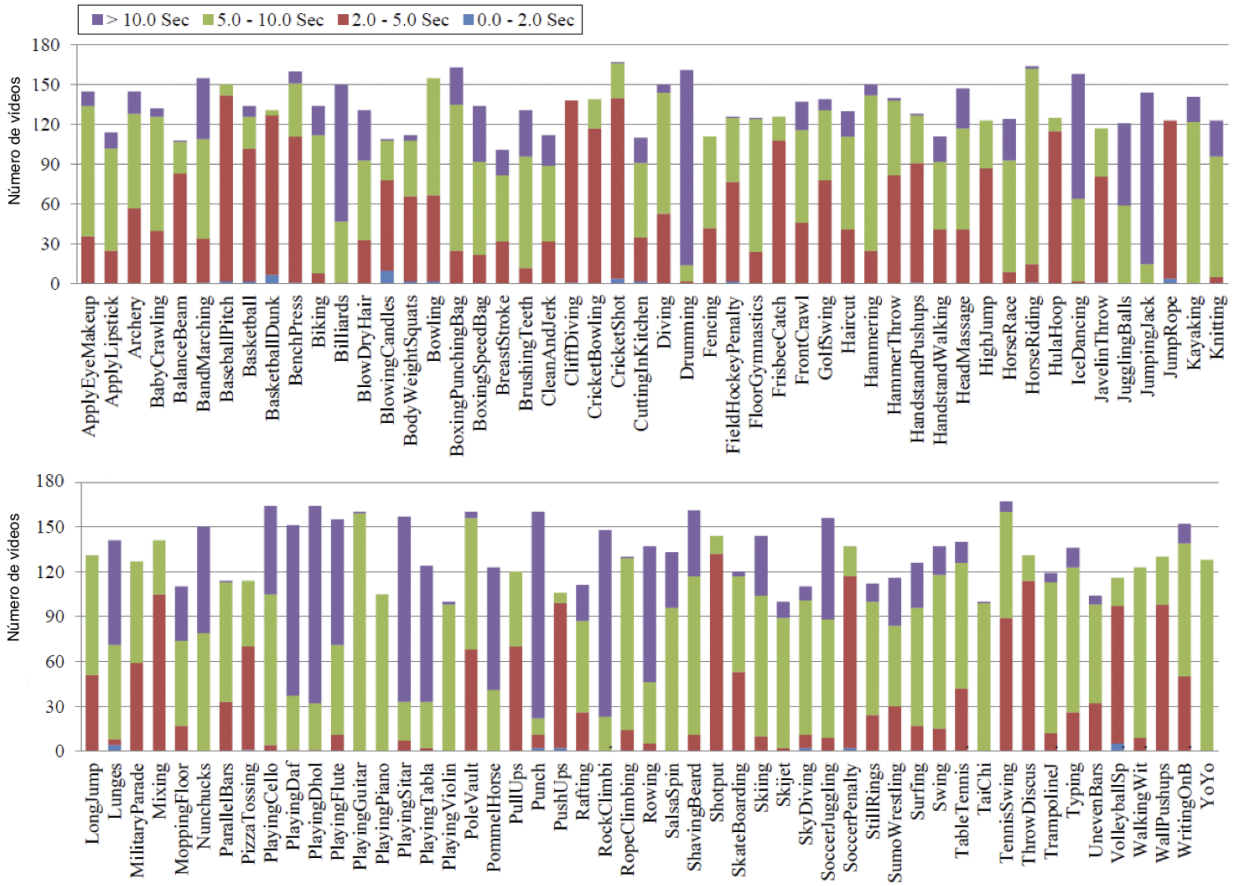


**Figura 3.2:** Duración y número de vídeos por clases [51].

### 3.1.2. Evaluación y Resultados previos

El mecanismo de evaluación para la tarea de clasificación de acciones en esta edición contempla el promedio de la precisión para cada clase, y posteriormente, el promedio de cada partición propuesta. Cada clase se compone de 25 grupos de vídeos, que se dividen en 18 grupos para entrenamiento y 7 para *test*. El mejor resultado obtenido en la edición del 2013 [51] fue un 85.90 % de precisión media sobre las tres particiones, por el equipo de LEAR-INRIA[1], mediante la extracción de características iDT y *Fisher Vectors* (FV) como mecanismo de codificación. Para la clasificación se empleó SVM lineal, como en la mayoría de equipos participantes. En la Tabla 3.2 se reflejan los resultados y algunos detalles sobre las características, la codificación y las técnicas de clasificación empleadas, para los 10 primeros equipos.

| **Equipo** | **Características de Bajo Nivel** | **Codificación** | **Clasificación** | **Resultados** |
|---|---|---|---|---|
| INRIA | iDT(video stab.) | Fisher Vector | linear SVM | 85.90 % |
| Florence | DTF, STIP, P-SIFT | Fisher Vector, BOVW | HK SVM and Linear SVM | 85.50 % |
| Cancerba | iDTF | Fisher Vector | linear SVM | 85.43 % |
| CAS-SIAT | iDTF | VLAD + Fisher Vector | linear SVM | 84.16 % |
| Nanjing | DTF | VLAD + bimodel encoding | PmSVM | 83.97 % |
| UCF_Tappen | DTF, STIP | BOVW, Fisher Vector | Kernel Map + 2-layer NN | 82.82 % |
| UCSD_MSRA_SJTU | DTF | LSAQ coding | linear SVM | 80.89 % |
| USC | DTF, MoSIFT | Fisher Vector | RBF Kernel - SVM | 77.36 % |
| NII | DTF, SIFT, MFCC | Fisher Vector (PCA) | linear SVM | 73.38 % |
| UNITN | 3D-HOG y 3D-HOF | Fisher Vector | SVM | 70.50 % |

**Tabla 3.2:** Resultados y detalles de la edición THUMOS 2013 [51].

Recientemente, en [45], se han publicado mejores resultados utilizando redes convolucionales 3D. Esta técnica, junto con la combinación de las características iDT-FV, ha obtenido un 90.4 % de precisión sobre las tres particiones.

En el siguiente apartado se llevará a cabo la descripción detallada de la experimentación realizada para la tarea de reconocimiento de vídeos.

## 3.2. Herramientas

Para llevar a cabo la experimentación se ha empleado en primer lugar, el código utilizado en [48] para la extracción de las características iDT, accesible a través de su sitio *web*[2]. Para el procesamiento de las características

[1] http://lear.inrialpes.fr/index.php - Consulta: 10 de julio de 2015

[2] http://lear.inrialpes.fr/people/wang/improved_trajectories - Consulta: 10 de julio de 2015

extraídas, se ha utilizado el lenguaje de programación *Python* 2.7.6, con librerías adicionales para el cálculo numérico y estadístico como Numpy[3] y scikit-learn[4].

Para la clasificación, se ha empleado el *toolkit* DeepNet[5], desarrollado en la Universidad de Toronto [16]. Este *toolkit* está implementado integramente en *Python*, es moderadamente fácil de utilizar y contempla varios modelos de aprendizaje automático: Redes neuronales, *Restricted Boltzmann Machines, Deep Belief Nets, Autoencoders, Deep Boltzmann Machines* y *Convolutional Nets*.

Hace uso de librerías de cálculo numérico y matricial como Eigen[6] además de Numpy y, proporciona, mediante la implementación de Vlad Mnih [31] y Alex Krizhevsky [23], librerías para el trabajo con unidades GPU mediante CUDA[7]. Sin embargo, en el desarrollo de este trabajo, solamente se ha podido hacer uso de las ventajas que ofrece la librería Eigen, ya que no se disponía de GPU compatible con CUDA.

En cuanto a los recursos computacionales, la última parte de la experimentación ha sido llevada a cabo en el *cluster* habilitado dentro del Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia. Este *cluster* dispone de 250 GB de memoria RAM y 4 nodos que suman un total de 120 *cores*.

## 3.3. Experimentación

La experimentación parte de explorar, dadas las limitaciones temporales, una de las tres particiones que proponen en la tarea. El objetivo principal de este proceso es encontrar la mejor combinación de arquitecturas y características locales para obtener la mayor precisión media sobre la primera partición y finalmente evaluar los resultados con las dos restantes.

Esta partición divide los 13.320 vídeos en dos conjuntos, una parte de entrenamiento de 9.537 vídeos y 3.783 para test. La división propuesta co-

[3]http://www.numpy.org/ - Consulta: 10 de julio de 2015
[4]http://scikit-learn.org/stable/ - Consulta: 10 de julio de 2015
[5]https://github.com/nitishsrivastava/deepnet - Consulta: 10 de julio de 2015
[6]http://eigen.tuxfamily.org - Consulta: 10 de julio de 2015
[7]https://github.com/cudamat - Consulta: 10 de julio de 2015

rresponde al uso de los grupos del 1 al 18 como vídeos de entrenamiento y del 19 al 25 como vídeos de test. A su vez, la parte de entrenamiento se ha subdivido para dar lugar a una parte para validación, siendo los grupos del 1 al 15 los vídeos de entrenamiento, y los grupos del 16 al 18 para validación, 7.944 y 1.593 vídeos respectivamente. La Tabla 3.3 resume los datos de esta partición.

| | |
|---|---|
| Clips totales | 13.320 |
| Grupos por acción | 25 |
| Grupos para entrenamiento | 15 |
| Grupos para validación | 3 |
| Grupos para test | 7 |
| # vídeos de entrenamiento | 7.944 |
| # vídeos de validación | 1.593 |
| # vídeos de entrenamiento totales | 9.537 |
| # vídeos de test | 3.783 |

**Tabla 3.3:** Características de la partición uno propuesta para el conjunto UCF101.

### 3.3.1. Extracción de características

Mediante el *script* proporcionado por los desarrolladores, se han computado las características para cada uno de los 13.320 vídeos que conforman el *dataset*. Los parámetros utilizados para la extracción de los descriptores iDT han sido los mismos que en [48], trabajo donde han demostrado un buen resultado en la tarea del reconocimiento de la acción en vídeos no restringidos. En cuanto a la dimensionalidad obtenida con esta configuración, la Tabla 3.4 desglosa las diferentes partes de cada característica. Los paramétros para la extracción han sido:

- $S$ y $T$: Se extraen características desde el principio del vídeo hasta el final.
- $L$: 15 *frames*.
- $W$: 5 píxeles.
- $N$: 32 píxeles.
- $s$: 2 celdas.

| | |
|---|---|
| Trayectoria | 39 (9 + 30) |
| HOG | 96 (8 x 2 x 2 x 3) |
| HOF | 108 (9 x 2 x 2 x 3) |
| MBH | 192 (8 x 2 x 2 x 2 x 3) |
| Dimensionalidad Total | 435 |

**Tabla 3.4:** Dimensionalidad de los vectores iDT

- $t$: 3 celdas.

Con este proceso finalizado, se le ha aplicado normalización a los datos con el objetivo de centrarlos mediante *z-score*. Esto deja las características con $\mu \approx 0$ y $\sigma \approx 1$, para facilitar el aprendizaje en la etapa posterior, como recomiendan en [28].

Cabe destacar que el número de características locales depende del movimiento presente en el vídeo y por ello, tras la extracción, se han detectado vídeos con muy pocas características. Como ejemplo, en un vídeo dentro de la clase "*PlayingFlute*", donde el movimiento más destacable es el de un dedo, se ha podido obtener solamente 15 características frente a, por ejemplo, las más de 3500 características obtenidas para vídeos de la clase "*Punch*". Esto es debido al descarte que realiza la técnica iDT sobre las zonas homogéneas, que componen la mayoría de la escena en los vídeos donde se han obtenido pocas características. La Figura 3.3 muestra la suma de las características locales obtenidas para cada clase.

Una vez obtenidos los descriptores iDT de los vídeos, se continuará con el aprendizaje del modelo basado en redes neuronales.

### 3.3.2. Clasificación

#### Entrenamiento del modelo

Se efectuarán pruebas evaluando las variaciones en la selección de las características locales y el número de características por vídeo, en cuanto a la extracción. En cuanto a la arquitectura y parámetros de la red, se variará el número de capas ocultas y el número de unidades ocultas en cada capa. Se utilizarán estructuras regulares, es decir, que siempre tendrán el mismo número de unidades en cada capa. Además durante la experimentación se evaluarán las funciones de activación más utilizadas, así como el empleo de

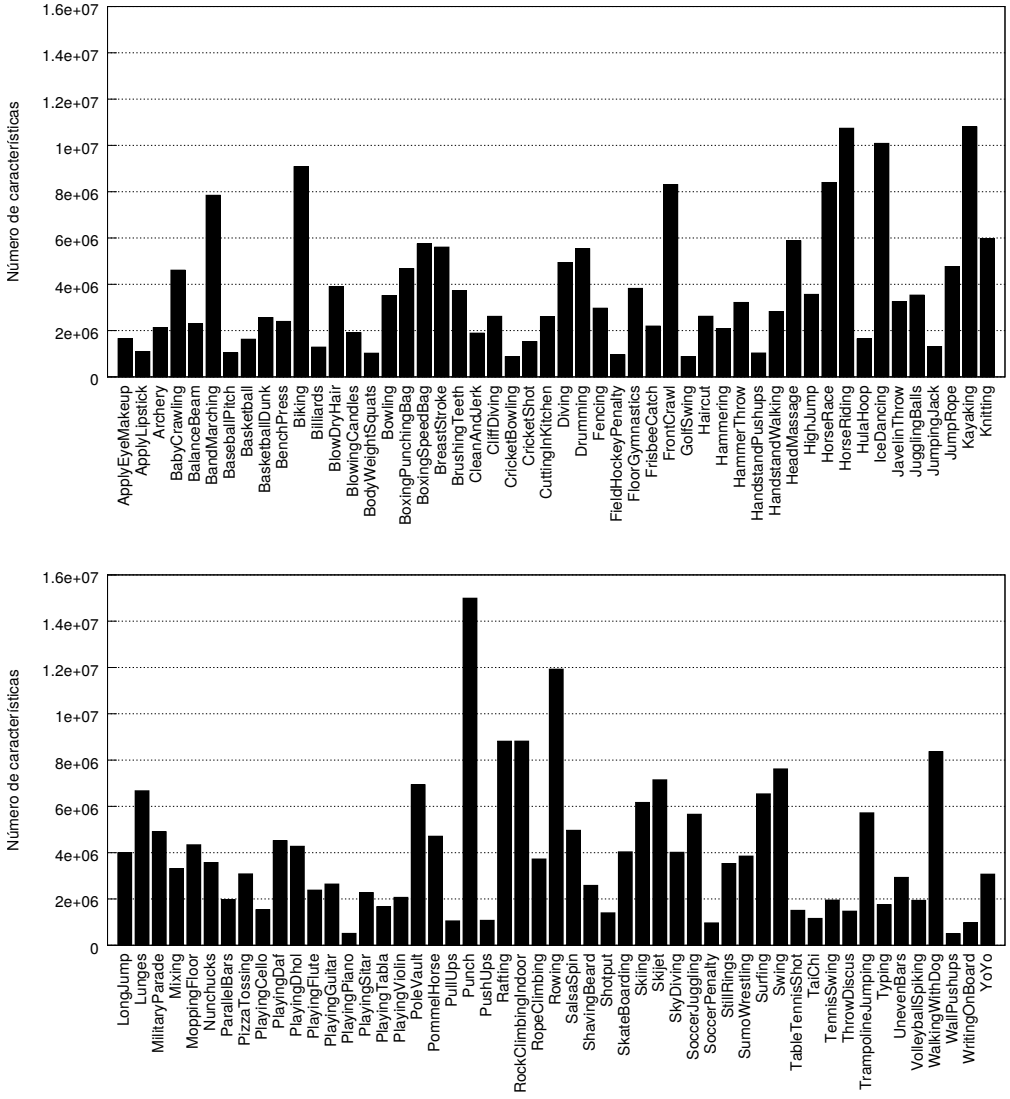


**Figura 3.3:** Distribución de características locales por clases.

*DropOut* [16]. Para el entrenamiento, se utilizarán las técnicas recomendadas por [28], como la inicialización y regularización de los pesos de la red.

Para el ajuste de los hiperparámetros se ha utilizado el conjunto de validación. El esquema de aprendizaje seguido es el siguiente:

- Entrenamiento mediante el conjunto compuesto por los 15 grupos y validación del modelo con el rest, 3 grupos, hasta el número de iteraciones

definido.

- Entrenamiento mediante el conjunto completo de 18 grupos, hasta que el valor de la *Cross-Entropy* sea menor que el mejor modelo de la etapa anterior.

Tras el entrenamiento, obtenemos la salida de la red neuronal en forma de matriz $m_{n\ x\ d}$, donde $n$ es el número de características locales de un vídeo y $d$ es la dimensión de salida, es decir, el número de categorías, 101. Esta matriz representa, en cada fila, la probabilidad $p(c_j|v_{[i]})$, donde $j$ es el índice de la columna, correspondiente a la categoría $c_j$, e $i$ hace referencia a la $i$-ésima característica local del vídeo, es decir, la fila $i$-ésima de la matriz.

A continuación, se lleva a cabo la etapa de fusión sobre la salida de la red neuronal. En esta etapa, se han utilizado las dos estrategias de fusión propuestas en el modelo. La aplicación de la fusión dará lugar a un conjunto de etiquetas, una por vídeo, sobre la que se calculará finalmente la precisión media sobre cada clase.

## 3.4. Resultados

En primer lugar, se muestran los resultados obtenidos tras evaluar la función de activación *Sigmoid*, o la utilización de la función de activación *Rectified Linear Unit* (ReLu) [8], que ha demostrado mejores resultados en trabajos recientes [32]. Este experimento se ha llevado a cabo escogiendo aleatoriamente 100 de las características locales iDT computadas previamente. No todos los vídeos han obtenido el mismo número de características, ya que como se ha expuesto anteriormente, la cantidad de éstas depende del movimiento que refleje el vídeo, por ello, existen vídeos con las 100 características y algunos con menos. El número total de características escogiendo 100 como máximo han sido 951.593 y 377.798 para entrenamiento y *test*, respectivamente.

Se muestra en la Tabla 3.5 y la Tabla 3.6, tanto el acierto a nivel de característica local, en la tercera columna, como el acierto a nivel de fusión, mediante el esquema de votación y suma de probabilidades *a posteriori*, en la cuarta y la quinta columna respectivamente.

Aunque a nivel de característica local, la función *Sigmoid* ha proporcionado mejores resultados, en la fusión se ha comportado generalmente peor.

| Capas | Unidades | Prec. % C.L | Prec. % Vot. | Prec. % Sum. |
|---|---|---|---|---|
| 1 | 128 | 17.98 | 46.76 | 49.51 |
| | 256 | 19.57 | 51.17 | 53.50 |
| | 512 | 21.08 | 53.29 | 54.66 |
| | 1024 | 21.95 | 55.85 | 57.65 |
| | 2048 | 22.34 | 55.96 | 57.22 |
| 2 | 128 | 19.17 | 49.22 | 51.86 |
| | 256 | 20.97 | 54.87 | 57.73 |
| | 512 | 22.55 | 56.67 | 57.89 |
| | 1024 | 22.55 | **61.61** | **63.78** |
| | 2048 | 20.87 | 54.25 | 55.90 |
| 3 | 128 | 19.39 | 49.69 | 52.07 |
| | 256 | 21.46 | 54.45 | 57.22 |
| | 512 | 22.73 | 58.23 | 60.90 |
| | 1024 | 23.20 | 60.53 | 62.17 |
| | 2048 | **24.09** | 58.19 | 60.73 |

**Tabla 3.5:** Resultados con la función *Sigmoid*, obteniendo 100 c.l./vídeo, mediante selección aleatoria.

| Capas | Unidades | Prec. % C.L | Prec. % Vot. | Prec. % Sum. |
|---|---|---|---|---|
| 1 | 128 | 16.75 | 43.51 | 47.39 |
| | 256 | 18.47 | 50.96 | 53.42 |
| | 512 | 20.01 | 55.16 | 57.81 |
| | 1024 | 20.73 | 60.48 | 62.70 |
| | 2048 | 20.94 | 59.74 | 61.74 |
| 2 | 128 | 17.84 | 45.91 | 48.61 |
| | 256 | 20.48 | 55.00 | 57.94 |
| | 512 | 21.40 | 59.23 | 61.14 |
| | 1024 | **21.78** | 61.61 | 63.78 |
| | 2048 | 21.56 | **62.27** | **64.81** |
| 3 | 128 | 18.41 | 47.47 | 50.83 |
| | 256 | 20.17 | 52.73 | 55.35 |
| | 512 | 20.93 | 57.54 | 60.11 |
| | 1024 | 21.22 | 61.11 | 63.20 |
| | 2048 | 21.38 | 59.18 | 61.43 |

**Tabla 3.6:** Resultados con la función ReLu, obteniendo 100 c.l./vídeo, mediante selección aleatoria.

Los mejores resultados a nivel local para la *Sigmoid* quizá se deban a la poca profundidad de la red, y de que no sufra del problema de los "*vanishing gradients*" [2], que implica que no se retropropague el error lo suficiente como para que la red aprenda, siendo este el escenario idóneo de uso de las funciones de activación *Rectified Linear*. En cualquier caso, a la vista de los resultados favorables a nivel de fusión, se optará por la función de activación ReLu.

En vista de los resultados, parece que la función *Sigmoid* tiende a dispersar más la masa de probabilidad que la función ReLu, ya que aunque ha obtenido un mejor acierto a nivel de característica, en la fusión se ha dispersado la ganancia. Esto provoca que en el momento de escoger el máximo o de sumar las probabilidades *a posteriori*, existan características locales que, aun siendo clasificadas correctamente, aporten cierta probabilidad a la clase errónea, declinando la balanza en el cómputo global por la clase equivocada.

En el siguiente apartado se comparará la selección aleatoria de las características frente a un criterio establecido. El criterio viene motivado por el contenido del HOF, ordenando las características según la cantidad de información almacenada en los histogramas de este tipo, representativos del movimiento, tratando de capturar las zonas más dinámicas. Los resultados se muestran con la misma estructura de las tablas previas, en este caso para 2 o 3 capas ocultas y 512, 1024 y 2048 unidades por capa, en la Tabla 3.7.

| Capas | Unidades | Prec. % C.L | Prec. % Vot. | Prec. % Sum. |
|---|---|---|---|---|
| 2 | 512 | 27.38 | **59.91** | **60.32** |
| | 1024 | 27.31 | 59.62 | 60.21 |
| | 2048 | **27.88** | 58.12 | 59.82 |
| 3 | 512 | 27.29 | 58.45 | 59.95 |
| | 1024 | 27.49 | 53.21 | 54.42 |
| | 2048 | 26.88 | 59.12 | 60.04 |

**Tabla 3.7:** Resultados con la función ReLu, obteniendo 100 c.l./vídeo, mediante selección basada en HOF.

De nuevo, la precisión a nivel de característica local ha aumentado, de 24.09 % como mejor resultado obtenido en los experimentos previos a 27.88 %, sin embargo, este incremento no ha repercutido en un aumento de la precisión a nivel de fusión, obteniéndose un 60.32 % con un 27.38 % a nivel de locales, frente al 64.81 % y 21.56 % a nivel de fusión y características respec-

tivamente. De nuevo, no parece existir una relación clara entre la precisión a nivel de característica.

Para paliar el error inducido por las clases que, aun no siendo ganadoras, aportan votos o masa de probabilidad, así como para comprobar la relación entre la precisión y el número de características obtenidas por vídeo, en el siguiente experimento se evaluá la respuesta del modelo ante la extración 300, 500 y 1.000 características por vídeo. La Tabla 3.8 muestra el número de características locales totales obtenidas para cada uno de los niveles de selección.

| Características extraídas por vídeo | Total entrenamiento | Total *test* |
|---|---|---|
| 100 | 951K | 377K |
| 300 | 2.845K | 1.128K |
| 500 | 4.752K | 1.882K |
| 1.000 | 9.465K | 3.744K |

**Tabla 3.8:** Número de características totales según la cantidad establecida para la extracción.

Las tablas contemplan los experimentos de 2 a 3 capas ocultas y de 128 a 2048 neuronas por capa, salvo en el caso de 1.000 características, que por motivos temporales, solo ha sido llevado a cabo con el modelo que mejor resultado ha dado para los anteriores experimentos. Las Tablas 3.9, 3.10 y 3.11 muestran los resultados obtenidos.

| Capas | Unidades | Prec. % C.L | Prec. % Vot. | Prec. % Sum. |
|---|---|---|---|---|
| 2 | 128 | 18.52 | 47.55 | 49.82 |
| | 256 | 21.57 | 52.36 | 54.53 |
| | 512 | 23.27 | 61.72 | 63.30 |
| | 1024 | 23.84 | 66.71 | 67.77 |
| | 2048 | 25.10 | **68.70** | **69.68** |
| 3 | 128 | 20.68 | 53.07 | 54.95 |
| | 256 | 22.07 | 57.30 | 59.34 |
| | 512 | 23.48 | 62.19 | 63.62 |
| | 1024 | **26.16** | 65.82 | 66.53 |
| | 2048 | 26.11 | 67.64 | 67.90 |

**Tabla 3.9:** Resultados con la función ReLu, obteniendo 300 c.l./vídeo, mediante selección aleatoria.

| Capas | Unidades | Prec. % C.L | Prec. % Vot. | Prec. % Sum. |
|---|---|---|---|---|
| 2 | 128 | 18.48 | 48.98 | 50.93 |
| | 256 | 21.58 | 55.53 | 58.04 |
| | 512 | 23.74 | 62.27 | 63.52 |
| | 1024 | 25.28 | 66.77 | 68.09 |
| | 2048 | **27.10** | **69.11** | **70.09** |
| 3 | 128 | 20.34 | 50.56 | 52.39 |
| | 256 | 21.55 | 55.77 | 57.36 |
| | 512 | 23.16 | 61.56 | 63.15 |
| | 1024 | 23.03 | 62.33 | 65.37 |
| | 2048 | 26.23 | 66.45 | 67.06 |

**Tabla 3.10:** Resultados con la función ReLu, obteniendo 500 c.l./vídeo, mediante selección aleatoria.

| Capas | Unidades | Prec. % C.L | Prec. % Vot. | Prec. % Sum. |
|---|---|---|---|---|
| 2 | 2048 | 27.21 | 71.60 | 71.98 |

**Tabla 3.11:** Resultados con la función ReLu, obteniendo 1.000 c.l./vídeo, mediante selección aleatoria.

Como cabría esperar, la precisión aumenta, tanto a nivel de característica local como a nivel de fusión, conforme el número de características aumenta, apreciándose en este caso como si existe una relación entre el incremento de precisión del acierto a nivel local y la precisión a nivel de fusión. Se ha obtenido el mejor resultado hasta el momento con la red de 2 capas ocultas, con 2048 neuronas por capa, utilizando la función ReLu y mediante la selección aleatoria de 1.000 características por vídeo.

En la Figura 3.4 podemos ver gráficamente como repercute el aumento de características locales para la precisión del sistema a nivel de característica y a nivel de fusión, en este caso, mediante el esquema de suma de probabilidades *a posteriori*. En las Figuras 3.5, 3.6, 3.7 y 3.8 se muestran las comparativas de los resultados con los mejores modelos, para cada una de las cantidades elegidas. En cuanto al acierto a nivel de categoría, los resultados quedan reflejados en la Figura 3.9.

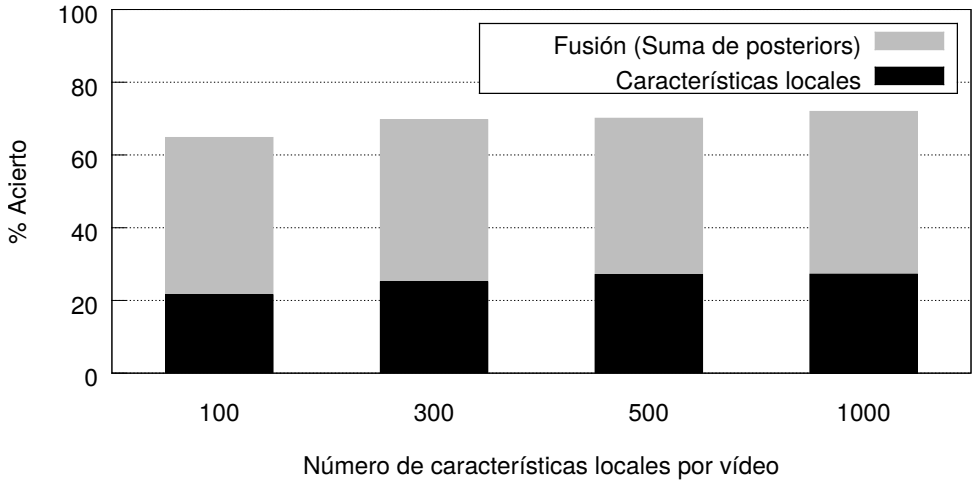


**Figura 3.4:** Diagrama ilustrando la variación del acierto en relación al aumento de características locales.

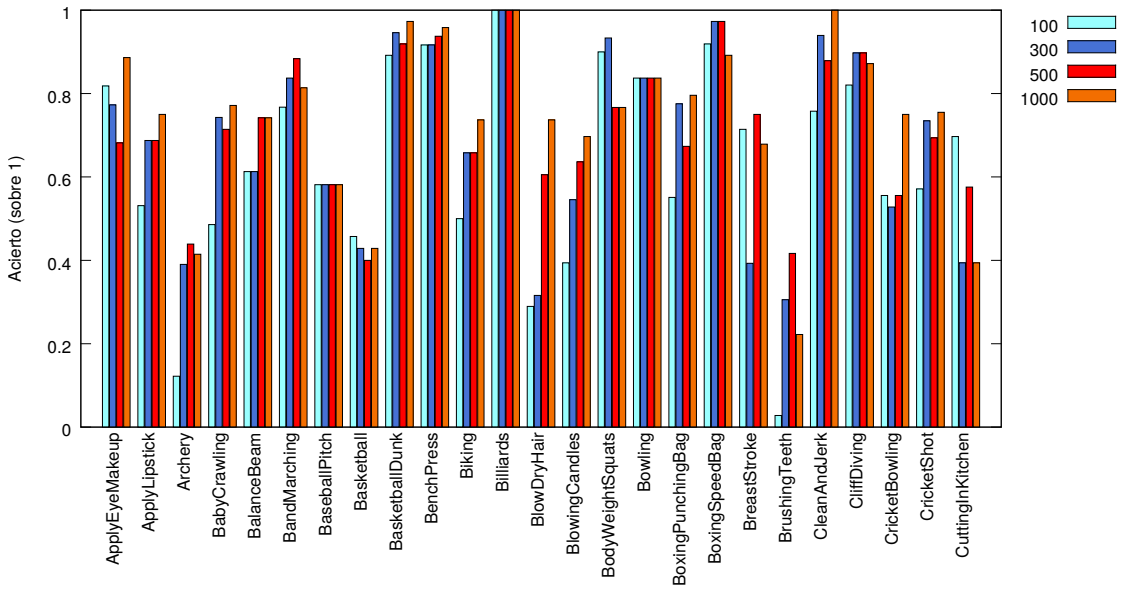


**Figura 3.5:** Resultados a nivel de clase (I)

El siguiente experimento volverá a evaluar la selección de características mediante el criterio de máximo HOF con el objetivo de comprobar el comportamiento de este esquema frente a la selección de 1.000 característica

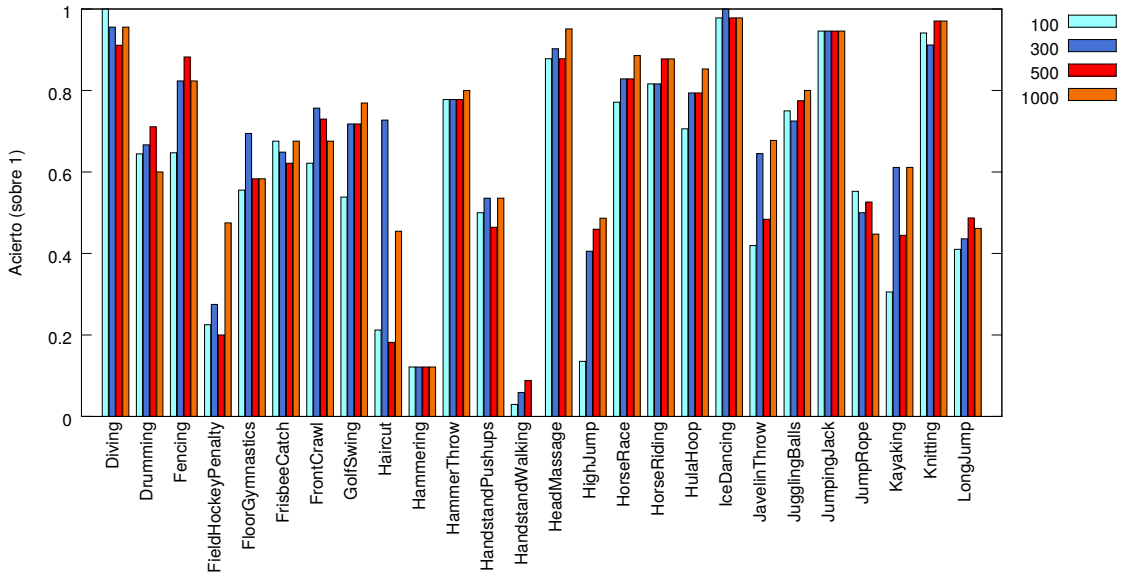


**Figura 3.6:** Resultados a nivel de clase (II)

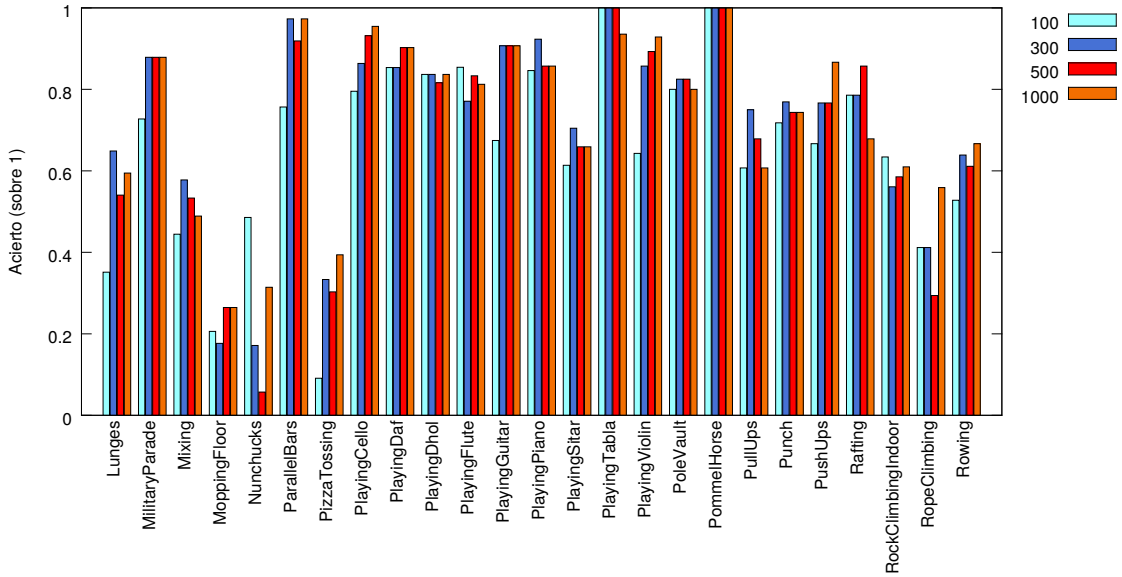


**Figura 3.7:** Resultados a nivel de clase (III)

mediante este criterio, para comprobar si el aumento ha conseguido paliar la diferencia en la precisión entre locales y la fusión. Para ello, se realizará

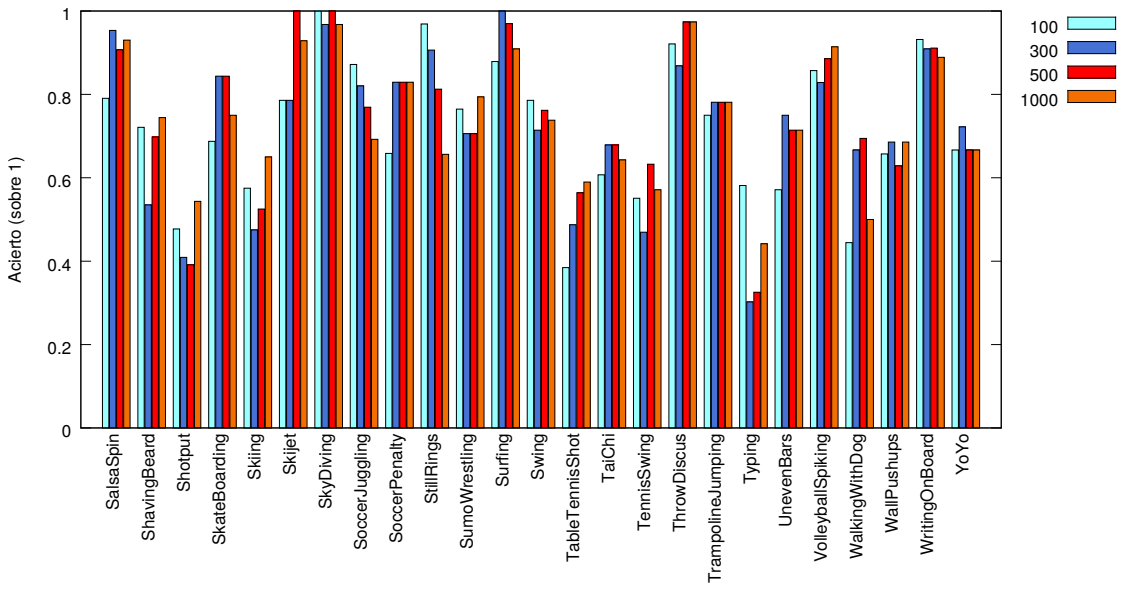


**Figura 3.8:** Resultados a nivel de clase (IV)

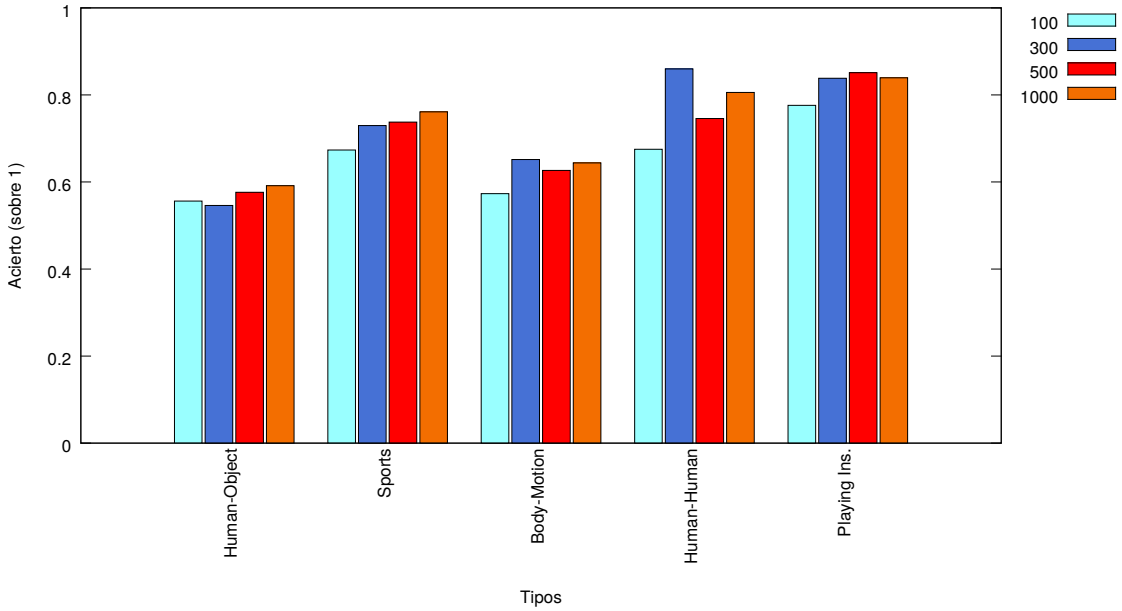


**Figura 3.9:** Resultados a nivel de categoría

el experimento con el mejor modelo hasta el momento, la red descrita en el experimento previo. La Tabla 3.12 refleja los resultados obtenidos.

| Esquema | Prec. % C.L | Prec. % Vot. | Prec. % Sum. |
|---|---|---|---|
| Selección aleatoria | 27.21 | 71.60 | 71.98 |
| Selección mediante HOF | 29.11 | 64.23 | 65.41 |

**Tabla 3.12:** Resultados con la función ReLu, obteniendo 1.000 c.l./vídeo.

El aumento del número de característica por vídeo ha significado un aumento en la precisión a nivel de locales, de 21.56 % a 29.11 %, sin embargo, esta sustancial mejora no ha implicado un incremento en la precisión de las mismas proporciones, aumentando menos de un punto sobre la precisión obtenida en el experimento inicial, pasando de 64.81 % a 65.41 %. De nuevo, este esquema de selección, aunque favorable en cuanto a la precisión de la red neuronal a nivel de las características, no proporciona buenos resultados a nivel de fusión.

Se incluirá en el siguiente experimento la técnica de *DropOut*[16]. Esta técnica favorece el poder de generalización de la red, emulando un entrenamiento promediado entre varias redes "ficticias", mediante la anulación, según cierta probabilidad, de neuronas de la red. Estas anulaciones emulan cambios en la arquitectura que facilita que las neuronas no se especialicen en profundidad en ciertas características del problema, debiendo ocupar el papel de otras cuando éstas son anuladas, favoreciendo la generalización.

Para el experimento que se muestra a continuación, se ha utilizado el mejor modelo hasta el momento a nivel de fusión y se ha establecido un factor de *DropOut* de 0.2 para las neuronas de entrada y 0.5 para las neuronas internas, proporcionando las posibles ventajas de la técnica en la capa de representación y en las capas internas de discriminación, respectivamente. La Tabla 3.13 muestra la precisión obtenida, a nivel de las características y de fusión.

| Capas | Unidades | Prec. % C.L | Prec. % Vot. | Prec. % Sum. |
|---|---|---|---|---|
| 2 | 2048 | 29.82 | 63.09 | 64.86 |

**Tabla 3.13:** Resultados con la función ReLu, obteniendo 1.000 c.l./vídeo, mediante selección aleatoria y con *DropOut*

El poder de generalización de la red se ha visto aumentado, como refleja

el incremento en la precisión a nivel de característica, obteniendo la mejor precisión hasta el momento con un 29.82 %, frente al 27.21 % del modelo con las mismas características pero sin *DropOut*. Sin embargo, la precisión a nivel de fusión se ha visto perjudicada, perdiendo más de 7 puntos, del 71.98 % del modelo previo hasta el 64.86 % obtenido por este modelo.

Tras este experimento, cabría evaluar el esquema de fusión de las locales, con el objetivo de poder enlazar el crecimiento de las dos partes del modelo, la precisión a nivel de características locales y a nivel de fusión, ya que no parece favorecedor obtener mejores resultados a nivel de locales con los esquemas de fusión planteados.

Por otro lado, se ha observado como el esquema de votación, aproximadamente en todos los casos, ha obtenido peores resultados que el esquema de suma de probabilidades *a posteriori*. Este resultado es coherente, dado que establecer el voto de una local mediante la maximización de la probabilidad *a posteriori* para esa característica es una aproximación al máximo de la suma de todas las probabilidades *a posteriori* de las características locales de un vídeo.

Con los resultados obtenidos y el mejor modelo alcanzado, se evaluará a continuación cómo se comporta frente a las restantes particiones propuestas, para obtener finalmente la precisión entre los tres conjuntos, y poder establecer la comparativa con los resultados de la competición de THUMOS 2013. La Tabla 3.14 contiene los resultados finales con el mejor modelo obtenido tras la experimentación. Las Figuras 3.10, 3.11, 3.12 y 3.13 muestran los resultados comparativos a nivel de clase del acierto por cada partición.

### Evaluación final

| Partición | Prec. % C.L | Prec. % Vot. | Prec. % Sum. |
|---|---|---|---|
| 1 | 27.21 | 71.60 | 71.98 |
| 2 | 27.73 | 72.23 | 72.63 |
| 3 | 27.52 | 72.50 | 72.80 |
| Promedio | 27.48 | 72.10 | 72.47 |

**Tabla 3.14:** Resultados con la función ReLu, obteniendo 1.000 c.l./vídeo, mediante selección aleatoria, para las tres particiones propuestas y el promedio final

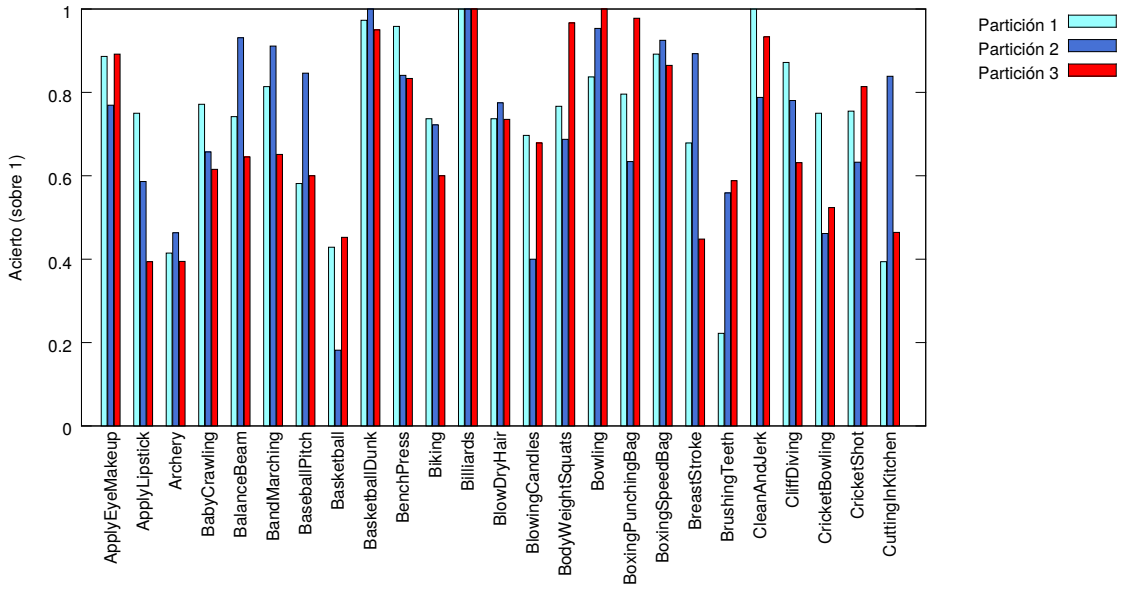


**Figura 3.10:** Resultados a nivel de clase para cada partición (I)

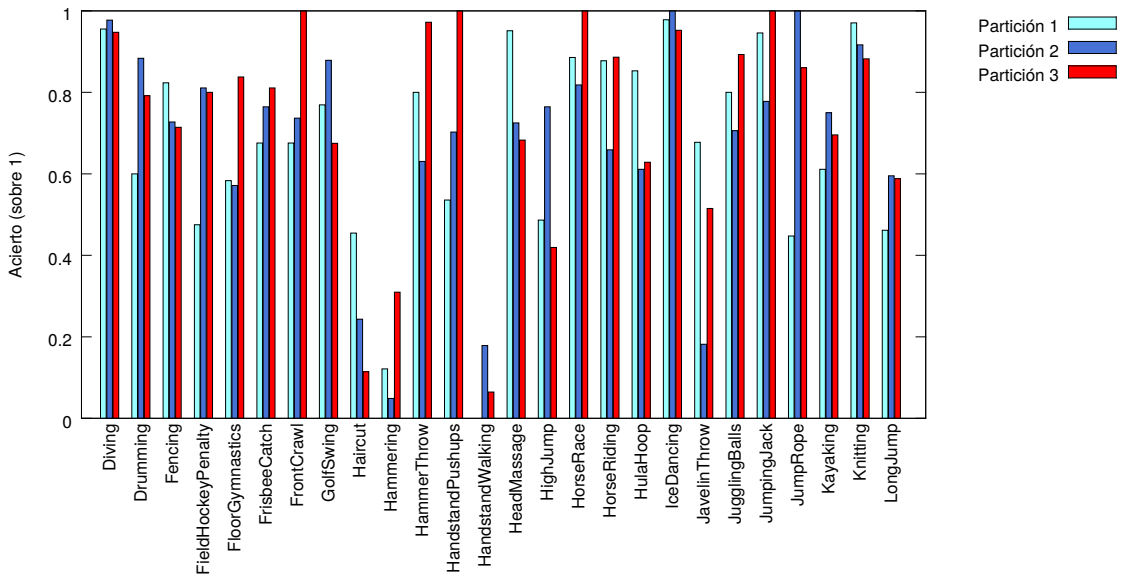


**Figura 3.11:** Resultados a nivel de clase para cada partición (II)

Para evaluar la precisión del modelo desde otro punto de vista, se ha llevado a cabo un experimento para cada partición donde se consideran di-

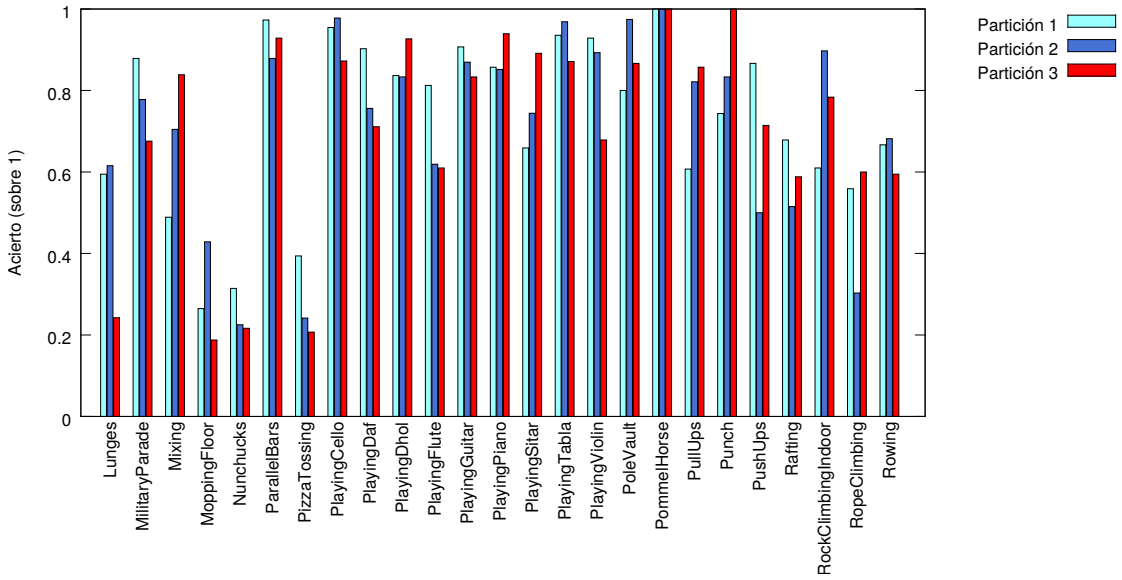


**Figura 3.12:** Resultados a nivel de clase para cada partición (III)

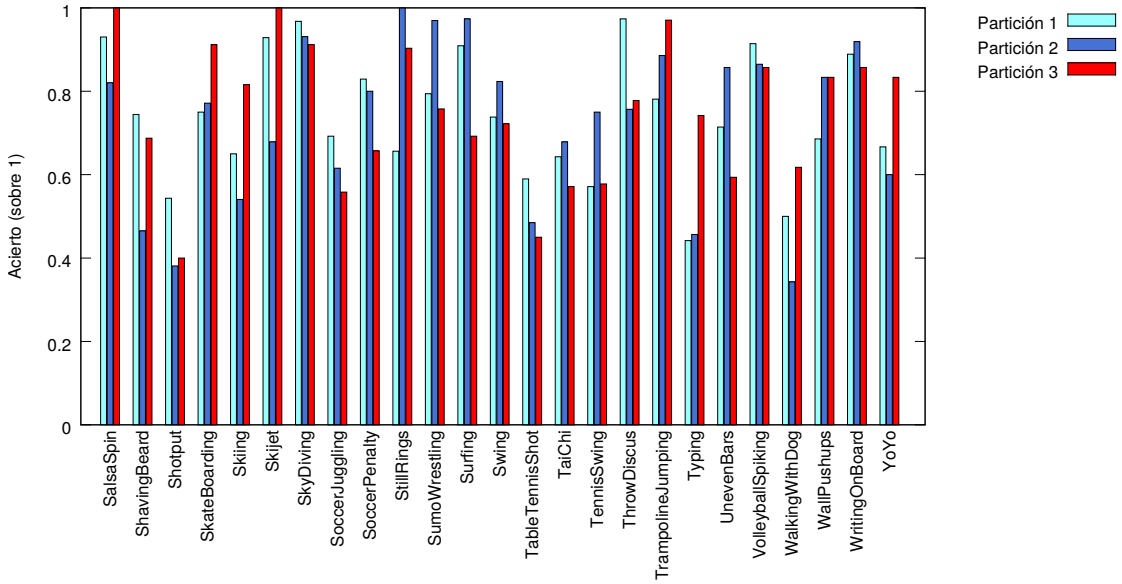


**Figura 3.13:** Resultados a nivel de clase para cada partición (IV)

ferentes *rankings*, es decir, contempla como un acierto si la clase correcta se encuentra entre los $k$ primeras clases predichas. El resultado de estos expe-

rimentos para diferentes valores de $k$ queda reflejado en la Tabla 3.15.

Partición 1

| Top | Acc Sum. % (voting) | Prec. % Sum. |
|---|---|---|
| 1 | 71.19 | 71.89 |
| 2 | 80.51 | 80.83 |
| 3 | 84.24 | 84.82 |
| 4 | 87.02 | 87.23 |
| 5 | 88.60 | 88.55 |

Partición 2

| Top | Acc Sum. % (voting) | Prec. % Sum. |
|---|---|---|
| 1 | 72.23 | 72.63 |
| 2 | 81.22 | 81.65 |
| 3 | 85.85 | 86.39 |
| 4 | 88.69 | 89.04 |
| 5 | 90.32 | 90.40 |

Partición 3

| Top | Acc Sum. % (voting) | Prec. % Sum. |
|---|---|---|
| 1 | 72.50 | 72.80 |
| 2 | 81.97 | 81.97 |
| 3 | 85.95 | 86.54 |
| 4 | 88.41 | 88.95 |
| 5 | 89.87 | 90.25 |

**Tabla 3.15:** Resultados con la función ReLu, obteniendo 1.000 c.l./vídeo, mediante selección aleatoria, mostrando la precisión para diferentes $rank - k$

Por los resultados obtenidos, se puede observar que la clasificación no va muy desencaminada, aumentando aproximadamente nueve puntos la precisión en todos los casos al considerar la segunda opción y proponiendo la clase correcta con 90 % de precisión al considerar las cinco primeras clasificadas en dos de las tres particiones. Esto impulsa de nuevo la idea de trabajar en el esquema de fusión de la salida de la red neuronal, tratando de llevar a cabo algún procesado posterior que facilite al modelo discriminar entre estas opciones, cercanas a la propuesta por el modelo básico, la clase correcta.

Se muestra en la Figura 3.14 el acierto a nivel de clase, ordenado, clasificado mediante el modelo de 2 capas, 2048 neuronas por capa, ReLu y entrenado extrayendo 1.000 características por vídeo. La Figura 3.15 mues-

tra la matriz de confusión de este modelo.

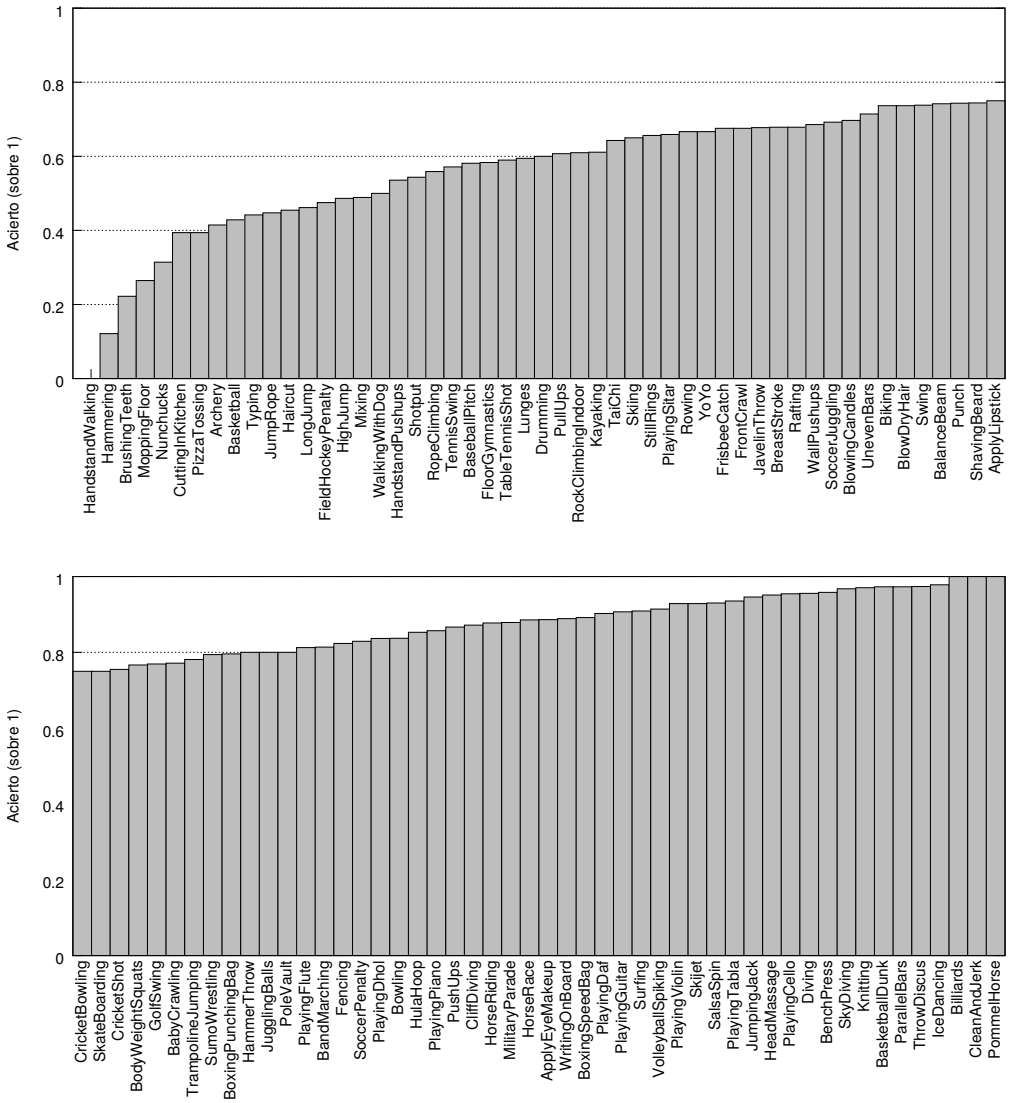


**Figura 3.14:** Resultados a nivel de clases, ordenado.

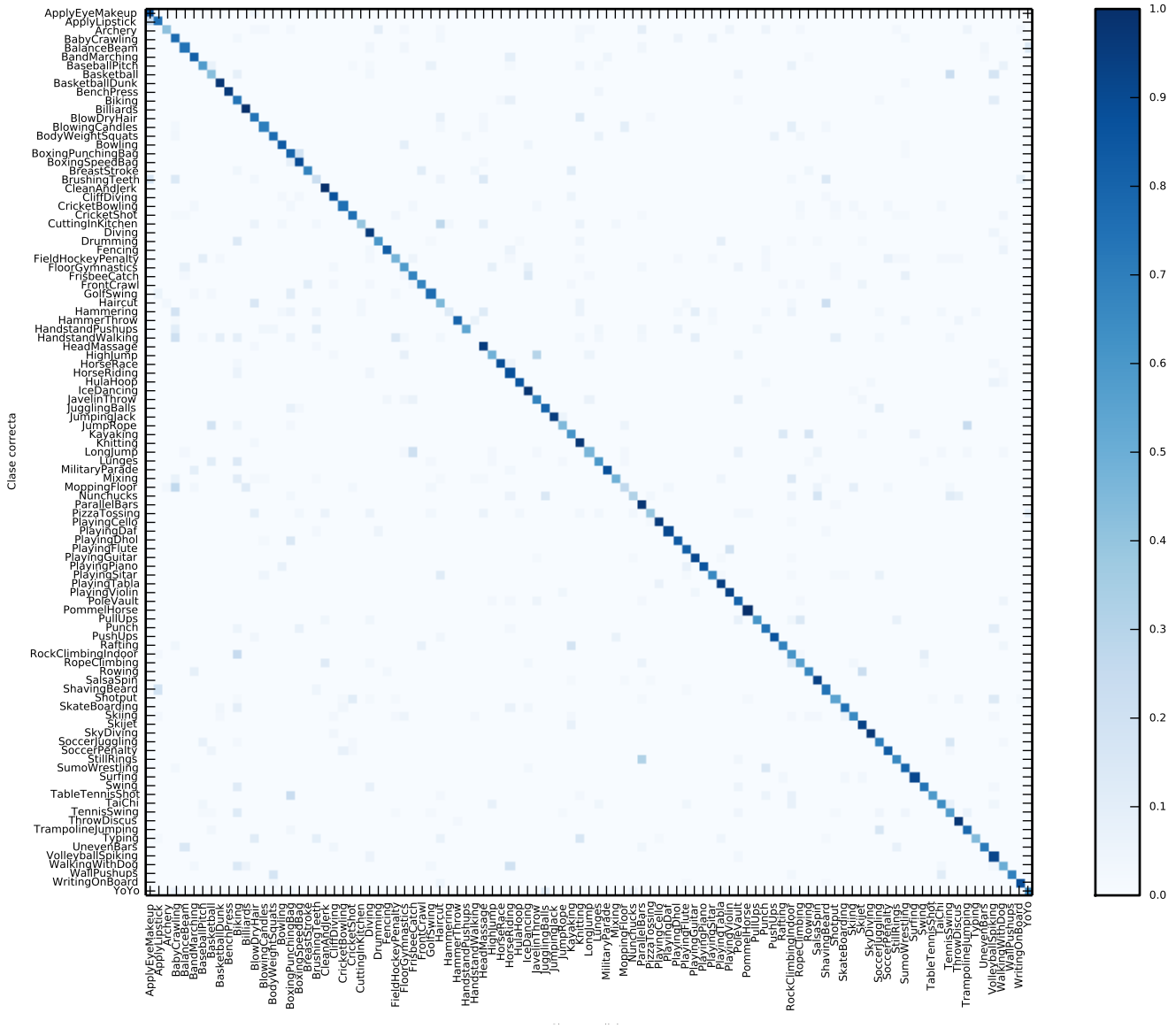


**Figura 3.15:** Matriz de confusión del modelo de 2 capas, 2048 neuronas por capa, ReLu y entrenado extrayendo 1.000 características por vídeo mediante selección aleatoria, en el eje horizontal se encuentran las clases correctas y en el eje vertical las clases predichas.

La Tabla3.16 refleja el resumen de los resultados obtenidos por los 10 primeros equipos. Con la propuesta presentada en este proyecto, el modelo de extracción de características iDT y la clasificación mediante redes neuronales artificiales para las características locales se posicionaría en la décima posición, formando parte del *top-ten*, siendo la única aproximación que utiliza

| Equipo | Resultados |
|---|---|
| INRIA | 85.90 % |
| Florence | 85.50 % |
| Cancerba | 85.43 % |
| CAS-SIAT | 84.16 % |
| Nanjing | 83.97 % |
| UCF_Tappen | 82.82 % |
| UCSD_MSRA_SJTU | 80.89 % |
| USC | 77.36 % |
| NII | 73.38 % |
| **iDT + RNA** | **72.47 %** |

**Tabla 3.16:** Resultados de la edición 2013 de THUMOS[51] y la marca obtenida con la aproximación de este proyecto.

directamente las características de bajo nivel.

### Ampliación del modelo

Para tratar de solucionar el pequeño error evidenciado en los experimentos *rank-k*, con el objetivo de recuperar los vídeos mal clasificados por poco, se ha experimentado con dos ideas relacionadas con la representación proporcionada por la red neuronal antes de llevar a cabo la fusión.

En el primer experimento se trató de llevar a cabo la fusión, mediante la suma de probabilidades *a posteriori* y sometiendo a normalización a todas las características locales, para que cada vídeo quedase respresentado por este vector, y poder aplicar de nuevo un aprendizaje basado en redes neuronales. Sin embargo, dado que en el entrenamiento el modelo aprende bastante bien el problema, deja poco margen para el aprendizaje por parte de la nueva red, y por tanto, los resultados en el test son ligeramente peores. Se experimentó con una red que obtuvo peores resultados en entrenamiento, pero tampoco dió buenos resultados, descartándose esta vía por el momento.

Posteriormente, se evalúo la combinación de redes, de entre las ya entrenadas. Este experimento se llevo a cabo sobre la partición uno, que es la única partición donde se disponía de varios modelos diferenciados. Esto supuso una mejora y los resultados se reflejan en la Tabla 3.17. El modelo que mejores resultados ha obtenido mencionado en los puntos previos, se combinó con otro modelo de la misma arquitectura pero entrenado durante

| Modelos | Prec. % (Sum.) |
|---|---|
| Modelo Básico 1 | 71.89 % |
| MB1 + Modelo Básico 2 (menos iteraciones) | 72.19 % |
| MB1 + MB2 + Modelo Básico (*DropOut*) | 72.86 % |

**Tabla 3.17:** Evaluación del experimento de combinación de las salidas de las redes neuronales entrenadas previamente.

menos iteraciones, aumentando ligeramente la precisión total. Por último, se combinó a estos dos modelos, el modelo del experimento con *DropOut*, incrementando de nuevo ligeramente el acierto global. La combinación se basaba en acumular sobre la misma matriz las diferentes salidas de las redes, y se procedía posteriormente a la fusión mediante la suma de probabilidades a posteriori.

Por otro lado, también se evalúo, entrenar el modelo con pocas características por vídeo, por ejemplo 100, pero en fase de test utilizar la extracción de 1.000. Esto tampoco proporcionó ninguna mejoría.

## 3.5. Discusión

Se ha llevado a cabo la experimentación, dentro de las restricciones temporales del proyecto, con el objetivo de obtener el mejor modelo para la tarea de clasificación de THUMOS 2013. El mejor modelo utiliza la función de activación ReLu, que no ha obtenido mejores resultados inicialmente a nivel de característica local, probablemente por la poca profundidad de la red, a diferencia de otros trabajos [23], pero sin embargo ha obtenido mejor resultado a nivel de fusión en los dos esquemas propuestos.

A diferencia de los trabajos llevados a cabo sobre esta tarea, no se han empleado todas las características obtenidas por la extracción mediante *improved Dense Trajectories*. En lugar de ello, se ha evaluado el modelo según un número de características por vídeo concreto, desde 100 hasta 1.000, comprobándose empirícamente que proporcionar un mayor número de características influye positivamente en la precisión del sistema. El uso directo de las características de bajo nivel no ha sido la propuesta de ningún equipo de entre los 10 primeros de la competición, empleando la codificación mediante *Fisher Vector* para no tener que trabajar directamente con la enorme cantidad de características totales.

Este modo de acotamiento de las características ha proporcionado la oportunidad, de evaluar una heurística de selección durante el proceso de extracción, abriendo otro nuevo horizonte para futuros trabajos donde se evalúen diferentes criterios. Concretamente, se ha evaluado la maximización de la característica HOF, relacionada íntimamente con el movimiento presente en la trayectoria. Esta característica ha demostrado un buen funcionamiento a nivel de local, pero no ha tenido una repercusión positiva en cuanto a la fusión para la clasificación final, por lo tanto finalmente se ha decantado por las características seleccionadas de forma aleatoria. Se ha vuelto sobre esta idea, aumentando el número de características, pero se han mantenido las mismas conclusiones.

Coincidiendo con la literatura, *DropOut* ha proporcionado mayor poder de generalización al modelo. Sin embargo, tampoco esta mejora de la red ha tenido repercusión en la precisión final en la fase de fusión, por lo que se ha descartado su inclusión en el modelo final.

En términos generales, el esquema de fusión que mejor ha funcionado ha sido la suma de probabilidades *a posteriori*, sin embargo, en vista de los resultados, cabe evaluar otros métodos o procesos posteriores con el objetivo de mejorar la etapa de fusión, en lugar de seguir trabajando en aumentar la precisión de la red, que no ha proporcionado buenos resultados a nivel de fusión. Esta idea se ve reforzada por la cercanía de la predicción correcta que ha evidenciado el experimento del $rank - k$.

De igual manera que la mayoría de equipos que participaron en la edicion THUMOS 2013, la primera partición es la que ha obtenido peores resultados, obteniéndose un resultado final de 72.47 % en promedio, que colocaría este modelo entre los diez primeros equipos de la competición.

Cabe destacar que, se han encontrado clases en las que no ha importado el aumento de las características, como por ejemplo "*Playing guitar*" o "*Jumping Jack*", manteniendo la misma precisión. Estos son algunos ejemplos donde las características se concentran en un punto, y posiblemente con la elección de unas pocas, ya se consigue obtener las discriminativas. Por otro lado, los resultados en cuanto a dificultad de los vídeos han coincido casi idénticamente con los mostrados por los organizadores de THUMOS 2013, estableciendo "*Billiards*" como la más fácil, acertada por completo con el modelo planteado, y "*Hammering*" y "*HandStandWalking*" como las más díficiles, coincidiendo de nuevo con lo obtenido en la aproximación plantea-

da. En cuanto a los grupos contemplados el modelo planteado también ha coincidido con determinar como más complejos los vídeos relacionados con *Human-Object Interaction* y *Body-Motion*, respectivamente. Estos conjuntos de vídeos cuentan con menor movimiento que el resto, además de no disponer de elementos tan diferenciadores como los intrumentos o los accesorios de deportes. Por ello, obtener descriptores sobre el movimiento resulta complicado.

Se han evaluado algunas extensiones del modelo, pero no se han obtenido beneficios significativos, sin embargo, en futuros proyectos, resultaría interesante analizar en profundidad qué proceso aplicar tras la fusión, con el objetivo de poder discernir entre los vídeos que se encuentran en la frontera entre varias categorías y que finalmente son clasificados de forma errónea.

# Capítulo 4
# Conclusiones

Para comenzar con la clausura del presente proyecto, se enunciarán los objetivos alcanzados durante el desarrollo del mismo. Se ha llevado a cabo el estudio empírico de la aplicación de redes neuronales artificiales para la tarea de clasificación de vídeos según la actividad que las personas desarrollan en él, obteniéndose resultados y datos que darán pie a continuar con la investigación en esta línea. Por otro lado, se han alcanzado los siguientes objetivos secundarios planteados inicialmente:

- Se ha realizado el análisis del conjunto de datos UCF-101, sobre el que se ha evaluado la técnica.
- Se ha hecho uso de la extracción de características locales de los vídeos mediante técnicas del estado del arte.
- Se ha desarrollado y entrenado un sistema de aprendizaje automático basado en Redes Neuronales Artificiales para la clasificación de las características locales, para aplicar posteriormente un esquema de fusión para la clasificación final.
- Se ha evaluado el sistema desde diferentes puntos de vista, tanto a nivel de categoría como a nivel de conjunto de vídeos.
- Se ha realizado una comparativa con los resultados obtenidos para el mismo conjunto de datos por otros trabajos en el marco de la competición que proponía la tarea, posicionándose en el décimo puesto con un 72.47 % de acierto en promedio.

## 4.1. Trabajos futuros

Como posibles líneas de continuación, se podrían plantear los siguientes puntos:

- Aumento del número de características, continuando con la experimentación aumentando hasta utilizar todas las proporcionadas por la extracción.
- Evaluar el modelo utilizando la codificación proporcionada por los *Fisher Vectors* como paso intermedio antes de la clasificación.
- Combinar el modelo con otros modelos de la literatura, utilizando varias vías, es decir, que varios clasificadores de diferente índole lleven a cabo la tarea y posteriormente se consensúe la clasificación.
- Seguir trabajando en el proceso de fusión, con el objetivo de obtener funciones que den mejor cuenta de los progresos en el entrenamiento de la red neuronal.
- Evaluar otro tipo de característica de bajo nivel, como por ejemplo STIP.

Estas ampliaciones han demostrado buenos resultados en la literatura y resultaría interesante comprobar la respuesta del modelo a estas inclusiones, que no han sido evaluadas por las limitaciones temporales.

# Bibliografía


[1] Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

[3] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1395–1402. IEEE, 2005.

[4] Hakan Boyraz, Syed Zain Masood, Baoyuan Liu, Marshall Tappen, and Hassan Faroosh. Action recognition by weakly-supervised discriminative region localization. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[5] Matteo Bregonzio, Shaogang Gong, and Tao Xiang. Recognising action as clouds of space-time interest points. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1948–1955. IEEE, 2009.

[6] Rizwan Chaudhry, Arunkumar Ravichandran, Georg Hager, and René Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1932–1939. IEEE, 2009.

[7] Oana G Cula and Kristin J Dana. Compact representation of bidirectional texture functions. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–1041. IEEE, 2001.

[8] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8609–8613. IEEE, 2013.

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[10] Trevor Darrell and Alex Pentland. Space-time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 335–340. IEEE, 1993.

[11] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. Springer, 2003.

[12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[13] Dennis Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93(26):429–441, 1946.

[14] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://www.thumos.info/`, 2015.

[15] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[16] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[17] Yuri Ivanov, Aaron F Bobick, et al. Recognition of visual activities and interactions by stochastic parsing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):852–872, 2000.

[18] A. Roshan Zamir J. Liu and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/index.html`, 2014. [Online; accedido 20-Julio-2015].

[19] Tommi Jaakkola, David Haussler, et al. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999.

[20] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[21] Y.-G. Jiang, J. Liu, A. Roshan Zamir, I. Laptev, M. Piccardi, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/ICCV13-Action-Workshop/`, 2013.

[22] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[24] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011.

[25] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[26] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[27] B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.

[28] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[29] Xin Lei, Andrew Senior, Alexander Gruenstein, and Jeffrey Sorensen. Accurate and compact large vocabulary speech recognition on mobile devices. In *INTERSPEECH*, pages 662–665, 2013.

[30] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[31] Volodymyr Mnih. Cudamat: a cuda-based matrix class for python. *Department of Computer Science, University of Toronto, Tech. Rep. UTML TR*, 4, 2009.

[32] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[33] Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 3–8. IEEE, 2002.

[34] Roberto Paredes, JC Pérez, Alfons Juan, and Enrique Vidal. Local representations and a direct voting scheme for face recognition. In *In Workshop on Pattern Recognition in Information Systems*. Citeseer, 2001.

[35] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *arXiv preprint arXiv:1405.4506*, 2014.

[36] Alessandro Prest, Cordelia Schmid, and Vittorio Ferrari. Weakly supervised learning of interactions between humans and objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):601–614, 2012.

[37] Kishore K Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[38] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[39] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[40] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR*

*2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[41] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, pages 357–360. ACM, 2007.

[42] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[43] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[44] Ju Sun, Xiao Wu, Shuicheng Yan, Loong-Fah Cheong, Tat-Seng Chua, and Jintao Li. Hierarchical spatio-temporal context modeling for action recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2004–2011. IEEE, 2009.

[45] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[46] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[47] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, Colorado Springs, United States, June 2011.

[48] 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.

[49] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*, pages 124–1. BMVA Press, 2009.

[50] Zhongwen Xu, Linchao Zhu, Yi Yang, and Alexander G Hauptmann. Uts-cmu at THUMOS 2015. *CVPR THUMOS Challenge*, 2015, 2015.

[51] A. Roshan Zamir I. Laptev M. Piccardi M. Shah Y.-G. Jiang, J. Liu and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes. `http://http://crcv.ucf.edu/ICCV13-Action-Workshop/`, 2013. [Online; accedido 20-Julio-2015].

[52] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992.

Apéndice A

# Parámetros de la extracción *improved Dense Trajectories* y formato de las características iDT

Para el cómputo de los descriptores, hay que especificar una serie de parámetros:

- $S$: *Frame* de comienzo
- $E$: *Frame* de finalización
- $L$: Longitud de la trayectoria.
- $W$: Desplazamiento para el *dense sampling*.
- $N$: Tamaño del vecindario.
- $s$: Número de celdas en el eje $XY$.
- $t$: Número de celdas en el eje temporal.

Los descriptores obtenidos mediante esta técnica están compuestos por las siguientes características, divididas entre las relacionadas con la trayectoria y las relacionadas con las características HOG,HOF y MBH. En cuanto a la trayectoria:

- Número de *frame*: *Frame* donde termina la trayectoria.
- Media $X$: Valor medio de la trayectoria en el eje $X$.

- Media $Y$: Valor medio de la trayectoria en el eje $Y$.
- Varianza $X$: Varianza de la trayectoria en el eje $X$.
- Varianza $Y$: Varianza de la trayectoria en el eje $Y$.
- Longitud: Duración en el eje temporal de la trayectoria.
- Escala: Escala espacial en la que ha sido computada.
- Posición $X$: Posición normalizada respecto al vídeo.
- Posición $Y$: Posición normalizada respecto al vídeo.
- Posición $T$: Posición normalizada respecto al vídeo.

En cuanto a la forma de la trayectoria, HOG, HOF y MBH:

- Trayectoria: Información relativa a la forma de la trayectoria
- HOG: Características HOG con 8 *bins*, dando lugar a una dimensión de $8 \cdot s \cdot s \cdot t$
- HOF: Características HOF con 9 *bins*, dando lugar a una dimensión de $9 \cdot s \cdot s \cdot t$
- MBHx: Características MBH en el eje $X$ con 8 *bins*, dando lugar a una dimensión de $8 \cdot s \cdot s \cdot t$
- MBHy: Características MBH en el eje $Y$ con 8 *bins*, dando lugar a una dimensión de $8 \cdot s \cdot s \cdot t$