
# Universidad Nacional Autónoma de México

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

## Clasificación multi-etiqueta de videos cortos usando unidades recurrentes reguladas

Que para optar por el grado de:

**Maestra en Ciencias de la Computación**

Presenta:

**Berenice Montalvo Lezama**

Tutor:

Dr. Gibran Fuentes Pineda
IIMAS-UNAM

CDMX, MÉXICO. ENERO 2018

# Índice general

# Resumen

Las redes neuronales con arquitecturas profundas se han aplicado principalmente a problemas de reconocimiento de imágenes obteniendo los mejores resultados en tareas como clasificación, detección o segmentación. Alentados por estos resultados, en este trabajo se proponen y evalúan diferentes arquitecturas de redes neuronales profundas para realizar clasificación multi-etiqueta de videos en una nueva base de datos creada para este proyecto llamada Trailers15k. Esta base de datos contiene 15,000 videos con 10 categorías diferentes que corresponden a géneros de películas. Primeramente, se estudia la clasificación considerando únicamente las características espaciales de los cuadros de video. Para esto, se presentan dos arquitecturas de redes convolucionales que emplean la técnica de transferencia de conocimiento. Posteriormente, se estudia la clasificación incorporando características espacio-temporales. Para esto, se presentan dos arquitecturas basadas en una red convolucional que emplea la técnica de transferencia de conocimiento y unidades recurrentes reguladas. Los resultados obtenidos con estas últimas arquitecturas muestran un rendimiento superior en comparación con las arquitecturas que hacen uso únicamente de las características espaciales de los cuadros de video.

# Agradecimientos

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a las personas que me han acompañado para hacer posible el presente trabajo.

En primer lugar quiero agradecer a la Universidad Nacional Autónoma de México y en especial al IIMAS por brindarme la oportunidad de realizar mis estudios de maestría.

A mi madre por brindarme su comprensión y apoyo incondicional en todo momento. A mis hermanos por ser el mejor ejemplo a seguir.

Agradezco de manera muy especial al Dr. Gibran Fuentes Pineda por su orientación y supervisión constante. Sus ideas y sugerencias fueron un aporte invaluable para terminar esta Tesis.

Esta Investigación fue realizada gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM IA104016 Generación de resúmenes de video basado en redes neuronales profundas de principio a fin. Agradezco a la DGAPA-UNAM la beca recibida.

# Índice de figuras

# Índice de tablas

# 1 Introducción

Las imágenes y videos se han vuelto vitales en la Internet. En general, el contenido multimedia ha crecido exponencialmente en los últimos años. Gracias a los teléfonos inteligentes y dispositivos móviles, se ha hecho posible que las personas se conviertan en creadores de contenidos multimedia, en lugar de ser solo consumidores. Específicamente, los videos han ganado relevancia hasta convertirse en uno de los contenidos predilectos. Su función principal es enriquecer y hacer más atractivo todo aquel material que pueda ser relevante para el usuario. YouTube® se ha convertido en la plataforma por excelencia para difundir este tipo de contenido y actualmente aloja millones de videos con temáticas muy diversas, que van desde tutoriales, avances cinematográficos, *videoblogs* y noticias hasta documentales, series, videos deportivos y musicales. Sin embargo, existen muchas otras plataformas destinadas a compartir videos, algunas de ellas son: `Vimeo`, `Dailymotion`, y `Tu.tv`. Asimismo, existen aplicaciones para la transmisión de videos en vivo como Snapchat®, Facebook Live®, Periscope® e Instagram Live®, que cada vez son más comunes entre los usuarios de la Internet. Aunado a ello, el fácil acceso a los servicios de almacenamiento masivo, como Dropbox®, Google Drive® y One Drive®, ha permitido que poco a poco prolifere más este tipo de contenido. Cisco®, una empresa líder en dispositivos y redes de computadoras, presentó un informe en el 2017 de predicciones del tráfico global en la Internet para el periodo comprendido entre 2016 y 2021 [1]. Su estudio reveló que el tráfico global en 2021 será equivalente a 127 veces el volumen que fue en 2005. También reveló que los videos son una tendencia que continúa en pleno crecimiento y se estima que para el año 2021 el 82 % del tráfico a nivel mundial será video. La Figura 1.1 muestra la estimación del crecimiento del tráfico de aplicaciones de manera mensual en la Internet para el periodo de 2016 a 2021. De la gráfica se puede apreciar que desde el año 2016 el video es el tipo de contenido dominante en la red.

Debido al crecimiento exponencial de los videos, surge también la necesidad de desarrollar métodos que de forma automática permitan analizar el enorme volumen de datos disponible. Desde hace varias décadas el análisis automático de imágenes ha sido ampliamente estudiado y actualmente existen sistemas de reconocimiento, segmentación, detección y descripción [2, 3, 4, 5], que se utilizan exitosamente en productos comerciales. Por ejemplo, la aplicación informática de intercambio de fotografía Google Photos®, de acuerdo a los metadatos que tiene cada imagen y según el contenido, es capaz de organizarlas por ubicación, fecha o por lo que se ha fotografiado. Sin embargo, el análisis de video es una tarea que ha sido menos explorada y presenta aún más retos en comparación al análisis de imágenes debido a que los videos agregan una componente temporal, además de que demandan mayor capacidad de procesamiento y almacenamiento [6, p. 345]. Más específicamente, ha

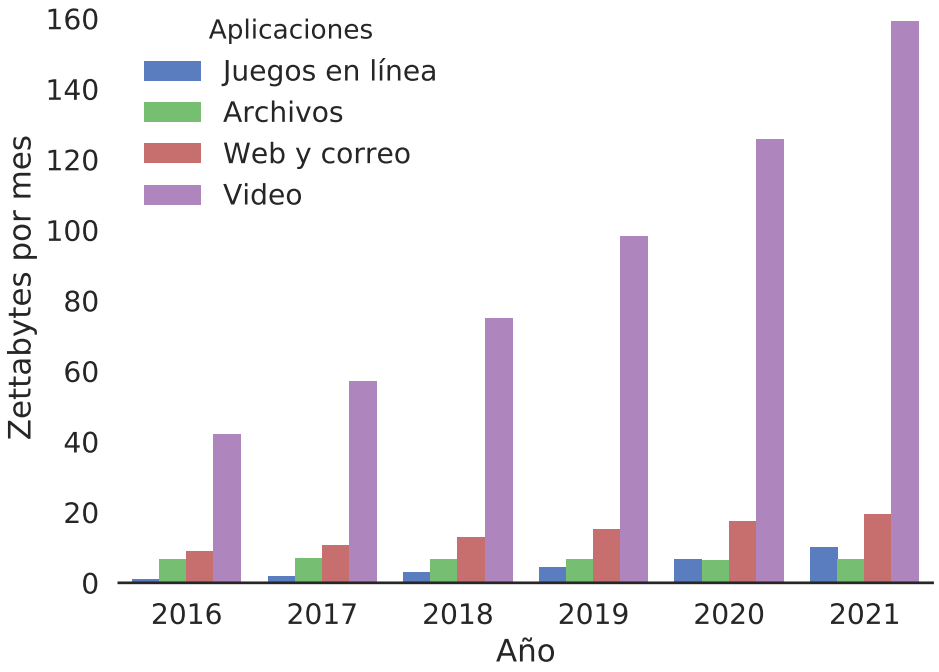



Figura 1.1. Estimación del crecimiento del tráfico mensual (Zettabytes por mes) en la Internet por aplicación para el periodo de 2016 a 2021. Fuente Cisco®[1]

surgido la necesidad de organizar de forma automática los videos basados en su contenido, para proporcionar navegación, búsqueda y recuperación eficiente de este tipo de material.

En este capítulo se plantean los objetivos y la motivación del proyecto de investigación para la tarea de clasificación multi-etiqueta de avances cinematográficos. Además, se describen los principales desafíos que enfrenta la clasificación de video y la contribución del proyecto de investigación.

## 1.1. Planteamiento del problema

La clasificación es uno de los problemas más relevantes en el área de aprendizaje automático. La tarea consiste en determinar una o varias etiquetas de clase para un ejemplo de entrada de un conjunto de datos [7, p. 3]. Cuando se tienen únicamente dos clases se trata de clasificación binaria, de caso contrario el problema se conoce como clasificación multi-clase. La clasificación multi-clase puede ser de dos tipos. Cuando se tiene una entrada que debe estar asociada a una etiqueta exactamente, se denomina clasificación uni-etiqueta. Un ejemplo de este tipo de clasificación se muestra en la Figura 1.2, donde la entrada es una imagen que pasa a un clasificador que otorga exactamente una clase. Por otra parte, la clasificación multi-etiqueta se da cuando el ejemplo de entrada puede estar asociado a más de una etiqueta de clase como se muestra en la Figura 1.3, donde la entrada es la misma imagen de la


Figura 1.2. Ejemplo de clasificación uni-etiqueta. La clase en color azul es la clasificación otorgada para la imagen de entrada.

Figura 1.2 pero en este caso es asociada a dos clases.

Para definir formalmente el problema consideremos un conjunto de datos de $m$ ejemplos representados por parejas $(\mathbf{x}_i, \mathbf{y}_i)$ y un conjunto de clases $C$ con más de dos elementos. La tarea de clasificación multi-clase [8, p. 100] consiste en el aprendizaje de una función $f\colon \mathbb{R}^n \to C^k$ que tome como entrada las características $\mathbf{x}_i$ e infiera un subconjunto $\mathbf{y}_i$ de $k$ clases en $C$. Cuando $\mathbf{y}_i$ consiste siempre de una sola clase se dice que se trata de clasificación uni-etiqueta, mientras que cuando $\mathbf{y}_i$ consiste de al menos una clase se dice que se trata de clasificación multi-etiqueta.

En la actualidad, los sistemas basados en características aprendidas por medio de modelos de arquitectura profunda han mostrado excelentes resultados en el análisis de imágenes. Particularmente, las redes neuronales convolucionales (CNN, por sus siglas en inglés) se han convertido en el método más exitoso para las tareas de clasificación y detección de imágenes [3, 9], pues han logrado tasas de reconocimiento sorprendentes. Incluso en tareas muy específicas, los resultados han igualado o superado los de un humano [10, 11, 12, 13]. Esto ha sido posible en cierta medida por las grandes bases de datos etiquetadas disponibles para el entrenamiento, así como también por el incremento en la capacidad de cómputo, ya que las CNN requieren largos períodos de entrenamiento para optimizar la gran cantidad de pesos que emplean. Sin embargo, la clasificación de video puede considerarse un problema abierto. A pesar de que se han querido extender los mismos modelos empleados para la clasificación de imágenes no se han obtenido resultados comparables, pues estos modelos están diseñados para aprender las características 2D en imágenes. Por ello, se necesitan de modelos más complejos que permitan capturar las características espacio-temporales y analizar el impacto que estas tienen en la tarea de clasificación.


Figura 1.3. Ejemplo de clasificación multi-etiqueta. Las clases en color azul son la clasificación otorgada para la imagen de entrada.

En particular, para la industria del entretenimiento, las películas, son de gran importancia. Existen gran cantidad de sitios web en los cuales se encuentra disponible la película completa y además su avance cinematográfico. Los géneros cinematográficos son las categorías en que se clasifican formalmente las películas según sus características temáticas y narrativas. La clasificación de género es fundamental para la distribución y promoción comercial de las películas. Sin embargo, hasta ahora, la clasificación se realiza de forma manual y el proceso no tiene estandarización. El problema a tratar en el presente trabajo es la clasificación de avances de películas con base en su género. La clasificación de avances es un problema multi-etiqueta debido a que en la mayoría de los casos una película se encuentra asociada a más de un género.

## 1.2. Objetivos

### Objetivos generales

El objetivo principal del proyecto es diseñar y desarrollar un modelo de clasificación multi-etiqueta de video, incorporando características espacio-temporales. En particular, el modelo será empleado para clasificar avances de películas de acuerdo a su género.

### Objetivos específicos

- Revisar y analizar el estado del arte en la tarea de clasificación de video.
- Recopilar una base de datos multi-etiqueta de avances de cinematográficos.
- Preprocesar la base de datos recopilada.
- Diseñar y desarrollar diversos modelos que permitan realizar clasificación multi-etiqueta de los videos de la base de datos recopilada, sin incorporar la componente temporal.
- Proponer e implementar diversos modelos que permitan realizar clasificación multi-etiqueta de la base de datos recopilada, incorporando las características espacio-temporales.
- Analizar el impacto que la componente temporal tiene en la clasificación de los videos recopilados de avances de cinematográficos con base en los modelos propuestos.

## 1.3. Retos al trabajar con videos

Trabajar con video presenta diversos retos. En particular, cuando de clasificación se trata existen ciertos desafíos, los cuales se detallan a continuación:

- Obtener una base de datos de videos para el análisis. Como se mencionó, el éxito de las redes neuronales convolucionales para el reconocimiento de imágenes ha sido posible hasta cierto punto gracias a las enormes bases de datos etiquetadas disponibles como ImageNet [14]. ImageNet ha permitido entrenar las arquitecturas más exitosas y representativas de las redes neuronales convolucionales como son Inception-v3 [15] y ResNet [16]. El papel que los conjuntos de datos juegan en tareas como clasificación es muy importante dado que cuando se tiene una porción considerable de datos que son representativos para la tarea en cuestión, es más sencillo que los modelos de aprendizaje profundo aprendan las características relevantes a la tarea y con ello puedan obtener un buen rendimiento. No obstante, una de las limitantes para el avance en análisis de video ha sido encontrar una base de datos a gran escala, pues, a pesar de que existe una enorme cantidad de videos en la Internet, en su mayoría no se encuentran etiquetados. Por otra parte, construir una base de datos de video a gran escala es muy laborioso. Uno de los factores que más influyen cuando se requiere de recolectar una base de datos de video son los requerimientos de almacenamiento, ya que, dependiendo de la cantidad de datos y la complejidad de ellos, se pueden necesitar desde el orden de los *gigabytes* hasta los *terabytes*. Otra de las dificultades que se presenta es la accesibilidad de los datos, debido a que existen múltiples plataformas que carecen de mecanismos para proveerlos y, aunque existen métodos para acceder a ellos, se requiere de un mayor esfuerzo para obtenerlos. Además, existen limitantes como la privacidad o propiedad de los datos. La obtención de las etiquetas es otro de los factores que intervienen en la recolección de una base de datos, puesto que etiquetar miles de videos requiere de mucho tiempo y en muchos casos no es factible.

- Complejidad del contenido del video. Un aspecto muy importante en clasificación de videos es el dominio de entrada, es decir, el tipo de video que se está clasificando. De manera general en imágenes, se dice que el dominio de entrada varía entre dominios estrechos y amplios. De acuerdo a Smeulders [17] "Un dominio estrecho tiene una variabilidad limitada y predecible en todos los aspectos relevantes de su apariencia". Es decir, cuando se trata de este tipo de dominio el contenido semántico está bien definido y esta propiedad se utiliza para diseñar sistemas muy precisos. Por ejemplo, los videos destinados a la inspección médica y diagnostico médico, así como también los industriales como los empleados en líneas de manufactura, son considerados de dominio estrecho dado que se dan bajo condiciones deliberadamente controladas. Por otro lado, de acuerdo a Smeulders [17] "Un dominio amplio tiene una variabilidad ilimitada e imprevisible en su apariencia incluso para el mismo significado semántico". Los avances cinematográficos se pueden considerar de dominio amplio, ya que se caracterizan por cambios bruscos de escenas con la intención de impactar en el público, lo que dificulta su análisis.

- Captura de características espacio temporales. La clasificación de videos presenta desafíos importantes debido a que posee características adicionales e interesantes a las presentes en imágenes. Las redes convolucionales 2D tienen

un buen desempeño en el análisis de imágenes puesto que están diseñadas para aprovechar su estructura bidimensional. Y si bien inicialmente un video podría considerarse como un conjunto de imágenes aisladas, la temporalidad presente en los videos juega un papel muy importante como se ilustra en la Figura 1.4. La primera columna muestra un conjunto de cuadros extraídos de un avance cinematográfico en orden cronológico, la segunda muestra una breve descripción del contenido de cada uno de ellos y la tercera presenta una deducción del género para el cuadro actual con base en su contenido y el de los cuadros previos. Es decir, se muestra la evolución de la percepción del género del avance cinematográfico a medida que se considera cada vez mayor información con los cuadros de video. Este ejemplo remarca la importancia de la evolución temporal para determinar una clasificación adecuada de un video, pues no basta con analizar un solo cuadro. Si se considera únicamente el segundo cuadro para otorgar una clasificación al video completo se podría asumir que el género es comedia. No obstante, es la evolución de la historia completa la que puede determinar una clasificación de género más apropiada. Por lo tanto, es necesario desarrollar métodos robustos que permitan aprovechar las características espacio-temporales del video. Estas se componen de las características espaciales en los cuadros de video y las relaciones entre estos generadas por su orden secuencial. La captura de estas características puede aportar mayor información a la tarea de clasificación, lo cual podría tener un impacto importante al determinar la clase a la cual pertenece el video. A pesar de que han existido trabajos en clasificación de video [18, 19], estos han sido uni-etiqueta y aún no es clara la importancia de la componente temporal en la tarea de clasificación.

- Recursos computacionales. Los modelos de aprendizaje profundo, como las redes convolucionales, son modelos con requisitos computacionales considerables. Dependiendo de la complejidad del modelo y el tamaño del conjunto de datos pueden tardar meses de entrenamiento. Las unidades de procesamiento gráfico (GPU, por sus siglas en inglés) han contribuido considerablemente al avance de modelos de aprendizaje profundo, pues aceleran el procesamiento. Se han desarrollado bibliotecas generales que permiten explotar las capacidades de cómputo de las tarjetas gráficas exponiendo un modelo de programación para implementar modelos con arquitecturas profundas [20, 21]. Asimismo, se han desarrollado bibliotecas específicas para redes convolucionales [22, 23, 24] y para redes recurrentes [25]. Por otra parte, estos modelos requieren de una buena cantidad de almacenamiento, puesto que como se menciona en secciones anteriores es necesario de una gran cantidad de datos para aprender características complejas y con ello obtener buenos rendimientos. El contar con los dos aspectos previamente mencionados es muy favorable cuando se trata de procesar y almacenar un conjunto de datos de video a gran escala incluso cuando se trata de videoclips cortos, puesto que cada uno de ellos contiene cientos de cuadros de video, lo cual aumenta considerablemente la demanda de recursos computacionales

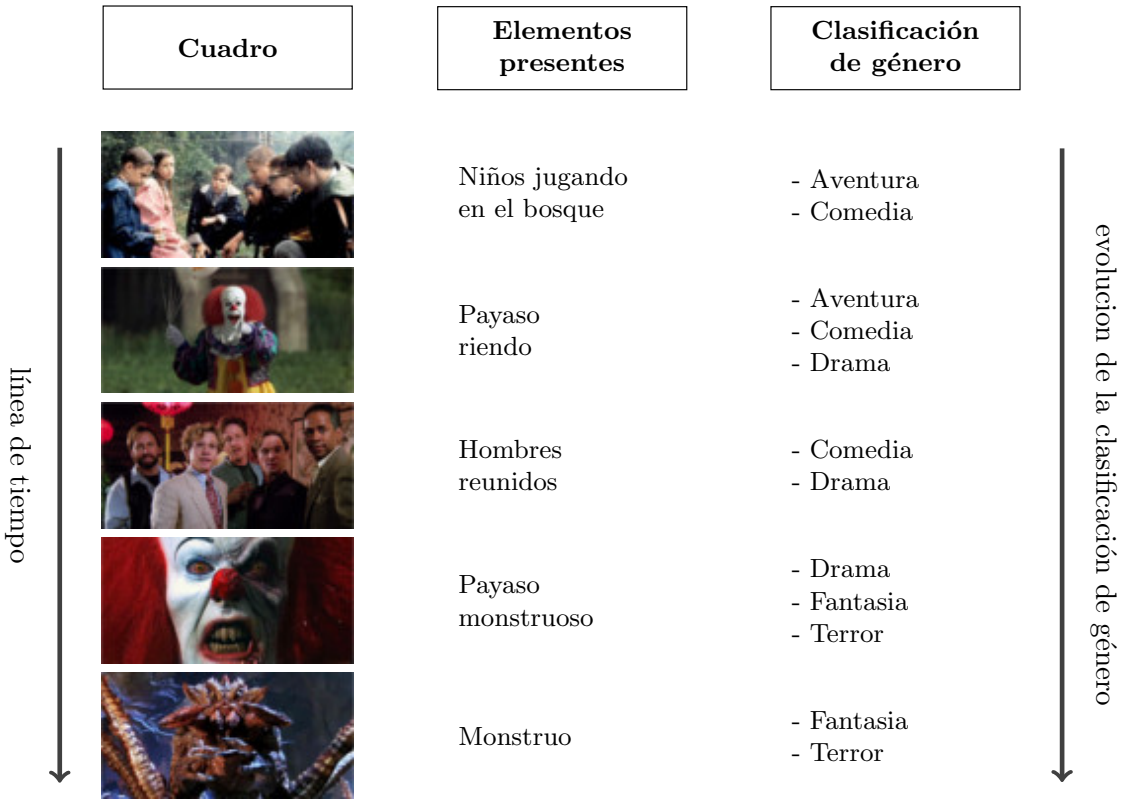



Figura 1.4. Ejemplo de la evolución en la percepción de género con base en los cuadros de un video.

## 1.4. Metodología

La metodología empleada para la clasificación de los avances cinematográficos considerando únicamente las características espaciales de los cuadros de video consistió en diferentes arquitecturas con una red neuronal convolucional pre-entrenada debido a que este tipo de redes están diseñadas para aprovechar las características de las imágenes. Por otro lado, la metodología usada para aprovechar las características espacio-temporales del video y con ello que aprender una descripción global de la evolución temporal, consistió en una combinación de una red neuronal convolucional pre-entrenada y una red neuronal recurrente (RNN, por sus siglas en inglés). Se usó una red recurrente debido a que hace uso de la información secuencial, pues incluye la capacidad de mantener la memoria interna con retroalimentación y, por lo tanto, respaldar el comportamiento temporal. En particular, se usó un tipo de red neuronal recurrente llamada Unidad Recurrente Regulada (GRU, por sus siglas en inglés).

## 1.5. Aportes de la investigación

Los aportes de este proyecto de investigación son los siguientes:

- Modelos de redes neuronales profundas para clasificación de video multi-etiqueta. Los modelos propuestos en este proyecto consisten principalmente en dos tipos. Los primeros son modelos para realizar la clasificación multi-etiqueta de videos sin considerar la componente temporal, es decir, en estas arquitecturas se explotan las características espaciales de los cuadros de video. Los segundos modelos propuestos incorporan la componente temporal para realizar la tarea de clasificación.

- Análisis del impacto de la componente temporal en la clasificación de video. Se analiza con base en los modelos propuestos y para la base de datos de avances cinematográficos, el impacto que la componente temporal tiene en la tarea de clasificación.

- Base de datos multi-etiqueta de avances cinematográficos. Para este proyecto se recolectó una base de datos de avances cinematográficos. La base fue recopilada, dado que al inicio del desarrollo de este proyecto no existían muchas bases de datos de video que estuvieran disponibles para el análisis y aquellas disponibles poseían tan solo cientos de videos. Además, en su mayoría eran conjuntos de videos de acciones, los cuales, si bien permiten realizar un análisis, este sería limitado. La base de datos recopilada se hará disponible a la comunidad de investigación para que pueda ser utilizada para profundizar aun más en el análisis de video.

## 1.6. Organización de la tesis

Esta tesis se encuentra organizada en 7 capítulos, cada uno de ellos se describe brevemente a continuación:

- Capítulo 2: se discute el estado del arte de los métodos de clasificación de video. Además, se describen las bases de video existentes disponibles.

- Capítulo 3: se describen los fundamentos para el proyecto. Se describen las redes perceptrón multicapa, las redes neuronales convolucionales y las redes neuronales recurrentes .

- Capítulo 4: se describe el proceso de recopilación de la base de datos de avances cinematográficos y se muestran estadísticas elementales de los videos de acuerdo a los 10 géneros diferentes. Además se proporcionan características específicas de los videos en la base de datos como son formato, resolución, duración, etcétera.

- Capítulo 5: se describen los dos modelos propuestos para clasificación de video multi-etiqueta.

- Capítulo 6: se describen las métricas empleadas para problemas de clasificación multi-etiqueta. Además, se muestran los resultados de los experimentos realizados con cada uno de los modelos propuestos en el Capítulo 5

- Capítulo 7: se concluye realizando una comparación de los modelos propuestos, primeramente entre aquellos que no incorporan la componente temporal, después entre los modelos que si la incorporan y finalmente se realiza una comparación entre el mejor modelo con y sin componente temporal. Además, se sugieren posibles direcciones del trabajo a futuro en la clasificación multietiqueta de video.

# 2 Estado del arte

En este capítulo se analizan los antecedentes en el área de clasificación de video y las bases de datos de video disponibles. Primeramente se presentan de forma general los métodos de extracción de características locales que han sido empleados para el análisis de video. Posteriormente, se realiza un estudio de las nuevas técnicas basadas en aprendizaje de características en la tarea de clasificación de video y los trabajos relacionados con la clasificación de avances de películas. Finalmente, se presentan las bases de video existentes hasta el momento y sus principales características.

## 2.1. Clasificación usando características locales

La investigación en reconocimiento de video ha sido impulsada en gran medida por los avances en los métodos de reconocimiento de imágenes que han sido adaptados y ampliados para que sean capaces de lidiar con los videos. Se han realizado varios avances en la investigación de clasificación de video empleando métodos tradicionales del área de visión computacional. En particular estas técnicas han mostrado avances en las tareas como reconocimiento de escenas [26, 27] y acciones [28, 29, 30]. El enfoque tradicional para la clasificación del contenido de video se ilustra en la Figura 2.1 .

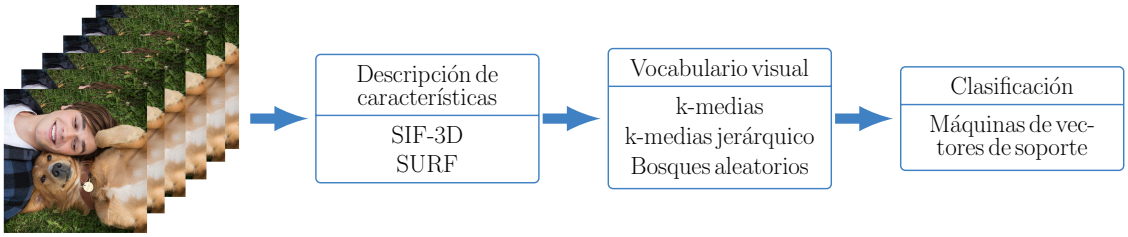



Figura 2.1. Etapas en la clasificación de video usando extracción de características locales.

Esta metodología implica principalmente tres etapas:

- Detección y descripción de características locales.
- Construcción de un vocabulario visual.
- Clasificación con máquinas de vectores de soporte.

### 2.1.1. Detección y descripción de características locales.

El principal objetivo de la detección y descripción de características locales en una imagen es proveer una representación general y compacta de la información contenida en dicha imagen, de manera que esta pueda ser utilizada para la detección de objetos y clasificación. Las características locales son patrones o estructuras distintivas que se encuentran en una imagen, como un punto, borde o una textura. Por lo general, son asociadas con uno o varios cambios en las propiedades de la imagen que difieren de su entorno por textura, color o intensidad. Encontrar correspondencias visuales entre imágenes se divide en dos partes importantes. Primeramente se detectan puntos de interés en la imagen y después se describen dichos puntos de interés.

Los detectores de características se encargan de encontrar regiones sobresalientes en la imagen. Un buen detector debe ser robusto a las perturbaciones y transformaciones. Esto último garantiza que dichos puntos se puedan detectar de forma confiable y con mucha exactitud al realizar esta tarea de forma repetitiva. Algunas de estas perturbaciones mencionadas son las rotaciones, el escalamiento y los cambios de iluminación. Existen diversos algoritmos de detección de características que se basan en la detección de esquinas, como el algoritmo Harris [31], que es invariante a las rotaciones, pero no al escalamineto. Por otra parte, el algoritmo de Harris-Laplace [32] es un algoritmo que es invariante a rotación y escalamiento. Uno de los detectores empleados en el análisis de video es Harris 3D [33], el cual se utiliza para detectar esquinas espaciales y temporales que pueden detectar en una secuencia de cuadros. Este detector encuentra esquinas espaciales con cambio de velocidad en el espacio y en el tiempo.

Una vez que se encuentra un conjunto distintivo de puntos de interés con características contrastantes, es posible construir un descriptor para cada uno de ellos. Un descriptor es una abstracción de las características visuales elementales que se encuentran contenidas en las imágenes. Los descriptores proporcionan un vector que contiene la información codificada de cada uno de los puntos de interés encontrados en la imagen. Un buen descriptor debe ser capaz de capturar la información más importante y distintiva de una región sobresaliente de manera que, si la misma estructura es detectada posteriormente, pueda ser reconocida. Algunos de los descriptores más empleados en clasificación de video son HOG [34, 35] y ST-MP7EH [36]. Por otra parte, existen algoritmos como 3D-SIFT que es un detector y descriptor de características locales que es invariante tanto a la rotación como al escalamiento [37].

### 2.1.2. Construcción de un vocabulario visual.

Una vez obtenidos los vectores de características de las imágenes, se combinan en una descripción a nivel de video. La manera clásica de transformar un conjunto de descriptores visuales locales en un solo vector de longitud fija es mediante el uso de un vocabulario visual. Es decir, una vez que todos los puntos de interés de las imágenes han sido descritos, se genera un vocabulario representativo de las

características que aparecen en los datos. Esto se logra con métodos de agrupamiento que organizan los datos en ciertos grupos que corresponden a palabras visuales de dicho vocabulario. Cada palabra visual, por lo tanto, representa la caracterización de un patrón visual. En otras palabras, una secuencia de video se representa como una bolsa de características locales [35], acuatizadas en palabras visuales y un video se representa entonces como el histograma de frecuencia sobre las palabras visuales. En particular el algoritmo de agrupamiento k-medias ha sido empleado en el análisis de imágenes y videos [38], aunque existen otros métodos que han sido empleados como los bosques aleatorios [39].

### 2.1.3. Clasificación

Finalmente, se entrena un método de clasificación para distinguir entre las clases de interés. Uno de los métodos ampliamente usados son las máquinas de vectores de soporte (SVM, por sus siglas en inglés) [35, 38, 40].

## 2.2. Clasificación de video usando redes neuronales convolucionales.

En los últimos años, el campo del aprendizaje automático ha avanzado enormemente en el tratamiento de problemas complejos como el reconocimiento, segmentación, detección y clasificación de imágenes. En particular, se ha encontrado que las redes neuronales convolucionales pueden lograr un muy buen desempeño en este tipo de tareas [2, 3, 15, 41, 42]. Por ejemplo, con la base de datos de imágenes de dígitos escritos a mano MNIST [43] se ha alcanzado un porcentaje de reconocimiento del 99.7 % [44]. En la tarea de clasificación de imágenes han surgido arquitecturas de redes neuronales convolucionales que de forma sucesiva han mostrado mejoras en el desempeño. Algunas de estas arquitecturas son:

- AlexNet [3].
- Inception [42].
- BN-Inception-v2 [45].
- Inception-v3 [15].

El desempeño de estos modelos ha sido validado con un subconjunto de datos provenientes de ImageNet [41], una base de datos emblemática en cuanto a problemas de clasificación de imágenes se trata. ImageNet está formada por 15 millones de imágenes de alta resolución recolectadas de la Internet y etiquetadas por personas empleando la herramienta Amazon Mechanical Turk [46], que es una plataforma que propone trabajos simples para los humanos pero que una máquina no puede realizar.

Debido a los buenos resultados obtenidos con las redes convolucionales en el dominio de las imágenes, se ha estudiado el rendimiento de ellas en la clasificación de video uni-etiqueta [18]. Uno de los trabajos más representativos de clasificación de

video empleando esta metodología fue el realizado por Andrej Karpathy *et al.* cuyo objetivo fue realizar clasificación uni-etiqueta de videos de deportes sobre la base de datos $Sports-1M$ creada para el mismo trabajo. La arquitectura más básica que se presenta permite realizar la clasificación del video tomando únicamente un cuadro del video completo. Posteriormente, presenta arquitecturas más complejas que involucran extraer un mayor número de cuadros consecutivos del video con el objetivo de aprender características temporales entre ellos. Sin embargo, los resultados de este importante trabajo arrojaron que las arquitecturas que consideran un mayor número de cuadros en comparación con la arquitectura más básica no muestran una mejora significativa. Esto podría indicar que no es suficiente el usar esta metodología para capturar las características espacio-temporales.

Por otro lado, se han realizado trabajos para la tarea de reconocimiento de acciones que con el objetivo de capturar la información de movimiento entre los cuadros de video en una red convolucional, se incorpora el flujo óptico [47]. La principal desventaja es que usan únicamente hasta 10 cuadros consecutivos, lo cual solo considera parte de la historia del video completo. También para la tarea de reconocimiento de acciones se han empleado redes convolucionales 3D [48, 49, 50], los cuales son modelos de que extraen la características de las dimensiones espaciales y temporales realizando convoluciones en 3D, capturando así la información de movimiento codificada en múltiples cuadros. Este tipo de modelos han sido empleados en videoclips cortos, por lo general de unos pocos segundos, para aprender implícitamente las funciones de movimiento desde cuadros de los videos.

## 2.3. Clasificación de video usando redes neuronales recurrentes

La redes neuronales recurrentes son redes neuronales artificiales diseñadas para hacer uso de la información secuencial. Específicamente una de las redes neuronales más empleadas es la memoria de corto y largo plazo [51] (LSTM, por sus siglas en inglés) ya que es capaz de aprender dependencias en una secuencia a corto y largo plazo. Por ello, han sido empleadas para tareas como reconocimiento de escritura a mano [52, 53], reconocimiento de voz [54, 55] y reconocimiento de emociones [56]. Asimismo, este tipo de redes neuronales han sido utilizadas para el análisis de video [19, 57, 58]. En particular, en clasificación de video se han tenido diversas aproximaciones. Por ejemplo, en el trabajo realizado por Moez Baccouche *et al.* [57], cuyo objetivo fue la clasificación de acciones en videos de fútbol, se realizó una representación del video por medio de secuencias de imágenes de las cuales se extrajeron características de manera tradicional, es decir empleando un algoritmo que permitiera detectar y describir características locales. Las secuencias de imágenes fueron representadas por un conjunto de descriptores, uno por imagen. Finalmente, se utilizó una LSTM para realizar la clasificación, con el objetivo de modelar la componente temporal del conjunto de descriptores. Otros de la trabajos empleando LSTM son [2, 59], que hacen uso del flujo óptico para incorporar información del movimiento. El flujo óptico es la distribución de las velocidades aparentes de los ob-

jetos en una imagen. Al estimarlo entre los cuadros de un video, se pueden estimar las velocidades de los objetos en este. Sin embargo, [19] mostró que el uso de flujo óptico no siempre es útil, sobre todo si los videos no fueron bien cuidados, es decir, no son profesionales o cuentan con pocos recursos.

## 2.4. Clasificación de avances cinematográficos.

Existen algunos trabajos relacionados con análisis de video. Específicamente, en avances cinematográficos o películas. Algunos de ellos son clasificación de acciones en escenas de películas [35]. Por otro lado, se ha realizado la clasificación de avances cinematográficos [60] en dos categorías, avances que contienen acción y aquellos que no tienen. Esto se ha realizado considerando las características visuales y las características de audio. Otro de los trabajos de clasificación uni-etiqueta para avances de películas fue realizado sobre un conjunto de datos de 223 avances en 7 clases diferentes, para ello se consideraron las características de audio y visuales [60].

## 2.5. Bases de datos de video

En esta sección se describen las bases de video disponibles que han sido empleadas para análisis de video. En su mayoría se trata de conjuntos de datos de acciones humanas cuyo tamaño es de apenas cientos de videos. Por otra parte, se describen dos de las más recientes bases de datos que no solo proporcionan una mayor cantidad de videos, sino también proporcionan mayor variedad pues no son destinadas para análisis de acciones humanas.

### *KTH*

*KTH* es una base de datos de videos en blanco y negro de acciones humanas etiquetadas [38]. Esta contiene seis tipos de acciones, cada una de las cuales fue repetida por 25 personas diferentes en cuatro escenarios distintos. Las características principales de esta base de datos son las siguientes:

- Número de videos: 600 (192 entrenamiento, 192, validación y 256 prueba).
- Número de clases: 6.
- Etiquetas: caminar, trotar, correr, boxear, agitar las manos y aplaudir.
- Resolución: $160 \times 120$.

El número de archivos de video está dado por la combinación de las 25 personas, 6 acciones y 4 escenarios distintos ($25 \times 6 \times 4 = 600$).

### $WEIZMANN$

$WEIZMANN$ es una base de datos etiquetada de video de acciones humanas, la cual fue recolectada por el $Weizmann\ Institute$ [61]. Todos lo videos que contiene la base de datos fueron tomados al aire libre. Las características principales de esta base de datos son las siguientes:

- Número de videos: 90.
- Número de clases: 10.
- Etiquetas: caminar, correr, saltar, galope de lado, girar, mover una mano, mover las dos manos, saltar en el lugar, salto de tijera, saltar.
- Resolución: $180 \times 144$.
- Imágenes por segundo: 50.

Para la recolección de esta base de datos participaron 9 personas diferentes.

### $UCF - Sports$

$UCFSports$ es una base de datos de acciones provenientes de diferentes deportes. Los videos fueron recolectados de varios canales televisivos. Las características principales de esta base de datos son las siguientes:

- Número de videos: 182.
- Número de clases: 9.
- Etiquetas: bucear (14 videos), balanceo en golf (18 videos), patear (20 videos), levantar pesas (6 videos), montar a caballo (12 videos), correr (13 videos), patinar (12 videos), balanceo en béisbol (20 videos), caminar (22 videos).
- Resolución: $720 \times 480$.
- Imágenes por segundo: 50.

### $UCF - 50$

$UCF50$ es un conjunto de datos de reconocimiento de acciones que cuenta con 50 categorías diferentes. Los videos de los cuales se compone la base fueron tomados de la plataforma YouTube® [62]. Este conjunto de datos es una extensión del conjunto de datos de acciones $UCF11$ el cual solo contiene 11 categorías. Las características principales de esta base de datos son las siguientes:

- Número de videos: 6,676.
- Número de clases: 50.

- Etiquetas: 50 etiquetas diferentes relacionadas con acciones realizadas en deportes.
- Resolución: $320 \times 240$.
- Imágenes por segundo: 25.
- Formato: *Audio Video Interleave* (AVI).

Cada clase tiene como mínimo 100 videos.

### $UCF-101$

$UCF-101$ es un conjunto de datos de video de acciones humanas realistas, los cuales fueron recopilados de la plataforma YouTube®. Este conjunto de datos fue publicado por el Centro de Investigación en Visión por Computadora de la Universidad de Florida Central en el año 2012 [63], cuenta con 101 categorías y es una extensión de $UCF50$ que tiene 50 categorías de acciones. Las características principales de esta base de datos son las siguientes:

- Número de videos: 13,320.
- Número de clases: 101.
- Etiquetas: 101 etiquetas que a su vez pueden ser divididas en 5 grupos: interacción hombre-objeto, movimiento del cuerpo humano, interacción humano-humano, interacción con instrumento musical, acciones en deportes.
- Resolución: $320 \times 240$.
- Imágenes por segundo: 25.
- Formato: *Audio Video Interleave* (AVI).

### *Hollywood*

Este conjunto de datos provee videos de acciones humanas realistas recolectadas de largometrajes, programas de comedia o segmentos de noticias [64]. Las características principales de esta base de datos son las siguientes:

- Número de videos: 430.
- Número de clases: 8.
- Etiquetas: contestar el teléfono, salir de un coche, saludar de mano, abrazar a una persona, besar, sentarse, pararse.
- Resolución: Variable.

### *Hollywood2*

Esta base es una extensión de *Hollywood*, que consiste en 430 videos y 8 clases. Las características principales de esta base de datos son las siguientes:

- Número de videos: 2,859.
- Número de clases: 12.
- Etiquetas: contestar el teléfono, conducir un coche, comer, pelear, salir de un coche, saludar de mano, abrazar a una persona, besar, correr, sentarse, pararse, enderezarse.
- Resolución: Variable.

### $Sports - 1M$

El conjunto de datos $Sports-1M$ consta de 1,133,158 de ligas a videos de deportes de la platafoma de YouTube® etiquetados. Esta base de datos fue publicada en el año 2014 [18]. Las características principales de esta base de datos son las siguientes:

- Número de videos: 1,133,158.
- Número de clases: 487.
- Etiquetas: 487 etiquetas que a su vez están divididas en 6 categorías : deportes acuáticos, deportes de equipo, deportes de invierno, deportes de pelota, deportes de combate y deportes con animales.

Cada una de las clases poseen de 1,000 a 3,000 videos aproximadamente y el 5 % de ellos pertenecen a más de una categoría. Las etiquetas de los videos fueron obtenidas de manera automática, analizando los metadatos de texto que se encontraban en el sitio de YouTube®. Este conjuto de datos está dividido en 70 % para entrenamiento, 10 % para validación y 20 % para pruebas. Sin embargo, dado que en el sitio de YouTube® existen videos duplicados, es posible que un mismo video pueda aparecer tanto en el conjunto de entrenamiento como en el de prueba o validación.

### $YouTube - 8M$

Esta es la base de datos de video más reciente hasta la fecha. Fue desarrollada por el grupo de *Video Understanding* de Google® y publicada en el 2016 [65]. Se trata de un conjunto de datos multi-etiqueta el cual fue etiquetado mediante el sistema de anotación de YouTube®. En promedio el número de clases a los cuales puede pertenecer un video es de 3.4. Los enlaces a los videos de la base de datos están disponibles pero dado que el tamaño del conjunto de datos de video es un *petabyte*, existe otra alternativa que es obtener las características de los videos sin necesidad de descargar el archivo de video. Las características de los videos fueron extraídas con una red neuronal convolucional pre-entrenada. El tener acceso a estas

características representa una gran ventaja ya que no se requiere de tanto espacio de almacenamiento y parte del procesamiento ya fue realizado. Las características principales de esta base de datos son las siguientes:

- Número de videos: 8,264,650.
- Número de clases: 4,716.
- Etiquetas: 4,716, agrupados en 24 categorías diferentes.

# 3 Redes Neuronales

Una red neuronal artificial (ANN, por sus siglas en inglés) es un modelo computacional que se compone de un gran número de unidades básicas de cálculo llamadas neuronas artificiales que están conectadas entre sí en una red de comunicación que permite realizar cálculos de gran complejidad. Desde su aparición han sido ampliamente usadas en tareas como clasificación [66], reconocimiento de patrones [67, 68, 69] y reconocimiento del habla [70]. Recientemente han tenido un gran auge con el surgimiento de nuevos modelos y técnicas que han permitido obtener muy buenos resultados en tareas de suma importancia como clasificación, reconocimiento y segmentación de imágenes [2, 3, 17].

En este capítulo se aborda el modelo más básico de una red neuronal, la neurona artificial. Posteriormente, se describe el perceptrón multicapa y su algoritmo entrenamiento. Finalmente, se describen modelos más complejos de redes neuronales que son empleados en este proyecto como las redes convolucionales y las redes recurrentes.

## 3.1. Neurona artificial

Una neurona artificial es una unidad de cómputo que toma un vector de entradas $\mathbf{x} \in \mathbb{R}^n$ asociado a un vector de pesos $\mathbf{w} \in \mathbb{R}^n$ y un peso extra $b \in \mathbb{R}$ conocido como sesgo. La salida o activación $h$ de la neurona está definida por la Ecuación 3.1:

$$h(\mathbf{x}; \mathbf{w}, b) = f(\sum_{i=1}^{n} x_i w_i + b). \tag{3.1}$$

donde $f\colon \mathbb{R} \to \mathbb{R}$ es la función de activación.

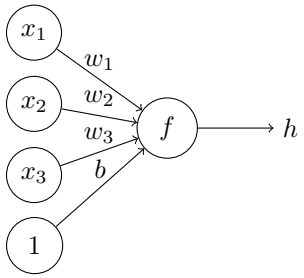



Figura 3.1. Neurona artificial de tres entradas $x_1, x_2, x_3$ ilustradas por círculos. El sesgo $b$ se considera el peso de una entrada adicional con valor constante 1. Los pesos $w_1, w_2, w_3$ son las conexiones de las entradas y la función de activación $f$.

El primer modelo matemático de una neurona artificial fue presentado por McCulloch y Pitts [71]. La Figura 3.1 muestra el diagrama de una neurona artificial. Es útil definir la salida de la neurona en términos de vectores como en la Ecuación 3.2:

$$h(\mathbf{x}; \mathbf{w}, b) = f(\mathbf{w}^T \mathbf{x} + b). \tag{3.2}$$

Las función de activación puede ser definida de muchas formas, las más comunes son las siguientes.

- Función sigmoide.

$$f(z) = \frac{1}{1 + e^{-z}}. \tag{3.3}$$

- Función tangente hiperbólica.

$$f(z) = \frac{e^x - e^{-x}}{e^{-x} + e^{-x}}. \tag{3.4}$$

  Esta función de activación puede verse como la función sigmoide escalada en el intervalo $(-1, 1)$.

- Función exponencial normalizada o softmax.

$$f_j(\mathbf{z}) = \frac{e^{z_j}}{\sum_{i=1}^{n} e^{z_i}}, \quad \text{para } j = 1 \dots n. \tag{3.5}$$

- Función rectificada lineal (ReLU, por sus siglas en inglés).

$$f(z) = max(0, z). \tag{3.6}$$

## 3.2. Perceptrón multicapa

El perceptrón multicapa es una red neuronal formada por grupos de neuronas llamados capas.

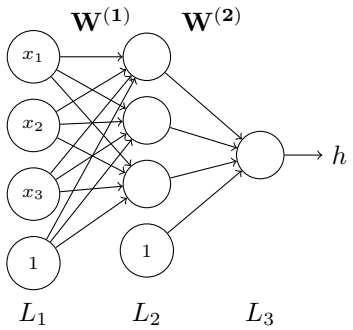



Figura 3.2. Perceptrón multicapa con una capa de entrada $L_1$, una capa oculta $L_2$ y una capa de salida $L_3$ (con una sola neurona). Los pesos $\mathbf{W}^{(1)}$ conectan las capas $L_1$ y $L_2$; mientras que los pesos $\mathbf{W}^{(2)}$ conectan las capas $L_2$ y $L_3$.

Como se muestra en la Figura 3.2, las conexiones entre neuronas se dan entre capas de izquierda a derecha. Por este motivo estas redes también toman el nombre de redes alimentadas hacia adelante. La capa más a la izquierda es la capa de entrada, la capa más a la derecha es la capa de salida y las capas intermedias son las capas ocultas. El número de capas en una red esta dado por $n_l$ y la capa número $l$ se denota por $L_l$.

Los parámetros de la red están dados por $(\mathbf{W}, \mathbf{b}) = ((\mathbf{W}^{(1)}, \mathbf{b}^{(1)}), (\mathbf{W}^{(2)}, \mathbf{b}^{(2)}))$, donde $w_{ij}^{(l)} \in \mathbf{W}^{(l)}$ es el peso asociado a la conexión entre la neurona $j$ en la capa $l$ y la neurona $i$ en la capa $l+1$; y $b_i^l$ es el sesgo asociado a la neurona $i$ en la capa $l+1$. Además, $s_l$ denota el número de neuronas en la capa $l$.

Para explicar el proceso de cómputo del perceptrón multicapa consideremos el ejemplo presentado en la Figura 3.2. La activación o salida de la neurona $i$ en la capa $l$ se denota $a_i^{(l)}$. Para la capa de entrada $l = 1$ se tiene que $a_i^{(1)} = x_i$. El cálculo $h(\mathbf{x})$ para la red está dado por las siguientes expresiones:

$$a_1^{(2)} = f(w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{13}^{(1)} x_3 + b_1^{(1)}), \tag{3.7}$$

$$a_2^{(2)} = f(w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2 + w_{23}^{(1)} x_3 + b_2^{(1)}), \tag{3.8}$$

$$a_3^{(2)} = f(w_{31}^{(1)} x_1 + w_{32}^{(1)} x_2 + w_{33}^{(1)} x_3 + b_3^{(1)}), \tag{3.9}$$

$$h(\mathbf{x}) = a_1^{(3)} = f(w_{11}^{(2)} a_1^{(2)} + w_{12}^{(2)} a_2^{(2)} + w_{13}^{(2)} a_3^{(2)} + b_1^{(2)}). \tag{3.10}$$

En lo siguiente se denotará $z_i^l$ como la suma pesada de las entradas a la neurona $i$ de la capa $l$ incluyendo el sesgo, de modo que $a_i^l = f(z_i^l)$. Esto permite utilizar una notación más compacta si se extiende la definición de la función $f(\cdot)$ para aplicarla a vectores por elemento. Entonces se pueden reescribir las ecuaciones anteriores como:

$$\mathbf{z}^{(2)} = \mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}, \tag{3.11}$$

$$\mathbf{a}^{(2)} = f(\mathbf{z}^{(2)}), \tag{3.12}$$

$$\mathbf{z}^{(3)} = \mathbf{W}^{(2)} \mathbf{a}^{(2)} + \mathbf{b}^{(2)}, \tag{3.13}$$

$$h(\mathbf{x}) = \mathbf{a}^{(3)} = f(\mathbf{a}^{(3)}). \tag{3.14}$$

En general, si se tiene en cuenta que $\mathbf{a}^{(1)} = \mathbf{x}$, la activación $\mathbf{a}^{(l+1)}$ para la capa $l+1$ se computa como:

$$\mathbf{z}^{(l+1)} = \mathbf{W}^{(l)} \mathbf{a}^{(l)} + \mathbf{b}^{(l)}, \tag{3.15}$$

$$\mathbf{a}^{(l+1)} = f(\mathbf{z}^{(l+1)}). \tag{3.16}$$

La Figura 3.2 muestra un ejemplo de red neuronal muy básico. No obstante, se pueden construir redes neuronales con diferentes patrones de conectividad entre neuronas, es decir, diferentes arquitecturas. Por ejemplo, una arquitectura más compleja puede ser formada con múltiples capas ocultas donde se tiene una capa de entrada $L_1$, una capa de salida $L_{n_l}$ y cada capa oculta $l$ está completamente conectada a la capa $l+1$. En esta configuración, el cómputo de la salida de la red se realiza calculando todas las activaciones de la capa $L_2$, después para la capa $L_3$ y así suce-

sivamente hasta la capa $L_{n_l}$ usando las ecuaciones 3.15 y 3.16. Este es un ejemplo de una red neurona alimentada hacia adelante, ya que el gráfico de conectividad no tiene ciclos o bucles dirigidos. Además, las redes neuronales pueden tener múltiples neuronas de salida. La Figura 3.3 muestra una red neuronal con dos neuronas de salida y dos capas ocultas.

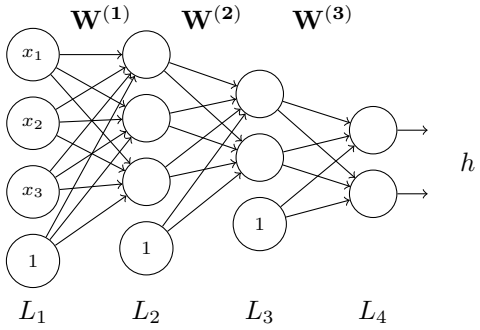



Figura 3.3. Perceptrón multicapa con una capa de entrada $L_1$; dos capas ocultas $L_2$ y $L_3$; y una capa de salida $L_4$ (con dos neuronas).

En este caso, se sobrecarga la definición de la función de salida de forma que $h(\mathbf{x}) \in \mathbb{R}^2$. En general, se considerará que la función de salida está sobrecargada, por lo que produce un vector correspondiente al número de salidas de la red neuronal.

### 3.2.1. Algoritmo de retropropagación

La idea general del algoritmo de retropropagación es minimizar una función de distancia o error $E$ entre las etiquetas reales del conjunto de datos de entrenamiento $\mathbf{y}$ y las etiquetas predichas $h(\mathbf{x})$ por la red para este conjunto. Consideremos un conjunto de entrenamiento $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}$ con $m$ ejemplos. Una de las funciones de error más comunes es el error cuadrático medio mostrado en la Ecuación 3.17:

$$E(\mathbf{W}, \mathbf{b}) = \frac{1}{2m} \sum_{i=1}^{m} ||h(\mathbf{x}^{(i)}; \mathbf{W}, \mathbf{b}) - \mathbf{y}^{(i)}||^2. \tag{3.17}$$

Otra función ampliamente utilizada es la entropía cruzada mostrada en la Ecuación 3.18:

$$E(\mathbf{W}, \mathbf{b}) = -\frac{1}{m} \sum_{i=1}^{m} \mathbf{y}^{(i)} \log(h(\mathbf{x}^{(i)}; \mathbf{W}, \mathbf{b})). \tag{3.18}$$

La explicación del algoritmo retropropagación se realizará utilizando el error cuadrático medio, no obstante, no es difícil extenderla a entropía cruzada. El objetivo es minimizar $E$ como una función de $\mathbf{W}$ y $\mathbf{b}$. El proceso de entrenamiento comienza inicializando cada peso $w_{ij}^{(l)}$ y $b_i^{(l)}$ con valores pequeños tomados de una distribución normal $\mathcal{N}(0, \epsilon)$. Para minimizar la función de error se emplea el método del gradiente descendiente, por lo que un paso de actualización de los pesos $\mathbf{W}$ y $\mathbf{b}$ se da de la forma siguiente:

$$w_{ij}^{(l)} := w_{ij}^{(l)} - \alpha \frac{\partial}{\partial w_{ij}^{(l)}} E(\mathbf{W}, \mathbf{b}), \tag{3.19}$$

$$b_{i}^{(l)} := b_{i}^{(l)} - \alpha \frac{\partial}{\partial b_{i}^{(l)}} E(\mathbf{W}, \mathbf{b}). \tag{3.20}$$

Donde $\alpha$ es la tasa de aprendizaje y := es asignación de variables. El paso clave es calcular estas derivadas parciales. Ahora se describirá la forma en que el algoritmo de retropropagación permite calcular estas derivadas de forma eficiente.

La intuición detrás del algoritmo es la siguiente. Dado un ejemplo de entrenamiento $(\mathbf{x}, \mathbf{y})$, durante la primera etapa (hacia adelante) se calculan todas las activaciones de la red, incluyendo las salidas $h(\mathbf{x}; \mathbf{W}, \mathbf{b})$. La segunda etapa (hacia atrás) consiste en calcular para cada nodo $i$ en la capa $l$, la fracción del error $\delta_i^{(l)}$ que determina la contribución del nodo al error total de la red. Para un nodo de salida, podemos calcular directamente la diferencia entre la activación de la red y el valor objetivo, el cual se denota por $\delta_i^{(n_l)}$. Más detalladamente el algoritmo puede ser descrito de la siguiente manera:

1. Realizar la propagación hacia adelante de toda la red. Se calculan las activaciones de cada capa con las ecuaciones 3.15 y 3.16.

2. Por cada nodo $i$ en la capa de salida se calcula:

$$\delta_i^{(n_l)} = \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} ||\mathbf{y} - h(\mathbf{x})||^2 = -(y_i - a_i^{(n_l)}) f'(z_i^{(n_l)}). \tag{3.21}$$

3. Por cada capa oculta se calcula:

$$\delta_i^l = (\sum_{j=1}^{s_{l+1}} w_{ji}^{(l)} \delta_j^{(l+1)}) f'(z_i^{(l)}). \tag{3.22}$$

4. Se actualizan los pesos de la red de acuerdo con las ecuaciones 3.19 y 3.20.

## 3.3. Deserción

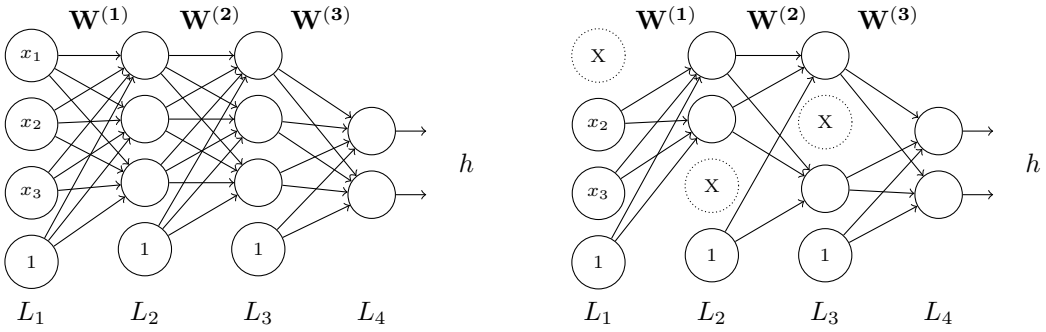
La gran cantidad de parámetros presentes en la redes neuronales profundas las hace susceptibles a sufrir sobreajuste. La deserción [72] es una técnica para abordar este problema. Su principio de funcionamiento consiste en desactivar temporalmente neuronas de la red neuronal junto con sus conexiones entrantes y salientes de acuerdo a una probabilidad $p$ fija. La idea detrás de esta técnica es obligar a la red neuronal a aprender múltiples representaciones independientes de los mismos datos mediante la desactivación alternada de neuronas en la fase de aprendizaje. Considerando una variable aleatoria $d_j^{(l)}$ que se distribuye según una distribución $Bernoulli(p)$, las ecuaciones 3.15 y 3.16 pueden reescribirse empleando la deserción de la siguiente manera:

$$\widetilde{\mathbf{a}}^{(l)} = \mathbf{d}^{(l)} * \mathbf{a}^{(l)}, \tag{3.23}$$

$$\mathbf{z}^{(l+1)} = \mathbf{W}^{(l)}\widetilde{\mathbf{a}}^{(l)} + \mathbf{b}^{(l)}, \tag{3.24}$$

$$\mathbf{a}^{(l+1)} = f(\mathbf{z}^{(l+1)}). \tag{3.25}$$

Donde $*$ denota el producto *Hadamard*, una operación binaria que toma dos vectores de las mismas dimensiones y produce otro vector donde cada elemento $i$ es el producto de los elementos $i$ de los dos vectores originales.




(a) Red neuronal estándar

(b) Red neuronal con deserción

Figura 3.4. Red neuronal con deserción. En la imagen (a) se muestra la arquitectura de una red neuronal con una capa de entrada $L_1$; dos capas ocultas $L_2$ y $L_3$; y una capa de salida $L_4$ (con dos neuronas). En la imagen (b) se muestra un ejemplo de la misma red con deserción.

Para cualquier capa $l$, $\mathbf{d}^{(l)}$ es un vector de variables aleatorias independientes de Bernoulli, cada una de las cuales tiene probabilidad $p$ de ser 1. Este vector se multiplica con el producto *Hadamard* por las salidas de la capa $\mathbf{a}^{(l)}$ para calcular las salidas con deserción $\widetilde{\mathbf{a}}^{(l)}$. Estas nuevas salidas se usan como entrada a la siguiente capa. Este proceso se aplica en cada capa. Esto equivale a muestrear una subred de una red más grande. La Figura 3.4 muestra una red neuronal estándar y la misma red al aplicar deserción.

## 3.4. Redes neuronales convolucionales

La arquitectura de una red neuronal convolucional consta principalmente de las tres siguientes capas:

- Capas convolucionales.
- Capas de submuestreo.
- Capas completamente conectadas.

La Figura 3.5 muestra la aquitectura de una red neuronal convolucional.

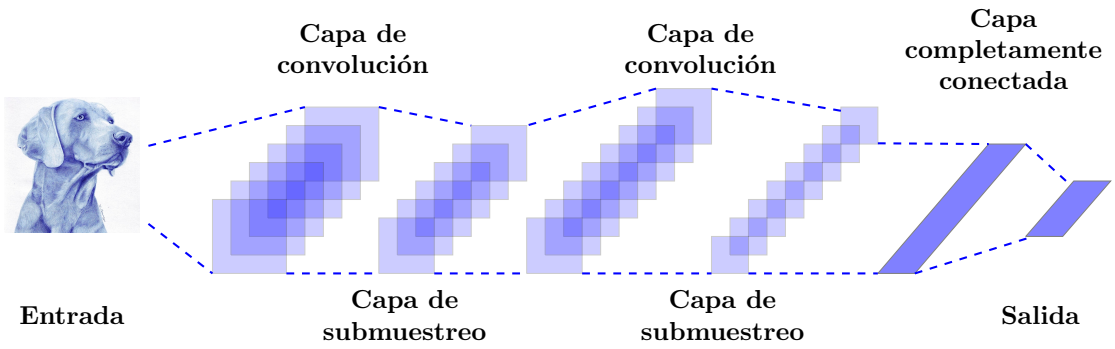



Figura 3.5. Diagrama de la arquitectura de una CNN.

### 3.4.1. Capas convolucionales

La entrada de las redes neuronales convolucionales en el caso de imágenes es una matriz que contiene los valores de cada pixel. Cada uno de los pixeles que conforman la imagen están codificados en un rango de valores de 0 a 255. Además, dado que las imágenes a color poseen tres canales estos agregan un campo de profundidad a los datos. Es decir, la representación de una imagen puede verse como una estructura tridimensional $r \times s \times p$, donde $r$ es el alto, $s$ el ancho de la imagen y $p$ es el número de canales. Por ejemplo, para la composición de color rojo, verde y azul (RGB, por sus siglas en inglés) $p = 3$.

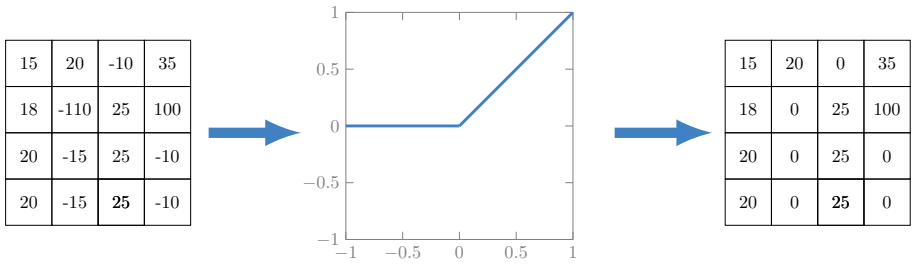



Figura 3.6. Representación del proceso de una capa no lineal de una CNN.

Una capa convolucional está compuesta de $k$ filtros de tamaño $r \times r \times p$ donde $r$ es más pequeño que la dimensión de la imagen y $p$ es igual al número de canales. En el contexto de las redes neuronales convolucionales, un filtro es un conjunto de pesos que realiza un recorrido, a partir de la esquina superior izquierda de la entrada hasta la esquina inferior derecha, se mueve de izquierda a derecha un número determinado de elementos y una vez que alcanza la esquina superior derecha, se mueve un elemento hacia abajo y de nuevo realiza un recorrido de izquierda a derecha. La convolución de una entrada con un filtro produce un mapa de características, es decir, el k-ésimo mapa de características en una capa está dado por la convolución

de los pesos del k-ésimo filtro y los valores de entrada, más el sesgo. Se puede pensar en las capas convolucionales como extractores de características de los datos de entada. La primera capa de convolución puede extraer características de bajo nivel como bordes, líneas y esquinas, mientras que las capas de nivel superior extraen características cada vez más complejas a partir de las características de más bajo nivel de la capa anterior.

Después de realizar la operación de convolución se aplica una función de activación, comúnmente se emplea la función *ReLU* de la Ecuación 3.6. La Figura 3.6 ejemplifica la aplicación de la función de activación a la salida de una capa convolucional.

### 3.4.2. Capas de submuestreo

Las capas de submuestreo son comúnmente empleadas después de las capas convolucionales. Existen diversas maneras de hacer el muestreo, las más comunes son el muestreo máximo y el promedio. Una capa de muestreo máximo regresa los valores máximos de regiones rectangulares de cierto tamaño determinado, mientras que una capa de muestreo promedio regresa el valor promedio de dicha región.

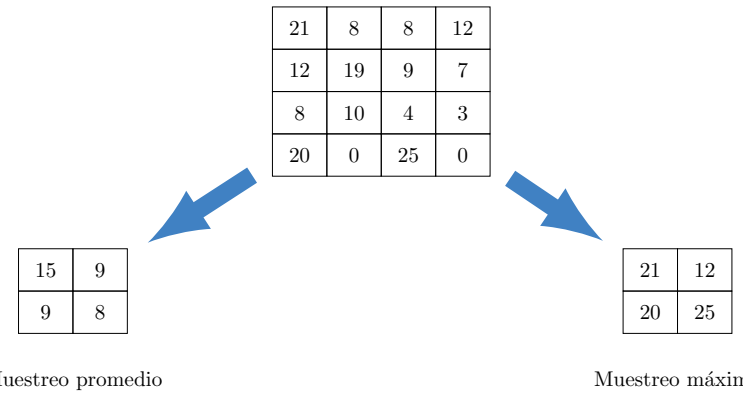



Figura 3.7. Ejemplo del proceso de muestreo promedio y máximo de una CNN.

La Figura 3.7 muestra el proceso de muestreo de una entrada de tamaño $4 \times 4$. Para el muestreo de $2 \times 2$, la imagen se divide en cuatro matrices no superpuestas de tamaño $2 \times 2$. En el caso de la muestreo máximo se toma el máximo de los cuatro valores y en el caso del muestreo promedio y se toma el promedio de los 4 valores y se redondea al entero más cercano. Este tipo de capas ofrecen algunas ventajas como, por ejemplo, reducir la dimensionalidad de las características y hacerlas más manejables, reducir el número de parámetros y cálculos en la red. Además, hacen más robusta a la red neuronal a translaciones y rotaciones.

### 3.4.3. Capas completamente conectadas

Las capas completamente conectada de una red convolucional son un perceptrón multicapa tradicional. La salida de las capas convolucionales y de muestreo representan características y son la entrada de las capas completamente conectadas. El

propósito de estas capas es utilizar las características, examinarlas y clasificarlas en aquella clase con la que tenga una mayor correlación. El entrenamiento se realiza por medio del algoritmo de retropropagación.

En particular, Inception-v3 es una arquitectura de una red neuronal convolucional para clasificación uni-etiqueta de imágenes ha mostrado muy buenos resultados y fue empleada en el proyecto de investigación.

### 3.4.4. Arquitectura Inception-v3

Inception-v3 [15] es una red neuronal convolucional con una arquitectura muy compleja que se ha ido modificando desde sus inicios de forma experimental con el objetivo de mejorar el desempeño en clasificación de imágenes. Esta red neuronal está diseñada para realizar clasificación de imágenes multi-clase pero no multi-etiqueta. Es decir, permite realizar la clasificación de una imagen entre más de dos clases pero hace la suposición de que a cada muestra solo se le puede asignar una y solo una etiqueta. La arquitectura de Inception-v3 puede dividirse en dos grandes etapas, las cuales son:

- Etapa de extracción de características. Esta etapa está formada por múltiples capas convolucionales con distintas características, capas de submuestreo máximo y promedio. También esta etapa está formada por capas de concatenación, las cuales se encargan de concatenar las salidas de las capas anteriores generando un único vector como entrada a la siguiente capa.
- Etapa de clasificación. Esta etapa se forma por una capa completamente conectada que finalmente es seguida de la función de activación *softmax* descrita por la Ecuación 3.5. La función de error que emplea esta red neuronal es la entropía cruzada que está definida por la Ecuación 3.18.

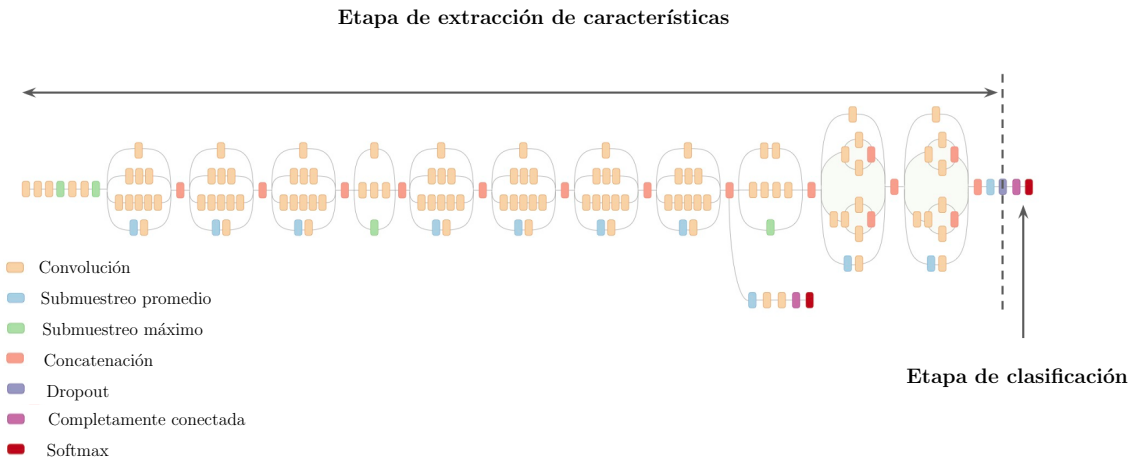


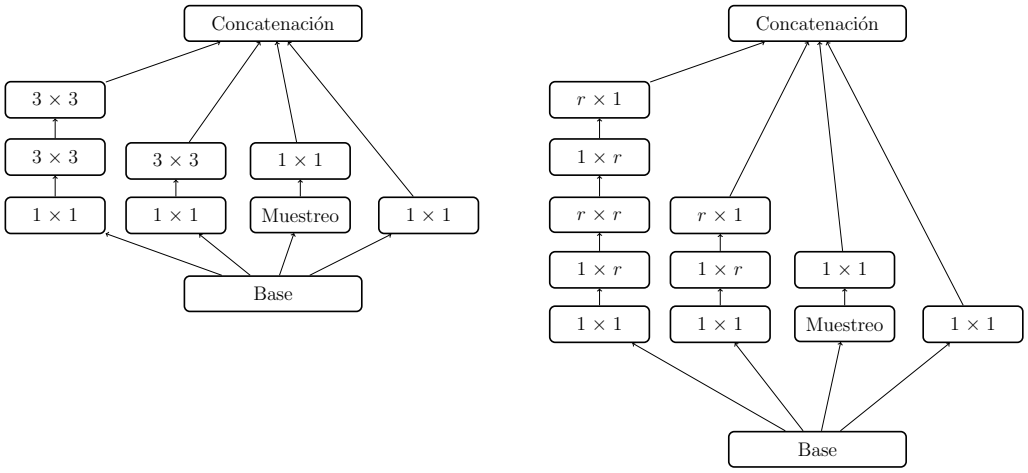

Figura 3.8. Capas de la arquitectura de la red neuronal convolucional Inception-v3.

La Figura 3.8 muestra las capas que componen la arquitectura de la red neuronal convolucional Inception-v3. Asimismo, muestra la división de la arquitectura en la

etapa de extracción de características y clasificación. De manera general, la clasificación con Inception-v3 se realiza de la siguiente manera. Se toma una imagen de entrada de tamaño $299 \times 299$ la cual pasa a través de la etapa de extracción de características y se genera un vector que corresponde a las características extraídas, esto por cada una de las imágenes del conjunto de datos de prueba. Este vector, se convierte en la entrada de la etapa de clasificación, la cual determina a qué clase pertenece la imagen. De manera más detallada se describe la arquitectura de Inception-v3 en la Tabla 3.1.

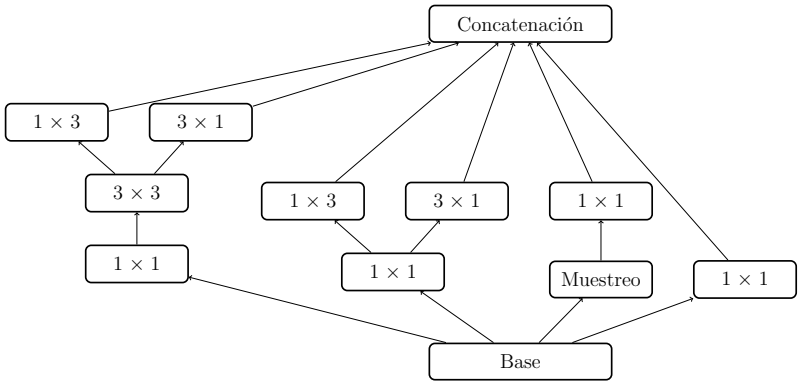
| **Inception-v3** | | |
|---|---|---|
| **Tipo de capa** | **Filtro-paso** | **Tamaño de entrada** |
| Convolución | $3 \times 3 - 2$ | $299 \times 299 \times 3$ |
| Convolución | $3 \times 3 - 1$ | $149 \times 149 \times 32$ |
| Convolución | $3 \times 3 - 1$ | $147 \times 147 \times 32$ |
| Muestreo máximo | $3 \times 3 - 2$ | $147 \times 147 \times 64$ |
| Convolución | $3 \times 3 - 1$ | $73 \times 73 \times 64$ |
| Convolución | $3 \times 3 - 2$ | $71 \times 71 \times 80$ |
| Muestreo máximo | $3 \times 3 - 1$ | $35 \times 35 \times 192$ |
| 3 Inception | Figura 3.9 (a) | $35 \times 35 \times 288$ |
| 5 Inception | Figura 3.9 (b) | $17 \times 17 \times 768$ |
| 2 Inception | Figura 3.10 (c) | $8 \times 8 \times 1280$ |
| Muestreo promedio | $8 \times 8$ | $8 \times 8 \times 2048$ |
| Completamente conectada | $3 \times 3 - 2$ | $1 \times 1 \times 2048$ |
| *Softmax* | Clasificador | $1 \times 1 \times 1000$ |

Tabla 3.1. Descripción de las capas de la arquitectura de Inception-v3.




(a) (b)

Figura 3.9. Descripción de los módulos (a) y (b) empleados en la arquitectura del modelo Inception-v3.




(c)

Figura 3.10. Descripción del módulo (c) empleado en la arquitectura del modelo Inception-v3.

### 3.4.5. Transferencia de conocimiento

La técnica de aprendizaje por transferencia de conocimiento consiste en tomar una red neuronal convolucional que ha sido entrenada para una tarea diferente a la de interés con un conjunto de datos muy grande y usarla como una inicialización o un extractor de características fijo. El aprendizaje por transferencia de conocimiento es una técnica que se ha empleado [65, 73, 5, 74, 75] debido a que los modelos de reconocimiento de objetos deben aprender millones de pesos y pueden tardar muchas semanas de entrenamiento incluso con equipos de alto desempeño que emplean GPUs de última generación, además que requieren de una gran cantidad de datos. En la práctica, es muy poco común entrenar una red neuronal convolucional desde cero debido a que por lo general no se dispone de conjuntos de datos etiquetados de gran tamaño. En particular, el buen desempeño logrado por Inception-v3 en ImageNet ha permitido emplear esta arquitectura para realizar transferencia de conocimiento. A continuación se describen las dos maneras de transferencia de conocimiento en las que puede ser usada Inception-v3.

#### Usada como inicialización

Para emplear Inception-v3 de esta manera se requiere utilizar los pesos de las capas convolucionales obtenidos en el entrenamiento con ImageNet para inicializar y entrenar desde cero las capas de clasificación, afinando los pesos de las capas convolucionales de la red pre-entrenada con la nueva base de datos mediante el algoritmo de retropropagación. Es posible ajustar los pesos de todas las capas convolucionales, o mantener algunas capas fijas, debido a que las primeras capas son responsables de obtener características muy generales como bordes y formas básicas.

#### Usada como extractor de características

Inception-v3 puede ser empleada como extractor de características, para ello basta con quitar la capa completamente conectada y tomar la salida de la última capa convolucional. El vector de características obtenido para una imagen es de una dimensión de 2048.

## 3.5. Redes neuronales recurrentes

Las redes neuronales recurrentes constituyen una herramienta muy apropiada para modelar series de tiempo. Se trata de un tipo de redes con una arquitectura que implementa una cierta memoria y, por lo tanto, un sentido temporal. En una red neuronal tradicional suponemos que todas las entradas son independientes entre sí. Sin embargo, para muchas aplicaciones no es la mejor aproximación. Por ejemplo, si se desean predecir la siguiente palabra en una oración, se requiere conocer las palabras predecesoras para realizar la predicción. Las RNN se llaman recurrentes ya que realizan el mismo procesamiento para cada elemento de una secuencia y la salida depende de los cálculos previos, es decir, las RNN poseen cierta memoria que captura la información sobre lo que se ha calculado hasta ahora. De manera teórica las redes recurrentes pueden usar información en secuencias largas, pero de manera práctica no es posible. Esto se debe al problema del desvanecimiento del gradiente [76] que es una dificultad que se encuentra en el entrenamiento de redes neuronales artificiales con métodos de aprendizaje basados en gradiente y retropropagación. Con estos métodos, cada peso recibe una actualización proporcional al gradiente de la función de error respecto al valor del peso en ese instante. El algoritmo de retropropagación calcula los gradientes. El problema es que, en algunos casos, el gradiente se vuelve infinitamente pequeño, lo que impide que los pesos de algunas capas cambien su valor y, por lo tanto esto puede ocasionar que la red neuronal deje de aprender.

### 3.5.1. Memorias de corto y largo plazo

Las redes recurrentes de memoria de corto y largo plazo (LSTM, por sus siglas en inglés) son ampliamente utilizadas ya que capturan mejor las dependencias temporales, es decir, es capaz de recordar la información durante considerables períodos de tiempo [51].

El componente fundamental de una LSTM es el bloque de memoria, el cual contiene una o más celdas con estado $\mathbf{C}_t$ en un tiempo $t$ y compuertas que tienen el objetivo de controlar el flujo de información. Estas últimas son unidades multiplicativas compartidas que pertenecen a un mismo bloque de memoria. En una LSTM existen de tres tipos: entrada $\mathbf{i}_t$, olvido $\mathbf{f}_t$ y salida $\mathbf{o}_t$.

La Figura 3.11 ilustra el bloque de memoria de la LSTM. La celda toma tres entradas: la entrada del paso de tiempo actual $\mathbf{x}_t$, la salida de la unidad LSTM anterior $\mathbf{h}_{t-1}$ y la memoria de la unidad anterior $\mathbf{C}_{t-1}$. Asimismo, la LSTM tiene dos salidas: la memoria del bloque $\mathbf{C}_t$ y la salida $\mathbf{h}_t$ actual. Por lo tanto, esta unidad

individual toma una decisión al considerar la entrada actual, la salida anterior y la memoria anterior. Y genera una nueva salida y altera su memoria.

El objetivo de la puerta de olvido es controlar qué información excluir de la memoria $\mathbf{f}_t$.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f). \tag{3.26}$$

Donde $[\mathbf{h}_{t-1}, \mathbf{x}_t]$ denota la concatenación de los vectores. La compuerta de entrada controla qué nueva información se agrega al estado de la celda $\tilde{\mathbf{C}}_t$ de la entrada $\mathbf{i}_t$.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i), \tag{3.27}$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C). \tag{3.28}$$

La actualización de la celda $\mathbf{C}_t$ es la suma de la memoria antigua regulada por la compuerta de olvido y la nueva memoria regulada por la compuerta de entrada.

$$\mathbf{C}_t = \mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{C}}_t. \tag{3.29}$$

Finalmente, la salida final del bloque de memoria $\mathbf{h}_t$ se calcula multiplicando la activación de la compuerta de salida $\mathbf{o}_t$ con el estado de la celda actualizado $\mathbf{C}_t$ pasado a través de una función tangencial hiperbólica.

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o), \tag{3.30}$$

$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{C}_t). \tag{3.31}$$

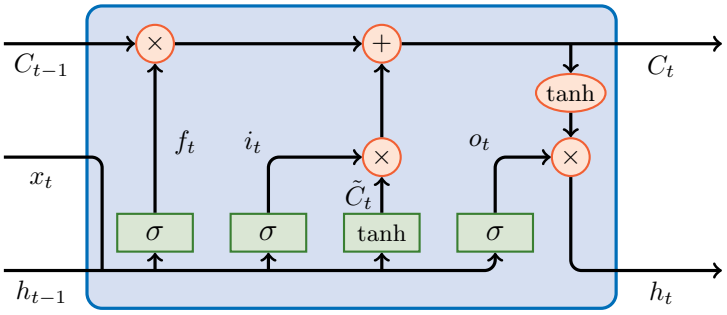



Figura 3.11. Diagrama del bloque de memoria de una LSTM.

Como se mencionó anteriormente, las redes neuronales recurrentes son un tipo de red neuronal donde los resultados de los pasos de tiempo anteriores se toman como entradas para el paso de tiempo actual. A lo que comúnmente se le conoce como desenrollar la red neuronal recurrente es el escribir la red completa para la secuencia de entrada.

El entrenamiento de este tipo de redes se realiza con el algoritmo de retropropagación a través del tiempo que es una ligera modificación del algoritmo de retropropagación anteriormente descrito. Esto es debido a que los pesos son compartidos

por todos los pasos de tiempo en la red, por lo tanto, el gradiente en cada salida depende no solo de los cálculos del paso de tiempo actual, sino también de los pasos de tiempo anteriores.

### 3.5.2. Unidad recurrente regulada

La unidad recurrente regulada es un tipo de red recurrente que posee solo dos compuertas a diferencia de la LSTM: una compuerta de reinicio $\mathbf{r}_t$ y una compuerta de actualización $\mathbf{z}_t$. La compuerta de reinicio determina cómo combinar la nueva entrada con la memoria anterior, mientras que la compuerta de actualización define cuánto de la memoria anterior se debe mantener. La Figura 3.12 ilustra el bloque de memoria de la unidad recurrente.

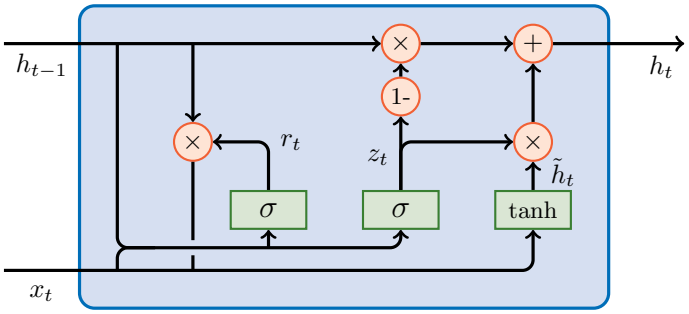



Figura 3.12. Diagrama del bloque de memoria de una GRU.

La salida $\mathbf{h}_t$ de la GRU está dada por una combinación entre la salida previa $\mathbf{h}_{t-1}$ y la nueva propuesta de salida $\tilde{\mathbf{h}}_t$.

$$\mathbf{h}_t = (1 - \mathbf{z}_t) * \mathbf{h}_{t-1} + \mathbf{z}_t * \tilde{\mathbf{h}}_t. \tag{3.32}$$

La compuerta de actualización $\mathbf{z}_t$ como su nombre lo indica determina qué tanto se debe actualizar, combinando la compuerta de entrada y olvido de una LSTM en una sola como sigue:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_z). \tag{3.33}$$

Finalmente, la nueva propuesta de salida $\mathbf{h}_t$ y la puerta de reinicio $\mathbf{r}_t$ están dadas por:

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_{\tilde{h}} \cdot [\mathbf{h}_{t-1} * \mathbf{r}_t, \mathbf{x}_t] + \mathbf{b}_{\tilde{h}}), \tag{3.34}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_r). \tag{3.35}$$

Las GRU fueron empledas para este trabajo debido a que tienen menos parámetros y por lo tanto, el entrenamiento es más rápido.

# 4 Base de datos

Una de las contribuciones de este proyecto es proporcionar a la comunidad de investigación una base de datos multi-etiqueta de avances cinematográficos la cual llamaremos $Trailers15k$.

$Trailers15k$ es una base de datos que cuenta con 15,000 videos obtenidos de la plataforma YouTube® con etiquetas provenientes de la base de datos cinematográfica en línea *Internet Movie Database* [77]. IMDb proporciona información de películas como géneros, sinopsis, año y equipo de producción. El código para la creación de la base de datos y los enlaces a los videos se encuentran disponible en [78]. En este capítulo se describe el proceso de cómo se realizó la construcción de la base de datos $Trailers15k$.

## 4.1. Construcción de la base de datos

La construcción de la base de datos se realizó por varias etapas las cuales se muestran en la Figura 4.1:


Figura 4.1. Diagrama de flujo de las etapas de creación de la base de datos.

### 4.1.1. Selección y obtención de los datos

La selección de los géneros de la base de datos se realizó con base en el número de registros disponibles en IMDb. Los géneros elegidos para formar la base de datos de avances de películas fueron las diez siguientes:

- Acción.
- Aventura.

- Animación.
- Comedia.
- Crimen.
- Drama.
- Fantasía.
- Horror.
- Ciencia ficción.
- Suspenso.

Es importante mencionar que se trata de una base de datos multi-etiqueta, es decir, cada uno de los títulos puede estar relacionado a uno o más géneros.

El etiquetado de estos géneros sigue siendo un proceso manual, el cual consiste en la recopilación de sugerencias de usuarios mediante correos electrónicos, es decir, se proporciona una definición de cada uno de los géneros de películas y, de acuerdo a estos, los usuarios recomiendan las etiquetas más apropiadas.

Debido a que el sitio no proporciona interfaz de programación de aplicaciones (API, por sus siglas en inglés), se necesitó realizar un programa cuyo objetivo fue inspeccionar y minar la información deseada del sitio. De este proceso, se obtuvieron los 2,000 títulos más recientes por género que fueron ingresados a IMDb hasta el momento de la inspección. La información que se obtuvo de IMDb para cada uno de los títulos fue la siguiente:

- Identificador.
- Título.
- Géneros.

Por otro lado, al acceder a una página web o sitio determinado, este hace previamente una petición que contiene la dirección IP, ubicación y un conjunto de datos técnicos y con base en ello regresa un resultado. En el caso de IMDb, la mayoría de los títulos originales de las películas disponibles están en inglés. Al realizar la descarga de la información desde una IP con ubicación en México, algunos de estos resultados de los títulos eran mostrados en inglés y otros en español. Por lo tanto, se hizo uso de un servidor virtual privado (VPS, por sus siglas en inglés) ubicado en Estados Unidos con el objetivo de que la información fuera lo más consistente posible. El total de títulos obtenidos fue de 18,881. En la Figura 4.2 se muestra la distribución de títulos por género.

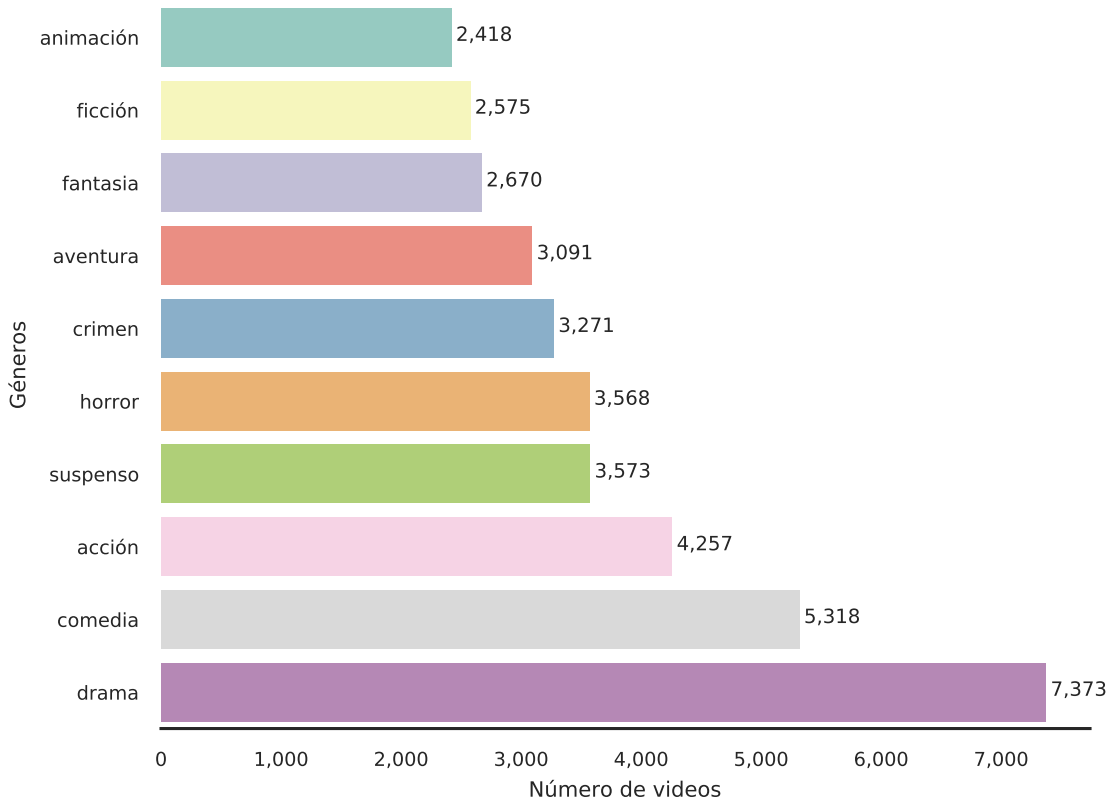



Figura 4.2. Distribución por género de los títulos obtenidos de IMDb.

### 4.1.2. Creación de la base de datos

Se creó una base de datos relacional en MySQL, la cual se compone de las siguientes tres tablas:

- *genres*.
- *movies*.
- *movies_genres*.

La tabla principal *movies* está formada por los siguientes campos:

- *id*: número de registro en la base de datos.
- *title*: título de la película.
- *imdbid*: identificador asociado a la película por el sitio de IMDb.
- *yturl*: enlace al avance de la película.
- *checked*: indicador de verificación manual.

- *downloaded*: identificador de descarga.

La tabla *genres* está formada por los siguientes campos:

- *id*: identificador de género.
- *name*: nombre de los diez distintos géneros.

La tabla *movies_genres* está formada por los siguientes campos:

- *movieid*: número de registro en la base de datos.
- *genreid*: identificador de género.

Una vez creada la base de datos se insertaron los datos obtenidos de IMDb. Posteriormente, se empleó un programa que manipuló la API de YouTube® para realizar la búsqueda de los enlaces a los avances cinematográficos de los títulos insertados. Una vez realizada la búsqueda de forma automática, se tomó el enlace al primer resultado arrojado y este fue insertado en el campo *yturl*. Para algunos de los títulos, el programa no encontró un enlace, por lo tanto el campo *yturl* quedó con un valor nulo.

### 4.1.3. Verificación de enlaces

Esta etapa consistió en revisar de forma manual y, para cada registro en la base de datos, que los enlaces a los avances cinematográficos obtenidos en la etapa previa correspondieran con el título e identificador de IMDb. Se insertó 1 en el campo *checked* en caso de que el enlace fuera correcto. Si el enlace no era el adecuado o el campo *yturl* estaba con valor nulo, se realizó una búsqueda manual del video en la plataforma de YouTube® con el objetivo de remplazar el enlace. En caso de no encontrar un video adecuado para el título se insertó 0 en el campo *checked*. Este proceso se realizó para cada una de las entradas en la base de datos. Después de realizar la verificación de los enlaces en la base de datos, el número de videos disponibles para los títulos fue de 15,000.

### 4.1.4. Descarga de videos

El objetivo de esta etapa fue descargar los videos mediante los enlaces previamente verificados. Para ello fue necesario realizar un programa que interactuó con el API de YouTube®. Sin embargo, algunos de los enlaces ya no se encontraban disponibles, pues los usuarios o videos habían sido eliminados. Además, al intentar descargar ciertos videos, el sitio no lo permitió debido a que contaban con restricciones de edad. Por lo tanto, se realizó una segunda verificación manual sobre los títulos cuyos enlaces fueron problemáticos con el objetivo de reemplazarlos e intentar descargarlos nuevamente. Por otra parte, algunos de los videos contaban con información ajena al avance cinematográfico como reseñas y publicidad, por lo cual estos archivos de video fueron recortados para solo conservar la información relevante.

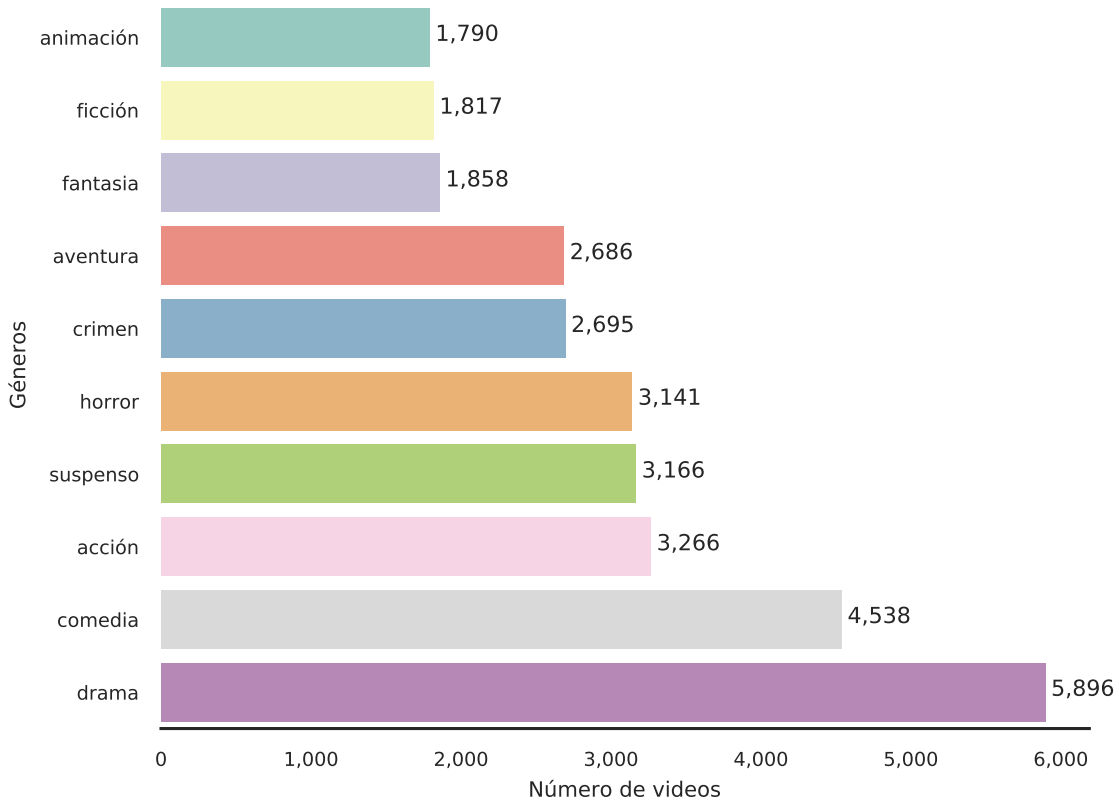



Figura 4.3. Distribución por género de los videos descargados.

El número total de videos descargados y del cual se compone la base de datos es de 15,000. La Figura 4.3 muestra el número de videos por género que fueron descargados, mientras que la Figura 4.4 muestra la distribución del número de etiquetas por video.

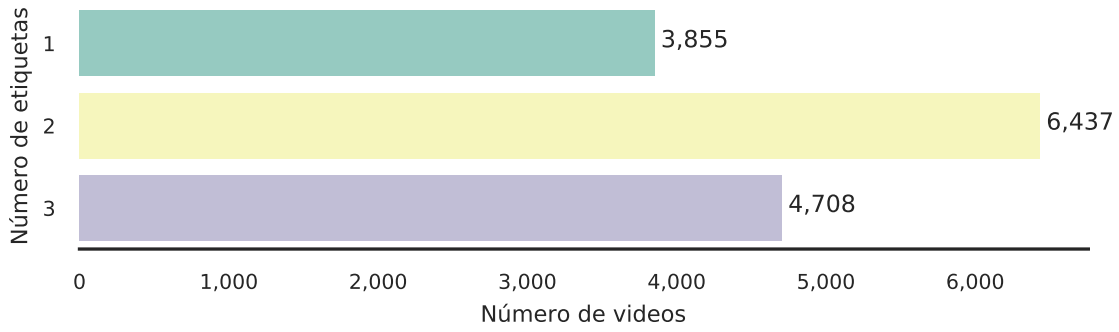



Figura 4.4. Distribución por número de etiquetas de los videos descargados.

De la Figura 4.4 se puede apreciar que existen videos que poseen asociados hasta tres géneros diferentes y que la mayoría de videos poseen dos géneros. La Tabla 4.1

muestra una comparativa de la base de datos $Trailers15K$ con las bases de datos de video más representativas.

| Base de datos | Clases | Videos | | | Etiquetado |
|---|---|---|---|---|---|
| | | Número | Resolución | Contenido | |
| $KTH$ | 5 | 600 | $160 \times 120$ | acciones humanas | uni-etiqueta |
| $WEIZMANN$ | 10 | 90 | $180 \times 144$ | acciones humanas | uni-etiqueta |
| $UCF - Sports$ | 9 | 182 | $720 \times 480$ | acciones en deportes | uni-etiqueta |
| $UCF - 50$ | 50 | 6,676 | $320 \times 240$ | acciones humanas | uni-etiqueta |
| $UCF - 101$ | 101 | 13,320 | $320 \times 240$ | acciones humanas | uni-etiqueta |
| $Hollywood$ | 8 | 430 | Variable | acciones películas | uni-etiqueta |
| $Hollywood2$ | 12 | 6,676 | Variable | acciones películas | uni-etiqueta |
| $Sports - 1M$ | 486 | 1,133,158 | Variable | deportes | uni-etiqueta |
| $YouTube - 8M$ | 4,716 | 8,264,650 | Variable | diverso | multi-etiqueta |
| **Trailers15K** | **10** | **15,000** | **$640 \times 360$** | **avances de películas** | **multi-etiqueta** |

Tabla 4.1. Tabla comparativa de las bases de video más representativas.

## 4.2. Características de los datos

La características de los videos pertenecientes a la base de datos son descritas a continuación:

- Duración. La duración de los videos depende del título de la película. La duración mínima de un video de esta base de datos es de 50 segundos mientras que la máxima es 240 segundos.
- Formato y resolución. El formato de descarga de los videos fue *MPEG-4*, que es un fomato de video muy empleado pues permite codificar el video en casi cualquier resolución. La plataforma YouTube® tiene disponibles las siguientes resoluciones de video:
  - 2160 pixeles: $3840 \times 2160$.

- 1440 pixeles: $2560 \times 1440$.
- 1080 pixeles: $1920 \times 1080$.
- 720 pixeles: $1280 \times 720$.
- 480 pixeles: $854 \times 480$.
- 360 pixeles: $640 \times 360$.
- 240 pixeles: $426 \times 240$.

Para la descarga de los videos se optó por la resolución de $640 \times 360$ pixeles. Se eligió esta resolución debido a que es suficiente para la tarea de clasificación propuesta y para otro tipo de tareas relacionadas.

- Almacenamiento. El espacio necesario para realizar el almacenamiento de los videos de la base es de 200 *gigabytes*.
- Número de países. Los títulos de películas disponibles en el sitio de IMDb corresponden a películas provenientes de todo el mundo. Sin embargo, los títulos de películas obtenidos para esta base de datos provienen de 41 países diferentes.
- Número de imágenes por segundo. El número de imágenes por segundo también llamados fotogramas por segundo (FPS, por sus siglas en inglés), es la velocidad a la cual un dispositivo muestra imágenes llamadas cuadros o fotogramas. En la industria del cine lo más común ha sido emplear 24 FPS. Sin embargo, poco a poco se ha ido incorporando y probando el uso de 48 FPS.

## 4.3. Partición de los datos

La distribución de los géneros en los videos de la base de datos fue mostrada en la Figura 4.3 del Capítulo 4. La división de la base de datos para los conjuntos de entrenamiento, validación y prueba fue 80 %, 10 % y 10 %, respectivamente.

Para respetar la distribución original de los géneros en los subconjuntos de entrenamiento, validación y prueba se empleó el siguiente procedimiento. Por cada género se creó una sublista con sus videos correspondientes ordenados aleatoriamente. Estas sublistas se ordenaron de menor a mayor con base en el número de videos. Para realizar división de la base de datos en los porcentajes mencionados de acuerdo a los géneros, se comenzó por la sublista con la menor cantidad de videos hasta terminar con la mayor.

Este método para la división de la base permitió generar conjuntos con un balance similar a la distribución original de los datos. La Tabla 4.2 muestra el porcentaje de videos destinados a entrenamiento, validación y prueba por género.

| Género | Entrenamiento % | Validación % | Prueba % |
|---|---|---|---|
| Acción | 79.85 | 10.16 | 9.98 |
| Aventura | 80.04 | 10.34 | 9.60 |
| Animación | 80.22 | 9.60 | 10.16 |
| Comedia | 79.46 | 10.68 | 9.85 |
| Crimen | 80.37 | 9.23 | 10.38 |
| Drama | 80.03 | 9.87 | 10.09 |
| Fantasía | 80.08 | 10.06 | 9.84 |
| Horror | 79.08 | 10.53 | 10.37 |
| Ficción | 79.69 | 10.45 | 9.85 |
| Suspenso | 79.12 | 10.20 | 10.67 |

Tabla 4.2. Porcentajes de videos por género para los conjuntos de entrenamiento, validación y prueba.

Las figuras 4.5, 4.7 y 4.6 muestran la distribución de lo videos por género para los conjuntos de entrenamiento, validación y prueba.

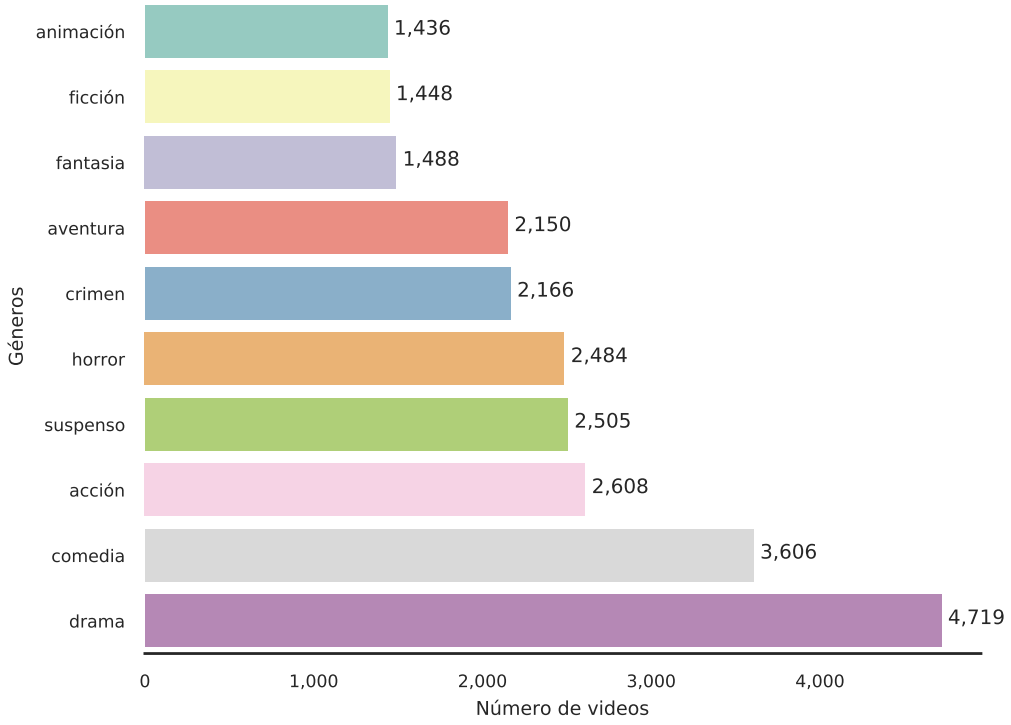



Figura 4.5. Distribución por género de los videos en el conjunto de entrenamiento.

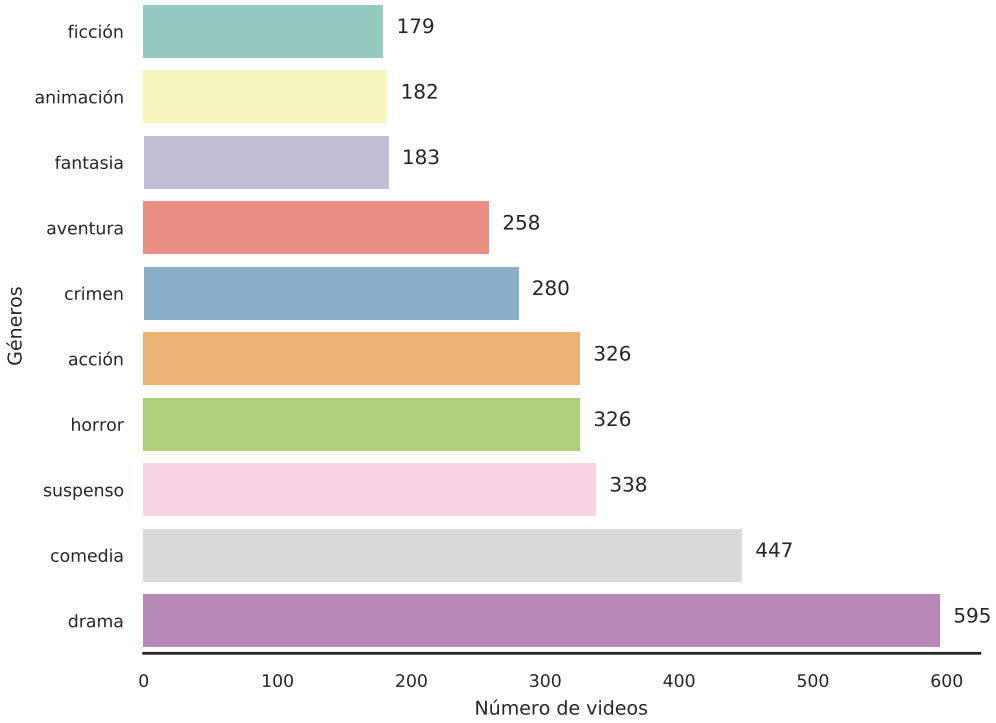



Figura 4.6. Distribución por género de los videos en el conjunto de validación.

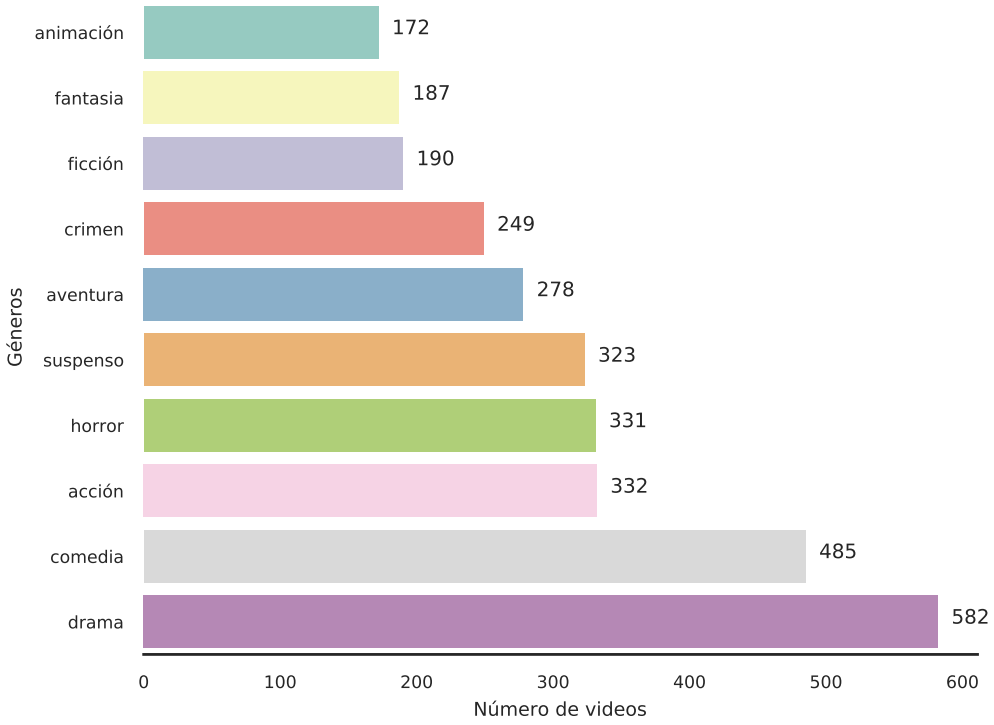



Figura 4.7. Distribución por género de los videos en el conjunto de prueba.

### 4.3.1. Selección de cuadros

Para la seleción de los 100 cuadros que representan un video en la base de datos *Trailers15k* se utilizó el algoritmo de medias de histograma de color. Considerando que un video originalmente tiene una secuencia de $n$ cuadros, el algoritmo particiona esta secuencia en 100 subsecuencias preservando el orden. Para seleccionar el cuadro más representativo en una subsecuencia, se calculan los histogramas de color de los cuados en la subsecuencia y se elige el más cercano a la media de histogramas de color utilizando mínimos cuadrados como medida de distancia. La base de datos solo incluyendo los archivos de video requiere un espacio de almacenamiento de 200 *gigabytes*.

# 5 Modelos de clasificación

En este capítulo se describen los modelos propuestos para realizar clasificación multi-etiqueta empleando la base de datos *Trailers15k*. Primeramente, se muestran dos modelos que explotan las características espaciales de los cuadros de vídeo sin tomar en cuenta la componente temporal. Posteriormente, se muestran dos modelos que incorporan las características espacio-temporales en la tarea de clasificación.

Para los modelos se utilizó la técnica de transferencia de conocimiento, tomando como modelo base la red neuronal convolucional Inception-v3. Se realizaron modificaciones en la arquitectura original de está red debido a que esta diseñada para realizar clasificación uni-etiqueta, mientras que el problema de clasificación de avances de películas es multi-etiqueta. Dentro de las modificaciones a la arquitectura original se realizó el remplazo de la función de activación *softmax* descrita en la Ecuación 3.5 en la capa de clasificación por múltiples funciones de activación *sigmoide* la cual fue descrita en la Ecuación 3.1. Por otra parte, se remplazó la función de error original descrita en la Ecuación 3.18 por la Ecuación 5.1 para medir el error de clasificación multi-etiqueta en un conjunto de clases $C$.

$$E(\mathbf{W}, \mathbf{b}) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{|C|} y_k^{(i)} \log(z_k^{(i)}) + (1 - y_k^{(i)}) \log(1 - z_k^{(i)}). \tag{5.1}$$

## 5.1. Clasificación espacial

En esta sección se presentan los modelos de clasificación multi-etiqueta que ignoran la información temporal en los videos. Uno de los objetivos de este proyecto de investigación es explotar la relaciones encontradas con la componente temporal. A fin de contar con un punto de referencia, se propusieron dos modelos diferentes de clasificación en los que no se tomó en cuenta esta componente.

### 5.1.1. Modelo Uni-Imagen-CNN

La primera aproximación de este proyecto es un modelo que intenta simplificar la tarea de clasificación de video a una de imagen, su arquitectura se muestra en la Figura 5.1.

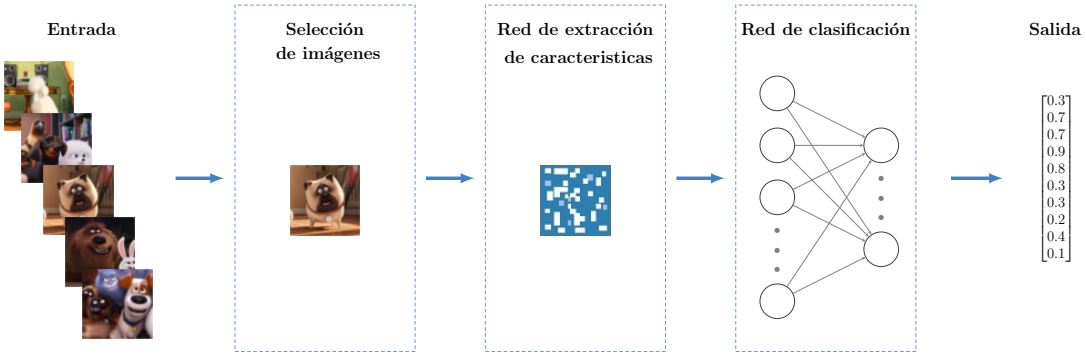



Figura 5.1. Arquitectura del modelo Uni-Imagen-CNN.

Las etapas de procesamiento de este modelo a partir de un ejemplo de video se describen a continuación:

- Entrada. Un video es representado por cien imágenes de $299 \times 299$ pixeles en RGB.
- Selección de imágen. Se toma el conjunto de cien cuadros de video y se selecciona el cuadro que se encuentra a la mitad de la secuencia. Al realizar esto, se simplifica la tarea de clasificación de video a una de clasificación de imágenes, ya que el video se representa a través de una sola imagen.
- Red de extracción de características. A partir de la imagen se obtiene un vector de características de 2048 empleando el modelo Inception-v3 pre-entrenado con ImageNet.
- Red de clasificación. El vector de características obtenido mediante la red de extracción se pasa a través de una capa completamente conectada con una capa de salida con 10 neuronas con función de activación *sigmoide* para obtener un vector con diez números reales en el intervalo $(0, 1)$, que representan las probabilidades de cada una de las etiquetas para el cuadro de entrada.

Este primer modelo omite la componente temporal de la tarea, lo que permite establecer un punto de referencia inicial para evaluar los modelos más sofisticados presentados en este proyecto. Además, permite comparar la dificultad de la tarea de clasificación de video multi-etiqueta con su contraparte uni-etiqueta, donde los modelos del estado del arte tienen una exactitud del 60 % empleando una aproximación de clasificación muy similar [18].

La métrica de exactitud clasificador uni-etiqueta [**sokolova˙systematic˙2009**] está dada por la siguiente expresión:

$$A = \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}. \tag{5.2}$$

Donde para la clase $i$, $tp_i$ es el número de verdaderos positivos, $tp_i$ es el número de verdaderos negativos, $fn_i$ es el número de falsos negativos y $fp_i$ es el número de falsos positivos.

### 5.1.2. Modelo Multi-Imagen-CNN-Promediación

El segundo modelo propuesto se muestra en la Figura 5.2. La idea detrás de este modelo es realizar clasificación sobre los cien cuadros tomados de cada video y posteriormente otorgar una clasificación a nivel video promediando las cien predicciones individuales de los cuadros.

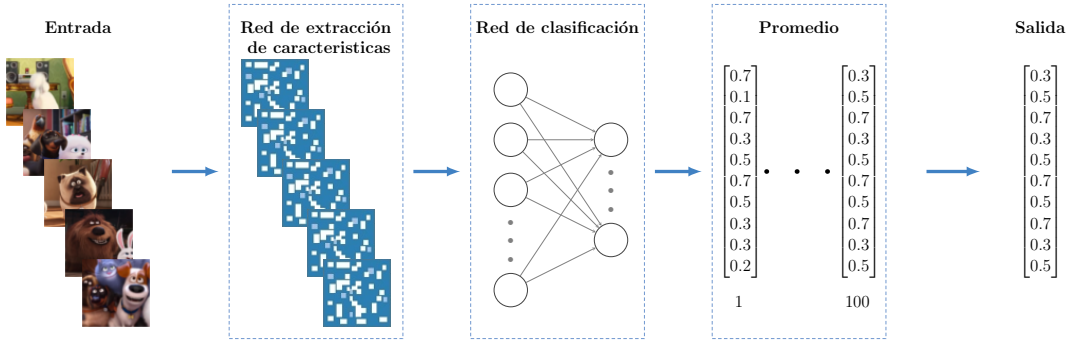



Figura 5.2. Arquitectura del modelo Multi-Imagen-CNN-promediación.

Las etapas de procesamiento de este modelo a partir de un ejemplo de video se describen a continuación:

- Entrada. Un video en este modelo es representado por cien cuadros. Cada cuadro es una imagen de $299 \times 299$ pixeles en RGB.
- Red de extracción de características. En este paso, cada una de las cien cuadros por video pasa a través del modelo Inception-v3, con lo que se obtienen cien vectores de características cada uno de tamaño de 2048.
- Red de clasificación. En esta etapa se toman los vectores de características extraídos y son clasificados mediante una capa completamente conectada con una capa de salida con 10 neuronas con función de activación *sigmoide*. Por cada vector de características que entra a la red se obtiene un vector con diez números reales en el intervalo $(0, 1)$, que representan las probabilidades de que cada cuadro pertenezca a cada una de las diez clases.
- Promedio. La salida de la etapa anterior es un conjunto de cien vectores donde cada entrada corresponde a un género distinto, es decir una etiqueta distinta. Por lo tanto, para otorgar una clasificación a nivel video se realizó un promedio por etiqueta de los cien cuadros.

Este modelo es una aproximación más completa para clasificar un video en comparación con el modelo anterior debido a que se toma en cuenta el conjunto total de cuadros que lo representan. Sin embargo, tomar el promedio de las etiquetas de cada cuadro es una operación que ignora las relaciones temporales entre los cuadros.

## 5.2. Clasificación espacio-temporal

En esta sección se muestran dos modelos de clasificación de video que a diferencia de los mostrados en la sección anterior incorporan la componente temporal. Con estos modelos se pretende aprender una descripción global de la evolución temporal del video para obtener una clasificación más acertada. Para aprender esta descripción global se empleó una red neuronal recurrente, pues este tipo de redes permiten hacer uso de la información secuencial. En particular para los modelos de este proyecto se empleó la red recurrente GRU.

### 5.2.1. Modelo Multi-imagen-CNN-PC-GRU

Este modelo es el primero en considerar la componente temporal para la tarea empleando una red recurrente GRU presentada en el Capítulo 3, su arquitectura se muestra en la Figura 5.3.

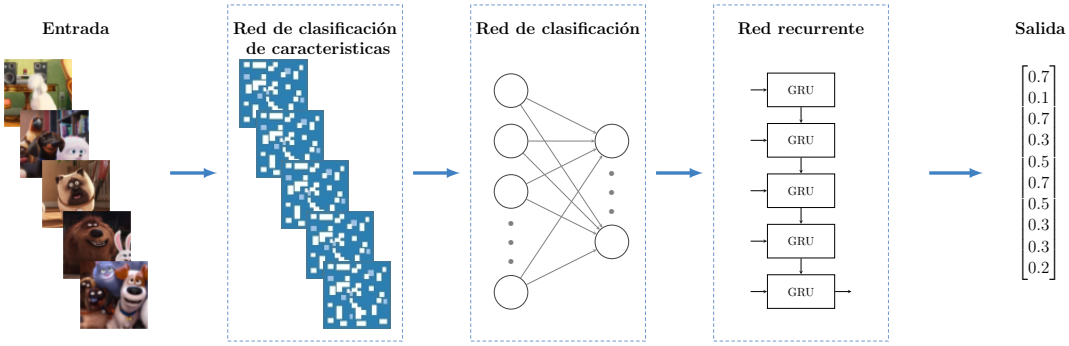



Figura 5.3. Arquitectura del modelo Multi-imagen-CNN-PC-GRU.

Las etapas de procesamiento de este modelo a partir de un ejemplo de video se describen a continuación:

- Entrada. Un video en este modelo es representado por cien cuadros. Cada cuadro es una imagen de $299 \times 299$ pixeles en RGB.
- Red de extracción de características. Por cada cuadro del video se produce un vector de 2048 características utilizando el modelo Inception-v3 pre-entrenado con ImageNet.
- Red de clasificación. En esta etapa se toman los vectores de características extraídos y son clasificados mediante una capa completamente conectada. Por cada vector de características que entra a la red se obtiene un vector con diez números reales que representan las probabilidades o puntajes de confianza (PC) de que cada cuadro pertenezca a cada una de las diez clases.
- Red recurrente. Cada vector de probabilidades es ingresado a una red recurrente GRU que está conectada a la unidad correspondiente del siguiente vector de probabilidades. La salida de la última unidad recurrente regulada pasa por

una función de activación *sigmoide* para obtener un vector con diez números reales en el intervalo $(0, 1)$.

En este modelo los vectores de probabilidades producidos por la capa de clasificación se ingresan como entrada a la red recurrente obteniendo nuevos vectores de probabilidades. Estos últimos han sido modificados por las unidades GRU de acuerdo a las relaciones temporales representadas por esta red, es decir, cada unidad GRU actúa como un regulador de las probabilidades de entrada tomando en cuenta la salida de la unidad asignada al vector de probabilidades anterior. De esta manera, este modelo explota la componente temporal de los cuadros de un video de una manera simplificada.

### 5.2.2. Modelo Multi-imagen-CNN-GRU

Este modelo se basa en el anterior eliminando la capa de clasificación intermedia. Su arquitectura se muestra en la Figura 5.4.

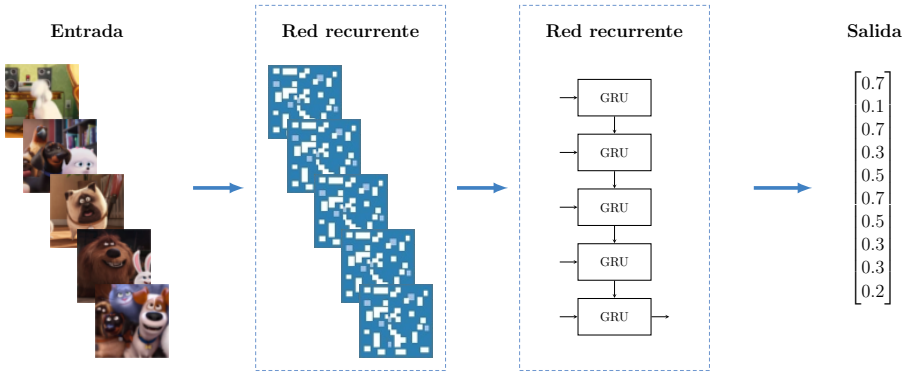



Figura 5.4. Arquitectura del modelo Multi-imagen-CNN-GRU.

Las etapas de procesamiento de este modelo a partir de un ejemplo de video se describen a continuación:

- Entrada. Un video en este modelo es representado por cien cuadros. Cada cuadro es una imagen de $299 \times 299$ pixeles en RGB.
- Red de extracción de características. Por cada cuadro del video se produce un vector de 2048 características utilizando el modelo InceptionV3.
- Red recurrente. Cada vector de características es ingresado a una unidad recurrente GRU que esta conectada a la unidad correspondiente del siguiente vector de características. La salida de la última unidad recurrente regulada pasa por una función de activación *sigmoide* para obtener un vector con diez números reales en el intervalo $(0, 1)$, el cual representa las probabilidades de cada una de las etiquetas para el cuadro de entrada.

En el modelo anterior, la red de clasificación limita severamente la cantidad de información que llega a la red recurrente, esto es, los vectores de características se reducen a vectores de probabilidades. Este nuevo modelo busca subsanar esta pérdida de información, conectando directamente la red de extracción de características a la red recurrente. De esta manera, se buscó que las unidades GRU tuvieran acceso a representaciones más ricas de los cuadros y pudieran explotarlas como parte de la tarea de clasificación.

# 6 Resultados experimentales

En este capítulo se describen los experimentos más significativos para cada uno de los modelos descritos en el Capítulo 5. Asimismo, se describen las métricas empleadas para evaluar el desempeño de los modelos propuestos. La implementación de los modelos se realizó en TensorFlow® [79] que es una biblioteca de código abierto orientada principalmente a la implementación eficiente de redes neuronales. El código de todos los modelos desarrollados se encuentra disponible en [80].

## 6.1. Evaluación de clasificación multi-etiqueta

La evaluación de un clasificador multi-etiqueta es más complicada que la de uno uni-etiqueta. En este último la predicción para un ejemplo consiste en una única etiqueta, por lo que la predicción se puede evaluar como correcta o incorrecta. Sin embargo, al evaluar un clasificador multi-etiqueta se debe tener en cuenta que la predicción para un ejemplo es un conjunto de etiquetas. Debido a esto, las diferentes estrategias de evaluación pueden considerar la predicción como totalmente correcta, parcialmente correcta o totalmente incorrecta. Para evaluar los modelos en este trabajo se optó por métricas con una estrategia de evaluación parcial [81], ya que estas proporcionan niveles granulares de corrección. En particular se eligieron las tres principales métricas usadas en la literatura [82] que son exactitud, precisión y exhaustividad.

Consideremos un conjunto de $m$ ejemplos, donde para el ejemplo $i$ se tiene que $Y_i$ es el conjunto de etiquetas reales y $\hat{Y}_i$ es el conjunto etiquetas predichas. Las métricas empleadas son las siguientes.

- Exactitud (A). Para cada ejemplo, se define como la razón del número de etiquetas predichas correctamente sobre el número total de etiquetas para esa instancia.
$$A = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|}. \tag{6.1}$$

- Precisión (P). Para cada ejemplo, se define como la razón del número de etiquetas predichas correctamente sobre el número total de etiquetas predichas para esa instancia.
$$P = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap \hat{Y}_i|}{|\hat{Y}_i|}. \tag{6.2}$$

- Exhaustividad (R). Para cada ejemplo, se define como la razón del número de etiquetas predichas correctamente sobre el número total de etiquetas reales

para esa instancia.

$$R = \frac{1}{m} \sum_{i=1}^{m} \frac{|Y_i \cap \hat{Y}_i|}{|Y_i|}. \tag{6.3}$$

## 6.2. Configuración de entrenamiento

De manera general todos los modelos se entrenaron con 200 épocas. El algoritmo de optimización empleado para los modelos fue el gradiente descendente. La probabilidad de deserción para los modelos que emplean la red de clasificación fue del 50 % como se recomiendan en [72].

## 6.3. Modelo Uni-imagen-CNN

Este modelo fue entrenado tomando una sola imagen por video en el conjunto de entrenamiento. El número total de imágenes con los que fue entrenado el modelo fue de 12,000, mientras que se emplearon 1,500 para la validación y 1,500 imágenes de prueba.

En particular, para este modelo se realizaron dos experimentos con tasa de aprendizaje diferentes. La Tabla 6.1 resume los resultados para la métrica de exactitud para ambos experimentos.

| Tasa de apendizaje | Exactitud % | | |
|---|---|---|---|
| | Entrenamiento | Validación | Prueba |
| .01 | 17.87 | 14.68 | 15.48 |
| **.001** | **17.00** | **14.90** | **15.39** |

Tabla 6.1. Resultados para la métrica de exactitud para el modelo Uni-imagen-CNN.

| Tasa de aprendizaje | Precisión % | | |
|---|---|---|---|
| | Entrenamiento | Validación | Prueba |
| .01 | 30.20 | 24.84 | 26.39 |
| **.001** | **29.20** | **25.94** | **26.65** |

Tabla 6.2. Resultados para la métrica de precisión para el modelo Uni-imagen-CNN.

Las tablas 6.2 y 6.3 resumen los resultados para los experimentos para la métricas de precisión y exhaustividad respectivamente.

Los resultados con este modelo sugieren que el problema de clasificación multi-etiqueta posee un mayor nivel de complejidad en comparación con su contraparte uni-etiqueta debido a que con una arquitectura muy similar estos últimos tienen una exactitud del 60 % [18].

| Tasa de aprendizaje | Exhaustividad % | | |
|---|---|---|---|
| | Entrenamiento | Validación | Prueba |
| .01 | 19.12 | 16.07 | 16.52 |
| **.001** | **18.28** | **16.18** | **16.45** |

Tabla 6.3. Resultados para la métrica de exhaustividad para el modelo Uni-imagen-CNN.

## 6.4. Modelo Multi-imagen-CNN-promediación

Este modelo se entrenó con 1,200,000 cuadros, 100 por cada video del conjunto de entrenamiento. A diferencia del modelo anterior, para obtener una predicción a nivel video se promedian las predicciones de todos los cuadros. Por lo tanto, para este modelo el conjunto de validación fue de 150,000 imágenes, al igual que el conjunto de prueba.

Para este modelo se realizaron dos experimentos con las mismas tasas de aprendizaje del modelo anterior. La Tabla 6.4, resume los resultados para la métrica de exactitud para ambos experimentos.

| Tasa de aprendizaje | Exactitud % | | |
|---|---|---|---|
| | Entrenamiento | Validación | Prueba |
| .01 | 17.61 | 17.04 | 17.84 |
| **.001** | **24.29** | **24.27** | **23.70** |

Tabla 6.4. Resultados para la métrica de exactitud para el modelo Multi-imagen-CNN-promediación.

| Tasa de aprendizaje | Precisión % | | |
|---|---|---|---|
| | Entrenamiento | Validación | Prueba |
| .01 | 29.08 | 28.60 | 29.77 |
| **.001** | **37.05** | **37.28** | **36.91** |

Tabla 6.5. Resultados para la métrica de precisión para el modelo Multi-imagen-CNN-promediación.

Las tablas 6.5 y 6.6, resumen los resultados para los experimentos del modelo para la métricas de precisión y exhaustividad respectivamente.

| Tasa de aprendizaje | Exhaustividad % | | |
|---|---|---|---|
| | Entrenamiento | Validación | Prueba |
| .01 | 21.09 | 20.33 | 21.39 |
| **.001** | **30.67** | **30.68** | **29.89** |

Tabla 6.6. Resultados para la métrica de exhaustividad para el modelo Multi-imagen-CNN-promediación.

## 6.5. Modelo Multi-imagen-CNN-PC-GRU

Este modelo se entrenó con 1,200,000 cuadros, 100 por cada video del conjunto de entrenamiento. El conjunto de validación y prueba para este modelo fue de 150,000 imágenes.

Para este modelo como se mencionó en el Capítulo 3, se empleó la red recurrente GRU. En los experimentos realizados para este modelo se varió tanto el número de capas de las redes recurrentes GRU como el tamaño de la memoria de las celdas. La tasa de aprendizaje para este modelo fue de 0.00005. Las redes recurrentes GRU tuvieron una deserción del 50 %. La Tabla 6.7 resume los experimentos y resultados obtenidos para este modelo para la métrica de exactitud.

| GRU | | Exactitud % | | |
|---|---|---|---|---|
| Capas | Memoria | Entrenamiento | Validación | Prueba |
| 2 | 16 | 31.25 | 30.93 | 31.07 |
| **2** | **32** | **32.62** | **32.62** | **33.09** |
| 4 | 16 | 29.11 | 28.69 | 28.45 |
| 4 | 32 | 29.32 | 28.99 | 29.00 |

Tabla 6.7. Resultados para la métrica de exactitud para el modelo Multi-imagen-CNN-PC-GRU.

| GRU | | Precisión % | | |
|---|---|---|---|---|
| Capas | Memoria | Entrenamiento | Validación | Prueba |
| 2 | 16 | 50.42 | 49.37 | 50.97 |
| **2** | **32** | **53.74** | **52.95** | **54.61** |
| 4 | 16 | 48.22 | 47.66 | 48.43 |
| 4 | 32 | 49.61 | 48.80 | 49.76 |

Tabla 6.8. Resultados para la métrica de precisión para el modelo Multi-imagen-CNN-PC-GRU.

Las tablas 6.8 y 6.9 resumen los resultados para los experimentos del modelo para la métricas de precisión y exhaustividad respectivamente.

| GRU | | Exhaustividad % | | |
|---|---|---|---|---|
| Capas | Memoria | Entrenamiento | Validación | Prueba |
| 2 | 16 | 33.40 | 33.40 | 33.07 |
| **2** | **32** | **34.74** | **35.03** | **35.42** |
| 4 | 16 | 30.76 | 30.70.66 | 30.28 |
| 4 | 32 | 30.66 | 30.37 | 30.51 |

Tabla 6.9. Resultados para la métrica de exhaustividad para los experimentos del modelo Multi-imagen-CNN-PC-GRU.

De los resultados se puede apreciar que incrementar el número de capas de la red recurrente GRU no contribuyó a mejorar la exactitud del modelo como lo hizo el incrementar el tamaño de la memoria de las celdas.

## 6.6. Modelo Multi-imagen-CNN-GRU

Este modelo se entrenó con 1,200,000 cuadros, 100 por cada video del conjunto de entrenamiento. El conjunto de validación y prueba para este modelo fue de 150,000 imágenes. La diferencia de este modelo con respecto al anterior es que la entrada de la red recurrente son los vectores de características extraídos mediante la red convolucional.

Para este modelo, como en el anterior, se empleó una red recurrente GRU. En los experimentos realizados para este modelo se varió tanto el número de capas de la GRU como el tamaño de la memoria de las celdas, al igual que en el modelo Multi-imagen-CNN-PC-GRU. La tasa de aprendizaje para este modelo fue de .001. Las redes recurrentes GRU tuvieron una deserción del 50 %. La Tabla 6.10 resume los experimentos y los resultados obtenidos para este modelo para la métrica de exactitud.

| GRU | | Exactitud % | | |
|---|---|---|---|---|
| Capas | Memoria | Entrenamiento | Validación | Prueba |
| 2 | 16 | 53.87 | 44.49 | 44.48 |
| 2 | 32 | 63.18 | 45.22 | 45.07 |
| 4 | 16 | 52.04 | 43.31 | 43.82 |
| **4** | **32** | **66.23** | **44.22** | **45.68** |

Tabla 6.10. Resultados para la métrica de exactitud para el modelo Multi-imagen-CNN-GRU.

La tablas 6.11 y 6.12 resumen los resultados para los experimentos del modelo para la métricas de precisión y exhaustividad respectivamente.

| GRU | | Precisión % | | |
|---|---|---|---|---|
| Capas | Memoria | Entrenamiento | Validación | Prueba |
| 2 | 16 | 76.20 | 66.29 | 65.41 |
| 2 | 32 | 80.80 | 64.72 | 63.39 |
| 4 | 16 | 74.59 | 65.49 | 65.26 |
| **4** | **32** | **82.92** | **62.76** | **63.21** |

Tabla 6.11. Resultados para la métrica de precisión para el modelo Multi-imagen-CNN-GRU.

| GRU | | Exhaustividad % | | |
|---|---|---|---|---|
| Capas | Memoria | Entrenamiento | Validación | Prueba |
| 2 | 16 | 60.13 | 50.55 | 51.75 |
| 2 | 32 | 71.37 | 54.56 | 55.06 |
| 4 | 16 | 58.62 | 50.54 | 50.89 |
| **4** | **32** | **74.18** | **53.48** | **56.11** |

Tabla 6.12. Resultados para la métrica de exhaustividad el modelo Multi-imagen-CNN-GRU.

En particular, para este modelo el incrementar el tamaño de la memoria de las celdas y el número de capas mejoró el desempeño del modelo.

## 6.7. Discusión de resultados

De los resultados anteriores se puede concluir que el tipo de arquitectura de la red neuronal y la complejidad del modelo en conjunto contribuyen a capturar de manera adecuada las características espacio-temporales de los videos y así producir predicciones más correctas. Las tablas 6.13 y 6.14 resumen los mejores resultados obtenidos para el mejor experimento de cada modelo para el conjunto de validación y prueba, respectivamente. De las tablas se puede apreciar que el modelo Multi-imagen-CNN-GRU obtuvo los mejores resultados.

| Modelo | Métricas % | | |
|---|---|---|---|
| | Exactitud | Precisión | Exhaustividad |
| Uni-imagen-CNN | 14.90 | 25.94 | 16.18 |
| Multi-imagen-CNN-promediación | 24.27 | 37.28 | 30.68 |
| Multi-imagen-CNN-PC-GRU | 32.62 | 52.95 | 35.03 |
| **Multi-imagen-CNN-GRU** | **44.22** | **62.76** | **53.48** |

Tabla 6.13. Resultados del conjunto de validación para las métricas para todos los modelos.

| Modelo | Métricas % | | |
|---|---|---|---|
| | Exactitud | Precisión | Exhaustividad |
| Uni-imagen-CNN | 15.39 | 26.65 | 16.45 |
| Multi-imagen-CNN-promediación | 23.70 | 36.91 | 29.89 |
| Multi-imagen-CNN-PC-GRU | 33.09 | 54.61 | 35.42 |
| **Multi-imagen-CNN-GRU** | **45.68** | **66.21** | **56.11** |

Tabla 6.14. Resultados del conjunto de prueba para las métricas para todos los modelos.

En particular, para cada modelo podemos hacer los siguientes comentarios.

- Modelo Uni-imagen-CNN. Los resultados obtenidos con este modelo indican que realizar la clasificación de video multi-etiqueta tomando únicamente un cuadro de video no es suficiente debido a que al tomar solamente un cuadro se pierde gran cantidad de información descriptiva en términos espaciales y temporales puesto que el cuadro seleccionado usualmente no incluye todas las características de los distintos géneros a los cuales puede estar asociado el video completo. Esto significaría querer simplificar el problema de clasificación de video a un problema de clasificación de una imagen.

- Modelo Multi-imagen-CNN-promediación. La arquitectura de la red neural convolucional empleada para la transferencia de conocimiento ha sido ampliamente usada para problemas de clasificación de imágenes mostrando excelentes resultados. Sin embargo, a pesar de que este modelo mostró un porcentaje de mejora en comparación con los dos previos, los resultados obtenidos con video no fueron los mejores. Por lo tanto, es necesario emplear otros métodos que permitan explotar las características que el video proporciona en comparación con las imágenes, es decir, se requiere emplear un modelo que permita incorporar la componente temporal con el objetivo de aportar mayor información a la tarea de clasificación.

- Modelo Multi-imagen-CNN-PC-GRU. Este modelo, siendo el primero que incorpora las características temporales con el uso de una red neuronal recurrente, presentó una mejora con respecto a los modelos que hacen uso únicamente de las características espaciales. Los resultados de este modelo sugieren que la red recurrente aprende características temporales de la secuencias de la entrada. No obstante, la entrada de este modelo es muy limitada dado que los vectores de características extraídos con la red convolucional se reducen a vectores de probabilidades mediante la capa completamente conectada.

- Modelo Multi-imagen-CNN-GRU. Este modelo recibe como entrada a la red recurrente las características extraídas mediante la red convolucional. A diferencia del modelo anterior, no existe una perdida de información entre la red convolucional y la red recurrente. Es decir, las características obtenidas con Inceptión-v3 proveen de una representación más completa de los cuadros de video a la red recurrente y con ello aprende una descripción temporal más rica.

# 7 Conclusiones

En este proyecto se presentan cuatro modelos diferentes para realizar la clasificación automática de video multi-etiqueta de avances cinematográficos clasificados en 10 géneros distintos. En primer lugar se presentan dos modelos que hacen uso únicamente de las características espaciales del video. Es decir, se simplifica la tarea original a una tarea de clasificación de imágenes y se explotan las características bidimensionales de los cuadros de video. Por otro lado, se presentan dos modelos que combinan las características espaciales y temporales. Esto se logró mediante una combinación de redes convolucionales y redes recurrentes GRU. Además, se detalla paso a paso el proceso que se realizó para recolectar la base de datos de avances de películas empleada en este proyecto y la cual es una de las contribuciones.

La base de datos recopilada en este proyecto cuenta con de 15,000 avances cinematográficos. Esta base de datos puede ser empleada para profundizar en el análisis de video y puede ser espacialmente útil para tareas específicas como lo es el resumen automático de video de manera supervisada, donde se requiere tener el video completo y un resumen del mismo.

Por otro lado, en cuanto a los modelos de clasificación. Los resultados indican que los modelos de clasificación que emplean únicamente las características espaciales son demasiado sencillos y presentan resultados de baja exactitud, esto sin importar el tamaño del conjunto de entrenamiento, lo cual insinúa que las características espacio-temporales son importantes. Por otro lado, se encontró que un modelo de cuadro único como Uni-imagen-CNN no es una buena aproximación como lo es en la tarea de clasificación uni-etiqueta debido a que la clasificación de múltiples etiquetas es una tarea más compleja y por lo tanto requiere más contexto. Además, el movimiento local de los cuadros puede ser críticamente importante para obtener una clasificación más apropiada.

Por otra parte, los dos modelos que emplean una red recurrente en su arquitectura, ambos motivados por la idea de que al incorporar información temporal a través de secuencias de cuadros permite una mejor clasificación de video. Los resultados indican que, si bien la exactitud no fue particularmente sensible con este tipo de arquitecturas que incorporan la información temporal, un modelo como Multi-imagen-CNN-GRU tiene un rendimiento mejor que las alternativas mostradas en Uni-imagen-CNN y Multi-imagen-CNN-promediación. Es importante notar que si bien existen dependencias temporales en los avances cinematográficos la información de las relaciones temporales plasmadas por los cuadros de estos videos no son lineales, es decir, una subsecuencia de cuadros puede representar una secuencia lineal o cronológica de un punto en la historia de la película completa. Sin embargo, los subgrupos de cuadros suelen estar fuera de orden temporalmente entre sí. Por ejemplo, una escena puede estar formada por un subgrupo de cuadros y estos pre-

sentan una secuencia lineal o cronológica; sin embargo, la siguiente escena puede no estar relacionada con la anterior lo cual dificulta el obtener la descripción temporal a nivel global del video.

## 7.1. Trabajo a futuro

A partir de la experiencia y los resultados obtenidos durante el desarrollo de este proyecto, las siguientes líneas de trabajo a futuro se consideran interesantes.

- Video cronológicamente lineal. Los modelos desarrollados en este proyecto fueron probados en un conjunto de datos de avances cinematográficos donde el orden cronológico no es lineal. Por lo tanto, sería interesante probar los mismos modelos en un conjunto de datos donde las relaciones secuenciales entre las escenas sean lineales.
- Redes recurrentes bidireccionales. Este tipo de redes permiten aprender relaciones a largo plazo hacia delante y hacia atrás en la secuencias de entrada, por lo que un modelo de este tipo podría contribuir a aprender mejores características espacio-temporales.
- Redes convolucionales recurrentes. Este tipo de redes permite incorporar la información secuencial dentro de las redes convoluciones, por lo que un modelo de este tipo podría contribuir a generar mejores características espacio-temporales.

# Bibliografía


[1] *Cisco Visual Networking Index: Forecast and Methodology, 2016–202*.

[2] Ross Girshick, Jeff Donahue, Trevor Darrell y Jitendra Malik. «Rich feature hierarchies for accurate object detection and semantic segmentation». En: *arXiv:1311.2524 [cs]* (11 de nov. de 2013). arXiv: 1311.2524.

[3] Alex Krizhevsky, Ilya Sutskever y Geoffrey Hinton. «ImageNet Classification with Deep Convolutional Neural Networks». En: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. USA: Curran Associates Inc., 2012, págs. 1097-1105.

[4] Andrej Karpathy y Li Fei-Fei. «Deep Visual-Semantic Alignments for Generating Image Descriptions». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.4 (abr. de 2017), págs. 664-676. ISSN: 0162-8828.

[5] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan y Stefan Carlsson. «CNN Features off-the-shelf: an Astounding Baseline for Recognition». En: *arXiv:1403.6382 [cs]* (23 de mar. de 2014). arXiv: 1403.6382.

[6] Charu Aggarwal, ed. *Data classification: algorithms and applications*. Chapman & Hall/CRC data mining and knowledge discovery series. Boca Raton: CRC Press, Taylor & Francis Group, 2014. 671 págs. ISBN: 978-1-4665-8674-1.

[7] Kevin Murphy. *Machine learning: a probabilistic perspective*. Adaptive computation and machine learning series. Cambridge, MA: MIT Press, 2012. 1067 págs. ISBN: 978-0-262-01802-9.

[8] Ian Goodfellow, Yoshua Bengio y Aaron Courville. *Deep learning*. OCLC: 987005922. 2016. ISBN: 978-0-262-33743-4.

[9] Yann LeCun, Yoshua Bengio y Geoffrey Hinton. «Deep learning». En: *Nature* 521 (27 de mayo de 2015), pág. 436.

[10] Joon Son Chung, Andrew Senior, Oriol Vinyals y Andrew Zisserman. «Lip Reading Sentences in the Wild». En: *arXiv:1611.05358 [cs]* (16 de nov. de 2016). arXiv: 1611.05358.

[11] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg y Demis Hassabis. «Human-level control through deep reinforcement learning». En: *Nature* 518.7540 (feb. de 2015), pág. 529. ISSN: 1476-4687.

[12] Emerging Technology from the arXiv. *Google Unveils Neural Network with "Superhuman" Ability to Determine the Location of Almost Any Image*. URL: https://www.technologyreview.com/s/600889/google-unveils-neural-network-with-superhuman-ability-to-determine-the-location-of-almost/ (visitado 13-12-2017).

[13] Johannes Stallkamp, Marc Schlipsing, Jan Salmen y Charles Igel. «Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition». En: *Neural Networks: The Official Journal of the International Neural Network Society* 32 (ago. de 2012), págs. 323-332. ISSN: 1879-2782.

[14] Jia Deng, Wei Dong, Richard Lippmann, Li-Jia Li, Kai Li y Li Fei-Fei. «ImageNet: A larg-scale hierarchical image database». En: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Jun. de 2009, págs. 248-255.

[15] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens y Zbigniew Wojna. «Rethinking the Inception Architecture for Computer Vision». En: *arXiv:1512.00567 [cs]* (1 de dic. de 2015). arXiv: 1512.00567.

[16] *Deep Residual Learning for Image Recognition - IEEE Conference Publication*. (Visitado 05-12-2017).

[17] Arnold Smeulders, Marcel Worring, Simone Santini, Anshul Gupta y Ramesh Jain. «Content-based image retrieval at the end of the early years». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.12 (dic. de 2000), págs. 1349-1380. ISSN: 0162-8828.

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar y Li Fei-Fei. «Large-Scale Video Classification with Convolutional Neural Networks». En: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Jun. de 2014, págs. 1725-1732.

[19] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga y George Toderici. «Beyond Short Snippets: Deep Networks for Video Classification». En: *arXiv:1503.08909 [cs]* (31 de mar. de 2015). arXiv: 1503.08909.

[20] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu y Xiaoqiang Zheng. «TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems». En: *arXiv:1603.04467 [cs]* (14 de mar. de 2016). arXiv: 1603.04467.

[21] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley y Yoshua Bengio. «Theano: new features and speed improvements». En: *arXiv:1211.5590 [cs]* (23 de nov. de 2012). arXiv: 1211.5590.

[22] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro y Evan Shelhamer. «cuDNN: Efficient Primitives for Deep Learning». En: *arXiv:1410.0759 [cs]* (3 de oct. de 2014). arXiv: 1410.0759.

[23] Andrew Lavin y Scott Gray. «Fast Algorithms for Convolutional Neural Networks». En: *arXiv:1509.09308 [cs]* (30 de sep. de 2015). arXiv: 1509.09308.

[24] Andrew Lavin. «maxDNN: An Efficient Convolution Kernel for Deep Learning with Maxwell GPUs». En: *arXiv:1501.06633 [cs]* (26 de ene. de 2015). arXiv: 1501.06633.

[25] Jeremy Appleyard, Tomas Kocisky y Phil Blunsom. «Optimizing Performance of Recurrent Neural Networks on GPUs». En: *arXiv:1604.01946 [cs]* (7 de abr. de 2016). arXiv: 1604.01946.

[26] Rob Fergus, Pietro Perona y Andrew Zisserman. «Object class recognition by unsupervised scale-invariant learning». En: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 2. Jun. de 2003, II–264–II-271 vol.2.

[27] Svetlana Lazebnik, Cordelia Schmid y Jean Ponce. «Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories». En: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. 2006, págs. 2169-2178.

[28] Jingen Liu, Jiebo Luo y Mubarak Shah. «Recognizing realistic actions from videos ïn the wild"». En: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Jun. de 2009, págs. 1996-2003.

[29] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev y Cordelia Schmid. «Evaluation of local spatio-temporal features for action recognition». En: British Machine Vision Association, 2009, págs. 124.1-124.11. ISBN: 978-1-901725-39-1.

[30] Heng Wang, Alexander Klaser, Cordelia Schmid y Cheng-Lin Liu. «Action recognition by dense trajectories». En: IEEE, jun. de 2011, págs. 3169-3176. ISBN: 978-1-4577-0394-2.

[31] Chris Harris y Mike Stephens. «A combined corner and edge detector». En: *In Proc. of Fourth Alvey Vision Conference*. 1988, págs. 147-151.

[32] Krzysztof Mikolajczyk y Cordelia Schmid. «Indexing based on scale invariant interest points». En: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001.* Vol. 1. Jul. de 2001, 525-531 vol.1.

[33] Guo Lv, Zhe Kong y Yue Li. «An Improved Algorithm for Extracting Video Frame Interest Point». En: *Journal of Software Engineering* 9.2 (1 de feb. de 2015), págs. 401-409. ISSN: 18194311.

[34] Jasper Uijlings, Cosmin Ionut Duta, Negar Rostamzadeh y Nicu Sebe. «Real-time Video Classification using Dense HOF/HOG». En: ACM Press, 2014, págs. 145-152. ISBN: 978-1-4503-2782-4.

[35] Ivan Laptev, Marcin Marszalek, Cordelia Schmid y Benjamin Rozenfeld. «Learning realistic human actions from movies». En: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008 IEEE Conference on Computer Vision and Pattern Recognition. Jun. de 2008, págs. 1-8.

[36] Claudiu Tanase y Bernard Merialdo. «Efficient Spatio-Temporal Edge Descriptor». En: *Advances in Multimedia Modeling*. Ed. por Klaus Schoeffmann, Bernard Merialdo, Alexander G. Hauptmann, Chong-Wah Ngo, Yiannis Andreopoulos y Christian Breiteneder. Vol. 7131. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, págs. 210-221.

[37] Paul Scovanner, Saad Ali y Mubarak Shah. «A 3-dimensional sift descriptor and its application to action recognition». En: ACM Press, 2007, pág. 357.

[38] Christian Schuldt, Ivan Laptev y Barbara Caputo. «Recognizing human actions: a local SVM approach». En: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. Ago. de 2004, 32-36 Vol.3.

[39] Leo Breiman. «Random Forests». En: *Machine Learning* 45.1 (1 de oct. de 2001), págs. 5-32. ISSN: 0885-6125, 1573-0565.

[40] Marcin Marszalek, Ivan Laptev y Cordelia Schmid. «Actions in context». En: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Jun. de 2009, págs. 2929-2936.

[41] Clement Farabet, Camille Couprie, Laurent Najman y Yann LeCun. «Learning Hierarchical Features for Scene Labeling». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (ago. de 2013), págs. 1915-1929. ISSN: 0162-8828.

[42] Christian Schuldt, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke y Andrew Rabinovich. «Going deeper with convolutions». En: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Jun. de 2015, págs. 1-9.

[43] *MNIST*. URL: `http://yann.lecun.com/exdb/mnist/`.

[44] Dan Cireşan, Ueli Meier y Juergen Schmidhuber. «Multi-column Deep Neural Networks for Image Classification». En: *arXiv:1202.2745 [cs]* (13 de feb. de 2012). arXiv: `1202.2745`.

[45] Sergey Ioffe y Christian Szegedy. «Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift». En: *arXiv:1502.03167 [cs]* (10 de feb. de 2015). arXiv: `1502.03167`.

[46] *Amazon Mechanical Turk*. URL: https://www.mturk.com/mturk/welcome.

[47] Karen Simonyan y Andrew Zisserman. «Two-Stream Convolutional Networks for Action Recognition in Videos». En: *arXiv:1406.2199 [cs]* (9 de jun. de 2014). arXiv: 1406.2199.

[48] Wei Xu y Ji Shuiwang. «3D Convolutional Neural Networks for Human Action Recognition». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (), págs. 221-231. ISSN: 0162-8828.

[49] Mihir Jain, Hervé Jégou y Patrick Bouthemy. «Better Exploiting Motion for Better Action Recognition». En: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Jun. de 2013, págs. 2555-2562.

[50] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia y Atilla Baskurt. «Sequential Deep Learning for Human Action Recognition». En: *Human Behavior Unterstanding*. Ed. por Albert Ali Salah y Bruno Lepri. Vol. 7065. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, págs. 29-39.

[51] Sepp Hochreiter y Jürgen Schmidhuber. «Long Short-Term Memory». En: *Neural Comput.* 9.8 (nov. de 1997), págs. 1735-1780. ISSN: 0899-7667.

[52] Alex Graves y Jürgen Schmidhuber. «Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks». En: *Proceedings of the 21st International Conference on Neural Information Processing Systems*. NIPS'08. USA: Curran Associates Inc., 2008, págs. 545-552. ISBN: 978-1-60560-949-2.

[53] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horts Bunke y Jurgen Schmidhuber. «A Novel Connectionist System for Unconstrained Handwriting Recognition». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.5 (mayo de 2009), págs. 855-868. ISSN: 0162-8828.

[54] Alex Graves, Abdel-rahman Mohamed y Geoffrey Hinton. «Speech recognition with deep recurrent neural networks». En: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Mayo de 2013, págs. 6645-6649.

[55] Ying Zhang, Mohammad Pezeshki, Philemon Brakel, Saizheng Zhang, Cesar Laurent Yoshua Bengio y Aaron Courville. «Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks». En: *arXiv:1701.02720 [cs, stat]* (10 de ene. de 2017). arXiv: 1701.02720.

[56] Martin Wöllmer, Moritz Kaiser, Florian Eyben, Björn Schuller y Gerhard Rigoll. «LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework». En: *Image and Vision Computing* 31.2 (feb. de 2013), págs. 153-163. ISSN: 02628856.

[57] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia y Atilla Baskurt. «Action Classification in Soccer Videos with Long Short-term Memory Recurrent Neural Networks». En: *Proceedings of the 20th International Conference on Artificial Neural Networks: Part II*. ICANN'10. Berlin, Heidelberg: Springer-Verlag, 2010, págs. 154-159. ISBN: 978-3-642-15821-6.

[58] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney y Kate Saenko. «Translating Videos to Natural Language Using Deep Recurrent Neural Networks». En: *arXiv:1412.4729 [cs]* (15 de dic. de 2014). arXiv: 1412.4729.

[59] Nitish Srivastava, Elman Mansimov y Ruslan Salakhutdinov. «Unsupervised Learning of Video Representations using LSTMs». En: *arXiv:1502.04681 [cs]* (16 de feb. de 2015). arXiv: 1502.04681.

[60] Zeeshan Rasheed y Mubarak Shah. «Movie genre classification by exploiting audio-visual features of previews». En: *Object recognition supported by user interaction for service robots*. Vol. 2. 2002, 1086-1089 vol.2.

[61] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani y Ronen Basri. «Actions As Space-Time Shapes». En: *IEEE Trans. Pattern Anal. Mach. Intell.* 29.12 (dic. de 2007), págs. 2247-2253. ISSN: 0162-8828.

[62] Kishore Reddy y Mubarak Shah. «Recognizing 50 Human Action Categories of Web Videos». En: *Mach. Vision Appl.* 24.5 (jul. de 2013), págs. 971-981. ISSN: 0932-8092.

[63] Khurram Soomro, Amir Roshan Zamir y Mubarak Shah. «UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild». En: *arXiv:1212.0402 [cs]* (3 de dic. de 2012). arXiv: 1212.0402.

[64] Dirk Holste, George Huo, Vivian Tung y Christopher Burge. «HOLLYWOOD: a comparative relational database of alternative splicing». En: *Nucleic Acids Research* 34 (Database issue 1 de ene. de 2006), págs. D56-62. ISSN: 1362-4962.

[65] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan y Sudheendra Vijayanarasimhan. «YouTube-8M: A Large-Scale Video Classification Benchmark». En: *arXiv:1609.08675 [cs]* (27 de sep. de 2016). arXiv: 1609.08675.

[66] Stephan Heermann y Nahid Khazenie. «Classification of multispectral remote sensing data using a back-propagation neural network». En: *IEEE Transactions on Geoscience and Remote Sensing* 30.1 (ene. de 1992), págs. 81-88. ISSN: 0196-2892.

[67] Reza Gharoie Ahangar y Mohammad Farajpoor Ahangar. «Handwritten Farsi Character Recognition using Artificial Neural Network». En: *arXiv:0908.4386 [cs]* (30 de ago. de 2009). arXiv: 0908.4386.

[68] Stefan Knerr, Léon Personnaz y Gérard Dreyfus. «Handwritten digit recognition by neural networks with single-layer training». En: *IEEE Transactions on Neural Networks* 3.6 (nov. de 1992), págs. 962-968. ISSN: 1045-9227.

[69] Singh Vijendra, Nisha Vasudeva y Hem Jyotsana Parashar. «Recognition of Text Image Using Multilayer Perceptron». En: *arXiv:1612.00625 [cs]* (2 de dic. de 2016). arXiv: 1612.00625.

[70] Richard Lippmann. «Review of Neural Networks for Speech Recognition». En: *Neural Computation* 1.1 (mar. de 1989), págs. 1-38. ISSN: 0899-7667.

[71] Warren McCulloch y Walter Pitts. «A logical calculus of the ideas immanent in nervous activity». En: *Bulletin of Mathematical Biology* 52.1 (1 de dic. de 1943), págs. 115-133. ISSN: 0007-4985, 1522-9602.

[72] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever y Ruslan Salakhutdinov. «Dropout: A Simple Way to Prevent Neural Networks from Overfitting». En: *J. Mach. Learn. Res.* 15.1 (ene. de 2014), págs. 1929-1958. ISSN: 1532-4435.

[73] Otávio Penatti, Keiller Nogueira y Jefersson dos Santos. «Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?» En: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Jun. de 2015, págs. 44-51.

[74] Daniel Sonntag, Michael Barz, Jan Zacharias, Sven Stauden, Vahid Rahmani, Áron Fóthi y András Lőrincz. «Fine-tuning deep CNN models on specific MS COCO categories». En: *arXiv:1709.01476 [cs]* (5 de sep. de 2017). arXiv: 1709.01476.

[75] Nimba Tajbakhsh, Jae Shin, Suryakanth Gurudu, Todd Hurst, Christopher Kendall, Michael Gotway y Jianming Liang. «Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?» En: *IEEE Transactions on Medical Imaging* 35.5 (mayo de 2016), págs. 1299-1312. ISSN: 0278-0062.

[76] Yann Bengio, Patrick Simard y Paolo Frasconi. «Learning long-term dependencies with gradient descent is difficult». En: *IEEE Transactions on Neural Networks* 5.2 (mar. de 1994), págs. 157-166. ISSN: 1045-9227.

[77] *IMDb - Movies, TV and Celebrities*. IMDb. URL: http://www.imdb.com/ (visitado 29-12-2017).

[78] *Trailers15k*. URL: https://bitbucket.org/Bere/trailers-tools (visitado 17-01-2018).

[79] *TensorFlow*. URL: https://www.tensorflow.org/ (visitado 29-12-2017).

[80] *Clasificación multi-etiqueta de videos cortos usando unidades recurrentes reguladas*. URL: https://bitbucket.org/Bere/mltc (visitado 17-01-2018).

[81] Shantanu Godbole y Sunita Sarawagi. «Discriminative Methods for Multi-Labeled Classification». En: *Advances in Knowledge Discovery and Data Mining*. Vol. vol. 3056. 18 de ago. de 2004.

[82] Mohammad Sorower. *A literature survey on algorithms for multi-label learning*. 2010.