

CARRERA: ESPECIALIZACIÓN EN CIENCIA DE DATOS

TRABAJO FINAL INTEGRADOR

“Identificación de actores en materiales filmográficos mediante el uso de reconocimiento facial.”

Nombre y Apellido del Alumno/a: Santiago Sabalain

Título de grado o posgrado (último): Ingeniero Industrial

Profesora:

Dra. Maria Juliana Gambini

Lugar y Fecha: Dec 13, 2022 , Madrid, España.

1. Introducción

La creciente demanda de productos cinematográficos en plataformas digitales (apalancada, en cierta parte, por la pandemia de COVID-19) se traduce en mayor cantidad de usuarios expuestos rutinariamente a una considerable cantidad de actores, lo que conlleva mayor desafío a la hora de identificar a los últimos por parte de los primeros.

A modo de mejorar la experiencia del usuario, se plantea una solución que permita facilitar la identificación de un actor en un determinado material filmográfico, contando únicamente con el nombre del material y el minuto en el cual aparece dicho actor. Esto es posible gracias al acceso a una base de datos que contenga esta información, la cual es obtenida mediante el uso de técnicas de reconocimiento facial y modelos de aprendizaje automático.

2. Antecedentes - Estado del arte - Marco teórico/Conceptual

En la actualidad, la industria del entretenimiento vinculada a los servicios de *Streaming* se encuentra en pleno crecimiento. El confinamiento mundial ocurrido en 2020, producto de la pandemia de COVID-19, provocó que muchísimas personas debieran permanecer en sus hogares y, en la mayoría de los casos, permitió contar con mayor tiempo de ocio (ya sea por merma en la intensidad laboral, ahorro en tiempos logísticos, etc). Esto se vió reflejado en un aumento de un 26% en los ingresos de estos servicios durante el año 2020, y se espera que esta tendencia continúe en los próximos años, performando una tasa de crecimiento anual compuesta de 21% desde el 2021 al 2028^[1].

Un mayor consumo de materiales cinematográficos se traslada en mayores ingresos para las compañías productoras, quienes destinan gran parte de estos ingresos en inversiones para nuevas producciones que puedan satisfacer la creciente demanda^[2]. Esto implica un incremento en el volumen de productos, los cuales requerirán más capital humano, entre ellos, actores.

Desde el punto de vista del consumidor, éste presenciará un incremento en el material disponible y, por ende, observará a una mayor cantidad de actores, muchos de los cuales no contarán con una gran trayectoria por los motivos explicados anteriormente. Esta combinación de factores hará que el espectador empiece a relacionar a un cierto actor de un determinado material con otros materiales en donde lo ha

visto recientemente, pero muy probablemente no pueda recordar varios datos (ya sea su nombre, otras series/películas donde ha trabajado, con qué otros actores lo ha hecho, etc).

En las últimas décadas, y gracias al crecimiento y la universalización de la tecnología, surgieron distintas plataformas que sirven de albergue para todo tipo de información relativa al ambiente cinematográfico. Uno de los repositorios de información con mayor difusión ha sido IMDb (“Internet Movie Database”), un sitio creado en 1990 que cuenta con datos (ya sea sobre producción, reparto, sinopsis, resúmenes, recaudaciones, etc), sobre películas, series, videojuegos y servicios de streaming, entre otros. Esta gran base de datos provee la posibilidad de identificar a los distintos actores que componen el reparto de un determinado material, lo que resulta útil siempre y cuando ese título se encuentre relevado en la base, así como también sus actores (tanto nombres como rostros). Hay que tener en cuenta que realizar un reconocimiento utilizando este método implica comprometer tiempo y recursos del espectador, ya que no solo debe encontrar en primera instancia la película/serie que está observando, sino que debe luego recorrer manualmente el listado de actores y asociar las fotografías allí expuestas con el rostro del actor que se desea reconocer (actividad que no siempre resulta exitosa).

La posibilidad de identificar de manera rápida y accesible la identidad de un actor, así como poder vincular al mismo con otros de sus trabajos realizados, implica una mejor experiencia de usuario al espectador. Actualmente, Amazon Prime incluye una función en algunos de sus títulos llamado “Amazon X-Ray”, el cual utiliza los servicios de reconocimiento facial de Amazon^[3] para identificar a los actores en un determinado momento del material cinematográfico y poder acceder de manera rápida a su perfil de IMDb. Este servicio se encuentra actualmente en desarrollo y no está disponible para todos los títulos disponibles en la plataforma.

3. Definición del problema

El problema consiste en la incapacidad de poder identificar personas en un determinado material cinematográfico (películas, series, largometrajes, etc), contando únicamente con el nombre del material y el instante temporal en el cual esas personas son exhibidas.

La ocurrencia de dicho problema depende de dos factores: por un lado, la experiencia del espectador y sus conocimientos en lo que refiere al mundo cinematográfico, y por otro lado, la popularidad o notoriedad que tenga un determinado grupo de actores. A modo de lograr la correcta identificación de los intérpretes de un determinado material, la relación entre ambos factores debe ser

inversamente proporcional: a menor popularidad del grupo a identificar, mayores deberán ser los conocimientos del espectador en la materia, y viceversa.

Comprender esto implica entender que el problema puede manifestarse en un amplio espectro de casos. La capacidad de solucionar la mayor cantidad de los mismos estará ligada a los recursos que se posean al momento de encarar el problema.

4. Justificación del estudio

El presente estudio busca analizar las distintas maneras en las cuales se puede llevar a cabo la identificación de actores en un material audiovisual, y evaluar de qué modo se logra arribar a una solución eficaz y escalable.

A diferencia de otros desarrollos actuales (como la solución implementada por Amazon Prime), este proyecto no busca construir un sistema que se encuentre integrado a un servicio de streaming en particular, sino que pretende estructurar un modelo que pueda ser alimentado con diversos materiales filmográficos y devolver, a modo de resultado, una base de datos con los mapeos entre actores, identidades y línea de tiempo obtenidos.

Por último, se busca también lograr un cierto grado de estandarización en la manera en la que se implementará la solución. Lograr definir procesos de manera clara, así como parámetros de entrada y salida claros y detallados, permiten la escalabilidad y reproducción de las técnicas descritas en este trabajo y establecen una metodología de trabajo por sobre el producto final.

5. Alcances del trabajo y limitaciones

El presente trabajo busca proveer una herramienta disponible para el público en general, intentando siempre que la preparación, calibración y puesta a punto de la misma sea sencilla y reproducible en los distintos ambientes en los que pueda ser utilizada. La construcción de una base de datos universal donde se vincule cada minuto de un material con los actores allí presentes, implica desde su planteamiento la necesidad de grandes capacidades de procesamiento, y es por este motivo en que se orienta a este estudio como el puntapié inicial de un proyecto colaborativo, donde cada usuario aporte recursos de procesamiento que expandan los registros disponibles en la base.

A su vez, existe un potencial interés de empresas que busquen mejorar la experiencia de usuario de los consumidores de materiales filmográficos. Sería posible crear un producto que se base en la solución planteada en el presente estudio y que permita a sus usuarios realizar la identificación de actores valiéndose sólo del nombre del material y el minuto que se desea analizar.

Tal y como se mencionó anteriormente, uno de los fundamentos de este estudio es definir un protocolo a llevar a cabo por todos los posibles usuarios. Para ello, es necesario establecer de antemano las técnicas y los algoritmos particulares que aseguren la confiabilidad del método implementado, evaluando la eficacia y eficiencia de los mismos. Por una limitante de tiempo y recursos, este estudio analizará los resultados aplicados a un universo reducido de películas, en particular enfocándose en las obras del director Christopher Edward Nolan.

6. Hipótesis

Es posible identificar a todos los actores que son exhibidos en un determinado instante de tiempo y para un determinado material cinematográfico.

6.1 Variables de la hipótesis

La hipótesis está compuesta por tres variables:

- Un determinado material cinematográfico: Es una variable **independiente** de tipo **contextual**, ya que delimita el ámbito de imágenes o cuadros a analizar.
 - *Definición nominal*: Un material cinematográfico es cualquier producción audiovisual realizada por el hombre con un fin artístico, educativo, publicitario o cultural.
 - *Definición operacional*: Se define así a una secuencia de imágenes cuya proyección en secuencia y a una determinada velocidad crean una ilusión óptica de movimiento.
- Un determinado instante de tiempo: Es una variable **independiente** de tipo **contextual**, ya que delimita un momento determinado dentro del material cinematográfico.
 - *Definición nominal*: Se trata de un momento particular de un video, que conduce a una imagen en particular.

- *Definición operacional:* Se define como un “timestamp” compuesto por las horas, minutos, segundos y milisegundos que han transcurrido desde el punto de partida (en el caso del presente estudio, el inicio del material cinematográfico).
- Los actores exhibidos: Se trata de una variable **dependiente, cualitativa** y **nominal**, cuyos valores serán un listado de nombres o identificadores, los cuales dependen de las dos variables antes mencionadas.
 - *Definición nominal:* Se trata del conjunto de personas que caracterizan a los distintos personajes que aparecen en un material audiovisual.
 - *Definición operacional:* Se compone de un listado de números que se corresponde con los identificadores individuales (claves) de una tabla con datos de personas humanas, ya sea el nombre completo o cualquier otro atributo que se considere pertinente.

La relación entre estas variables es de causa-efecto: Dado un determinado material cinematográfico y un determinado instante de tiempo del mismo, se obtiene un listado específico de actores exhibidos.

7. Objetivos

A continuación, se detallan los distintos objetivos planteados para este trabajo, partiendo de un único objetivo general y descomponiendo al mismo en pequeños objetivos específicos que sirven como directrices para el desarrollo del estudio.

7.1 Objetivo general

- Construir una herramienta que permita, mediante el uso de técnicas de reconocimiento facial, identificar a los actores que son exhibidos en un determinado momento de un material cinematográfico.

7.2 Objetivos específicos

- ❖ Diseñar una base de datos donde se guarde la información que vincula a los actores y los distintos instantes de tiempo.
- ❖ Identificar las distintas maneras en las que se puede construir una base de rostros y actores y definir los métodos a utilizar (*web scraping, datasets* ya consolidados, etc).

- ❖ Evaluar las distintas técnicas de reconocimiento facial disponibles y elegir la adecuada.
- ❖ Diseñar y entrenar un modelo que identifique a una o más personas en una imagen estática, las cuales se encuentran previamente cargadas en la base de rostros.
- ❖ Aplicar la lógica del modelo a un proceso en el cual se reciba un video como input y se devuelvan datos con la estructura definida en la base de datos del objetivo 1.
- ❖ Validar y analizar los resultados y la efectividad del modelo.
- ❖ Diseñar la infraestructura adecuada para poder escalar la herramienta y lograr manejar grandes volúmenes de datos.

8. Metodología

8.1 Metodología de trabajo

La metodología de trabajo a emplear durante el transcurso del estudio consiste en un esquema Scrum convencional para desarrollo de software, donde se definen las actividades a realizar en base a los objetivos específicos planteados, se planifica la realización de las mismas en periodos de 3 semanas (“Sprints”) y se realizan análisis retrospectivos de lo cumplido y del status general del desarrollo.

8.2 Técnicas y tecnologías a utilizar

Existen diversas tecnologías y herramientas que son utilizadas a lo largo del desarrollo del trabajo. La principal técnica a implementar, y la cual es el eje de este proyecto, es el reconocimiento facial.

8.2.1 Reconocimiento facial - Conceptos y definiciones generales

El **reconocimiento facial** es un método para identificar o confirmar la identidad de una persona a través del análisis de su rostro. Los sistemas de reconocimiento facial se pueden utilizar para identificar a las personas en fotos, videos o en tiempo real.

El reconocimiento facial comienza con la **detección facial**, es decir, el poder identificar la presencia de rostros de personas dentro de imágenes digitales. Esta identificación se logra utilizando técnicas de aprendizaje automático que logran determinar si hay uno o más rostros en una determinada imagen, independientemente de la identidad de dichos rostros.

Una vez lograda la detección de los rostros, el paso siguiente consiste en asociar a estos últimos con la identidad de una persona. El proceso funciona utilizando técnicas y algoritmos que identifican las características faciales mediante la extracción de puntos de referencia, o características, de una imagen de la cara del sujeto. Puede entenderse como “características faciales” a la posición relativa, el tamaño y / o la forma de los ojos, la nariz, los pómulos y la mandíbula. Dichas características deben ser posteriormente traducidas a un código numérico que es conocido como **huella facial**.

Análogamente a la huella dactilar, la huella facial de un individuo es única dentro de un mismo sistema. Es importante aclarar que esta unicidad es solo válida dentro de un sistema dado ya que, a diferencia de la huella dactilar, no existe hoy por hoy una manera universal y estandarizada de construir la huella facial, por lo que mismos individuos podrían tener huellas faciales diferentes según el sistema y la información que este posea. Las **bases de datos faciales** (o bases de reconocimiento facial) se componen, entonces, por al menos un dato que funcione como clave para identificar a una persona (desde identificadores subrogados hasta nombres completos, números de documentos de identidad, etc) y la huella facial construida para dicha persona. Estas bases son un elemento clave para poder implementar los distintos **algoritmos de reconocimiento facial** que se utilizan para llevar a cabo la tarea homónima.

8.2.2 Algoritmos y técnicas de reconocimiento facial utilizados en la actualidad

Existen en la actualidad tres grandes categorías de técnicas de reconocimiento facial^[6]:

1. Técnicas que utilizan imágenes de intensidad
2. Técnicas que tratan secuencia de video
3. Técnicas que utilizan datos sensoriales

8.2.2.1 Técnicas que utilizan imágenes de intensidad

Las técnicas que utilizan **imágenes de intensidad** son aquellas que se aplican sobre imágenes tradicionales (fotografías). Éstas se dividen en dos enfoques principales: aquellos basados en características y los holísticos.

8.2.2.1.1 Enfoque basado en características

Los enfoques basados en características reciben una imagen de entrada y proceden a mapear, extraer y medir los rasgos faciales distintivos del sujeto (ojos, boca o nariz). Hecho eso, se calculan las relaciones geométricas entre esos puntos faciales para obtener un vector de características geométricas (la huella facial). La técnica de Histograma de Gradientes Orientados (HOG) se encuentra comprendida en este enfoque.

8.2.2.1.2 Enfoque holístico

Este enfoque utiliza la descripción completa de la imagen, en lugar de usar características locales del rostro. Estos esquemas pueden ser subdivididos a su vez en dos grupos: enfoques estadísticos y de Inteligencia Artificial.

El enfoque estadístico consiste en la versión más simple de los métodos holísticos. En estos, la imagen es representada como una matriz de valores de intensidad y el reconocimiento se realiza mediante comparaciones de correlación entre la cara de entrada y las otras caras en la base de datos.

Los enfoques de inteligencia artificial son aquellos que utilizan herramientas como redes neuronales y técnicas de aprendizaje automático para lograr el reconocimiento facial. Desde el uso de Support Vector Machines (SVM) para clasificar patrones hasta redes neuronales convolucionales (CNNs), estos enfoques crecen en el día a día en relevancia y popularidad, ya que la limitante de recursos se reduce con el avance y abaratamiento de los costos de procesamiento de grandes volúmenes de datos.

8.2.2.2 Técnicas que tratan secuencia de video

Al momento de analizar secuencias de video, es necesario realizar una serie de operaciones que permitan obtener una imagen que sea plausible de análisis vía técnicas de imágenes de intensidad.

En primer lugar, se debe detectar aquellos fotogramas que contienen un rostro. Una vez detectado, se debe rastrear a ese rostro en fotogramas contiguos a modo de poder seleccionar aquellos de mayor calidad, y por último se aplicará la técnica de imagen de intensidad seleccionada.

8.2.2.3 Técnicas que utilizan datos sensoriales

Estas técnicas consisten en el procesamiento y manejo de datos que no se encuentran en el formato tradicional de las imágenes, sino que provienen de sensores particulares (como 3D o infrarrojos).

La ventaja del uso de estos modelos radica en poder desprenderse de los ruidos tradicionales que pueden hallarse en una fotografía (problemas de iluminación, orientación, fondo,etc), pero en contrapartida implican una mayor complejidad tanto en su implementación como en la obtención de los datos a analizar, así como también acarrean un alto costo computacional.

9. Desarrollo

9.1 Herramientas utilizadas

Tal y como se expuso en la sección 8.2.1, el reconocimiento facial puede desagregarse en tres etapas:

1. Una etapa de detección, donde se identifica la presencia de un rostro en una imagen;
2. Una etapa de representación, donde se traduce ese rostro en un identificador único y comparable, y
3. Una etapa de reconocimiento, donde se busca relacionar a ese identificador con una determinada entidad ya conocida (un actor, en este caso).

Es importante entender que en cada una de estas etapas se pueden utilizar técnicas o algoritmos diferentes. En las siguientes secciones se exponen los métodos considerados para resolver el problema, señalando aquellos que finalmente se utilizan para el desarrollo del estudio.

9.1.1 Detección facial

9.1.1.1 HOG vs SSD

En lo que refiere a detección facial, existen actualmente dos técnicas que se destacan por sobre el resto: el uso de Histograma de Gradientes Orientados ('HOG') combinado con el entrenamiento de un clasificador supervisado, y el uso de *Single Shot Detectors* ('SSD').

El método de *HOG*^[8] es un método de descripción de características (*feature descriptor*) que consiste en representar a una imagen mediante un vector que guarda las estructuras básicas más importantes de la misma. Para ello, se parte de una imagen, se la lleva a escala de grises y se la divide en recuadros uniformes con una determinada cantidad de píxeles cada uno. Luego se calcula, para cada uno de los píxeles dentro de un recuadro, el **vector gradiente** respecto de los píxeles que lo rodean. Este vector representa la dirección y el sentido hacia el cual la imagen se vuelve más oscura. Todos los vectores de un determinado recuadro son dispuestos en un histograma donde se agrupan según su orientación, para luego asignar la orientación predominante al recuadro. Esto permite reducir la cantidad de información con la que se trabaja dado que ya no hay un vector por cada píxel, sino con uno por cada recuadro (el cual termina siendo una representación de todos los vectores gradiente de sus píxeles contenidos).

Finalmente, la imagen de partida es representada por un único **vector de características** compuesto por la información de cada recuadro (conocido como “*HOG Features*”), que representa la estructura del objeto deseado (un rostro en este caso) y se deshace de cualquier característica sin importancia en la imagen. En las *Figuras 1* y *2* se puede apreciar, de manera gráfica, la representación de un rostro mediante vectores de gradientes orientados.



Figura 1. Imagen inicial

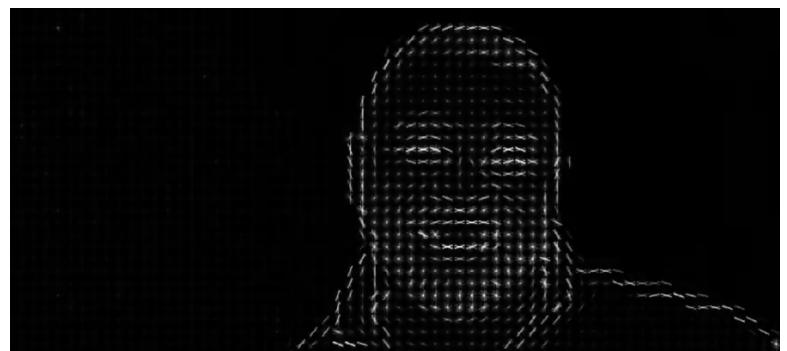


Figura 2. Imagen tras aplicar HOG

Estos *Features* pueden ser luego utilizados para alimentar cualquier algoritmo de aprendizaje automático que permita realizar una regresión o clasificación (como *Support Vector Machines* lineales, redes neuronales, etc.), y poder así detectar aquellos descriptores HOG que se asimilen a un rostro.

Por otro lado, existen los métodos basados **únicamente** en el uso de redes neuronales que son entrenadas con grandes cantidades de datos para identificar la presencia de rostros. Dentro de estos

modelos se encuentra un método de detección de objetos en imágenes denominado *Single Shot Detector (SSD)*^[9], el cual se basa en una única *Deep Neural Network (DNN)* que genera diferentes “cajas” o recuadros en una misma imagen y evalúa la posibilidad de que el recorte contenido en esa caja sea un determinado objeto. A través de distintas iteraciones, la posición y el tamaño de estas cajas se ajusta hasta obtener la mejor probabilidad de encontrar un determinado objeto en esa sección delimitada.

Finalmente, si se aplica sobre una imagen un modelo de este tipo entrenado para detectar rostros, el resultado será el recuadro que delimita los bordes del rostro (en caso de encontrarse uno).

9.1.1.2 Técnica seleccionada

En lo que respecta a este trabajo, se ha decidido utilizar para la detección facial una **red neuronal basada en SSD** y pre-entrenada por *OpenCV*^[10], una de las librerías públicas de *Computer Vision* más reconocidas y con mayor cantidad de colaboradores en la actualidad.

Los motivos que han determinado esta decisión han sido:

- La baja performance de los modelos *HOG* en detectar rostros que no se encuentran alineados o mirando de frente (lo cual resulta bastante inconveniente dada la naturaleza del proyecto, que es analizar películas, donde no todos son primeros planos frontales);
- La fácil escalabilidad que tienen los modelos basados en redes neuronales, al no tener que transformar imágenes en vectores y luego entrenar otros algoritmos (como el caso de *HOG*), sino poder realizar todo el aprendizaje en un mismo proceso, y
- La posibilidad de obtener un modelo ya entrenado con millones y millones de registros gracias al alcance del proyecto de *OpenCV*, y la fácil escalabilidad planteada en el punto anterior.

9.1.2 Representación facial

En lo que corresponde a la representación facial, y en línea con los argumentos desarrollados anteriormente, se decide utilizar un modelo basado en una *Deep Neural Network* pre-entrenada por *OpenFace*^[11], la cual se detalla en la publicación *FaceNet: A Unified Embedding for Face Recognition and Clustering*^[12].

Este modelo parte de imágenes de rostros, los cuales ya han sido detectados y recortados, y devuelve un vector de 128 dimensiones (conocido como “*Embeddings*”), el cual funciona como un identificador comparable del mismo. En la Figura 3 se puede apreciar un diagrama general del modelo.



Figura 3: Diagrama sobre cómo la DDN utilizada obtiene los embeddings de un rostro^[12]

El mecanismo utilizado por el modelo para computar este vector se centra en su proceso de entrenamiento en sí, a saber:

1. En primer lugar, cada “batch” de datos debe incluir tres imágenes, las cuales son:
 - a. El “ancla” (*‘Anchor’*): Esta imagen contiene el rostro a computar, identificado como “sujeto A”;
 - b. La “imagen positiva” (*‘Positive Image’*): Esta imagen también contiene el rostro del sujeto A, y
 - c. La “imagen negativa” (*‘Negative image’*): Contiene el rostro de otro sujeto que no sea el A (sujeto B, C, D, etc...).
2. La red, entonces, procesa este *input* y devuelve un primer vector por cada una de las imágenes. Estos vectores son luego sometidos a un proceso de normalización “L2” (donde cada vector obtenido es ajustado de manera tal que su módulo pase a ser la unidad), obteniendo así los primeros *embeddings*.
3. Finalmente, el modelo ajusta los pesos de la red buscando **minimizar** la distancia euclídea entre los *embeddings* del ancla y la imagen positiva, así como al mismo tiempo **maximizar** la distancia entre el ancla y la imagen negativa. En la Figura 4 vemos como luego de distintas etapas de entrenamiento se obtiene esta diferenciación. Este proceso de ajuste es muy similar al método de “Análisis de vecinos más próximos”, y se lo denomina ‘*Triplet Loss*’.

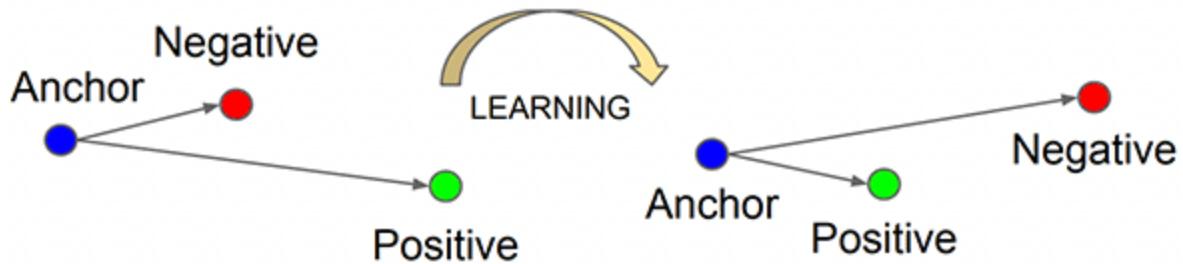


Figura 4: Proceso de ajuste de parámetros vía ‘Triplet Loss’.

De esta manera, la red logra cuantificar rostros y devuelve identificadores robustos y discriminatorios que sirven para llevar a cabo el reconocimiento facial.

9.1.3 Reconocimiento facial

Por último, el reconocimiento facial se realiza mediante el entrenamiento de un *SVM* que recibe una tabla de dos columnas generada a partir de varias imágenes de rostros de actores, y donde se enlistan en una columna los *embeddings* obtenidos a través de los pasos anteriores y un identificador único de la persona a la que pertenecen.

La implementación de modelo de aprendizaje supervisado se realiza mediante el uso de la librería de *Python* “*scikit-learn*” [13].

9.2 Infraestructura de trabajo

El desarrollo del trabajo se lleva adelante en un repositorio [14] con control de versión *GIT*, y se basa en una solución de contenedores de *software* provista por *Docker*. Esto proporciona una capa adicional de abstracción y permite definir distintos contenedores que simulan ser máquinas virtuales completamente personalizables, desde su sistema operativo hasta los programas instalados en cada uno de ellos. Dichas máquinas pueden comunicarse entre sí a través de redes internas predefinidas, y pueden también utilizar volúmenes de almacenamiento compartidos.

Dentro de las ventajas de utilizar *Docker*, encontramos:

- Desarrollar bajo un mismo ambiente predefinido en cada contenedor, sin tener que preocuparse por la compatibilidad de sistemas operativos o versiones de librerías;
- Trabajar de manera independiente en cada contenedor, pudiendo incluir u omitir ciertos módulos en caso de que haga falta, y
- Reducir considerablemente la barrera tecnológica en caso de querer ejecutar el proyecto en un servidor en la nube, y poder así escalar los recursos disponibles fácilmente.

En la Figura 5 es posible observar un diagrama con los contenedores utilizados. Dentro los distintas tecnologías utilizadas, encontramos:

- ❖ *Apache Airflow*, utilizada para orquestar y ejecutar procesos.
- ❖ *Apache Spark*, que posibilita el paralelismo y el cómputo distribuido.
- ❖ *Jupyter Notebooks*, a modo de obtener una interfaz de desarrollo en Python más amigable y orientada a las etapas de descubrimiento.
- ❖ *MySQL*, base de datos relacional donde almacenar distintos tipos de datos.
- ❖ *Selenium*, que permite realizar ‘*Web-Scraping*’ para automatizar tareas repetitivas.

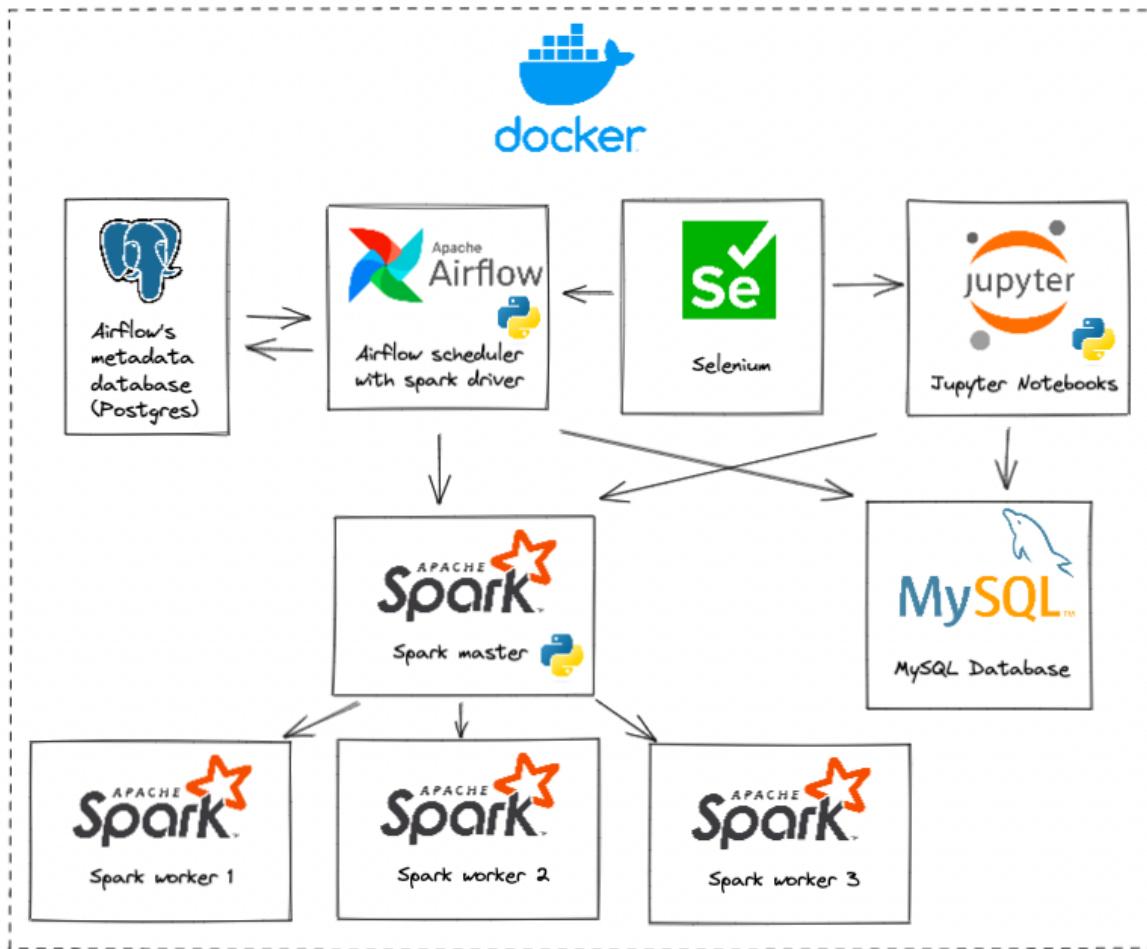


Figura 5: Infraestructura del proyecto

9.3 Implementación

9.3.1 Creación de la base de datos de actores

El punto de partida del proyecto consiste en generar una base de datos con información relativa a los actores que se encuentran en una determinada película. Esto permite acotar la cantidad de actores con la cual entrenar al modelo de reconocimiento a aplicar en un determinado filme, y de esta manera hacer un uso eficiente de los recursos disponibles.

Para lograr consolidar dicha base, se utiliza la API pública del sitio *The Movie Database*^[15], que posee toda la información necesaria relativa al elenco de un título. Utilizando *Python* para realizar la lectura de la API, se consume la información relativa a todas las películas de un determinado director, así

como a todos los actores que actúan en cada una de ellas. Dicha información se almacena en *MySQL*, creando una base de datos con el nombre del director seleccionado (ver Figura 6), y con tres tablas:

1. ‘Actors’, con información sobre los actores (nombres, género, popularidad, etc);
2. ‘Movies’, con información sobre las películas (título, fecha de lanzamiento, valoración de usuarios, etc), y
3. ‘Actors_by_movie’, donde se establece el vínculo entre actores y películas y se detalla el personaje interpretado en cada caso.

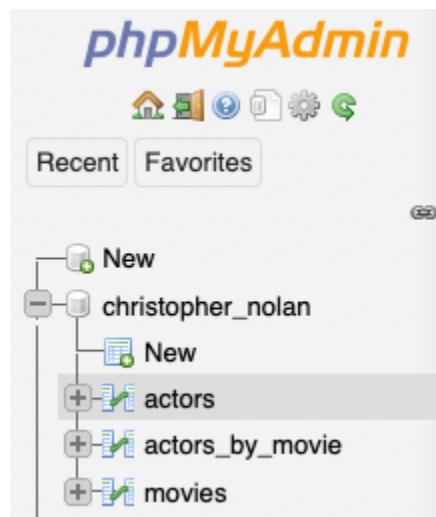


Figura 6: Tablas creadas para el director Christopher Nolan.

9.3.2 Descarga de imágenes faciales

Una vez creada la base de datos de un determinado director, se dispone de todos los nombres de los actores involucrados en sus distintas películas. Esta información resulta imprescindible para el desarrollo del modelo de reconocimiento facial del presente trabajo, ya que permite acotar con distintos niveles de granularidad (a nivel de director o a nivel de película) las personas a ser identificadas.

Con dicha información se procede a utilizar el módulo de *Selenium* para poder descargar imágenes de *Google Images*. *Selenium* es una herramienta de *Web Scrapping* que permite replicar el comportamiento de un humano al navegar por cualquier sitio de internet, y puede ser programada para realizar tareas repetitivas como descargar imágenes de *Google* (en este caso). Los parámetros que deben ser definidos para realizar dichas descargas son:

- Búsqueda a realizar (es decir, la frase a introducir en el buscador).
- Cantidad de imágenes a descargar por cada búsqueda.

Dado que en este caso se busca obtener imágenes del rostro de distintos actores, las búsquedas a realizar para un determinado actor tendrán la forma de “[Nombre del actor] face” o “[Nombre del actor] [Nombre de la película] face”.

Las imágenes descargadas en este paso son almacenadas en una carpeta compartida por todos los contenedores y compartimentadas de una manera estandarizada, utilizando los identificadores de los distintos actores en la base de datos (ver Figura 7).

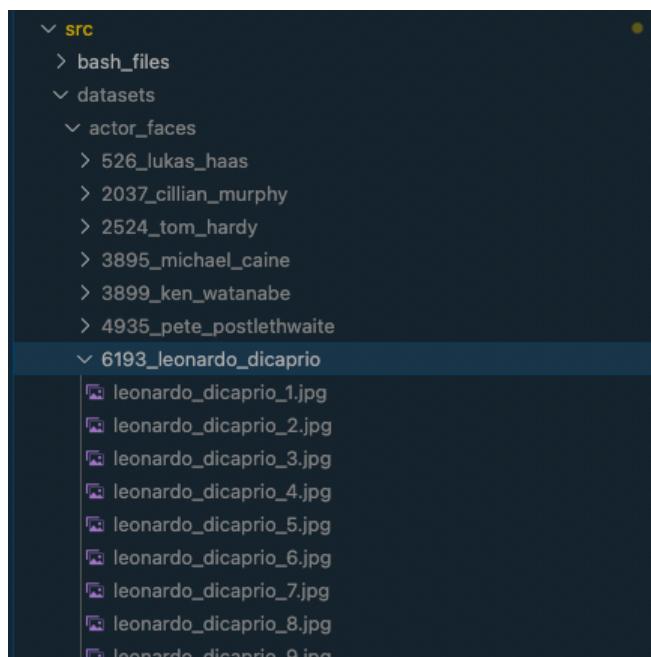


Figura 7: Estructura de ficheros de las imágenes descargadas

9.3.3 Entrenamiento del modelo de reconocimiento

9.3.3.1 Detección y representación facial

Una vez creados los ficheros con las distintas imágenes descargadas, es necesario traducir esa información en una base de datos facial (es decir, donde se establezcan una o más relaciones entre un identificador correspondiente a una persona conocida, y unos determinados “embeddings”). Hasta este punto, únicamente es posible asumir que cada una de las imágenes descargadas en el paso anterior para un determinado actor contienen el rostro del mismo.

A modo de obtener esta base de datos facial, el primer paso es analizar cada una de las imágenes descargadas e intentar detectar el o los rostros que puedan existir en ellas. Tal y como se expuso en el inciso 9.1.1.2, mediante el uso de una *DNN* pre-entrenada por *OpenCV*, es posible detectar los rostros presentes en una imagen, obteniendo no sólo sus coordenadas en la misma sino también la probabilidad de que se trate efectivamente de un rostro. Ambas características nos permiten configurar un determinado *threshold* o “probabilidad mínima” para considerar si una imagen posee efectivamente un rostro o no, así como también para quedarnos con el rostro de mayor probabilidad en caso de que exista más de uno en la imagen.

Una vez detectada la existencia de un rostro en la imagen, se procede a representar al mismo siguiendo la técnica expuesta en el inciso 9.1.2. Para ello, se extrae de la imagen la Región de Interés (*ROI*) detectada por el modelo anterior y se utiliza sólo esa parte para la representación. En la *Figura 8* se observan los resultados de procesar una imagen, dentro de los cuales encontramos:

- Los vectores “embeddings” de 128 dimensiones generados.
- El diccionario “embeddings_metadata”, que contiene la siguiente información:
 - Dirección en el directorio de la imagen analizada (‘img_path’).
 - Cantidad de rostros encontrados (‘scanned_faces’).
 - Probabilidad/nivel de confianza de que se trate de un rostro (‘face_confidence’).
 - Tiempos de ejecución de los procesos de escaneo y obtención de identificadores (‘scan_time’ y ‘emb_time’ respectivamente).

```
Number of embeddings generated: 1
Embeddings metadata:

{
  "img_path": "./datasets/actor_faces/2037_cillian_murphy/cillian_murphy_3.jpg",
  "scanned_faces": "1",
  "face_confidence": [
    "0.99999666"
  ],
  "scan_time": "0.04901623725891113",
  "emb_time": "0.01314854621887207"
}
```

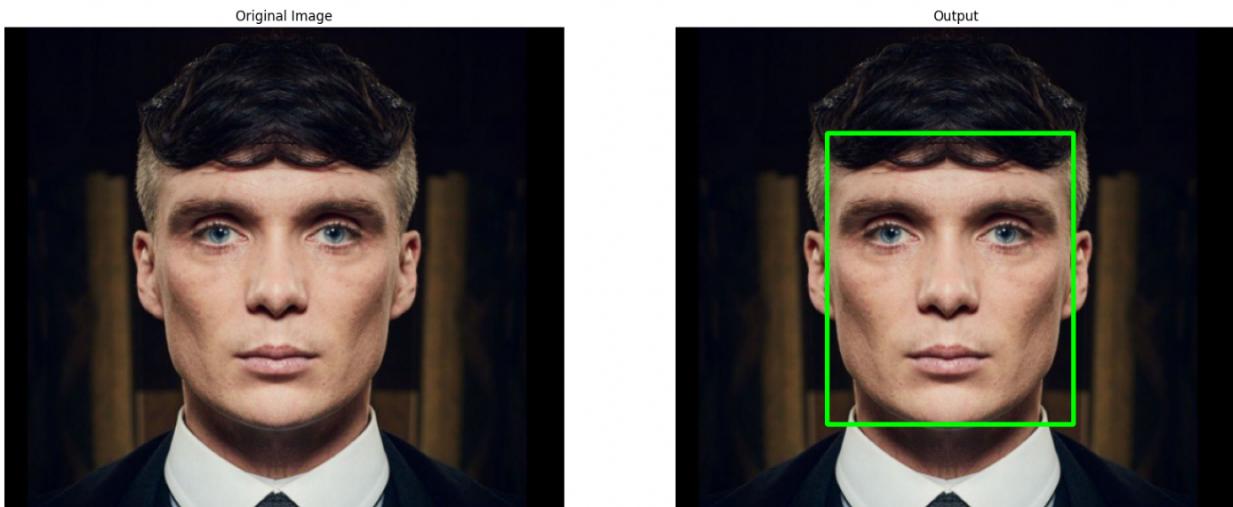


Figura 8: Resultados de escanear una imagen.

El resultado final de haber detectado y representado los rostros deriva en la obtención de una base de datos facial: un registro en el cual se relaciona al identificador único de cada actor, con los *embeddings* obtenidos para cada una de las imágenes descargadas para el mismo. Dicha información puede almacenarse tanto en una base de datos *SQL* como también en ficheros tradicionales. Dado el bajo volumen de datos a tratar en el alcance de este proyecto, se ha decidido utilizar la última opción en pos de un rápido acceso a la información.

9.3.3.2 Entrenamiento de modelo

Por último, esta base de datos facial es utilizada para entrenar un modelo *SVM* cuya configuración es tal que permite, dado un determinado *embedding*, devolver las probabilidades de que pertenezca a cada uno de los actores con los cuales ha sido entrenado. Esto resultará particularmente útil en el presente análisis ya que permite analizar mucha más información que la que se obtendría si simplemente el modelo asignara un único actor a un *embedding*.

Cuando un *SVM* es usado para clasificar registros en distintas clases (o asignar la probabilidad de pertenencia a cada una, como en este caso), se lo denomina *Support Vector Classifier (SVC)*. Un *SVC*

intenta encontrar al mejor hiperplano que permita separar a las distintas clases, maximizando la distancia entre todos los puntos y el hiperplano en sí.

Los parámetros que utiliza el clasificador SVC son los siguientes:

- **Kernel:** Determina el tipo de hiperplano a utilizar, pudiendo ser lineal ('*linear*') o no lineal ('*rbf*' o '*poly*').
- **C:** Establece la penalización para el error obtenido. Implica encontrar una solución de compromiso entre la “generalidad” del hiperplano encontrado y su ajuste a los datos de entrenamiento. Valores altos de *C* pueden llevar a un sobreajuste..
- **Gamma:** Se trata de un parámetro utilizado únicamente para *kernel* de tipo *rbf*, y define qué tanto se busca que el modelo se ajuste a los datos de entrenamiento o no (a mayor *gamma*, mayor ajuste).
- **Degree:** Aplicable sólo para *kernel* de tipo '*poly*', ya que define el grado del polinomio a utilizar.

El modelo puede ser entrenado con distintos tipos de *Kernel*, así como distintos valores de variables *K*, *Gamma*, etc; para luego poder realizar una comparación de desempeño de cada una de las configuraciones. En la sección 9.4.1 se ahondará más en este tema.

9.3.4 Procesamiento de video

El procesamiento de video consiste en partir de un determinado archivo de video y, dada una serie de parámetros, obtener una tabla que relacione distintos momentos o *timesteps* del mismo, con los *embeddings* presentes en dichos instantes.

Los principios utilizados para la parametrización de video son los mismos empleados al momento de construir una base de datos facial, ya que es posible tratar a los videos como una serie de imágenes sucedidas entre sí. La cantidad de cuadros por segundo a analizar es una variable susceptible de modificación al momento de procesar películas completas o escenas. A mayor cantidad de cuadros, mayor es el consumo de recursos para su procesamiento, pero también aumentan las posibilidades de identificar o no a un actor.

En lo que respecta al almacenamiento de los embeddings obtenidos para distintos videos, no hay diferencias respecto de lo implementado en la construcción de modelos de reconocimiento: puede hacerse en una base de datos relacional, como así también en ficheros tradicionales (esta última opción ha sido la utilizada al analizar escenas de corta duración).

9.3.4 Reconocimiento de video

El reconocimiento de video es el punto cúlmine de la ejecución del presente estudio, ya que entran en juego todos los elementos antes expuestos. Partiendo de una determinada película/escena, se parametrizan una cierta cantidad de cuadros a modo de obtener los *embeddings* correspondientes a los rostros presentes (en caso de existir). Esos *embeddings* son luego evaluados por un *SVM* entrenado con imágenes conocidas de los actores presentes en el video a analizar. Finalmente, se obtiene para cada cuadro las probabilidades de que el rostro presente pertenezca a cualquiera de los distintos actores.

El conjunto de datos resultantes deriva en un listado de distintos cuadros de un video (particularmente en los que se ha detectado al menos un rostro), con el instante de tiempo en el cual aparece ese cuadro en el video original, y con las distintas probabilidades que tienen los rostros detectados de pertenecer a los actores con los cuales se ha entrenado el modelo de reconocimiento (ver Figura 9).

Cuadro:	474
Timestamp:	0:00:15
Predicciones:	
6193_leonardo_dicaprio	51.92%
2524_tom_hardy	20.58%
3899_ken_watanabe	7.38%
3895_michael_caine	6.61%
4935_pete_postlethwaite	3.31%
24045_joseph_gordon-levitt	2.39%
526_lukas_haas	2.04%
95697_dileep_rao	1.86%
13022_tom_berenger	1.34%
2037_cillian_murphy	1.26%
27578_elliot_page	0.72%
8293_marion_cotillard	0.59%

Figura 9: Predicciones obtenidas para un cuadro en particular.

9.3.5 Interpretación de predicciones

Una vez obtenida la información del paso anterior (esto es, la probabilidad de cada actor de encontrarse en un cuadro específico), es necesario traducir a la misma en registros donde se designe un intervalo de tiempo en el cual se asume que se encuentra un determinado actor. Para ello, se debe analizar las probabilidades de los diferentes cuadros de manera conjunta, establecer límites e intervalos de aceptación para las probabilidades obtenidas, y definir criterios que contemplen la naturaleza de los datos a analizar, siendo en este caso datos relativos a películas y actores que aparecen en las mismas.

A continuación se describen las distintas etapas necesarias para la obtención de los resultados finales del presente estudio.

9.3.5.1 Estructuración de los datos

El primer paso consiste en transformar los datos de partida, los cuales son diccionarios de tipo “JSON” (no estructurados) en datos estructurados, a modo de poder facilitar el análisis de los mismos.

El resultado de este paso implica pasar del formato expuesto en la *Figura 9* al que se observa en la *Figura 10*, donde los datos son presentados en una tabla cuyos registros corresponden a los distintos cuadros evaluados, y donde es posible acceder a las diez predicciones obtenidas por el modelo.

cuadro	timestamp	nombre_pred_1	valor_pred_1	nombre_pred_2	valor_pred_2	nombre_pred_3	valor_pred_3	nombre_pred_4	valor_pred_4	...
120	0:00:06	4935_pete_postlethwaite	0.385225	13022_tom_berenger	0.175547	3899_ken_watanabe	0.146736	3895_michael_caine	0.060721	...
264	0:00:11	6193_leonardo_dicaprio	0.448619	2524_tom_hardy	0.216089	526_lukas_haas	0.079603	95697_dileep_rao	0.030307	...
...

Figura 10: Tabla de predicciones.

Los siguientes pasos se aplican a cada actor de manera individual, para luego unir los resultados en una instancia final.

9.3.5.2 Filtrado de datos

Ya con la información estructurada, se procede a descartar las predicciones desde el tercer puesto en adelante, a modo de utilizar únicamente la información relativa al primer y segundo lugar. Esto se hace con el fin de simplificar el conjunto de datos a analizar, y debido a que los valores en los puestos descartados suelen ser relativamente bajos en comparación a los primeros.

En la *Figura 11* se observa un gráfico de dispersión en el cual se muestra, para un determinado actor, los distintos valores de probabilidad obtenidos en un intervalo de cuadros específico. El color y la figura de los puntos indican, para cada cuadro, la posición relativa de cada predicción respecto de los demás actores. En la Figura 12 se observan los mismos datos, pero esta vez filtrados tal y como se explicó anteriormente.

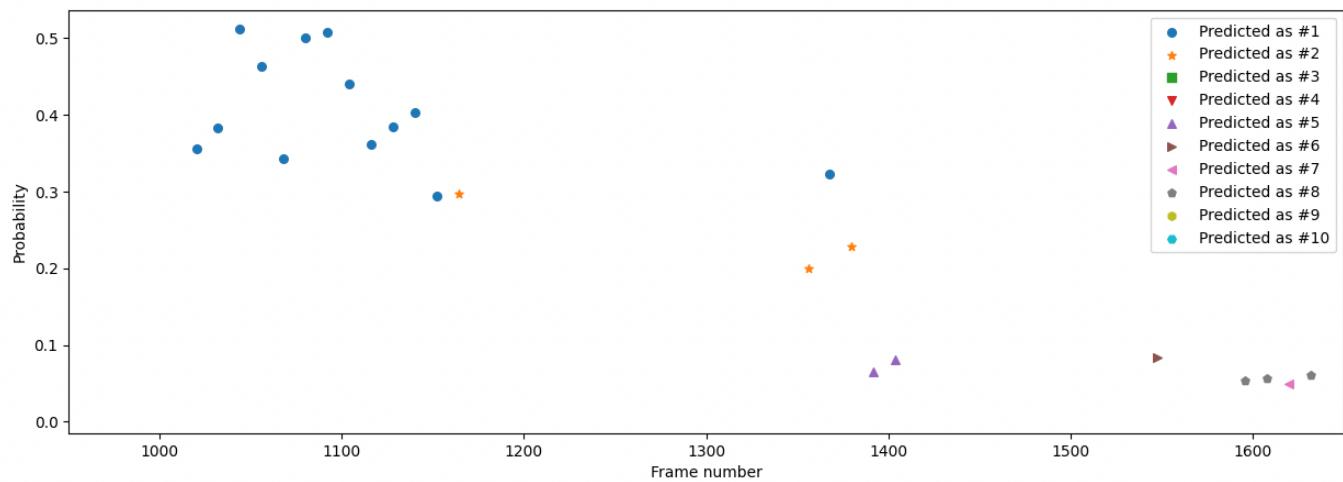


Figura 11: Probabilidades obtenidas para un determinado actor en un intervalo de tiempo, clasificadas según su posición relativa respecto del resto de actores.

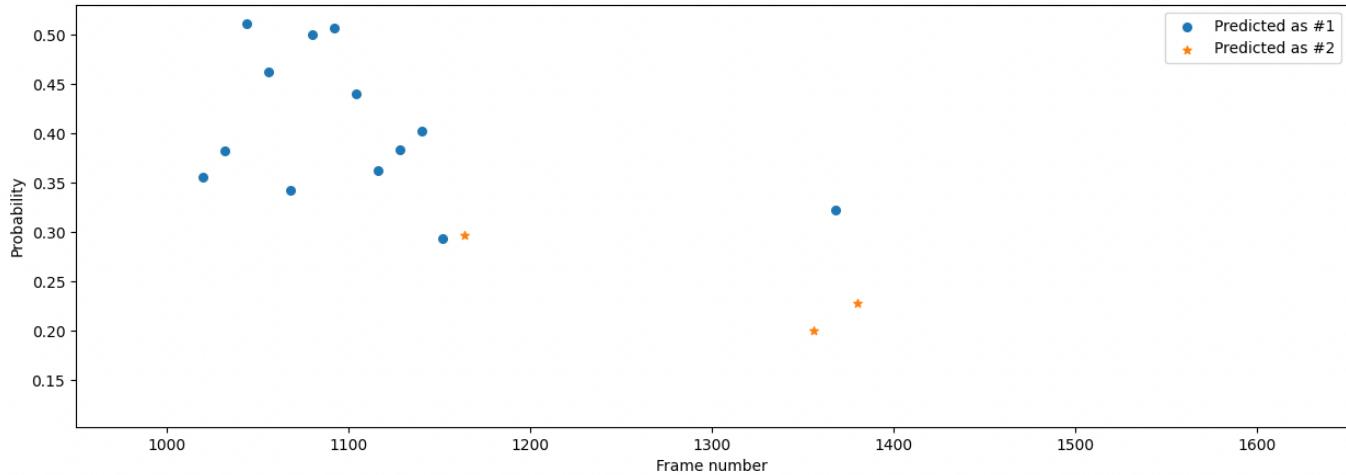


Figura 12: Probabilidades obtenidas para un determinado actor en un intervalo de tiempo, filtradas para las posiciones 1 y 2.

9.3.5.3 Definición de “zonas de identificación”

El siguiente paso consiste en determinar las “zonas de identificación” de un actor. El análisis exploratorio y gráfico de los resultados obtenidos hasta esta instancia evidencia una marcada concurrencia en la aparición de probabilidades de primer y segundo orden, estando éstas generalmente distribuidas en grupos o sectores marcados y aislados de los demás. Esto tiene sentido dada la naturaleza de los datos, ya que es esperable que el rostro de un actor se mantenga en escena una cantidad de cuadros suficiente como para poder ser percibido por el espectador, lo que se traslada en varios cuadros consecutivos con el mismo

rostro. A estos grupos de probabilidades se los denomina “zona de identificación”, y se definen como aquellos intervalos en los que existen dos o más probabilidades de primer o segundo grado, separadas por no más de tres cuadros de distancia.

En la *Figura 13* es posible observar las mismas probabilidades expuestas en la *Figura 12*, con el agregado de líneas que se extienden tres cuadros por delante y tres por detrás de cada probabilidad. En la *Figura 14* se proyectan los límites de dichas líneas, lo que evidencia la continuidad en la aparición de las probabilidades analizadas dado que la totalidad de los puntos se encuentra a no más de tres cuadros de distancia de otro punto del mismo sector. Finalmente, en la *Figura 15* se traza una curva que une los puntos de un determinado sector, delimitando así a las dos zonas de identificación del ejemplo.

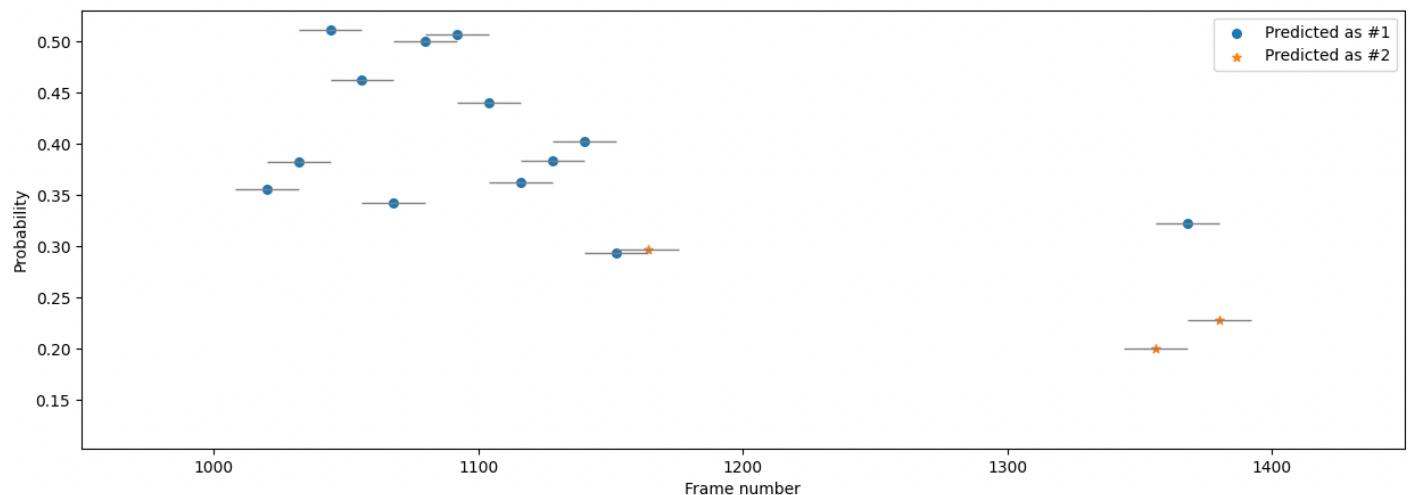


Figura 13: Extensión bilateral de tres cuadros para cada probabilidad de un actor.

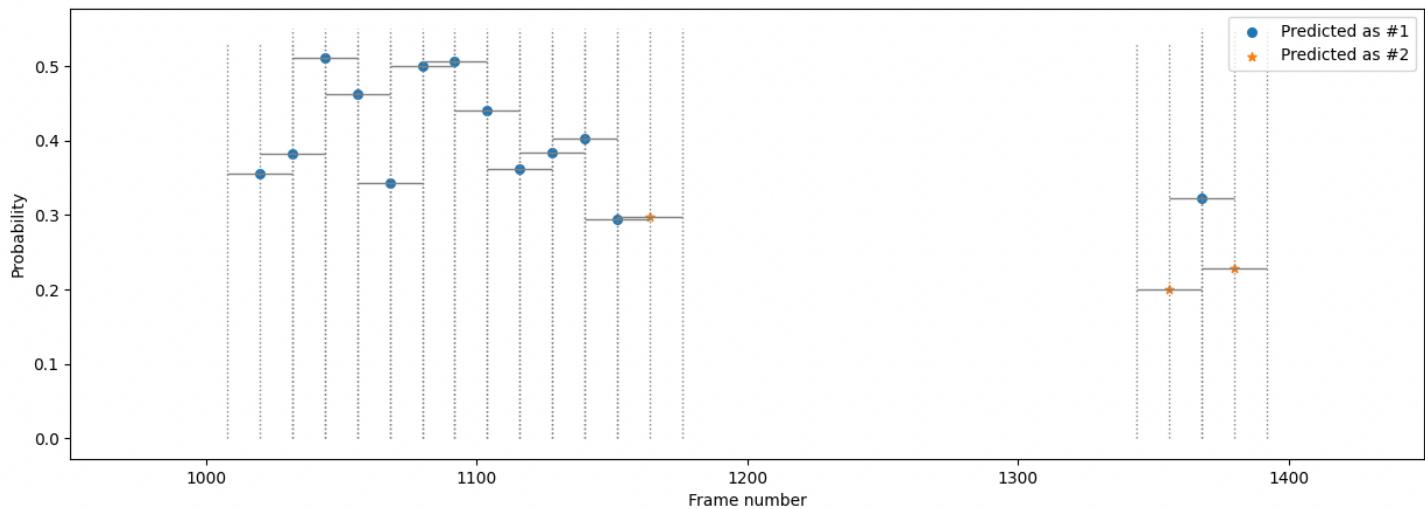


Figura 14: Proyección de los límites de las extensiones de cada probabilidad.

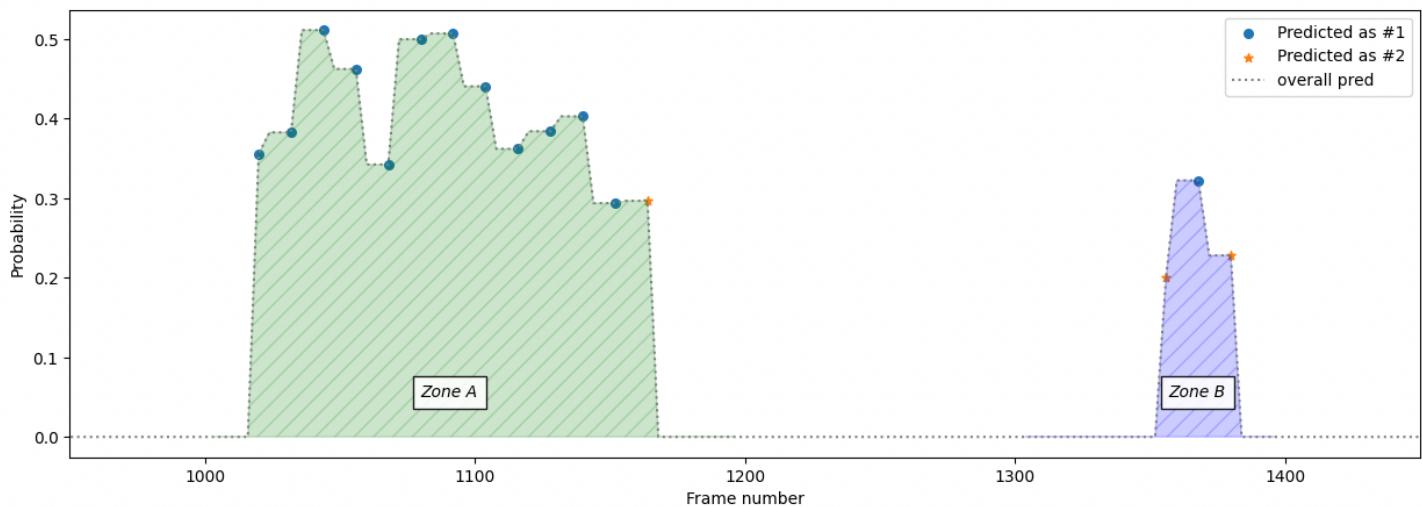


Figura 15: Unión de puntos y definición de “zonas de identificación”.

9.3.5.4 Calculo de métricas para las “zonas de identificación”

Habiendo definido las zonas de identificación de un actor, es necesario encontrar una manera de poder comparar cualitativamente a las mismas mediante el uso de métricas comunes y replicables. El objetivo detrás de esta comparación radica en poder descartar aquellas zonas que potencialmente indiquen falsos positivos, y utilizar solo aquellas donde se cumplen requisitos y límites determinados.

Para cada zona, entonces, se definen las siguientes métricas:

- Ancho: La distancia (en cuadros) entre los límites de una zona.
- Máxima probabilidad: La mayor probabilidad identificada en una zona.
- Suma de probabilidades: La suma de todas probabilidades de una zona (métrica muy similar al área dado que el ancho de los intervalos entre probabilidades suele ser constante).

La *Figura 16* muestra de manera gráfica las tres métricas antes descritas, para el mismo actor e intervalos analizados en los ejemplos anteriores.

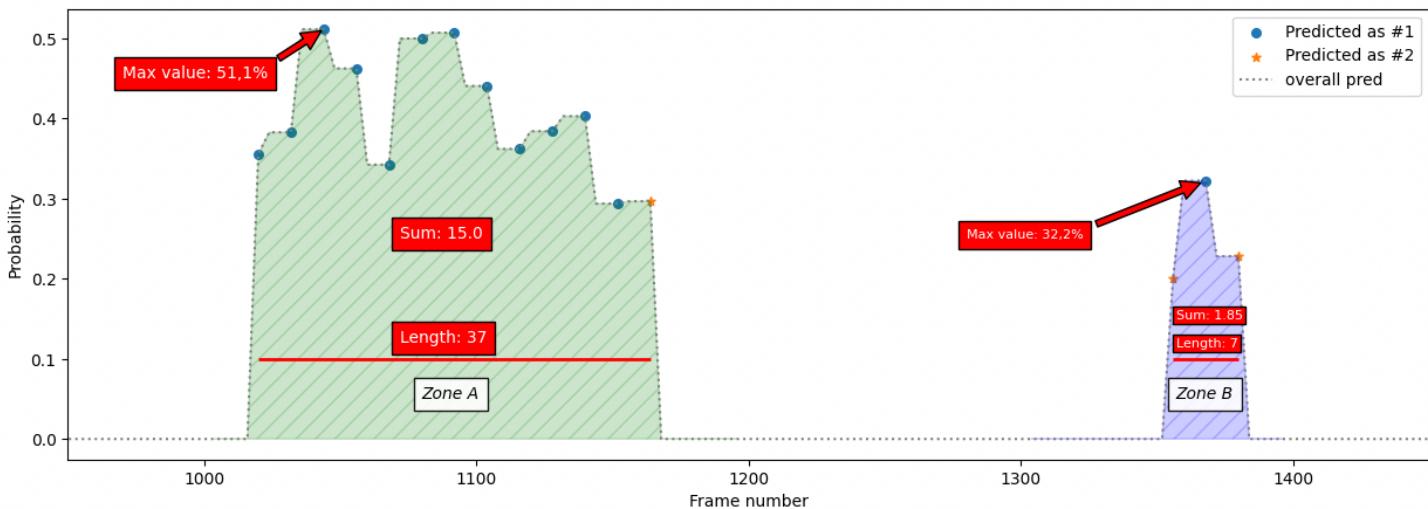


Figura 16: Métricas para las distintas “zonas de identificación” de un determinado actor.

9.3.5.5 Filtrado de zonas y estructura final de los datos

El paso final consiste en establecer cotas para las distintas métricas que permitan filtrar a las “zonas de identificación” y obtener así los resultados definitivos del estudio. Cuanto mayor sea la exigencia con los límites, disminuyen las probabilidades de reportar falsos positivos pero se incrementan las de no reportar apariciones genuinas, por lo que la definición de dichos parámetros debe ser una solución de compromiso entre ambos factores.

La *Figura 17* expone una tabla con las distintas “zonas de identificación” de distintos actores y sus métricas asociadas. Este formato es el que se trazó como objetivo al comienzo del estudio: dado un determinado instante de tiempo y un material filmográfico particular, listar los actores que se encuentren en pantalla.

material	actor	zona_de_identificacion	cuadro_inicio	timestamp_inicio	cuadro_fin	timestamp_fin	ancho	maxima_prediccion	suma_de_predicciones
inception_escena_4	2037_cillian_murphy	1	432	0:00:18	480	0:00:20	13	35.15%	3.642645
inception_escena_4	2037_cillian_murphy	2	540	0:00:23	612	0:00:26	19	28.93%	4.205183
inception_escena_4	2037_cillian_murphy	3	636	0:00:27	672	0:00:28	10	31.35%	2.296708
inception_escena_4	2037_cillian_murphy	4	1020	0:00:43	1164	0:00:49	37	51.10%	15.006024
inception_escena_4	2037_cillian_murphy	5	1356	0:00:57	1380	0:00:58	7	32.22%	1.850404
inception_escena_4	24045_joseph_gordon-levitt	1	276	0:00:12	288	0:00:12	4	17.44%	0.639503
inception_escena_4	24045_joseph_gordon-levitt	2	732	0:00:31	756	0:00:32	7	23.96%	1.442586
inception_escena_4	24045_joseph_gordon-levitt	3	1596	0:01:07	1608	0:01:08	4	21.79%	0.745673
...

Figura 17: Tabla de resultados finales para un determinado material.

9.4 Resultados

9.4.1 Entrenamiento del SVM

Con el fin de definir los hiperparámetros con los cuales entrenar al modelo de reconocimiento a del presente estudio, se realizan varias iteraciones con distintas combinaciones de los mismos. A cada ejecución se la denomina *pipeline*, y comprende los siguientes pasos:

1. Establecimiento de una semilla única y propia de cada *pipeline*.
2. División de la base de datos facial entre conjunto de entrenamiento y conjunto de prueba, utilizando la semilla en el proceso para asegurar el menor sesgo de datos posible. La división se establece en un 70% de los datos para entrenamiento y un 30% para prueba.
3. Obtención de *embeddings* para el conjunto de entrenamiento.
4. Entrenamiento del modelo utilizando los hiperparámetros dados.
5. Evaluación del conjunto de prueba con el modelo obtenido.

Éste último paso consiste en utilizar el modelo entrenado para predecir, por cada imagen del conjunto de prueba, las probabilidades de que se trate de los distintos actores. Dado que la clase a la que pertenece cada imagen evaluada es conocida (es decir, el actor real al que pertenecen), es posible definir si la predicción ha sido acertada o no. Pero con el fin de aprovechar el hecho de que los resultados de la aplicación del modelo sean un conjunto de probabilidades y no la probabilidad de mayor valor, se definen dos métricas para cada imagen evaluada:

1. “Acierto”: que será 1 en caso de que la predicción de mayor porcentaje coincida con la clase real de la imagen, y

2. “Acierto top 3”: que será 1 en caso de que la clase real se encuentre dentro de las primeras tres predicciones, ordenadas por su porcentaje de mayor a menor.

Finalmente, se puede calcular por cada *pipeline* el porcentaje de “aciertos” y “aciertos top 3”, sobre el total de imágenes de prueba disponibles. Dichas métricas se denominaron ‘*accuracy*’ y ‘*accuracy top 3*’, respectivamente.

Los hiperparámetros evaluados han sido:

- *Kernels*: ‘linear’, ‘rbf’, ‘poly’.
- *C*: 1, 10, 100, 1000.
- *Gamma* (solo para *kernel ‘rbf’*): 0.1 , 1, 10, 100.
- *Degree* (solo para *kernel ‘poly’*): 2, 3, 4.

Para cada combinación de hiperparámetros disponible, se realizaron 8 (ocho) iteraciones distintas con una semilla única en cada iteración. En la *Figura 18* se puede ver el total de pipelines entrenados, el cual evidencia que se han probado un total de 32 configuraciones distintas, lo que implicó 256 ejecuciones totales.

Kernel	Opciones disponibles			Subtotal	Iteraciones	Total
	C	Gamma	Degree			
<i>linear</i>	4	-	-	4	8	32
<i>rbf</i>	4	4	-	16	8	128
<i>poly</i>	4	-	3	12	8	96
Total				32	8	256

Figura 18: Pipelines entrenados

Dado que para cada una de las treinta y dos distintas configuraciones de parámetros se obtienen ocho *pipelines*, y por ende ocho valores de ‘*accuracy*’ y ‘*accuracy top 3*’, se procede a utilizar el promedio de ambas métricas para analizar y comparar las distintas configuraciones. Hecho esto, se asignan dos rankings para cada opción, los cuales permiten observar en qué posición se encuentra cada modelo respecto de los demás, y por último se crea un ranking compuesto que es simplemente la suma nominal de ambos. Al ordenar a los modelos por su ranking compuesto y de menor a mayor, se obtienen

en los primeros registros aquellos que han obtenido mejores resultados tanto en ‘accuracy’ como en ‘accuracy top 3’.

En la *Figura 19* se visualizan las cinco mejores configuraciones, según el ranking compuesto. Vemos que el modelo mejor puntuado es aquel con kernel *rbf*, $C = 10$ y $\gamma = 0.1$, el cual tuvo el mejor ‘accuracy top 3’ y el cuarto lugar en ‘accuracy’, por lo que es ésta la configuración elegida a implementar en el modelo para analizar video.

	parameters_uuid	kernel	C	gamma	degree	avg_accuracy	avg_accuracy_top3	accuracy_rank	accuracy_top3_rank	composed_rank
0	7ad97536-2a32-4751-95bf-b9efb8436b7e	rbf	10	0.1	NaN	0.638764	0.864719	4	1	5
1	62a34e92-0177-4384-abc4-7e77949e3793	rbf	1000	0.1	NaN	0.627080	0.847530	9	2	11
2	7f67df66-bf2b-4351-b3ed-74e71150c0e9	rbf	1	1.0	NaN	0.659171	0.836855	1	11	12
3	65671b66-fbc9-479b-ae42-2a99300fd74f	poly	1	NaN	3.0	0.635451	0.840496	7	6	13
4	a9e1f6cf-9539-48ba-8cce-fc5b532ca773	poly	10	NaN	3.0	0.646059	0.833400	3	12	15

Figura 19: Primeras cinco configuraciones de hiperparametros, ordenadas por ranking compuesto.

9.4.2 Aplicación en película

A continuación se exponen los resultados obtenidos al aplicar el proceso explicado en el punto “9.3 Implementación”.

9.4.2.1 Parametros utilizados

La película analizada es “*Inception*”, un título del director Christopher Nolan del año 2010 y que cuenta con una duración de 148 minutos. La infraestructura utilizada ha sido una virtualización vía *Docker Desktop*, con 4 CPUs, 8 GB de memoria RAM y un disco de 104 GB.

En primer lugar, se descargan 35 imágenes de rostros para los primeros doce actores de la película, ordenados según su popularidad en el sitio *The Movie Database*. Es de esta misma página de donde se obtiene la información relativa a la obra y su reparto. La consultas realizadas para obtener las 35 imágenes han sido:

- ❖ “[Nombre del actor] face” en 32 ocasiones, y
- ❖ “[Nombre del actor] [Nombre de la película] face” en 3 ocasiones.

Luego, y en base a lo expuesto en la sección “9.4.1 Entrenamiento del SVM”, se entrena un SVM partiendo de las imágenes previamente descargadas, con los siguientes hiperparametros:

- *Kernel*: RBF
- *C*: 10
- *Gamma*: 0.1

A continuación, se extraen los *embeddings* de la película. La misma cuenta con aproximadamente 148 minutos y ha sido grabada a una tasa de 24 cuadros por segundo, por lo que se compone de un total de 213.107 cuadros. La extracción se realiza buscando analizar únicamente 4 cuadros por cada segundo de video, por lo que finalmente se analizan 35.518 imágenes. Del subconjunto analizado, en 12.956 oportunidades (36,47%) se detectaron rostros con una confianza del 90% o superior, y es en estos casos únicamente donde se extraen los *embeddings* disponibles. El proceso de extracción se divide en 32 particiones a modo de evitar la sobrecarga de los recursos disponibles, y lleva un total de 5 horas y 11 minutos.

Por último, se aplica el SVM previamente entrenado para obtener las probabilidades de cada *embedding* extraído de pertenecer a los distintos actores con los que se entrena al modelo. Este proceso toma 18 segundos, y el resultado es un conjunto de datos no estructurados con las predicciones asociadas a cada uno de los cuadros analizados. Tal y como se explica en la sección “9.3.5 Interpretación de predicciones”, estos datos son finalmente transformados en “zonas de identificación” a las cuales es posible calcular las respectivas métricas expuestas en la sección “9.3.5.4 Cálculo de métricas para las ‘zonas de identificación’”, las cuales permiten manipular con mayor facilidad la información.

La película, entonces, es reducida a 2589 “zonas de identificación”, donde en cada una se establece un momento de inicio y de fin en el cual se presume se encuentra un actor, y para las cuales se calculan las distintas métricas. A modo de poder evaluar la eficacia del modelo, se decide trabajar con un subconjunto de zonas que cumplan con alguna de las siguientes características:

- ‘Ancho’ mayor o igual a 48 cuadros (dos segundos).
- ‘Máxima probabilidad’ mayor o igual 60%.
- ‘Suma de probabilidades’ mayor o igual 4.

Al realizar este filtrado se busca descartar aquellas zonas cuya predicción no sea lo suficientemente fiable, y conservar aquellas donde las probabilidades de acierto sean mayores. El conjunto de datos resultante se compone de 516 “zonas de identificación”. En la *Figura 20* se observa un mapa conceptual con los distintos números clave obtenidos en los distintos pasos del análisis.

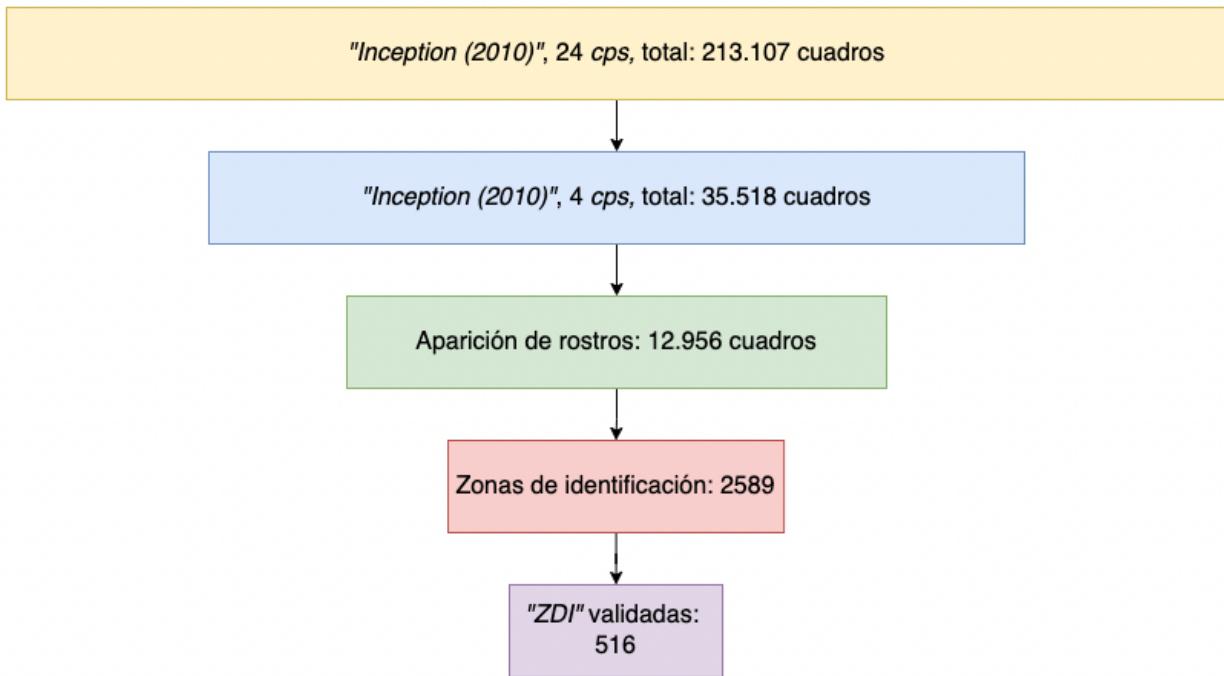


Figura 20: Reducción del número de datos en las distintas partes del proceso.

9.4.2.2 Proceso de validación de resultados

El paso final consiste en evaluar la eficacia de las “zonas de identificación” filtradas, a través de la validación de un usuario. Para ello, se utiliza un sistema en el cual se definen:

- El subconjunto de zonas a evaluar (en este caso, las que cumplen con las condiciones antes mencionadas),
- El número de zonas a evaluar por iteración, y
- El número de cuadros a exponer por cada zona.

De este modo, el usuario evaluador es expuesto a distintas imágenes pertenecientes a una misma “zona de identificación” y se le consulta si en alguna de las imágenes aparece el actor predecido por el modelo. El usuario debe realizar un análisis visual e ingresar su respuesta, la cual es almacenada en el sistema. En la *Figura 21* se exponen diversos ejemplos del funcionamiento del proceso de validación, para una iteración de treinta zonas a evaluar, con cuatro cuadros por zona.



Figura 21: Funcionamiento del proceso de validación de zonas.

9.4.2.3 Resultados obtenidos

En primer término, es importante remarcar que los resultados obtenidos forman parte de solo dos cuadrantes de una matriz de confusión tradicional: los “verdaderos positivos” (es decir, aquellos casos donde se predice que un actor se encuentra en un determinado intervalo de tiempo y efectivamente es así), y los “falsos positivos” (cuando esa predicción es errónea). La construcción del modelo no permite identificar a la parte negativa dado que esta situación es la que se asume por defecto: si en un intervalo no hay predicción alguna, es porque en ese momento no se encuentra presente ninguno de los actores objetivo (ya que caso contrario el modelo debería detectarlo y realizar una predicción). Es por ello que la métrica utilizada para evaluar los resultados es la “precisión”, que se representa por la proporción de verdaderos positivos dividido entre todos los resultados positivos (tanto verdaderos positivos, como falsos positivos).

Luego de validar las predicciones de las 516 “zonas de identificación” filtradas, se halla que en 186 casos (36,04%) la estimación es acertada. Es esta métrica (cantidad de aciertos sobre el total de zonas evaluadas) la que se denomina “precisión”.

En la *Figura 22* se observa cómo la precisión varía según la cantidad de “zonas de identificación” que se tomen, ordenando a las mismas por la métrica “Suma de probabilidades” de mayor a menor. A modo de ejemplo, y siguiendo la gráfica, si sólo se analizan las primeras 20 zonas de identificación con mayor valor de “Suma de probabilidades”, la precisión es de un 60%. Si se analizan las primeras 40

zonas, el indicador desciende a 45%, y si finalmente se observa el final de la gráfica (donde se comprenden las 516 zonas), el número es el expuesto anteriormente (36%). Este comportamiento es esperable dado que cuanto mayor sea la “suma de probabilidades”, mayor debería ser la confianza con la que el modelo predice la aparición de un actor, y por ende mayor debería ser la precisión. También se aprecia que a partir de las primeras 200 zonas y hacia la derecha de la gráfica se produce una estabilización en la franja de 36% al 38%, lo que indica una tendencia marcada.

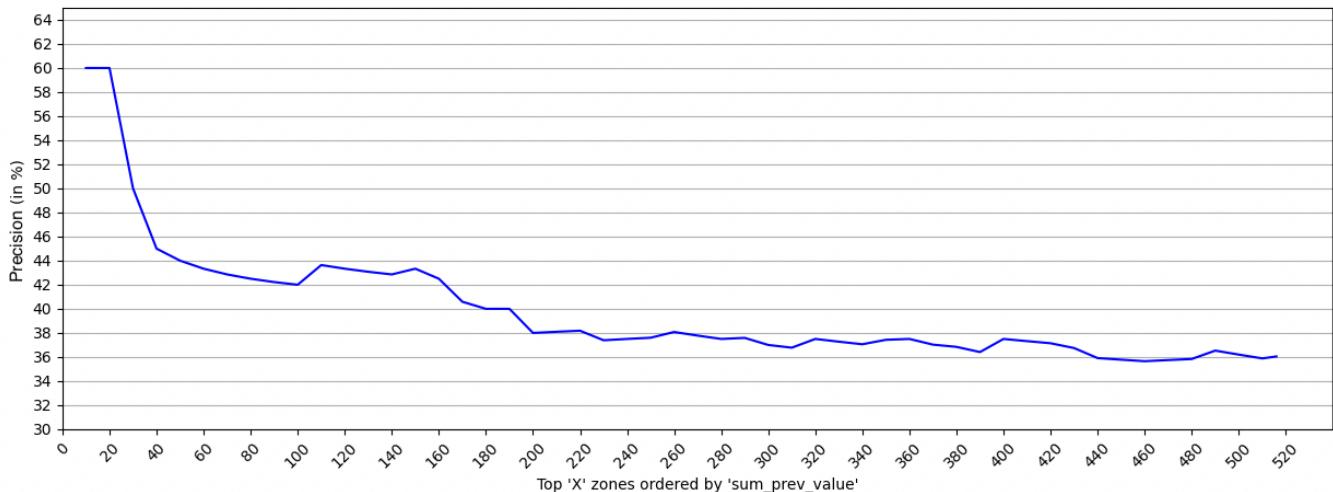


Figura 22: Distribución de la precisión a medida se incrementa la cantidad de “Zonas de identificación” evaluadas, ordenandolas por la métrica “suma de probabilidades”

Otra manera de visualizar los resultados es mediante lo expuesto en la tabla de la *Figura 23*, donde se analiza la precisión por cada uno de los actores con los que se entrena al modelo. Como información adicional, se incorpora la cantidad de segundos que posee cada actor en la película (recuperado de *IMDb* [16]), y se los ordena de mayor a menor siguiendo esta variable.

Existen diversas inferencias que pueden realizarse en base a lo expuesto en la tabla:

- La precisión varía fuertemente entre los diversos actores.
- Actores con relativamente poco tiempo en pantalla muestran los peores resultados (a excepción de “Dileep Rao”, caso analizado en la sección “9.5.1 Explicación de los resultados”).
- El tiempo real en pantalla no es directamente proporcional a la cantidad de detecciones que realiza el modelo (Ken Watanabe es predecido por el modelo varias veces más que Elliot Page o Joseph Gordon-Levitt, ambos con mayor tiempo en pantalla).

Actor	Tiempo en pantalla (seg)	Zonas totales	Aciertos	Precisión	Precisión acumulada
6193_leonardo_dicaprio	3645	84	74	88.10%	88.10%
27578_elliot_page	1635	15	15	100.00%	89.90%
24045_joseph_gordon-levitt	1290	8	3	37.50%	85.98%
3899_ken_watanabe	975	120	28	23.33%	52.86%
2037_cillian_murphy	930	41	26	63.41%	54.48%
8293_marion_cotillard	795	45	16	35.56%	51.76%
2524_tom_hardy	765	50	7	14.00%	46.56%
95697_dileep_rao	300	17	13	76.47%	47.89%
13022_tom_berenger	285	13	1	7.69%	46.56%
526_lukas_haas	120	24	1	4.17%	44.12%
3895_michael_caine	120	18	1	5.56%	42.53%
4935_pete_postlethwaite	75	81	1	1.23%	36.05%
Total		516	186	36.05%	

Figura 23: Precisión por actor.

9.5 Discusión

9.5.1 Explicación de los resultados

Es posible evidenciar que los resultados generales indican un bajo rendimiento del modelo, logrando predecir de manera correcta en apenas por encima de un tercio de las ocasiones. Existen diversos factores que influyen en los resultados del modelo y que explican los desvíos existentes.

- Naturaleza del conjunto de entrenamiento: Tal y como se detalla en la sección “9.3.2 Descarga de imágenes faciales”, el conjunto de imágenes de entrenamiento es obtenido mediante técnicas de *web-scraping* que descargan los resultados devueltos por un buscador. Esto implica que las imágenes descargadas para un actor pueden variar sensiblemente entre ellas, ya que pueden tratarse de distintos momentos de la vida del sujeto (en especial si posee una larga trayectoria actuando), distintos cortes de pelo, atuendos, maquillaje, o inclusive apariencias propias de una película que no es la que se busca analizar.

La alta variabilidad entre las imágenes se traduce en una mayor dificultad de recrear huellas faciales robustas y discriminatorias propias de cada individuo.

- Naturaleza del conjunto objetivo: Es importante recordar que las imágenes evaluadas en este proceso provienen de cuadros de video y no de fotografías particulares (como es el caso del conjunto de entrenamiento). Esta diferencia no es menor, dado que los cuadros de película no se

piensan de manera aislada sino como partes componentes de un conjunto, lo que supone algunas de las siguientes diferencias:

- Existencia de rostros de perfil o en ángulo, ocasionalmente de frente.
- Presencia de expresiones faciales atípicas en fotos y propias de la actuación (principalmente diálogos, pero también expresiones de emociones, ojos cerrados, etc).
- Iluminación heterogénea y variable, dependiente del ambiente en el que se sitúa una escena.
- Particularidades del caso analizado: Por último, cada película cuenta con un conjunto de variables que toman valores particulares y que pueden conservarse a lo largo de todo el material (cortes de cabello, maquillaje de caracterización, etc) y otros que inclusive pueden variar dentro del mismo (uso de accesorios como anteojos, sombreros, gorros, maquillajes circunstanciales, etc).

En el caso de la película analizada, es posible destacar el hecho de que gran parte del elenco cuenta con rasgos comunes (masculinos de entre 30 y 40 años, caucásicos, de contextura delgada y pelo lacio largo), lo que puede implicar mayor dificultad al momento de diferenciarlos. Esto es evidente con el caso de Dileep Rao, el cual posee un alto porcentaje de precisión a pesar de su baja cantidad de apariciones, lo que cobra sentido al considerar que es el único actor con ascendencia india y rasgos sustancialmente distintos al resto del elenco (pelo corto y rizado, contextura ancha y tez morena).

9.5.2 Oportunidades de mejora

Existen diversos puntos sobre los cuales trabajar a modo de poder contrarrestar el impacto de los factores antes mencionados:

1. Mejora del conjunto de imágenes de entrenamiento: Es necesario construir un conjunto de imágenes de entrenamiento acorde al material a analizar. Se debe priorizar las imágenes que representen la apariencia de los actores en la película objetivo, lo cual es difícil de conseguir en comparación a fotografías de los actores en otros ámbitos. Se debe buscar un equilibrio entre la intervención humana en el proceso de consolidado de imágenes de entrenamiento y la automatización del mismo.
2. Retroalimentación: Este punto se basa en utilizar la validación del usuario descrita en el apartado “9.4.2.2 Proceso de validación de resultados” para agregar nuevas imágenes con las que entrenar al modelo de reconocimiento. Estas imágenes deberán tener un peso mayor que las descargadas de internet dado que no solo cuentan con una verificación humana que confirma que

efectivamente se trata del actor deseado, sino que también incluyen cualquier tipo de particularidad que pudiera tener ese actor en la película objetivo. De agregarse este paso, el proceso pasaría a ser iterativo, dado que el modelo deberá ser re entrenado tantas veces como etapas de validación se esté dispuesto a realizar.

A su vez, retroalimentar el modelo una vez obtenidos los resultados permitiría focalizar los esfuerzos manuales del usuario validador en aquellos sectores donde más hiciera falta (por ejemplo, validando en primer lugar imágenes de actores con la precisión más baja, buscando mejorar el conjunto de entrenamiento para esos actores antes que para aquellos con buenos resultados).

9.6 Conclusiones

Una primera conclusión que se desprende del análisis de los resultados es que la marcada diferencia entre las imágenes del conjunto de entrenamiento y las imágenes objetivo extraídas de los distintos cuadros de una película, dificulta la correcta identificación de los actores. El hecho de que las fotografías utilizadas para cada actor difieren no sólo de la imagen de éste último en la película analizada, sino también entre sí (ya que corresponden a distintas etapas de la vida del actor, distintas interpretaciones, etc.), no es un error del proceso, sino que es el resultado de perseguir lo expuesto en la sección “*5. Alcances del trabajo y limitaciones*”: crear una solución flexible, sencilla y escalable, en lugar de una herramienta personalizada y construida alrededor de una única película.

El proceso de construcción del conjunto de imágenes de entrenamiento presentado en este estudio es completamente independiente de la película o el director elegidos. Es posible indicar una o más obras y descargar cuántas imágenes se deseen de los actores presentes en ellas. Construir de manera manual el *dataset* de entrenamiento para cada película que se desea analizar implicaría un esfuerzo mucho mayor e iría en contra de la concepción universal de la solución.

De todas maneras, y retomando nuevamente lo enunciado en la sección 5, el proceso expuesto en presente trabajo se piensa como un proyecto de naturaleza colaborativa: la única manera de lograr abarcar grandes proporciones del vasto universo cinematográfico es mediante la escalabilidad en la implementación. Y dicha escalabilidad no solo refiere al aprovechamiento de los recursos computacionales disponibles por los distintos nodos componentes, sino también a la participación colectiva de usuarios validadores que aporten información valiosa al potencial proceso de

retroalimentación. Disponer de humanos validando si la predicción de un modelo es o no acertada y utilizar esa validación para mejorar al mismo no es algo impensado en el mundo del aprendizaje automático (por ejemplo, Google utiliza la validación de su servicio de reCAPTCHA como mecanismo de etiquetado de datos)^[17].

En base a las conclusiones y los resultados antes expuestos, se concluye que sin la implementación de un sistema de retroalimentación que utilice la evaluación humana para mejorar los datos de entrenamiento, no es posible diseñar una herramienta genérica que permita identificar a todos los actores que son exhibidos en un determinado instante de tiempo y para un determinado material cinematográfico.

10. Referencias bibliográficas

1. Grand View Research (Feb. 2021). *Video Streaming Market Size, Share & Trends Analysis Report By Streaming Type, By Solution, By Platform, By Service, By Revenue Model, By Deployment Type, By User, By Region, And Segment Forecasts, 2021 - 2028*. Recuperado de <https://www.grandviewresearch.com/industry-analysis/video-streaming-market>
2. C. Morales (2016). *Tesis de evaluación: Netflix*. (Tesis de Maestría, Universidad de San Andrés, Argentina). Recuperada de <https://repositorio.udesa.edu.ar/jspui/bitstream/10908/11911/1/%5BP%5D%5BW%5D%20T.M.%20Fin.%20Morales%2C%20Carolina.pdf>
3. Amazon Rekognition Service, recuperado de <https://aws.amazon.com/rekognition/?blog-cards.sort-by=item.additionalFields.createdDate&blog-cards.sort-order=desc>
4. S. Foucher, L. Gagnon (2007). *Automatic Detection and Clustering of Actor Faces based on Spectral Clustering Techniques*. 4th Canadian Conference on Computer and Robot Vision, Montreal. Recuperado de <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.330.9953&rep=rep1&type=pdf>
5. Vineet Gandhi, Rémi Ronfard. Detecting and Naming Actors in Movies using Generative Appearance Models. CVPR 2013 - International Conference on Computer Vision and Pattern Recognition, IEEE, Jun 2013, Portland, Oregon, United States. pp.3706-3713, ff10.1109/CVPR.2013.475ff. Ffhal-00814197. Recuperado de <https://hal.inria.fr/hal-00814197/document>

6. Jafri, Rabia & Arabnia, Hamid. (2009). A Survey of Face Recognition Techniques. *JIPS*. 5. 41-68. 10.3745/JIPS.2009.5.2.041. Recuperado de [Jafri, R., & Arabnia, H. R. \(2009\). A survey of face recognition techniques. Jips. 5\(2\), 41-68.](#)
7. Librerias para Python recuperadas en <https://pypi.org/>
8. Navneet Dalal and Bill Triggs. *Histograms of Oriented Gradients for Human Detection* (2005). Recuperado de <https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf>
9. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. *SSD: Single Shot MultiBox Detector*. Cornell University. Recuperado de: <https://arxiv.org/abs/1512.02325>
10. Modelo pre-entrenado por OpenCV, basado en DNN y SSD, recuperado de : <https://github.com/opencv/opencv/tree/master/modules/dnn>
11. OpenFace project, recuperado de: <https://cmusatyalab.github.io/openface/>
12. F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682. https://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf
13. Librería scikit-learn, recuperada de <https://scikit-learn.org/stable/>
14. Repositorio de GitHub del proyecto "Who is on screen?" https://github.com/ssabalain/who_is_on_screen
15. The Movie Database (TMBD) API, <https://www.themoviedb.org/documentation/api>
16. Tiempo en pantalla de los actores presentes en la filmografía completa de Christopher Nolan, recuperado de https://www.imdb.com/list/ls063178960/?sort=list_order.asc&mode=detail&page=1
17. Rugare Maruzani (2021). *Are You Unwittingly Helping to Train Google's AI Models?* Recuperado de <https://towardsdatascience.com/are-you-unwittingly-helping-to-train-googles-ai-models-f318dea53aee>