RECONOCIMIENTO FACIAL NO SUPERVISADO EN PROGRAMAS DE TELEVISIÓN

Néstor Llamas Llopis

- Universidad Politécnica de Cataluña (**UPC**)
- Escuela Superior de Ingenierías Industrial, Aeroespacial y Audiovisual de Tarrasa (**ESEIAAT**)
- Departamento de Teoría de la Señal y Comunicaciones (**TSC**)
- Grupo de Procesamiento de Imagen (**GPI**)

Barcelona, junio de 2017

A mi madre, por alentarme todos estos años
a esforzarme y seguir adelante en mis estudios.

AGRADECIMIENTOS

En primer lugar, quisiera agradecer la coordinación y supervisión del director de este proyecto, Josep Ramon Morros Rubió, por la guía de desarrollo que ha ido manejando, así como por sus reuniones magistrales. Cabe destacar también el agradable trato mantenido durante los diversos meses que ha durado la elaboración del proyecto, así como la paciencia demostrada ante un largo parón de trabajo debido a problemas personales que no vienen al caso.

En segundo lugar, también quiero dar gracias por el trato incondicional recibido por Albert Gil Moreno, ingeniero de *software* del Grupo de Procesamiento de Imagen (GPI) al cual he pertenecido durante mi estancia como alumno desarrollador de un Proyecto Final de Grado (PFG). Cada vez que me encontraba ante algún problema, él, como personal técnico del grupo, ha sabido dar con el error y me ha dado respuesta eficaz e inmediata.

En tercer lugar, es de agradecer el trato cordial y las cuestiones administrativas por parte del jefe de estudios, Jordi Voltas Aguilar, y del personal del Servicio de Información y Atención al Estudiantado (SIAE). Me han dado solución a todas las cuestiones burocráticas que me han ido surgiendo en relación al proyecto, así como también temas de papeleo que con carácter especial me han sido favorecidos.

En cuarto lugar y, no por ello menos importante, mis agradecimientos a mi propia familia, por apoyarme en la elaboración de este proyecto, así como por las continuas facilidades y herramientas que me han ido brindando en cada ocasión. A mi madre, por la parte emocional que conlleva elaborar un proyecto de este calibre. A mi padre, por llevarme y traerme en coche a las numerosas reuniones que he ido manteniendo con mi tutor. Y a mi hermano, por proporcionarme herramientas que me han facilitado el desarrollo del trabajo.

A todos ellos, sin más reparo, muchas gracias. Nada hubiera sido posible si por mí solo hubiera realizado este proyecto de tal envergadura.

RESUMEN

La enorme cantidad de datos visuales que se genera actualmente, especialmente vídeos, crea una fuerte necesidad de herramientas de anotación que hagan posible la búsqueda y recuperación de información presente en dichos datos visuales. La anotación automática en programas de televisión es una funcionalidad clave en aplicaciones de *video retrieval* sobre grandes bases de datos visuales. Una de las informaciones más relevantes es la identidad de las personas. En este contexto, la anotación consiste en determinar de forma automática la identidad y la localización temporal de las personas que aparecen en el programa de televisión utilizando técnicas de *video tracking* y de reconocimiento facial.

En este proyecto se realiza una anotación automática en programas de televisión, la cual consiste en ir creando automáticamente una base de datos con las identidades de las personas que van apareciendo, donde a cada identidad le asignamos todos los *frames* que pertenecen a la misma persona y que se han obtenido de distintos instantes de tiempo. Cada identidad tendrá asociado su correspondiente nombre.

El reconocimiento facial empleado para identificar no se supervisa puesto que no interviene ningún supervisor externo que previamente haya establecido unos determinados modelos de identidades con los que comparar para reconocer, sino que es la propia base de datos la que se va creando a partir de la información presente en los vídeos. Esta información es el nombre de las identidades que van apareciendo.

La técnica de reconocimiento facial utilizada es la llamada «*Sparse Representation*». Se basa en imágenes frontales y es robusta tanto a cambios de expresión facial y de iluminación, como a algún tipo de oclusión, corrupción u objeto de máscara o disfraz.

PALABRAS CLAVE | Reconocimiento facial; recuperación de vídeos; anotación automática; reconocimiento facial no supervisado; seguimiento de vídeos; *Sparse Representation*; MATLAB.

RESUM

L'enorme quantitat de dades visuals que es genera actualment, especialment vídeos, crea una forta necessitat d'eines d'anotació que facin possible la recerca i recuperació d'informació present en aquestes dades visuals. L'anotació automàtica en programes de televisió és una funcionalitat clau en aplicacions de *video retrieval* sobre grans bases de dades visuals. Una de les informacions més rellevants és la identitat de les persones. En aquest context, l'anotació consisteix a determinar de forma automàtica la identitat i la localització temporal de les persones que apareixen al programa de televisió utilitzant tècniques de *video tracking* i de reconeixement facial.

En aquest projecte es realitza una anotació automàtica en programes de televisió, la qual consisteix a anar creant automàticament una base de dades amb les identitats de les persones que van apareixent, on a cada identitat li assignem tots els *frames* que pertanyen a la mateixa persona i que s'han obtingut de diferents instants de temps. Cada identitat tindrà associat el seu corresponent nom.

El reconeixement facial emprat per identificar no se supervisa ja que no intervé cap supervisor extern que prèviament hagi establert uns determinats models d'identitats amb els quals comparar per reconèixer, sinó que és la pròpia base de dades la que es va creant a partir de la informació present en els vídeos. Aquesta informació és el nom de les identitats que van apareixent.

La tècnica de reconeixement facial utilitzada és l'anomenada «*Sparse Representation*». Es basa en imatges frontals i és robusta tant a canvis d'expressió facial i d'il·luminació, com a algun tipus d'oclusió, corrupció o objecte de màscara o disfressa.

PARAULES CLAU | Reconeixement facial; recuperació de vídeos; anotació automàtica; reconeixement facial no supervisat; seguiment de vídeos; *Sparse Representation*; MATLAB.

ABSTRACT

The enormous amount of visual data that is currently generated, especially videos, creates a strong need for annotation tools that make it possible to search for and retrieve information found in such visual data. Automatic annotation in television programs is a key functionality in video retrieval applications on large visual databases. One of the most relevant information is the identity of the people. In this context, the annotation consists of automatically determining the identity and the temporary location of those who appear in the television program using video tracking and face recognition techniques.

In this project an automatic annotation is made in television programs, which consists of automatically creating a database with the identities of the people that are appearing, where to each identity we assign all the frames that belong to the same person and have been obtained from different instants of time. Each identity will have associated its corresponding name.

Face recognition used to identify is not supervised since there is no external supervisor who had previously established certain identity models with which to compare in order to recognize, but it is the database itself that is created from the information contained in the videos. This information is the name of the identities that are appearing.

The face recognition technique used is the so-called "Sparse Representation". It is based on frontal images and is robust both to changes of face expression and illumination, as well as to some type of occlusion, corruption or object of mask or disguise.

KEYWORDS | Face recognition; video retrieval; automatic annotation; unsupervised face recognition; video tracking, Sparse Representation; MATLAB.

ÍNDICES

Índice de contenidos

Índice de figuras

Índice de tablas

Índice de siglas

2D	Dos dimensiones
3D	Tres dimensiones
AP	*Average Precision*
ASRC	*Adaptive Sparse Representation-based Classification*
AUC	*Area Under Curve*
CPU	*Central Processing Unit*
ECTS	*European Credit Transfer and Accumulation System*
ESEIAAT	Escuela Superior de Ingenierías Industrial, Aeroespacial y Audiovisual de Tarrasa
FN	*False Negative*
FP	*False Positive*
FPR	*False Positive Rate*
GPI	Grupo de Procesamiento de Imagen
GPU	*Graphics Processor Unit*
ID	Identificador
IVA	Impuesto sobre el Valor Añadido
JAFFE	*Japanese Female Facial Expression*
JSRC	*Joint Sparse Representation-based Classification*
LBP	*Local Binary Pattern*
MAP	*Mean Average Precision*
MSSRC	*Mean Sequence Sparse Representation-based Classification*
PCA	*Principal Component Analysis*
PFG	Proyecto Final de Grado
PR	*Precision-Recall*
RAM	*Random Access Memory*
RGB	*Red, Green, Blue*
ROC	*Receiver Operating Characteristic*
RR	*Recognition Rate*
RSRC	*Regularized Sparse Representation-based Classification*
SCI	*Sparsity Concentration Index*
SIAE	Servicio de Información y Atención al Estudiantado
SRC	*Sparse Representation-based Classification*
SRC-KNS	*Sparse Representation-based Classification on K-Nearest Subspace*
TB	*Terabyte*
TN	*True Negative*
TP	*True Positive*
TPR	*True Positive Rate*
TSC	Teoría de la Señal y Comunicaciones

TVC	Televisión de Cataluña
UPC	Universidad Politécnica de Cataluña
WSRC	*Weighted Sparse Representation-based Classification*

CAPÍTULO 1. INTRODUCCIÓN

Un sistema de reconocimiento facial es una aplicación informática capaz de identificar o verificar una persona a partir de una imagen digital o de un fotograma de vídeo. Una de las maneras de hacer esto es comparando las características faciales de la imagen con una base de datos previamente almacenada. Este sistema se puede comparar con otros sistemas biométricos tales como las huellas dactilares o el reconocimiento del iris de los ojos [1].

Actualmente es una de las aplicaciones de análisis de imagen más activas y exitosas. Es esencial para comunicaciones e interacciones efectivas entre personas y se utiliza en numerosas aplicaciones prácticas, tales como identificación de cuentas bancarias, control de acceso, búsqueda de fichas policiales, monitorización de seguridad y sistema de vigilancia [2], [3].

Este Proyecto Final de Grado (PFG) viene precedido de un trabajo de investigación donde se aprende el estado del arte actual, es decir, los estudios de reconocimiento facial realizados hasta la fecha. Posteriormente, se implementa el propio método de reconocimiento y se realizan los experimentos convenientes junto con sus correspondientes análisis de resultados, los cuales nos indicarán la fiabilidad y perfección del método. Dichos experimentos se realizan primero para un método de reconocimiento a partir de imágenes estáticas y, después, para el método completo de reconocimiento a partir de vídeos. Las imágenes se adquieren de bases de datos estándar, mientras que los vídeos corresponden a programas de televisión, los cuales han sido proporcionados por los organizadores de MediaEval [4]. Ésta es una iniciativa de *benchmarking* dedicada a evaluar nuevos algoritmos de acceso y recuperación multimedia.

La finalidad de este proyecto es realizar una anotación automática en programas de televisión [5], [6] a partir de unos datos de *video tracking* [7] obtenidos por los organizadores de MediaEval. Dicha anotación consiste en ir creando automáticamente una base de datos con las identidades de las personas que van apareciendo, donde a cada identidad le asignamos todos los *frames* que pertenecen a la misma persona y que se han obtenido de distintos instantes de tiempo. Cada identidad tendrá asociado su correspondiente nombre. Por otra parte, el *video tracking* es el proceso de estimar en el tiempo la ubicación de una identidad móvil, es decir, consiste en hacer un seguimiento de una identidad en concreto, *frame* por *frame*, en el tiempo y en el espacio. En este ámbito necesitaremos definir un concepto nuevo: el *track*. Éste es una secuencia de *frames* consecutivos delimitados por un intervalo de tiempo y correspondientes a una misma persona, donde para cada *frame* ha habido un proceso de detección de dicha identidad.

El reconocimiento facial empleado para identificar no se supervisa [8] puesto que no interviene ningún supervisor externo que previamente haya establecido unos determinados modelos de identidades con los que comparar para reconocer, sino que es la propia base de datos la que se va creando a partir de la información presente en los vídeos. Esta información es el nombre de las identidades que van apareciendo. Se acaba de citar un concepto nuevo: el

modelo. Éste es un conjunto de imágenes pertenecientes a una misma identidad. La técnica de reconocimiento facial [9] utilizada es la llamada «*Sparse Representation*» [10]–[12].

La implementación del método se ha realizado en el servidor del Grupo de Procesamiento de Imagen (GPI) [13] utilizando para ello el lenguaje de programación MATLAB [14].

1.1. Motivación del proyecto

La motivación principal de hacer real este proyecto es el de poder implementar un sistema que sea capaz por sí solo de llevar a cabo todo un proceso de anotación de forma automática mediante un reconocimiento facial no supervisado.

Numerosos estudios sobre métodos de anotación manuales, semiautomáticos y automáticos [15] se han ido realizando a lo largo de los años sobre diversos tipos de información visual. En este caso se trata el tema de la anotación de identidades de los interlocutores en los programas de televisión de una forma automática y no supervisada, hecho que *a priori* no se encuentra fácilmente por la red, por lo que supuestamente somos de los primeros en llevarlo a cabo. En [16] se comenta que recientes estudios se han realizado con proyectos de estas características, pero con una principal diferencia: la extracción de los nombres de las personas se hace a partir de dichos nombres pronunciados, es decir, a partir de una señal de audio. Se ha demostrado que esta fuente puede ser imprecisa y proporciona un nivel de confianza bajo, por lo que en este proyecto la extracción se realiza a partir de los nombres mostrados como texto, es decir, a partir de una señal de imagen, tal y como se propone en [16].

Además, existe especial interés en llevar a cabo este trabajo puesto que precisamente ésta es una tarea propuesta por el concurso ofrecido por MediaEval, donde se propone a participantes de todo el mundo llegar a obtener las mejores tasas de rendimiento [17].

También motiva para este proyecto la importancia de la técnica de reconocimiento empleada: *Sparse Representation*. En [10] se comenta que según este método ya no es tan importante la elección de la técnica de extracción de características [18], ni tampoco cuánta oclusión en la imagen pueda haber. El motivo por el cual esta técnica es tan exitosa también es debido principalmente a que las clases de señales importantes tales como las imágenes y el audio tienen representaciones dispersas (*sparse representations*) naturales con respecto a bases fijas (como por ejemplo Fourier o Wavelet) [11].

1.2. Objetivos del proyecto

Los objetivos principales que se deberán de poder solventar de un modo razonable en este proyecto son los siguientes:

- Utilizar *Sparse Representation* como técnica de reconocimiento facial.
- Implementar un sistema de anotación automática en vídeos.
- Implementar un sistema de reconocimiento facial no supervisado en vídeos.

- **Integrar funcionalidades** ya implementadas por otras personas. Por ejemplo, las referentes a *video tracking*.
- Implementar el sistema completo en lenguaje de programación **MATLAB**.
- Obtener buenas **tasas de rendimiento**, no solamente que verifiquen la fiabilidad del método, sino la perfección de éste.
- Obtener resultados óptimos en el menor **tiempo** posible implementando para ello el algoritmo de una forma eficiente.
- Testear el método implementado en imágenes mediante **varias bases de datos** para comprobar en qué condiciones el sistema de *Sparse Representation* es más robusto.
- Testear el método implementado en vídeos mediante **varios programas de televisión** para comprobar la robustez del sistema ante todos los posibles casos de reconocimiento facial que se puedan dar.

1.3. Estructura del documento

Este **PFG** se puede dividir en los siguientes ítems:

1. Lectura de la **bibliografía**: artículos científicos y fuentes de información electrónicas.
2. **Implementación** del método: reconocimiento en imágenes y reconocimiento en vídeos.
3. Redacción de la **memoria**: informe explicativo del proyecto.
4. Elaboración de la **presentación**: informe de defensa del proyecto.

Este documento hace referencia a la **memoria** del PFG y se estructura tal y como se detalla a continuación. Primero estudiamos el **estado del arte**, una parte literaria donde el lector puede encontrar una visión global de estudios llevados a cabo hasta le fecha relacionados con el reconocimiento facial. Se hace hincapié en aspectos directamente relacionados con el proyecto en el apartado de *background*. Posteriormente, se redacta el **método implementado**, es decir, el método de reconocimiento facial que se ha llevado a cabo para solventar los objetivos iniciales del proyecto. A continuación, se detallan las **pruebas experimentales** realizadas, así como también los resultados obtenidos para cada una de éstas. Estos test reflejan la fiabilidad y perfección del método previamente implementado. Luego, se realiza un análisis de la estimación del **presupuesto** necesario para llevar a cabo este trabajo como un proyecto real. Se toma en cuenta el tiempo invertido, así como también las herramientas de *hardware* y *software* utilizadas. Finalmente, se termina el proyecto con unas **conclusiones** extraídas tras finalizar el mismo. Conforman este apartado unas conclusiones desde un punto de vista técnico y unas líneas de futuro para promover la investigación sobre nuevos proyectos.

CAPÍTULO 2. ESTADO DEL ARTE

2.1. Revisión literaria

El **reconocimiento facial** actualmente es una de las aplicaciones de **análisis de imagen** más activas y exitosas, una tarea muy investigada en los últimos años. Es esencial para comunicaciones e interacciones efectivas entre personas y se utiliza en numerosas aplicaciones prácticas, tales como identificación de cuentas bancarias, control de acceso, búsqueda de fichas policiales, monitorización de seguridad y sistema de vigilancia. Todas ellas se pueden englobar en dos grupos generales: **aplicaciones comerciales** —como por ejemplo el control de acceso— y **aplicaciones del cumplimiento de la ley** —como por ejemplo la videovigilancia— [2], [3].

Gracias a la investigación desarrollada en los últimos años se dispone hoy en día de tecnologías bastante factibles. Aunque los sistemas de reconocimiento mediante computador han alcanzado ya un cierto nivel de madurez, aún están **lejos de la capacidad del sistema de reconocimiento humano**. Esto es debido a las **limitaciones** que presentan las aplicaciones reales, como son los cambios de **iluminación**, las **posturas**, o las **imágenes al aire libre**, los principales retos aún sin resolver de una forma convincente. Además, el reconocimiento también resulta complicado cuando se comparan imágenes de una misma persona adquiridas en **años distantes** [2].

Cabe destacar que en este tipo de métodos de identificación personal **no se requiere de cooperación o conocimiento de la persona a reconocer**, ya que únicamente son necesarias algunas imágenes frontales o de perfil para su análisis, a diferencia de otros métodos como el análisis de huellas dactilares o el escaneo de iris [2].

Los sistemas de técnicas de reconocimiento facial pueden hacer uso de **imágenes estáticas** o bien de **secuencias de vídeo**. A partir de estos datos de entrada, el reconocimiento consiste en **identificar o verificar** las personas presentes en la escena utilizando para ello una base de datos de caras previamente almacenada. Para ello, primero se realiza una detección de caras dentro de la escena, después se lleva a cabo un proceso de normalización de dichas caras, más tarde se realiza una extracción de características de las diferentes regiones faciales, y finalmente se lleva a cabo un proceso de reconocimiento facial comparando de alguna manera con la información de la base de datos [2].

La **identificación** consiste en **determinar la identidad de la persona desconocida** a reconocer asignando una de las identidades conocidas de la base de datos, mientras que la **verificación** consiste en **afirmar o negar la identidad de la persona que se hace saber** comparando con las identidades conocidas de la base de datos. En cualquier caso, se reconocen objetos en 3D a partir de imágenes en 2D. En el caso de secuencias de vídeo originalmente el reconocimiento se basaba en técnicas basadas en imágenes estáticas, de modo que el sistema automáticamente trataba *frames* individualmente. No obstante, **actualmente se utilizan técnicas**

más avanzadas que logran un mejor rendimiento basándose en *tracking*, donde se sintetizan imágenes frontales mediante estimación de posición y profundidad en el vídeo [2].

La identificación es biométrica puesto que se tienen en cuenta características fisiológicas de la cara y patrones de comportamiento, como por ejemplo la voz. Tiene la ventaja de ser un sistema pasivo, es decir, no necesita de la presencia del usuario, además de ser natural y amigable [3].

En [19] se explica que el conocimiento avanzado acerca de las maneras en que la gente se reconoce entre sí ayuda a desarrollar prácticos sistemas de reconocimiento facial automáticos. Un objetivo primordial para los investigadores en visión por computador es llegar a superar el rendimiento humano en cuanto al reconocimiento facial se refiere. Algunos aspectos importantes que se deben de tener en cuenta a la hora de implementar un sistema y que tienen que ver con la estrecha relación entre neurociencia visual y visión por computador son los siguientes [2], [19]:

- El sistema visual dedica recursos neuronales especializados en la percepción facial.
- El sistema visual empieza el reconocimiento con una preferencia por patrones faciales.
- Las características faciales son procesadas holísticamente con una visión global facial.
- Las cejas son las características faciales más importantes para el reconocimiento.
- La parte superior de la cara es más útil para el reconocimiento que la parte inferior.
- Las características internas son más importantes que las externas para caras familiares.
- La nariz es más importante que los ojos o la boca en imágenes de perfil.
- Cuanto más atractiva es una cara mayor es su tasa de reconocimiento.
- Somos capaces de reconocer caras familiares en imágenes de poca resolución.
- Con caras familiares se puede tolerar un cierto grado de degradación en la imagen.
- La información de alta frecuencia es insuficiente para un buen reconocimiento.
- Las señales de color son tan importantes como las señales de forma.
- Las señales de color son importantes sobre todo al degradarse las señales de forma.
- La inversión de contraste afecta al reconocimiento debido al uso de señales de color.
- Al aplicar compresión en anchura y altura los ratios de distancias no cambian.
- La identidad facial y la expresión deben ser procesados por sistemas separados.
- La forma de la cara puede ser codificada en forma de caricatura exagerando los rasgos.
- El movimiento de la cara facilita el posterior reconocimiento.

Es cierto que existen técnicas de reconocimiento para cada específica variación, como por ejemplo cambios de iluminación, pero no existe ninguna técnica global que ayude a reconocer bajo cualquier circunstancia. Es por ello que el reconocimiento de caras es aún un problema sin resolver, si además consideramos que en aplicaciones reales es posible que dispongamos de una base de datos algo limitada. Los retos más relevantes a estudiar son el cambio del ángulo de postura, el cambio del ángulo de iluminación incidente, y el cambio de la tonalidad de iluminación incidente [3].

Dependiendo de la metodología de adquisición de datos faciales, las técnicas de reconocimiento facial se pueden clasificar en general en dos categorías: las que tratan con imágenes estáticas y las que operan con secuencias de vídeo. En los siguientes apartados se estudiarán ambas categorías.

2.1.1. Reconocimiento en imágenes

En [2] se explica que el problema de reconocimiento facial automático puede ser dividido en tres pasos: detección de caras, extracción de características de caras, e identificación o verificación de caras.

Con tal de categorizar los sistemas de reconocimiento facial que se conocen hasta la fecha, podemos diferenciar entre varios tipos de métodos generales sugeridos por un estudio psicológico en cuanto a cómo los humanos utilizan características holísticas o locales: métodos holísticos, métodos basados en características, y métodos híbridos. Los métodos holísticos utilizan la región facial entera como la entrada al sistema de reconocimiento. En los métodos basados en características las características locales tales como los ojos, la nariz, o la boca son primero extraídas y sus localizaciones y estadísticas locales (geométricas y/o apariencia) se introducen en un clasificador estructural. Al igual que el sistema de percepción humano, los métodos híbridos utilizan tanto la región facial entera como las características locales. Estos últimos métodos, como sistemas de reconocimiento automático, deberían de ofrecer lo mejor de los dos tipos de métodos, tal y como hace también el sistema de percepción humano.

Como ya se ha comentado anteriormente, de toda la literatura existente hasta el momento nos centraremos en la técnica *Sparse Representation*. Dentro del ámbito de imágenes estáticas, de esta técnica podemos encontrar fácilmente por la red muchas variantes en cuanto al ámbito de reconocimiento facial se refiere, al igual como también pasa con otras muchas técnicas. En este apartado se intentarán reunir unas cuantas variantes.

En [20] se propone una variante de la técnica *Sparse Representation-based Classification* (SRC). Dicha variante se denomina *Weighted Sparse Representation-based Classification* (WSRC). Dada una muestra de test, WSRC calcula el peso para una muestra de entrenamiento del diccionario conforme a la distancia o relación de semejanza entre la muestra de test y la muestra de entrenamiento. Después, representa la muestra de test explotando las muestras de entrenamiento ponderadas basándose en la norma l_1, y clasifica la muestra de test usando los resultados de representación dispersa. El objetivo de WSRC es que, dada una muestra de test, esta técnica presta más atención a aquellas muestras de entrenamiento que son más similares a la muestra de test en representar dicha muestra de test. En general, el resultado de representación de WSRC es más disperso que el de SRC. Los experimentos realizados sobre varias bases de datos demuestran que el algoritmo propuesto puede lograr un rendimiento de reconocimiento deseable y llegar a obtener incluso mejores resultados que varios métodos actuales, incluyendo la técnica SRC.

En [21] se da a conocer otra variante de la técnica SRC, la llamada *Adaptive Sparse Representation-based Classification* (ASRC). El problema de SRC es que enfatiza demasiado la dispersión y pasa por alto la información de correlación, la cual ha demostrado ser crítica en problemas reales de reconocimiento facial. Por otro lado, otras técnicas consideran la correlación, pero pasan por alto la habilidad discriminativa de la dispersión. A diferencia de todas éstas, la técnica ASRC considera conjuntamente tanto la dispersión como la correlación. Concretamente, cuando las muestras de entrenamiento del diccionario son de baja correlación, ASRC selecciona las muestras más discriminatorias para la representación, como SRC, mientras que cuando las muestras de entrenamiento son de alta correlación, ASRC selecciona la mayoría de las muestras correlacionadas y discriminatorias para la representación, en lugar de elegir algunas muestras relacionadas al azar. En general, el modelo de representación es adaptable a la estructura de correlación, la cual se beneficia tanto de la norma l_1 como de la norma l_2. Los experimentos realizados sobre varias bases de datos verifican la efectividad y robustez del algoritmo propuesto comparándolo con métodos modernos e incluso con la propia técnica SRC.

En [22] se presenta una nueva variante de la técnica SRC, la denominada *Sparse Representation-based Classification on K-Nearest Subspace* (SRC-KNS). Un aspecto negativo que tiene SRC es que su complejidad computacional es muy alta debido a la resolución de un problema de minimización de la norma l_1 complejo. SRC-KNS surge con la idea principal de mejorar la eficiencia de este cálculo. Este nuevo método primero explota la distancia entre la imagen de test y el subespacio de cada modelo individual para determinar los k subespacios más cercanos, y entonces realiza la técnica SRC sobre los k modelos seleccionados. SRC-KNS es capaz de reducir en gran medida la complejidad del problema de representación dispersa, ya que el cálculo para determinar los k subespacios más cercanos es bastante simple. Por consiguiente, SRC-KNS tiene una complejidad computacional mucho más baja que la técnica original SRC. También se propone el algoritmo SRC-KSN modular para el caso del reconocimiento de caras ocluidas. En dicho método primero las imágenes son particionadas en bloques. Después, se propone un indicador con tal de eliminar los bloques contaminados y elegir los k subespacios más cercanos. Finalmente, se utiliza el método SRC para clasificar la muestra de test ocluida en el nuevo espacio de características. En comparación con la aproximación utilizada en el método SRC original, el método SRC-KNS modular puede reducir también en gran medida la carga computacional. Los experimentos realizados sobre varias bases de datos muestran que este método es al menos cinco veces más rápido en comparación con el método original SRC, y se logran tasas de reconocimiento comparables o incluso mejores. También se logran mejores resultados que otras varias técnicas modernas.

2.1.2. Reconocimiento en vídeos

En [2] se explica que un típico sistema de reconocimiento facial basado en vídeo detecta regiones faciales, extrae características faciales del vídeo y reconoce la identidad facial si una

cara es presente. En las aplicaciones de vigilancia, seguridad de la información, y control de acceso, el reconocimiento facial a partir de una secuencia de vídeo es un problema importante.

El reconocimiento facial basado en vídeo es preferible a usar imágenes estáticas, ya que el movimiento ayuda en el reconocimiento de caras familiares cuando las imágenes son negadas, invertidas, o cuando se aplica un umbral. También se ha demostrado que los humanos pueden reconocer caras animadas mejor que imágenes reordenadas aleatoriamente del mismo conjunto. El reconocimiento de caras a partir de una secuencia de vídeo es una extensión directa del reconocimiento basado en imágenes estáticas, de manera que las técnicas de reconocimiento facial basadas en vídeo deben de usar coherentemente la información espacial y temporal.

Hay algunos aspectos aún sin resolver de una forma convincente en secuencias de vídeo. Por ejemplo, que la calidad del vídeo suele ser baja y las personas no son cooperativas para el reconocimiento, de modo que existen grandes variaciones de iluminación y posición en las imágenes faciales, así como también son posibles oclusiones parciales y objetos de máscara o disfraz. También hay que tener en cuenta que las imágenes faciales en secuencias de vídeo son de tamaño reducido en comparación con imágenes estáticas debido a la adquisición de éstas.

Como ya se ha comentado anteriormente, de toda la literatura existente hasta el momento nos centraremos en la técnica *Sparse Representation*. Dentro del ámbito de imágenes en movimiento o secuencias de vídeos, de esta técnica podemos encontrar fácilmente por la red muchas variantes en cuanto al ámbito de reconocimiento facial se refiere. En este apartado se intentarán reunir unas cuantas variantes.

En [23] se propone una variante de la técnica SRC, la cual se denomina *Regularized Sparse Representation-based Classification* (RSRC). El principal objetivo del reconocimiento facial a partir de secuencias de vídeo es identificar *tracks* de personas conocidas utilizando un gran diccionario de imágenes faciales estáticas, al mismo tiempo que rechazar también *tracks* de personas desconocidas. Los métodos existentes usan modelos probabilísticos en base a *frame* por *frame* para identificar caras, lo que es computacionalmente costoso cuando el tamaño de los datos es amplio. Con tal de superar este inconveniente la técnica RSRC utiliza la aproximación de minimización de la norma l_2 en lugar de la tradicional minimización de la norma l_1, y obtiene un único vector de coeficientes para todos los *frames* en lugar de un vector por *frame*. Gracias a la minimización de la norma l_2 se logran ratios más dispersos y los errores residuales sobre los *frames* son reducidos. Los experimentos realizados utilizando secuencias de vídeo sin restricciones demuestran que, debido al mínimo error obtenido, el algoritmo propuesto puede lograr un rendimiento de reconocimiento deseable y llegar a obtener mejores resultados que varios métodos actuales de reconocimiento facial a partir de secuencias de vídeo, incluyendo la propia técnica SRC.

En [24] se da a conocer otra variante de la técnica SRC, la llamada *Joint Sparse Representation-based Classification* (JSRC). Un desafío clave en el reconocimiento facial a partir de secuencias de vídeo es explotar la información extra disponible en el vídeo, como por

ejemplo la cara o el cuerpo. Diferentes secuencias de vídeo del mismo sujeto pueden contener variaciones en resolución, iluminación, pose, y expresiones faciales, lo que contribuye a desafíos en diseñar un algoritmo de reconocimiento facial efectivo. La técnica JSRC tiene en cuenta simultáneamente tanto correlaciones como información de acoplamiento entre los *frames* del vídeo. El método representa conjuntamente todos los datos del vídeo mediante una combinación lineal dispersa de los datos de entrenamiento. Este modelo también es robusto ante la presencia de ruido y oclusión. Los experimentos realizados utilizando secuencias de vídeo sin restricciones verifican la eficacia del método y logran un mejor rendimiento que varios de los algoritmos actuales de reconocimiento facial a partir de secuencias de vídeo, incluyendo la propia técnica SRC.

En [25] se presenta una nueva variante de la técnica SRC, la denominada *Mean Sequence Sparse Representation-based Classification* (MSSRC). Una sencilla aplicación de la popular minimización de la norma l_1 para reconocimiento facial en base a *frame* por *frame* es prohibitivamente costoso. MSSRC realiza el reconocimiento facial a partir de secuencias de vídeo utilizando una optimización conjunta aprovechando todos los datos del vídeo disponibles y el conocimiento de que los *frames* de un *track* en cuestión pertenecen a la misma persona. Añadiendo una restricción temporal estricta a la minimización de la norma l_1 que fuerza a todos los *frames* individuales de un *track* en cuestión a reconstruir una única identidad, se muestra que la optimización se reduce a una única minimización sobre el promedio del *track*. Los experimentos realizados utilizando secuencias de vídeo sin restricciones muestran que este método consigue lograr un mejor rendimiento que varios métodos modernos de reconocimiento facial a partir de secuencias de vídeo, incluyendo la propia técnica SRC.

2.2. *Background*

2.2.1. Extracción de características

2.2.1.1. Técnica holística

En [2] se explica que las técnicas holísticas utilizan la región facial entera como entrada del sistema de reconocimiento, por lo que los descriptores que definen las características de cada imagen son globales. Una de las representaciones más extensamente usada de la región facial es *Eigenfaces*, la cual se basa en la técnica *Principal Component Analysis* (PCA), una aproximación estadística utilizada para reducir el número de variables en el reconocimiento facial. En [26] se comenta que en el método PCA cada imagen del conjunto de entrenamiento se representa como una combinación lineal de eigenvectores (vectores propios) ponderados llamados *Eigenfaces*. Los eigenvectores son obtenidos de una matriz de covarianza del conjunto de imágenes de entrenamiento, mientras que los pesos se descubren tras seleccionar las *Eigenfaces* más relevantes.

En el método propuesto la técnica escogida hace uso de todos los píxeles de la imagen para comparar caras entre sí. Ya veremos que en verdad al final cogemos menos píxeles de

información, ya que hemos de considerar de que se aplica una reducción de resolución para que todas las imágenes se puedan comparar entre ellas con el mismo número de píxeles. Así pues, tomaremos en cuenta todos los píxeles de las imágenes disminuidas de resolución.

2.2.1.2. Técnica *Random Pixels*

En la técnica *Random Pixels* que se propone en el método implementado los descriptores que definen las características de cada imagen se pueden considerar globales, ya que utilizan unos píxeles cualesquiera en función de toda la región facial entera.

En concreto, se escoge un determinado número de píxeles escogidos aleatoriamente de toda la imagen. Las coordenadas de los píxeles escogidos aleatoriamente en cada imagen deberán de ser los mismos para poder comparar caras entre sí en las mismas condiciones. Así pues, siempre se sigue el mismo patrón de coordenadas. Al igual que con la técnica holística, los píxeles aleatorios se escogen de cada imagen una vez ésta ha sido disminuida de resolución.

A priori, puede parecer que esta técnica sea poco fiable, pero ya veremos como con esta simple técnica que no requiere de gran implementación se obtienen mejores resultados que otras técnicas mucho más complejas, como las propuestas en esta implementación.

2.2.1.3. Técnica *Local Binary Pattern*

La técnica *Local Binary Pattern* (LBP) propuesta en este método se basa en construir un descriptor a nivel local en la imagen, por lo que se diferencia de las técnicas holísticas o globales anteriores en que no parte de la región facial entera.

En [27] se explica que este descriptor, como su nombre indica, se basa en un patrón binario local, el cual contiene características de textura. Los descriptores de textura se aplican a nivel local puesto que hacen el promedio de niveles de gris sobre un área de la imagen, de modo que son invariantes a translación y rotación. El descriptor LBP es robusto a cambios de expresión facial, iluminación, posición, y edad. Tolera cambios de escala de gris monótonos, no es necesaria ninguna previa normalización de escala de gris, y es computacionalmente eficiente.

La idea de usar este descriptor es que las caras se pueden entender como una composición de micro-patrones, los cuales quedan bien descritos por el llamado operador LBP. Se divide la imagen en regiones o bloques. Estas regiones pueden ser rectangulares o circulares, de tamaños diferentes, e incluso con superposición entre consecutivas regiones.

Figura 1: División de una imagen en regiones.
La imagen se divide en 7x7, 5x5, y 3x3 regiones rectangulares, respectivamente [27].

Por cada región, el operador asigna una etiqueta a cada píxel. Cada una de estas etiquetas se obtiene tras umbralizar generalmente una matriz rectangular de dimensiones 3x3 con el valor del píxel en cuestión en el centro de ésta y considerando el resultado como un número binario.

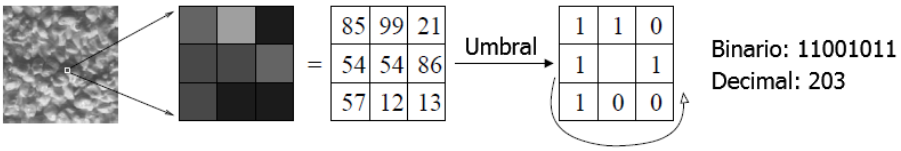

Figura 2: Umbralización de una matriz de dimensiones 3x3.
El valor del píxel central que se está analizando sirve como umbral para los píxeles que tiene a su alrededor [27].

El resultado no es más que la concatenación de números binarios alrededor de cada píxel, los cuales han sido umbralizados, es decir, su valor de nivel de gris original se ha forzado a 0 o 1 en función de un valor de umbral arbitrario. En la figura anterior este umbral es precisamente el valor del píxel central. Si el valor de nivel de gris está por debajo del umbral, éste se transforma en 0, mientras que, si es igual o mayor, éste se transforma en 1. La matriz puede tener otras dimensiones, o incluso diferente forma, como la circular.

Posteriormente, se realiza un histograma con todas las etiquetas obtenidas, de forma que éste será el descriptor de textura, el cual se utiliza para construir un descriptor local por cada región de la imagen. La concatenación de estos descriptores locales formará el descriptor global.

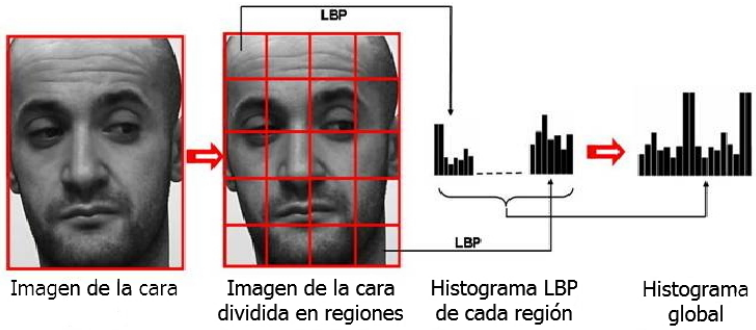

Figura 3: Formación del histograma del descriptor LBP.
El histograma global se forma por la concatenación de histogramas de las distintas regiones [28].

Cabe destacar que cada descriptor local tendrá asignado un peso en concreto en función de su importancia, la cual está ligada con el reconocimiento facial humano. Estos pesos, así como también la división de la imagen en regiones teniendo en cuenta que las caras pueden variar en orientación, se pueden obtener mediante técnicas de aprendizaje. El descriptor ponderado es mejor que el no ponderado tanto en tiempo de procesamiento como en tasa de reconocimiento.

Con tal de optimizar el rendimiento del descriptor se buscan los valores óptimos de parámetros como la forma y tamaño del operador LBP, o el número y pesos de regiones de la imagen. En cualquier caso, se trata de obtener, en la medida de lo posible, el mejor rendimiento de reconocimiento con el menor tamaño posible del descriptor global.

2.2.2. Clasificación

2.2.2.1. Técnica *Sparse Representation*

En [10] se explica que la técnica de clasificación de imágenes *Sparse Representation* se basa en imágenes frontales y es robusta tanto a cambios de expresión facial y de iluminación, como a algún tipo de oclusión, corrupción u objeto de máscara o disfraz.

El método se basa en representar una imagen de test como una «combinación lineal dispersa» de las imágenes correspondientes a los modelos de la base de datos más un cierto ruido denso, o errores debidos a oclusiones o corrupciones. El objetivo de la combinación es que intente utilizar únicamente imágenes del mismo modelo que el de test o, al menos, en su mayoría. De este modo, tan sólo se tiene en cuenta una mínima parte de la base de datos, gracias a un vector de coeficientes cuyos valores son nulos o de poca importancia excepto aquellos asociados a las imágenes correspondientes al mismo modelo que el de test.

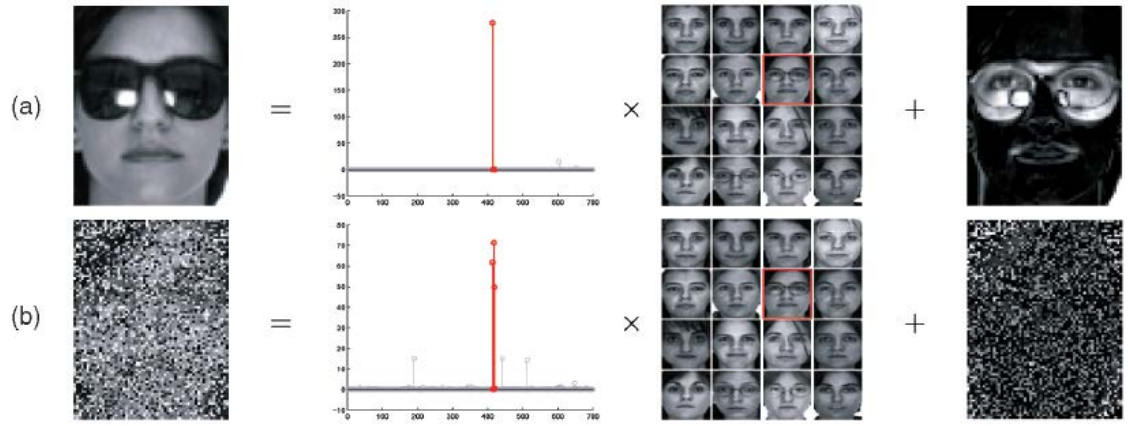

Figura 4: Visión general de la aproximación de *Sparse Representation*.
Una imagen de test se aproxima como una combinación lineal en función de las imágenes de los modelos [10].

En la figura anterior tanto la imagen de test (a) como la imagen de test (b) se representan como una combinación lineal de las imágenes de los modelos gracias a los vectores de coeficientes más un cierto error. En la imagen de test (a) este error es debido a la oclusión que presenta la cara, en este caso unas gafas de sol, ya que en la imagen de modelo la persona está llevando unas gafas normales. En la imagen de test (b) el error es debido a corrupción.

Como decíamos, para una imagen de test de entrada el primer paso consiste en obtener un vector de «coeficientes *sparse*» α' que represente dicha imagen con las muestras de imágenes correspondientes a los modelos de la base de datos. En concreto, se deberá de resolver el problema de minimización de la norma l_1 del vector de coeficientes *sparse* α' dentro del sistema de ecuaciones lineales siguiente:

$$\widehat{\alpha'}_1 = arg\,min_{\alpha'} \|\alpha'\|_1 \quad tal\ que \quad y' = A \cdot \alpha', \qquad (1)$$

donde la matriz A se corresponde con todas las imágenes de los modelos de la base de datos expresadas con vectores de características, y el vector y' se corresponde con la imagen de test, también expresada con un vector de características. Si extrapolamos esta ecuación por una

genérica que tome en cuenta varias imágenes de test de entrada, entonces tendremos la ecuación

$$y = A \cdot \alpha, \qquad (2)$$

donde la matriz y se corresponde ahora con todas las imágenes de test expresadas con vectores de características, y la matriz α se corresponde ahora con los vectores de coeficientes *sparse* calculados para cada una de las imágenes de test.

Antes de clasificar una nueva imagen de test, ésta se ha de someter a un proceso de validación para comprobar si el modelo al que pertenece es alguno de los disponibles en la base de datos. En caso de que la detección no se cumpla se rechazará la imagen de test. La imagen será válida si su representación dispersa tiene coeficientes no nulos concentrados principalmente en un modelo, como en la figura 5, mientras que será inválida si los coeficientes no nulos están propagados por múltiples modelos, como en la figura 6. Así pues, vemos que el proceso de validación depende de la distribución del vector de coeficientes.

Para poder llevar a cabo el proceso de validación primero se obtendrá un valor óptimo de umbral τ que permita diferenciar entre identidades de imágenes de test válidas e inválidas. Este valor se ha de entrenar de algún modo y puede adoptar valores dentro del intervalo $(0, 1)$. Una vez obtenido el valor de este parámetro, por cada imagen de test se calculará el valor llamado *Sparsity Concentration Index* (SCI), el cual se obtiene según la fórmula

$$SCI(\alpha') = \frac{k \cdot \frac{max_i \|\delta_i(\alpha')\|_1}{\|\alpha'\|_1} - 1}{k - 1}, \qquad (3)$$

donde el vector α' se corresponde con los coeficientes *sparse* calculados de una imagen test en concreto, y k se corresponde con el número de modelos existentes en la base de datos. δ_i es la función característica que selecciona los coeficientes *sparse* del vector α' asociados con el modelo i. El valor de SCI puede adoptar valores dentro del intervalo $[0, 1]$ y se compara con el valor óptimo de umbral τ. Una vez calculados ambos valores, una imagen de test se considerará válida si su valor de SCI asociado es mayor o igual que τ:

$$SCI(\alpha') \geq \tau. \qquad (4)$$

Si el valor de SCI es menor que τ, entonces la imagen de test asociado a ese valor de SCI será rechazada por el sistema y no seguirá el proceso de clasificación. Es decir, para las imágenes inválidas no se les reconocerá la identidad, mientras que para las válidas sí.

Después del proceso de validación el sistema debe de tener almacenadas la matriz de características y y la matriz de coeficientes *sparse* α únicamente de aquellas identidades de test validadas. El resto de vectores que pertenecían a ambas matrices habrán quedado descartados y no seguirán el proceso de clasificación.

Una vez validadas las imágenes de test, el método sigue por calcular las diferencias residuales entre cada imagen de test validada y su correspondiente aproximación como

combinación lineal. Esto se traduce en obtener un vector de «residuos *sparse*» por cada imagen de test y en función de cada modelo según la fórmula

$$r_i(y') = \left\| y' - A \cdot \delta_i\left(\widehat{\alpha'}_1\right) \right\|_2, \quad (5)$$

donde la matriz A se corresponde en cada caso con todas las imágenes del modelo i de la base de datos expresadas con vectores de características, el vector y' se corresponde con la imagen de test, también expresada con un vector de características, y el vector α' se corresponde con los coeficientes *sparse* calculados de la imagen de test. δ_i es la función característica que selecciona los coeficientes *sparse* del vector α' asociados con el modelo i. Del mismo modo que con el cálculo de coeficientes *sparse*, si extrapolamos esta ecuación por una genérica que tome en cuenta varias imágenes de test de entrada, entonces tendremos la ecuación

$$r_i(y) = \| y - A \cdot \delta_i(\hat{\alpha}_1) \|_2, \quad (6)$$

donde la matriz A se corresponde ahora con todas las imágenes de los modelos de la base de datos expresadas con vectores de características, la matriz y se corresponde ahora con todas las imágenes de test, también expresadas con vectores de características, y la matriz α se corresponde ahora con los vectores de coeficientes *sparse* calculados para cada una de las imágenes de test.

Así pues, por cada imagen de test validada tendremos asociado un vector de residuos *sparse* con un valor de residuo por cada modelo existente de la base de datos. Cuanto menor sea este valor asociado entre imagen de test y modelo, mayor similitud representará dicha imagen de test con la identidad de dicho modelo.

Finalmente, una vez calculados los residuos *sparse* para cada una de las imágenes de test validadas, el método determina las hipótesis de reconocimiento para cada una de éstas. Para cada imagen se asignará la identidad del modelo que tenga asociado un mínimo valor de residuo. Así pues, vemos que el proceso de clasificación depende de la distribución del vector de residuos. Se deberá de seguir la fórmula

$$identidad(y') = arg\,min_i\ r_i(y'), \quad (7)$$

donde el vector r se corresponde con los residuos *sparse* calculados de la imagen de test. Del mismo modo que con el cálculo de coeficientes *sparse* y residuos *sparse*, si extrapolamos esta fórmula por una genérica que tome en cuenta varias imágenes de test de entrada, entonces tendremos la fórmula

$$identidad(y) = arg\,min_i\ r_i(y), \quad (8)$$

donde la matriz r se corresponde con los residuos *sparse* calculados para cada una de las imágenes de test.

Para las imágenes que previamente hayan resultado inválidas por el sistema de validación se les asignará la identidad propiamente como inválida.

Una vez llegados a este punto ya tendremos resuelto el problema de identificación: a las imágenes que han sido determinadas como válidas se les ha asignado la hipótesis de reconocimiento, mientras que a las imágenes que han sido determinadas como inválidas se les ha asignado la identidad propiamente como inválida.

A continuación, veremos un caso de ejemplo de imagen de test válida:

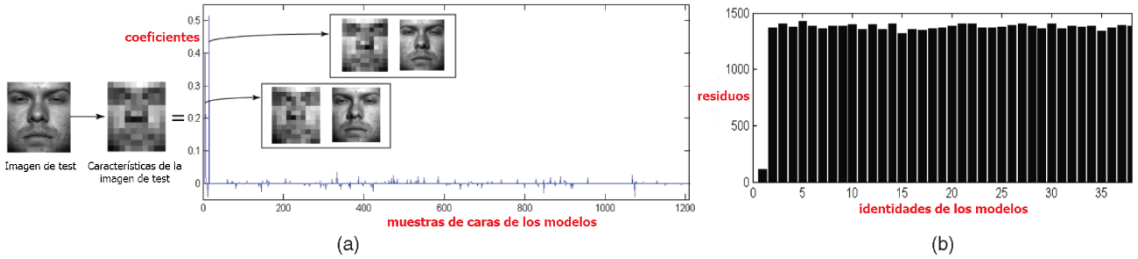

Figura 5: Coeficientes y residuos de una imagen de test válida.
Los coeficientes y los residuos muestran correspondencia con las imágenes de los modelos [10].

En la figura anterior se puede observar como en (a) los coeficientes no nulos están relacionados con muestras de caras pertenecientes a la misma identidad que la de test. En cuanto a los residuos se puede observar como en (b) en este caso el residuo mínimo se corresponde con la identidad 1, la cual es la misma que la de test. Así pues, el cálculo de coeficientes y de residuos ha resultado exitoso en este caso en concreto.

A continuación, veremos un caso de ejemplo de imagen de test inválida:

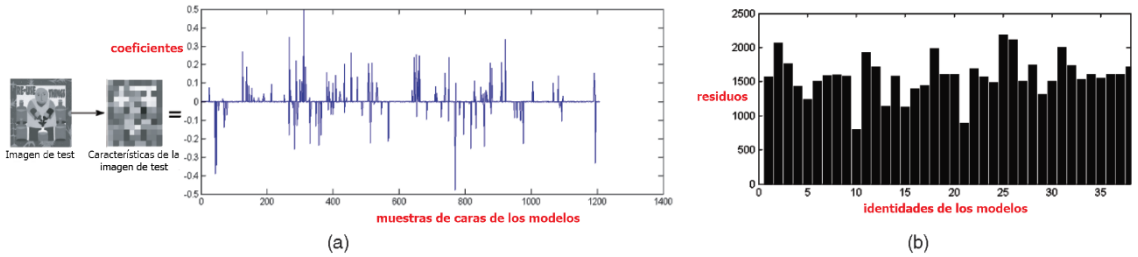

Figura 6: Coeficientes y residuos de una imagen de test inválida.
Los coeficientes y los residuos no muestran correspondencia con las imágenes de los modelos [10].

En la figura anterior se puede observar como en (a) los coeficientes no nulos no están relacionados con muestras de caras de modelos de la base de datos pertenecientes a la misma identidad que la de test. En cuanto a los residuos se puede observar como en (b) no hay ningún residuo claramente mínimo, de hecho, el ratio de los dos residuos menores es de 1:1.2. Así pues, la imagen de test de la figura 6 se considera inválida y en realidad no se deberían de llegar a calcular los residuos de dicha imagen, pero la gráfica sirve como dato para la comprensión.

Cabe comentar que se computa una combinación lineal para cada nueva imagen de test, no teniéndose en cuenta características que puedan servir para identificar otras imágenes de test. Así pues, para cada nueva imagen a reconocer, se buscan las imágenes de la base de datos que más se adaptan a ésta.

En cuanto a la extracción de características faciales, si la dispersión se consigue adecuadamente, la elección de características no es un problema crítico, sino que lo importante es que el número de características usadas sea elevado. Actualmente, existen muchas técnicas de extracción de características, cada una con sus pros y sus contras, de forma que potencian las ventajas para tratar de obtener buenas tasas de reconocimiento siempre con unos determinados requisitos. La elección de la técnica que se quiera utilizar dependerá de estos requisitos, y es difícil encontrar alguna que sirva para cualquier caso. Como comentábamos, en este caso la elección de características no es un asunto crítico.

2.2.3. Evaluación

2.2.3.1. Reconocimiento en imágenes

Para este apartado de evaluación con imágenes necesitaremos conocer unos determinados conceptos básicos en cuanto al sistema de validación se refiere, los cuales se comentan en [29]:

- *True Positive* (TP): son todas aquellas imágenes validadas por el sistema cuyas identidades realmente sí existen en la base de datos. Por tanto, son aciertos del sistema de validación. Se puede denominar como «acierto».
- *False Positive* (FP): son todas aquellas imágenes validadas por el sistema cuyas identidades realmente no existen en la base de datos. Por tanto, son fallos del sistema de validación. Se puede denominar como «falsa alarma».
- *False Negative* (FN): son todas aquellas imágenes invalidadas por el sistema cuyas identidades realmente sí existen en la base de datos. Por tanto, son fallos del sistema de validación. Se puede denominar como «fallo».
- *True Negative* (TN): son todas aquellas imágenes invalidadas por el sistema cuyas identidades realmente no existen en la base de datos. Por tanto, son aciertos del sistema de validación. Se puede denominar como «rechazo correcto».

Una vez conocidos estos datos ahora se citarán las fórmulas que se necesitan para calcular unas determinadas tasas de validación y de reconocimiento con tal de poder evaluar el sistema de validación implementado. Estos nuevos datos también se explican en [29].

Para las tasas de validación evaluamos el sistema mediante los indicadores de *accuracy* y F_1 *score*:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} = \frac{\sum img.\,validadas/invalidadas\ correctamente}{\sum img.\,totales}, \quad (9)$$

$$F_1\ score = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \quad (10)$$

$$precision = \frac{TP}{TP + FP} = \frac{\sum img.\,validadas\ correctamente}{\sum img.\,válidas}, \quad (11)$$

$$recall = \frac{TP}{TP + FN} = \frac{\sum img.\,validadas\ correctamente}{\sum img.\,cuyas\ identidades\ existen\ en\ la\ base\ de\ datos}. \quad (12)$$

Para las tasas de reconocimiento evaluamos el sistema mediante el indicador de *Recognition Rate* (RR):

$$RR = \frac{\sum img.\,válidas\ con\ reconocimientos\ correctos}{TP + FP} = \\ = \frac{\sum img.\,válidas\ con\ reconocimientos\ correctos}{\sum img.\,válidas}. \quad (13)$$

Cuando hablamos de imágenes validadas o invalidadas nos referimos a que el sistema de validación de imágenes ha determinado que las identidades de dichas imágenes son válidas o inválidas, respectivamente.

Para evaluar el sistema de validación implementado también haremos uso de unas gráficas. En concreto, basaremos nuestro análisis en la curva *Receiver Operating Characteristic* (ROC) [30] y la curva *Precision-Recall* (PR) [31].

Como se comenta en [30], la curva ROC es una gráfica que ilustra el rendimiento de un sistema clasificador binario en función de la variación de su umbral de discriminación. En nuestro caso, el sistema clasificador binario es el sistema de validación, puede adoptar los valores de «identidad válida» e «identidad inválida», y el umbral de discriminación es el valor de τ. Así pues, haciendo un barrido de la variable correspondiente al umbral τ, se obtendrá un resultado u otro. La curva se crea trazando el *True Positive Rate* (TPR) frente el *False Positive Rate* (FPR) para cada uno de los valores que toma en cuenta el umbral τ. A continuación, se detallan ambos conceptos:

$$TPR = \frac{TP}{TP + FN} = \frac{\sum img.\,validadas\ correctamente}{\sum img.\,cuyas\ identidades\ existen\ en\ la\ base\ de\ datos}, \quad (14)$$

$$FPR = \frac{FP}{FP + TN} = \frac{\sum img.\,validadas\ incorrectamente}{\sum img.\,cuyas\ identidades\ no\ existen\ en\ la\ base\ de\ datos}. \quad (15)$$

Varios ejemplos de curvas ROC se muestran en la siguiente figura:

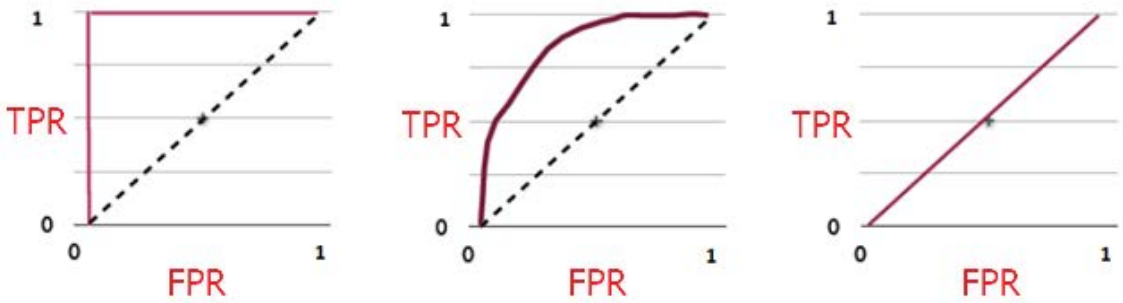

Figura 7: Ejemplos de curvas ROC.
Cuanto mayor es el valor de AUC, mejor rendimiento tiene el sistema clasificador binario [32].

Cuanto mayor es el valor del *Area Under Curve* (AUC), es decir, del área bajo la curva, mejor rendimiento se obtiene del sistema clasificador binario. El caso ideal se produce cuando $AUC =$

1, como se muestra en la curva ROC de la izquierda, mientras que un caso sin valor diagnóstico se produce cuando $AUC = 0{,}5$, como se apreciar en la curva ROC de la derecha. La curva ROC del medio se puede considerar que tiene un valor diagnóstico notable. Este índice se puede interpretar como la probabilidad de que el clasificador ordene o puntúe una instancia positiva elegida aleatoriamente más alta que una negativa. El análisis de la curva ROC se relaciona de forma directa con el análisis de coste/beneficio en toma de decisiones diagnósticas.

Como se comenta en [31], la curva PR es otra gráfica que ilustra el rendimiento de un sistema clasificador binario en función de la variación de su umbral de discriminación. Haciendo un barrido de la variable correspondiente al umbral τ, se obtendrá un resultado u otro. La curva se crea trazando el valor de *precision* frente al valor de *recall* para cada uno de los valores que toma en cuenta el umbral τ. Ambos conceptos ya han sido detallados anteriormente.

Varios ejemplos de curvas PR se muestran en la siguiente figura:

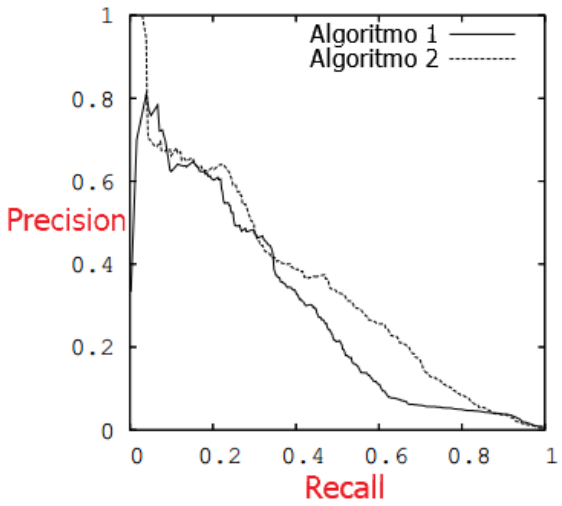

Figura 8: Ejemplos de curvas PR.
Los valores de *precision* y *recall* están inversamente relacionados [31].

Típicamente, los valores de *precision* y *recall* están inversamente relacionados. A medida que *precision* aumenta, *recall* disminuye, y viceversa. Se debe de lograr un equilibrio entre estos dos conceptos y es gracias a la utilidad de la curva PR que es posible comparar el rendimiento entre algoritmos diferentes. En esta gráfica también podemos considerar el AUC, al igual que con la gráfica de la curva ROC.

2.2.3.2. Reconocimiento en vídeos

Para este apartado de evaluación con vídeos continuaremos con la misma idea de conceptos básicos que el apartado anterior, cuya información se encuentra en [29]:

- TP: son todos aquellos *tracks* validados por el sistema cuyas identidades realmente sí existen en la base de datos y, además, la identidad de *ground truth* coincide con la identidad de la hipótesis de reconocimiento.
- FP: son todos aquellos *tracks* validados por el sistema cuyas identidades realmente sí existen en la base de datos, aunque la identidad de *ground truth* no coincide con la

identidad de la hipótesis de reconocimiento. También son todos aquellos *tracks* validados por el sistema cuyas identidades realmente no existen en la base de datos.

- FN: son todos aquellos *tracks* invalidados por el sistema cuyas identidades realmente sí existen en la base de datos.
- TN: son todos aquellos *tracks* invalidados por el sistema cuyas identidades realmente no existen en la base de datos.

Para este apartado se obtendrán los valores de *accuracy*, F_1 *score*, *precision*, *recall*, y MAP. Cada una de las cuatro primeras tasas se calculará para cada vídeo, cuyas fórmulas también se explican en [29] y son:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} = \frac{\sum tracks\ validados/invalidados\ correctamente}{\sum tracks\ totales}, \quad (16)$$

$$F_1\ score = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \quad (17)$$

$$precision = \frac{TP}{TP + FP} = \frac{\sum tracks\ validados\ correctamente}{\sum tracks\ válidos}, \quad (18)$$

$$recall = \frac{TP}{TP + FN} = \frac{\sum tracks\ validados\ correctamente}{\sum tracks\ cuyas\ identidades\ existen\ en\ la\ base\ de\ datos}. \quad (19)$$

Cuando hablamos de *tracks* validados o invalidados nos referimos a que el sistema de validación de *tracks* ha determinado que las identidades de dichos *tracks* son válidos o inválidos, respectivamente.

En cuanto al valor del MAP, ésta es una tasa que se calcula en global para todos los vídeos, por lo que sólo se calcula una vez. En el contexto de anotación de personas en secuencias de vídeo, el objetivo es determinar las apariencias de cada persona en concreto en una secuencia de vídeo. Más específicamente, dado un conjunto de *tracks*, el objetivo es devolver los *tracks* que corresponden a una persona en concreto. Cada persona hace referencia a una *query*. Para cada *query* se calcula el valor de *Average Precision* (AP) [33]. Finalmente, el valor de MAP será el promedio de valores de AP de todas las *queries*.

CAPÍTULO 3. MÉTODO IMPLEMENTADO

El método implementado en este proyecto se puede dividir en dos partes generales: una primera parte de reconocimiento de caras en imágenes estáticas y una segunda parte de reconocimiento de caras en imágenes en movimiento, es decir, vídeos.

En la parte de reconocimiento en imágenes se testea mediante varias bases de datos el correcto funcionamiento de la técnica de reconocimiento facial implementada: *Sparse Representation*.

En la parte de reconocimiento en vídeos se testea mediante varios vídeos correspondientes a programas de televisión el correcto funcionamiento del proceso de la anotación automática de dichos vídeos. En esta parte se hace uso también de la funcionalidad de la parte anterior, ya que para realizar el proceso de anotación también es necesario llevar a cabo un proceso de reconocimiento facial cada vez que queramos conocer la identidad de una nueva persona desconocida. Para solventar esta segunda parte del proyecto serán necesarias unas técnicas de *video tracking* y de reconocimiento facial no supervisado. El proceso de *video tracking* ya ha sido solventado por los organizadores de MediaEval, por lo que en este proyecto nos centramos en la no supervisión del reconocimiento facial.

3.1. Entorno y marco de trabajo

La elaboración de este PFG se ha realizado bajo la plataforma de desarrollo del GPI. Como se comenta en [34] el 99% de la investigación y desarrollo que se hace diariamente en el grupo se realiza en la propia nube de servidores que tiene a su disposición. En estos servidores se dispone de gran cantidad de recursos como CPUs, GPUs, RAM, y TBs de disco duro.

En octubre de 2014 Oriol Jaumà Lara publicó su PFG bajo el título «*Tècniques de reconeixement facial*» [12]. En él se centraba también en la técnica de reconocimiento facial *Sparse Representation*. El mismo autor del proyecto, Oriol, propuso en su memoria que para investigaciones futuras se podría seguir profundizando a partir de esta técnica. Y así se ha hecho, ya que se ha reutilizado su código en lenguaje MATLAB. Dicha implementación se ha adaptado a las necesidades de este proyecto. La diferencia principal es que su proyecto iba orientado al reconocimiento de imágenes, mientras que este proyecto va enfocado a la anotación automática de vídeos —sin olvidarnos del reconocimiento de imágenes—. No solamente se ha reutilizado la técnica de reconocimiento principal, sino que también se ha aprovechado parte de su código perteneciente a los sistemas de extracción de características.

En cuanto a *video tracking* también se ha reutilizado código en lenguaje MATLAB referente a la extracción de *tracks* de un vídeo cualquiera de entrada. Esta implementación fue solventada por alumnos del GPI y también se ha adaptado a las necesidades de este proyecto.

Por último, también se ha utilizado un *script* implementado en lenguaje Python para calcular el valor de un dato de evaluación para la parte de reconocimiento en vídeos. Este dato hace

referencia a *Mean Average Precision* (MAP) [33], el cual se explicará en apartados posteriores. El *script* es facilitado por terceras personas.

3.2. Reconocimiento en imágenes

En la parte de reconocimiento en imágenes el sistema ha de ser capaz de computar el reconocimiento facial de un conjunto de imágenes de test mediante la clasificación basada en *Sparse Representation*. Para ello serán necesarios también unos conjuntos de imágenes de modelos y de entrenamiento.

Este método primero realiza un proceso de validación para rechazar identidades inválidas y después un proceso de reconocimiento para reconocer cada una de las identidades de aquellas imágenes que han sido determinadas como identidades válidas. Las identidades inválidas son todas aquellas personas a las que no se les ha asignado ninguna identidad, puesto que, *a priori*, no pertenecen a ninguna de las identidades de la base de datos.

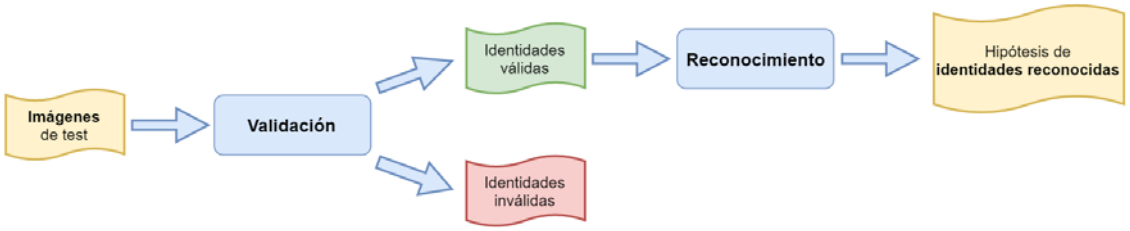

Figura 9: Clasificador basado en *Sparse Representation*.
A partir de unas imágenes de test el sistema rechaza identidades inválidas y sólo reconoce las identidades válidas.

Para llevar a cabo el reconocimiento facial de las imágenes de test se deberán seguir los siguientes pasos: definición de los conjuntos de imágenes, extracción de características de todas las imágenes, clasificación de las imágenes de test, y evaluación del sistema.

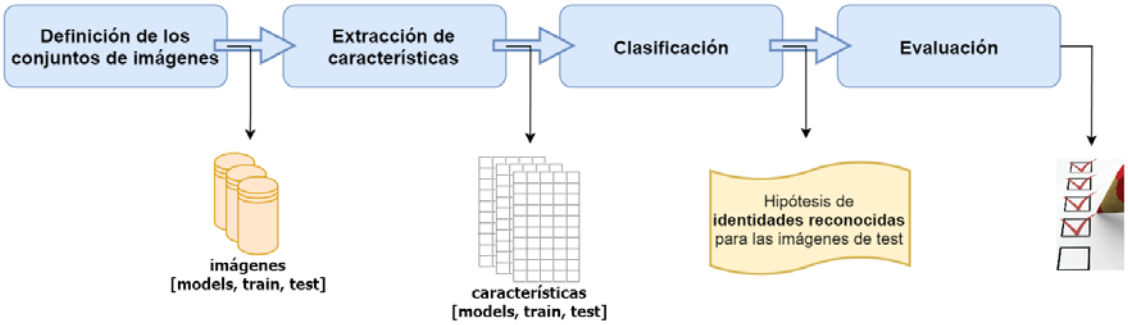

Figura 10: Implementación del reconocimiento facial de unas imágenes de test.
Estos son los pasos a seguir para computar el reconocimiento facial de cada nueva base de datos.

A nivel general, también podemos estructurar el proceso en dos fases: una primera fase de entrenamiento y una segunda fase de test. En la primera fase se calcula un valor óptimo de umbral que discrimine o diferencie las identidades válidas de las inválidas. Esta frontera será un valor numérico comprendido entre 0 y 1 y nos permitirá rechazar identidades inválidas en la segunda fase, de tal modo que sólo se reconocerán las identidades de aquellas imágenes que hayan sido validadas por el sistema.

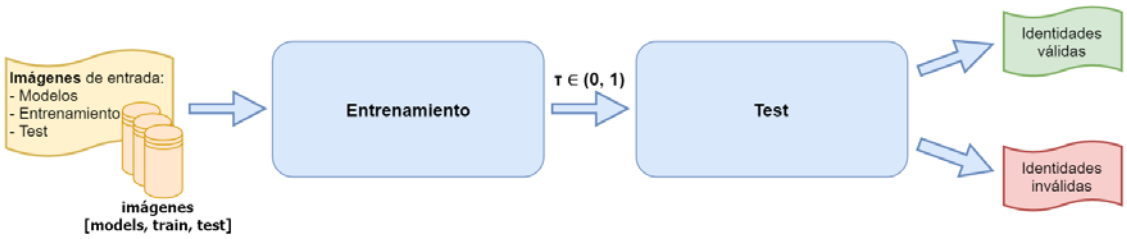

Figura 11: Fases generales del reconocimiento facial de unas imágenes de test.
En la primera fase de entrenamiento se obtiene un valor de umbral a ser utilizado por la segunda fase de test.

3.2.1. Definición de los conjuntos de imágenes

Para llevar a cabo la correcta computación del reconocimiento tres **conjuntos de imágenes** han de ser facilitados:

- Conjunto de **imágenes de base de datos**: conjunto de imágenes que contienen las **identidades válidas** del sistema. Se trata de los **modelos** del sistema, los cuales han de servir como referencia para determinar si una imagen de test contiene una identidad válida o no.
- Conjunto de **imágenes de entrenamiento**: conjunto de imágenes necesarias para entrenar el sistema y poder así llevar a cabo la validación de las imágenes de test. Estas imágenes de entrenamiento contienen tanto **identidades válidas** como **identidades inválidas**. Dichas imágenes nos servirán únicamente para determinar un **valor óptimo de umbral** que sirva como frontera para diferenciar entre identidades válidas e identidades inválidas de las imágenes de test.
- Conjunto de **imágenes de test**: conjunto de **imágenes a ser testeadas** por el sistema, es decir, aquellas imágenes de las que queremos conocer la identidad de las personas contenidas en ellas. Estas imágenes de test contienen tanto **identidades válidas** como **identidades inválidas**. En el caso de que tras realizar el reconocimiento facial resulten como identidades válidas se asignarán las identidades que se hayan reconocido, mientras que si resultan como identidades inválidas se asignarán las identidades propiamente como inválidas. En cualquiera de los casos el reconocimiento facial computado puede haber sido fallido.

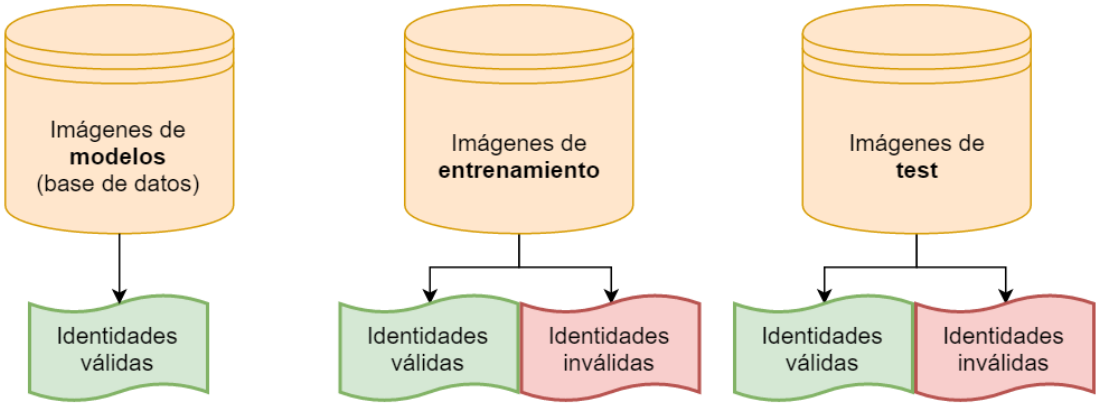

Figura 12: Definición de los conjuntos de imágenes.
Según el conjunto de imágenes éste podrá contener imágenes inválidas o no.

En esta implementación todas las imágenes deben de ser recortadas de forma que **cada cara ocupe toda la imagen entera**. Posteriormente, mediante implementación todas las imágenes **se normalizarán a unas dimensiones de anchura y altura determinadas**, por lo que al final todas las imágenes contendrán el mismo número de píxeles. Recordemos que estas imágenes se obtienen ya directamente de bases de datos estándar, por lo que en este proyecto nos olvidamos absolutamente de métodos de detección y extracción de caras.

Figura 13: Ejemplo de cara recortada.
La cara debe de ocupar toda la imagen, aunque no es necesario estar normalizada a unas dimensiones estándar.

En el método implementado, para indicar el comportamiento de cada una de las imágenes del sistema se utilizan unos **ficheros de texto**, los cuales contienen las rutas absolutas de cada una de estas imágenes. Gracias a estas rutas podremos realizar el análisis de las imágenes adecuadamente, tratándolas como **imágenes de modelos, de entrenamiento, o de test**. Es a partir de estos ficheros de los cuales se parte y sirven en realidad como unos parámetros más del sistema. Por cada nueva base de datos que se utilice, deberán de existir estos ficheros.

3.2.2. Extracción de características

Una vez facilitados estos datos, el método empieza por realizar una extracción de las **características para cada una de las caras** de los conjuntos de imágenes. Estas características nos servirán para poder **comparar caras entre sí**. En la práctica, cada cara de cada imagen tendrá asociado un vector con un determinado tipo de características. Antes de procesar la técnica de extracción de características escogida, cada imagen deberá de ser leída y transformada a **niveles de gris** en caso de que se encuentre codificada en formato RGB (*Red*, *Green*, *Blue*) [35].

Para llevar a cabo este proceso se deben de configurar dos parámetros:

- Valor de **redimensionamiento**: es el número de píxeles, tanto en anchura como en altura, aplicado en todas las imágenes con tal de redimensionarlas con iguales dimensiones.
- **Técnica de extracción de características**: puede ser de tres tipos:
  - Técnica holística: descriptor global.
  - Técnica *Random Pixels*: descriptor global.
  - Técnica LBP: descriptor local.

Al final, por cada conjunto de imágenes se dispondrá de una matriz con los vectores de características de cada cara. Así pues, tendremos tres matrices con las características de todas las identidades —tanto válidas como inválidas—.

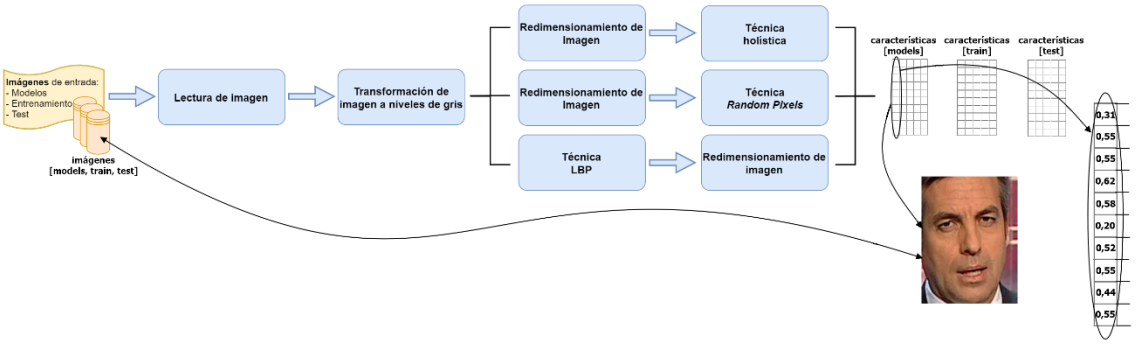

Figura 14: Extracción de características.
Por cada conjunto de imágenes se obtiene una matriz con las características de cada imagen.

3.2.3. Clasificación

Una vez llevada a cabo la extracción de características, el método sigue por realizar una clasificación basada en *Sparse Representation* tanto para el conjunto de imágenes de entrenamiento como de test. La clasificación del conjunto de imágenes de entrenamiento servirá principalmente para entrenar el sistema y calcular un valor óptimo de umbral a ser utilizado por la clasificación del conjunto de imágenes de test. Las imágenes de entrenamiento también nos servirán para simular como si se tratasen de imágenes de test. En esta simulación al final se obtendrán las identidades reconocidas como con el conjunto de test, pero éstas no se almacenarán, sino que únicamente servirán para calcular los valores de evaluación para este conjunto y valorar así la fiabilidad del método implementado.

Cada una de ambas clasificaciones se puede dividir en los siguientes pasos: cálculo de coeficientes, validación, cálculo de residuos, y reconocimiento.

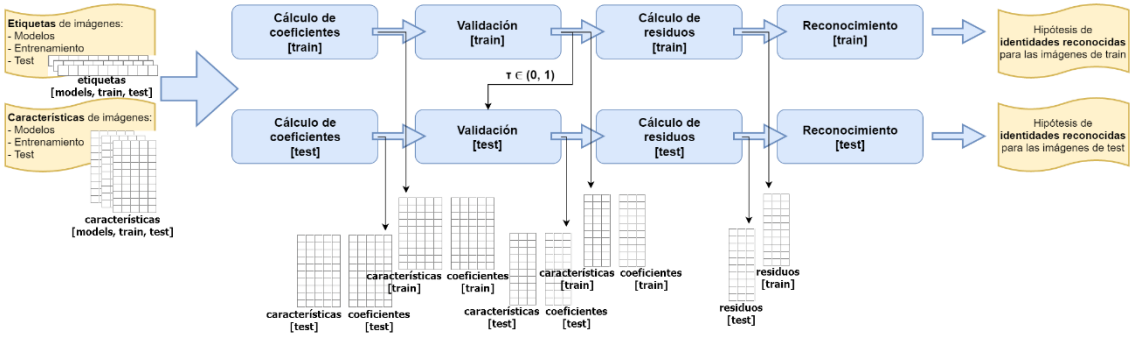

Figura 15: Clasificación.
Estos son los pasos a seguir para computar la clasificación de los conjuntos de entrenamiento y test.

3.2.3.1. Cálculo de coeficientes

El primer paso de la clasificación es el cálculo de coeficientes. Para calcular los coeficientes *sparse* del conjunto de imágenes en cuestión (entrenamiento o test) el valor de la matriz α debe de ser calculado en el sistema de ecuaciones lineales

$$y = A \cdot \alpha, \qquad (20)$$

donde la matriz A se corresponde con todas las imágenes de los modelos de la base de datos expresadas con vectores de características, y la matriz y se corresponde con todas las imágenes del conjunto de imágenes en cuestión (entrenamiento o test), también expresadas con vectores de características.

Para calcular los coeficientes *sparse* se debe de configurar un parámetro:

- Método de cálculo de coeficientes: puede ser de dos tipos:
  - Método «*Primal-Dual Logarithmic Barrier*» [36]
  - Método «*Accelerated Proximal Gradient*» [37]

Al final, por cada conjunto de imágenes, tanto de entrenamiento como de test, se dispondrá de una matriz con los vectores de coeficientes de cada cara. Así pues, tendremos dos matrices con los coeficientes *sparse* de todas las identidades —tanto válidas como inválidas—.

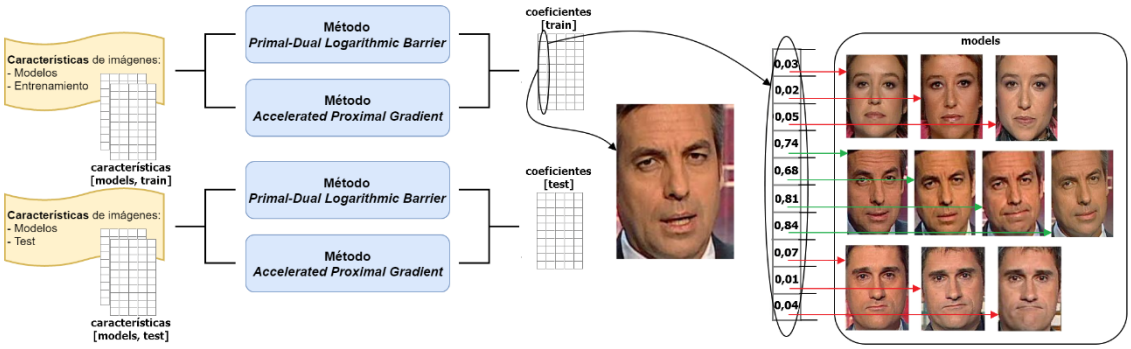

Figura 16: Cálculo de coeficientes.
Por cada conjunto de imágenes se obtiene una matriz con los coeficientes de cada imagen.

3.2.3.2. Validación

El segundo paso de la clasificación es la validación. Consiste en determinar si la identidad de la persona es considerada válida (se halla en la base de datos) o no. El conjunto de entrenamiento sirve para entrenar de alguna forma el sistema mediante el cálculo de un parámetro. Dicho parámetro consiste en un valor óptimo de umbral τ que es obtenido y pasado como parámetro de entrada en la validación del conjunto de test, así como también sirve para testear la validación del conjunto de entrenamiento. Este umbral nos servirá para establecer una frontera entre identidades válidas e identidades inválidas y se obtiene haciendo un barrido de la variable utilizada como umbral, es decir, para cada incremento de esta variable se simula el comportamiento del sistema para poder así determinar el valor óptimo. Cuando hablamos de comportamiento del sistema nos referimos a qué valor de $\mathbf{F_1}$ *score* se obtiene en cada caso, de modo que cuanto mayor sea este valor mejor será el valor del umbral seleccionado. Como ya se ha comentado, el valor óptimo de umbral también es utilizado por el conjunto de entrenamiento para valorar la fiabilidad del método, de modo que se simula como si de imágenes de test se tratase.

Para determinar si una imagen de entrenamiento o de test pertenece a una identidad válida o no debemos de **comparar con el valor óptimo de umbral** calculado. Esta comparación se realiza con el **valor de SCI**, el cual es calculado para cada una de las imágenes. Este valor se obtiene mediante la fórmula

$$SCI(\alpha') = \frac{k \cdot \frac{max_i \|\delta_i(\alpha')\|_1}{\|\alpha'\|_1} - 1}{k - 1}, \qquad (21)$$

donde el vector α' se corresponde con los coeficientes *sparse* calculados de una imagen en cuestión (entrenamiento o test), y k se corresponde con el número de modelos existentes en la base de datos. δ_i es la función característica que selecciona los coeficientes del vector α' asociados con el modelo i.

Si el **valor de SCI** obtenido de una imagen es **menor** al **valor óptimo de umbral** que se ha obtenido al entrenar el sistema, entonces la identidad que pertenece a esa imagen quedará **invalidada** por el sistema, de modo que se catalogará como identidad inválida. Por otro lado, si el **valor de SCI** es igual o **mayor** al **valor óptimo de umbral**, entonces la identidad que pertenece a esa imagen quedará **validada** por el sistema y en el posterior proceso de reconocimiento **se determinará la correspondiente identidad**:

$$SCI(\alpha') \geq \tau. \qquad (22)$$

El proceso de validación puede haber resultado fallido o no, de modo que una imagen catalogada como identidad válida en realidad puede no serlo, y una imagen clasificada como identidad inválida en realidad también puede no serlo.

Es en este punto de la implementación en el cual se calcula el número de **TP**, **FP**, **FN**, y **TN**, tanto para el conjunto de entrenamiento como de test, de modo que posteriormente se permita el cálculo de los valores de evaluación que permitan determinar la fiabilidad y perfección del método implementado.

Al final, por cada conjunto de imágenes, tanto de entrenamiento como de test, se dispondrá de las matrices de características y de coeficientes únicamente de aquellas imágenes que hayan sido validadas por el sistema. Los datos de las identidades inválidas habrán sido rechazados, por lo que posteriormente clasificaremos a éstas directamente como identidades inválidas.

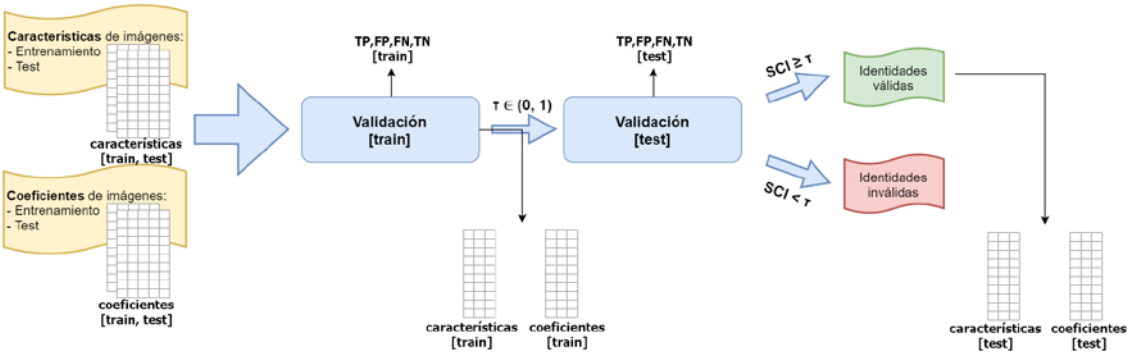

Figura 17: Validación.

En el proceso de validación se compara el valor SCI de cada imagen con el valor del umbral óptimo obtenido.

3.2.3.3. Cálculo de residuos

El tercer paso de la clasificación es el cálculo de **residuos**. Para calcular los residuos *sparse* del conjunto de imágenes en cuestión (entrenamiento o test) el valor de la matriz r debe de ser calculado en el sistema de ecuaciones lineales

$$r_i(y) = \|y - A \cdot \delta_i(\hat{\alpha}_1)\|_2, \qquad (23)$$

donde la matriz A se corresponde con todas las imágenes de los modelos de la base de datos expresadas con vectores de características, la matriz y se corresponde con todas las imágenes del conjunto de imágenes en cuestión (entrenamiento o test), también expresadas con vectores de características, y la matriz α se corresponde con los coeficientes *sparse* calculados del conjunto de imágenes en cuestión (entrenamiento o test). δ_i es la función característica que selecciona los coeficientes para cada vector α' asociados con el modelo i. Y recordemos que el vector α' se corresponde con los coeficientes *sparse* calculados de una imagen en cuestión (entrenamiento o test).

Así pues, **por cada imagen de test tendremos asociado un vector de residuos con un valor de residuo por cada modelo existente de la base de datos**. Cuanto menor sea este valor asociado entre imagen de test y modelo, mayor similitud representará dicha imagen de test con la identidad de dicho modelo.

La matriz r **se calcula únicamente para aquellas imágenes que han sido validadas** por el sistema. Por tanto, es lógico que las identidades inválidas no pasen el proceso de cálculo de residuos puesto que han quedado rechazadas por el sistema.

Al final, por cada conjunto de imágenes, tanto de entrenamiento como de test, se dispondrá de una matriz con los vectores de residuos de cada cara. Así pues, tendremos dos matrices con los residuos *sparse* de las identidades válidas.

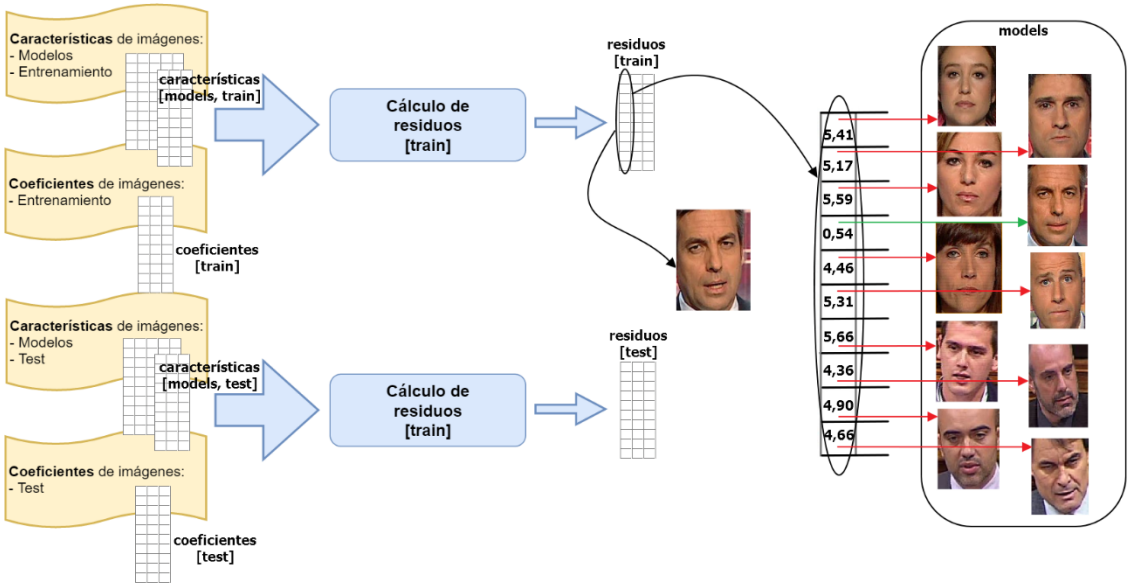

Figura 18: Cálculo de residuos.
Por cada conjunto de imágenes se obtiene una matriz con los residuos de cada imagen validada.

3.2.3.4. Reconocimiento

El cuarto paso de la clasificación consiste en asignar la **hipótesis de reconocimiento**. Se determina la identidad que pertenece a cada imagen del conjunto de imágenes en cuestión (entrenamiento o test). Cada una de estas identidades reconocidas tendrán asociadas un **mínimo valor de residuo *sparse***. Recordemos que una imagen tiene asociado un valor de residuo *sparse* por cada modelo de la base de datos. El valor mínimo de entre estos valores corresponderá con el modelo cuya identidad será la reconocida. **Para las imágenes que no se han validado previamente se les asignará la identidad como identidades inválidas.**

Para computar el reconocimiento seguiremos la fórmula

$$identidad(y) = arg\,min_i\ r_i(y), \qquad (24)$$

donde la matriz r se corresponde con los residuos *sparse* calculados del conjunto de imágenes en cuestión (entrenamiento o test).

Al final, por cada conjunto de imágenes, tanto de entrenamiento como de test, se dispondrá de las hipótesis de identidades reconocidas para cada una de las imágenes.

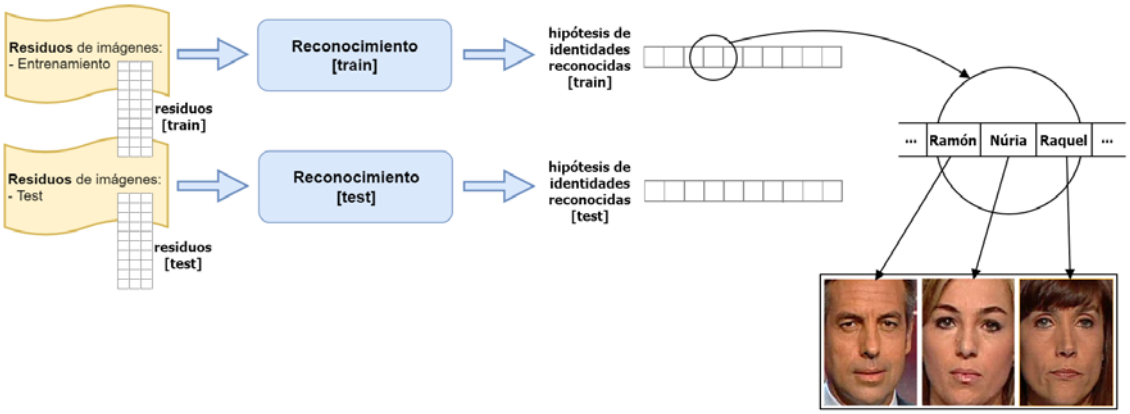

Figura 19: Reconocimiento.
En el proceso de reconocimiento se asignan las identidades con menores valores de residuos.

3.2.4. Evaluación

Una vez llevada a cabo la clasificación, el método termina por realizar una **evaluación** de los resultados obtenidos:

- Se calculan las **tasas de validación y reconocimiento tanto del conjunto de imágenes de entrenamiento como de test**. Para las tasas de validación se obtienen los valores de *accuracy* y F_1 *score*, donde este último se obtiene a partir de los valores calculados de *precision* y *recall*. Para la tasa de reconocimiento se obtiene el valor de **RR**.
- Los **resultados de reconocimiento del conjunto de imágenes de test** se guardan en un fichero de texto. Para ello, por cada imagen a testear (identidad de *ground truth*) se anotará la identidad reconocida seguida del veredicto de reconocimiento. Dicho veredicto podrá ser exitoso, en cuyo caso se asignará como resultado un valor lógico «1», o fallido, en cuyo caso se asignará como resultado un valor lógico «0».

- Las gráficas creadas en el proceso de validación del **conjunto de imágenes de entrenamiento** se guardan en un fichero de figura. Dichas gráficas hacen referencia a la **curva ROC** y la **curva PR** y se representan dentro de una misma figura.
- Se calcula el **tiempo de ejecución del método** completo.

Para este apartado será necesario explicar que previo a todo el proceso de reconocimiento facial se ha empezado por crear el *ground truth* para cada uno de los conjuntos de imágenes. Esto consiste en **asignar etiquetas asociadas a cada una de las imágenes**. Estas etiquetas contienen el identificador de las identidades a las que hacen referencia. Dicho identificador está formado únicamente por el nombre de la persona a la que se hace referencia. Las etiquetas nos sirven como referencia para **determinar la validez de las identificaciones** en cada caso, y es por eso que se explica este paso en este apartado, ya que el *ground truth* se usa para la evaluación.

Al final, por cada conjunto de imágenes se dispone de un vector con las etiquetas verdaderas de cada cara. Así pues, tendremos tres vectores con las etiquetas de todas las identidades —tanto válidas como inválidas—.

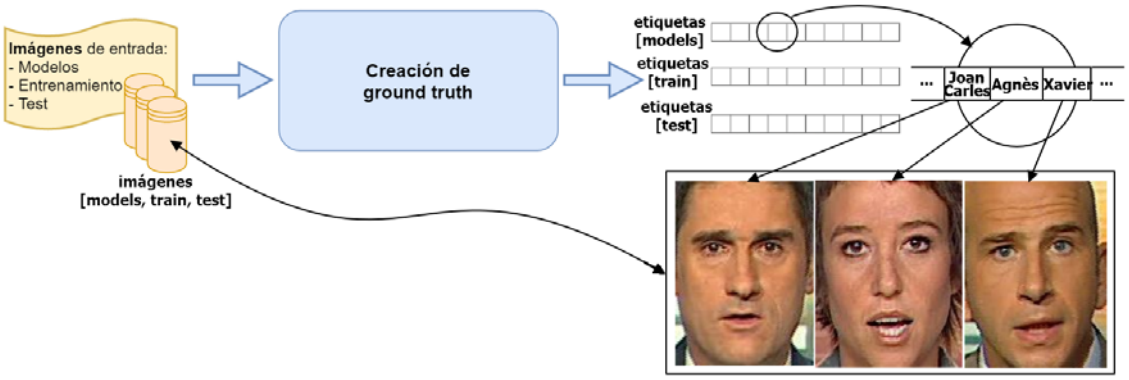

Figura 20: Creación de *ground truth*.
Por cada conjunto de imágenes se obtiene un vector con las etiquetas verdaderas de las identidades.

3.3. Reconocimiento en vídeos

En la parte de reconocimiento en videos el sistema ha de ser capaz de realizar todo un proceso de **anotación automática** de una serie de **vídeos**. Como ya se ha comentado anteriormente, en esta parte haremos uso de la parte anterior, por lo que se utiliza para esta implementación la técnica de reconocimiento facial *Sparse Representation*.

Este método crea para cada vídeo analizado sus **modelos** correspondientes a la base de datos de dicho vídeo. Al final, **cada modelo estará formado por un conjunto de *tracks***, los cuales se habrán ido extrayendo del vídeo a medida que éste se haya ido analizando. Para determinar en cada caso la identidad del modelo al que pertenecen los *frames* de cada *track*, un **reconocimiento facial** es necesario. Así pues, **los modelos se van creando secuencialmente** a medida que se va avanzando en el procesamiento del vídeo, así como también se pueden actualizar en cada paso al reconocerse la identidad del modelo para un determinado *track*.

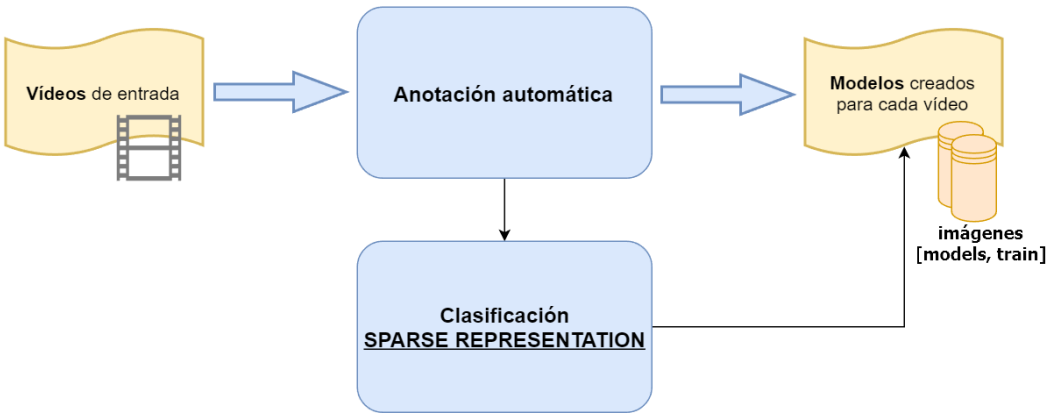

Figura 21: Sistema de anotación automática.
En este sistema se hace uso de la clasificación basada en *Sparse Representation* para realizar el reconocimiento facial.

Recordemos que las imágenes en este caso se obtienen de programas de televisión mediante técnicas de *video tracking* ya implementadas por otras personas, por lo que en este proyecto nos olvidamos absolutamente de métodos de detección y extracción de caras.

Para cada vídeo, el proceso a seguir es el siguiente: definición de los **datos**, **lectura de los datos**, **asociación entre tracks y nombres**, creación de **modelos**, y **evaluación** del sistema.

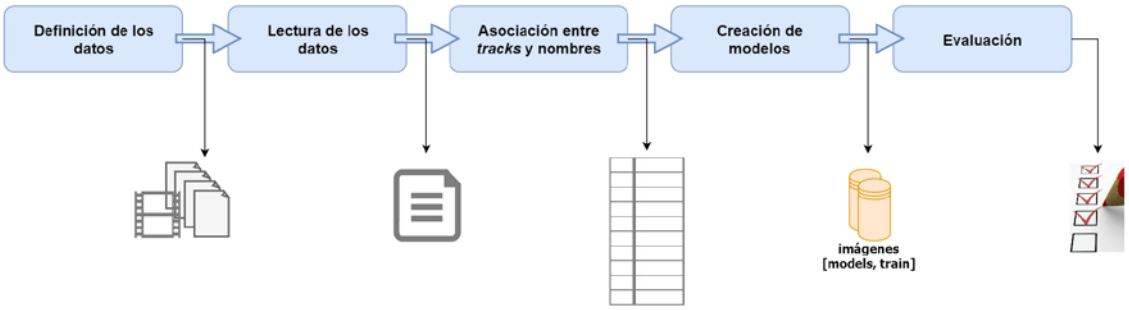

Figura 22: Implementación de la anotación automática de un vídeo.
Estos son los pasos a seguir para computar el proceso de anotación automática de cada nuevo vídeo.

3.3.1. Definición de los datos

Para llevar a cabo la correcta computación de la anotación se deben de definir primero los **datos** a partir de cinco ficheros:

- Fichero de **vídeo** (extensión «mp4»): vídeo del cual crearemos los modelos extrayendo los *tracks* uno por uno.
- Fichero de ***video tracking*** (extensión «seg»): fichero de texto que contiene los datos de *video tracking*. Entre estos datos encontramos los **delimitadores de *frames* de cada *track*** del vídeo correspondiente.
- Fichero de ***text tracking*** (extensión «MESeg»): fichero de texto que contiene los datos de *text tracking*. Entre estos datos encontramos los **nombres de las identidades** que van apareciendo a lo largo del vídeo, así como también los **delimitadores de *frames* de dichos nombres**. Este fichero es global y contiene la información perteneciente a todos los vídeos, por lo que solamente existe un único fichero de *text tracking*.

- Fichero de *face tracking* (extensión «facetrack»): fichero de texto que contiene los datos de *face tracking*. Entre estos datos encontramos las coordenadas en píxeles para extraer las caras de cada uno de los *frames* de cada uno de los *tracks* del vídeo correspondiente.
- Fichero de *ground truth* (extensión «gt»): fichero de texto que contiene los datos de *ground truth* para cada uno de los *tracks* del vídeo correspondiente, es decir, el nombre verdadero de la identidad que aparece a lo largo de cada *track*.

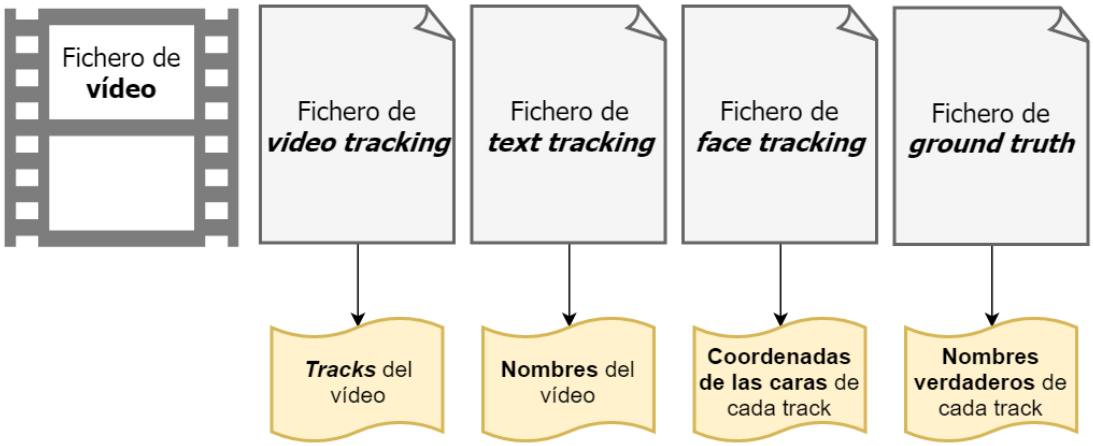

Figura 23: Definición de los datos.
Gracias a estos datos se podrá realizar la anotación automática del vídeo de entrada correspondiente.

3.3.2. Lectura de los datos

Una vez facilitados estos datos, el método empieza por realizar la lectura de éstos. Se deberá de importar toda la información perteneciente a los ficheros acabados de comentar: vídeo, *video tracking*, *text tracking*, *face tracking*, y *ground truth*.

3.3.3. Asociación entre *tracks* y nombres

Una vez llevada a cabo la lectura de los datos, el método sigue por realizar una asociación entre tracks y nombres de identidades. Para ello, se comprueba el solapamiento de *frames* entre *tracks* (a partir de los datos presentes en el fichero de *video tracking*) y nombres (a partir de los datos presentes en el fichero de *text tracking*).

A cada identificador de *track* (*track ID*) se le asigna el nombre de la identidad que tenga solapado.

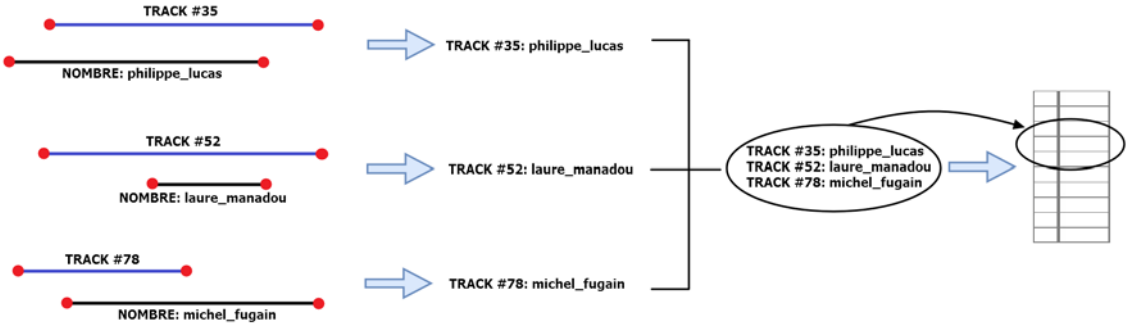

Figura 24: Ejemplos de solapamientos entre *tracks* y nombres.
Estos son algunos ejemplos de formas posibles de solapamiento entre los *tracks* y los nombres.

En caso de que un mismo **nombre esté solapado con más de un *track*** se añadirá un **sufijo numérico con el formato «_i» al final del nombre**, donde «i» es un número entero. Por ejemplo, imaginemos que el nombre «nicolas_sarkozy» tiene asociados los *tracks* 24, 25, y 26. Entonces estos *tracks* se asociarán con los nombres «nicolas_sarkozy_01», «nicolas_sarzoky_02», y «nicolas_sarkozy_03», respectivamente. Esto puede suceder en dos casos: cuando aparece un nombre y hay personas (*tracks* detectados) «de fondo» que no pertenecen a dicho nombre, y cuando el *track* que corresponde al nombre se detecta a trozos dividiéndose éste en varios *tracks*.

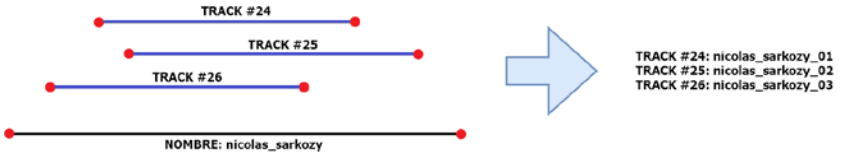

Figura 25: Ejemplo de solapamiento de un nombre con varios *tracks*.
Cuando un nombre se solapa con más de un *track*, entonces al nombre asociado a cada *track* se le añade un sufijo.

Para simplificar el problema, consideraremos que **un mismo *track* no puede estar solapado con más de un nombre**. Esto podría suceder en un caso hipotético: cuando en una misma escena hay dos personas (*tracks* detectados), primero habla una con el texto que le corresponde a ésta y después habla la otra con el texto que le corresponde a ésta otra. De este modo un mismo *track* tendría asociados dos nombres: uno que le pertenece y otro que no.

Al hacer estas asociaciones entre *tracks* y nombres se nos permitirá que **cada *track* se extraiga como un nuevo modelo en un directorio único** con el nombre de la identidad al que hace referencia.

3.3.4. Creación de modelos

Una vez llevada a cabo la asociación entre *tracks* y nombres, el método sigue por realizar la creación de los **modelos**. Para ello, se realizará una **extracción de *tracks***, uno por uno, realizando de esta forma una anotación automática del vídeo que se está procesando.

Al procesar los modelos tres casos pueden suceder:

- **Creación de modelo**: se crea un nuevo modelo.
- **Intento de actualización de modelo**: puede conllevar a la actualización de modelo o no.
- **No creación/intento de actualización de modelo**: ni se crea ni se actualiza ningún modelo.

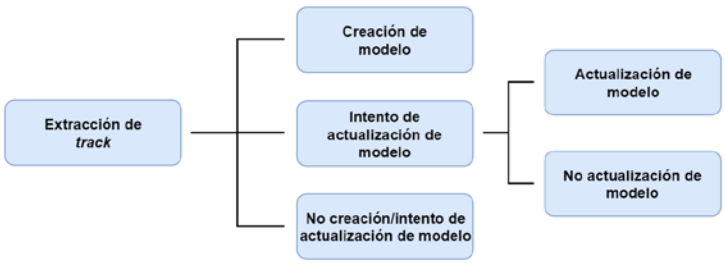

Figura 26: Proceso de anotación automática.
A la hora de crear los modelos extrayendo *track* por *track* pueden suceder tres casos posibles.

En este método se parte de la consideración consistente en que la identidad de una persona se considera como desconocida mientras no aparezca su nombre en el vídeo por vez primera. A partir de este momento, todos los *tracks* que pertenezcan a dicha identidad se considerarán como tal identidad.

Es en este punto de la implementación en el cual se calcula el número de TP, FP, FN, y TN, de modo que en la extracción de cada *track* estos valores son actualizados.

A continuación, se detallan los posibles casos acabados de citar que pueden suceder al procesar cada uno de los *tracks*.

3.3.4.1. Creación de modelo

Un nuevo modelo es creado si el *track* actual que se está procesando tiene asociado un nombre de identidad. Para crear un modelo se creará un nuevo directorio con el nombre de la identidad al que hace referencia dicho modelo. En dicho directorio se extraerán los *frames* que pertenecen al *track* que se está extrayendo, de donde dichos *frames* se extraerán las caras recortadas (a partir de los datos presentes en el fichero de *face tracking*).

Figura 27: Creación de modelo.
Un nuevo modelo es creado si el *track* que se procesa está asociado con un nombre.

3.3.4.2. Intento de actualización de modelo

El *track* actual que se está procesando no tiene asociado ningún nombre de identidad y se intenta actualizar algún modelo existente. Para intentar actualizar un modelo se creará un directorio temporal con el nombre «*unknown*», es decir, una identidad de momento desconocida. A efectos de reconocimiento facial se tratará como identidad de test. Al igual que en la creación de modelos en dicho directorio se extraerán las caras recortadas pertenecientes al *track* que se está extrayendo.

Posteriormente, se procederá al proceso de reconocimiento facial utilizando para ello la técnica *Sparse Representation* ya explicada en apartados anteriores. Es aquí donde entra en juego la primera parte del método implementado: reconocimiento en imágenes. La principal diferencia será que ahora no partimos de unos modelos preestablecidos, sino que la propia base de datos del vídeo se va creando a medida que éste se va anotando automáticamente, de modo que el reconocimiento facial es no supervisado.

En dicho reconocimiento se comparan los *frames* del *track* actual que se está procesando con los *frames* de los modelos ya existentes, es decir, los que se han creado hasta el momento. El reconocimiento puede resultar en que el *track* que se está reconociendo pertenece a alguna de las identidades ya existentes, o por el contrario que no se asemeje lo suficiente a ninguna de ellas, por lo que la identidad reconocida resultará desconocida.

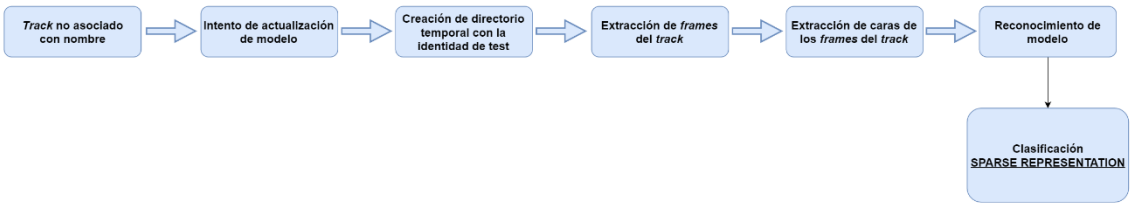

Figura 28: Intento de actualización de modelo.
Se intenta actualizar algún modelo existente cuando el *track* que se procesa no tiene asociado ningún nombre.

Para determinar si un *track* pertenece a alguna identidad se comprueba el número de *frames* válidos, es decir, aquellos que han pasado el proceso de validación exitosamente. En la implementación propuesta basta con que el 20% de *frames* sean válidos como para que el *track* también lo sea. Este umbral se ha escogido *ad-hoc*, es decir, para un conjunto de *tracks* se ha comprobado que este valor es el que aproximadamente mejor discrimina entre identidades válidas e identidades inválidas (identidades desconocidas, es decir, *unknowns*). En caso de que el *track* sea inválido, entonces se asignará la identidad como desconocida, por lo que no se procederá a actualizar ningún modelo existente. Considerando que el *track* haya pasado el proceso de validación entonces se asignará la identidad reconocida como aquella que más veces se repita de entre las identidades correspondientes a los *frames* que han sido validados, por lo que se procederá a actualizar el modelo reconocido.

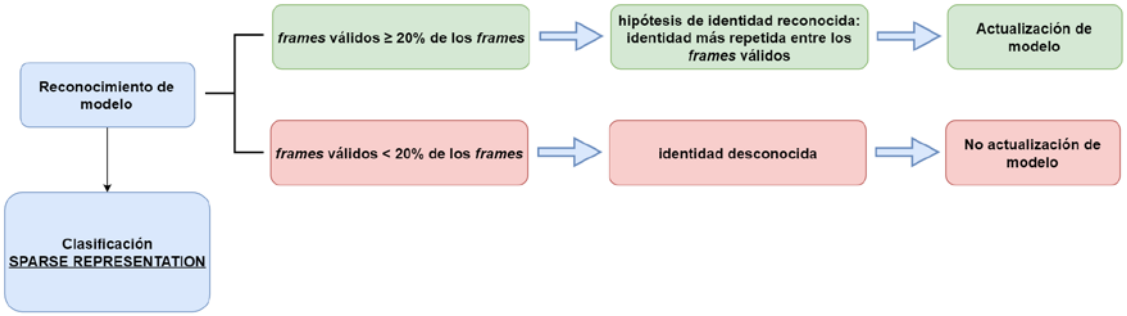

Figura 29: Decisión entre actualización o no actualización de modelo.
Según si el *track* que se procesa es válido o no, se procederá a actualizar un modelo o no.

Un aspecto a tener en cuenta en el reconocimiento facial es que el clasificador se ha de entrenar con algunas imágenes preestablecidas correspondientes a identidades inválidas, las cuales no aparecen en absoluto en el vídeo en cuestión. En este entrenamiento también se toman en cuenta imágenes de entrenamiento obtenidas de los modelos creados hasta el momento. Esto nos permitirá hallar el valor óptimo de umbral τ que discrimina en nuestro sistema entre identidades válidas e inválidas. Este valor se calcula en cada proceso de reconocimiento facial, por lo que al final es calculado tantas veces como intentos de actualización de modelos existentes son procesados.

Como ya se ha dejado entrever, después de llevar a cabo el proceso de reconocimiento facial, el intento de actualización puede conllevar a dos posibles casos:

- Actualización de modelo: un modelo existente es actualizado si el *track* actual que se está procesando pertenece a dicho modelo, ya que la identidad a la que hace referencia

es la misma. Esta identidad es la que se ha reconocido mediante el reconocimiento facial. Para actualizar un modelo se moverán las imágenes de test del *track* actual que se está procesando desde el directorio temporal con identidad previamente desconocida al directorio del modelo con la identidad reconocida.

- No actualización de modelo: ningún modelo existente es actualizado si el *track* actual que se está procesando no pertenece a ningún modelo, ya que la identidad a la que hace referencia es desconocida (*unknown*). Esta identidad desconocida es la que no se ha podido reconocer mediante el reconocimiento facial. Así pues, el directorio temporal contenedor de la identidad desconocida es eliminado, ya que al no pertenecer a ningún modelo no es de nuestro interés.

3.3.4.3. No creación/intento de actualización de modelo

Este caso se produce si el *track* actual que se está procesando no tiene asociado ningún nombre de identidad, ni tampoco hay modelos creados como para actualizar alguno de ellos. En este caso asignaremos directamente el resultado de reconocimiento de este *track* como identidad desconocida (*unknown*), ya que recordemos partimos de la premisa que una identidad es desconocida mientras no aparezca su correspondiente nombre por vez primera.

3.3.5. Evaluación

Una vez llevada a cabo la creación de los modelos, el método termina por realizar una evaluación de los resultados obtenidos:

- Se calculan las tasas de reconocimiento para cada uno de los vídeos analizados. En concreto, se obtienen los valores de *accuracy* y $\mathbf{F_1}$ *score*, donde este último se obtiene a partir de los valores calculados de *precision* y *recall*. Y también se calcula el valor de MAP, el cual se basa en todos los vídeos procesados.
- Los resultados de reconocimiento de los *tracks* del vídeo correspondiente se guardan en un fichero de texto. Para ello, por cada *track* a testear (identidad de *ground truth*) se anotará la identidad creada o reconocida seguida del veredicto de reconocimiento. Dicho veredicto podrá ser exitoso, en cuyo caso se asignará como resultado un valor lógico «1», o fallido, en cuyo caso se asignará como resultado un valor lógico «0». En caso que la identidad creada o reconocida acabe con el sufijo «_i», éste se suprimirá para poder comparar adecuadamente con el *ground truth*.
- Los resultados de reconocimiento de los *tracks* de todos los vídeos se guardan en otro fichero de texto. Para ello, por cada vídeo y por cada *track* a testear se anotará la identidad creada o reconocida seguida de una puntuación. Para las identidades reconocidas, esta puntuación es el producto de una puntuación del sistema de validación por una puntuación del sistema de reconocimiento. La puntuación del sistema de validación consiste en un promedio del valor de SCI por cada *track*. Recordemos que

el SCI se calcula para cada una de las imágenes de test, en este caso *frames*. La puntuación del sistema de reconocimiento consiste en el cociente del número de *frames* de la identidad que más veces se repite de entre los *frames* válidos dividido entre el número total de *frames* válidos. Para las identidades creadas como un nuevo modelo y para las identidades que se han reconocido como desconocidas (*unknowns*) el valor de la puntuación del *track* se fuerza a «1». Este fichero se edita de esta manera ya que nos servirá para poder evaluar luego la tasa de reconocimiento correspondiente al MAP. Si bien este fichero hace referencia a las hipótesis de reconocimiento, también serán necesarios para poder evaluar el MAP otros dos ficheros: uno que contenga todos los *tracks* de todos los vídeos, y otro que contenga el *ground truth* de todos los *tracks* de todos los vídeos. En concreto, como ya se ha comentado anteriormente, el MAP se calcula mediante un *script* escrito en lenguaje de programación Python, el cual es proporcionado por terceras personas, por lo que no se ha implementado para este método específico.

- Se calcula el tiempo de ejecución del procesamiento de cada vídeo, así como del método completo. Además, también se calcula el tiempo de ejecución cada vez que se realiza un intento de actualización de modelo, es decir, cada vez que hay una identidad de test a reconocer.

CAPÍTULO 4. EXPERIMENTOS Y RESULTADOS

Se han realizado pruebas experimentales primero con **imágenes** estáticas y, posteriormente, con **vídeos**, es decir, con imágenes en movimiento. Al final, para imágenes se han realizado experimentos con **una única base de datos**, mientras que para vídeos las pruebas se han llevado a cabo con **cuatro vídeos** correspondientes a programas de televisión. En concreto, la base de datos de imágenes es una mezcla entre la base de datos **Televisión de Cataluña** (TVC) y la base de datos *Japanese Female Facial Expression* (JAFFE) [38], mientras que los vídeos hacen referencia a **telenoticias de una cadena de televisión francesa**. TVC contiene imágenes de una cadena de televisión catalana, mientras que JAFFE contiene imágenes de unas mujeres japonesas.

Para imágenes haremos una comparativa de resultados **contrastando las diversas técnicas de extracción de características** implementadas. Así pues, daremos resultados para la técnica holística, para la técnica *Random Pixels*, y para la técnica LBP. Ya veremos que fijaremos los parámetros de valor de **redimensionamiento** y **método de cálculo de coeficientes** a unos valores óptimos, los cuales dan mejores resultados, sea cual sea la técnica de extracción de características. En concreto, los resultados que deberemos de dar para cada técnica de extracción de características son los siguientes: valor de *accuracy*, valor de F_1 *score*, valor de RR, curva ROC, y curva PR.

Para vídeos, no sólo fijaremos los parámetros de valor de **redimensionamiento** y **método de cálculo de coeficientes**, sino que también fijaremos la **técnica de extracción de características**. Ésta última será la que mejores resultados nos haya dado para imágenes estáticas. Ya veremos que en este caso se trata de la técnica *Random Pixels*. En concreto, los resultados que deberemos de dar para cada vídeo son los siguientes: valor de *accuracy* y valor de F_1 *score*. En conjunto para todos los vídeos también se deberá de dar el valor de MAP.

Tanto para imágenes como para vídeos, con tal de minimizar el tiempo de ejecución de los experimentos se ha hecho uso adecuado de los recursos del servidor del GPI que tenemos los estudiantes a nuestra disposición [39]. En concreto, para cada experimento se han solicitado **10 CPUs** y **8 GB de RAM**. Si hubiéramos pedido más, el resultado hubiera sido el mismo, ya que con estos recursos tenemos suficiente para solventar nuestra ejecución, por lo que, si pedimos más, estaremos desaprovechando la utilidad de estos recursos extra.

4.1. Reconocimiento en imágenes

Hay que diferenciar entre resultados obtenidos para el conjunto de imágenes de **entrenamiento** y resultados obtenidos para el conjunto de imágenes de **test**. Así mismo, también hay que diferenciar entre tipo de resultados de **validación** y tipo de resultados de **reconocimiento**. Los valores de *accuracy*, F_1 *score*, *precision*, y *recall* pertenecen a la fase de validación, mientras

que el valor de **RR** pertenece a la fase de reconocimiento. Recordar que también se obtendrán las gráficas de la **curva ROC** y la **curva PR**.

También hemos de tener en cuenta que como valores óptimos de parámetros se ha fijado el valor de **redimensionamiento** a **20 píxeles**, mientras que el **método de cálculo de coeficientes** se ha definido como *Primal-Dual Logarithmic Barrier*. Con los parámetros definidos así, para la base de datos utilizada siempre obtendremos mejores resultados, sea cual sea la técnica de extracción de características.

Dicho esto, es necesario definir las características que tiene la base de datos con la cual se han realizado los experimentos. Esta base de datos consta de **22 identidades: 12 pertenecen a TVC y son consideradas válidas, mientras que 10 pertenecen a JAFFE y son consideradas inválidas**. Estamos considerando, por tanto, que **las identidades de TVC son los modelos**, mientras que las identidades de JAFFE no lo son, por lo que se trata de identidades inválidas. De las identidades válidas, por cada una de éstas cogeremos ⅓ del número de imágenes que tenga para imágenes de modelos (base de datos), otro ⅓ para imágenes de entrenamiento, y otro ⅓ para imágenes de test. De las identidades inválidas, por cada una de éstas cogeremos ½ del número de imágenes que tenga para imágenes de entrenamiento, y otro ½ para imágenes de test. Es lógico que no tomemos ninguna imagen de ninguna identidad inválida para modelos, ya que los modelos únicamente contienen imágenes válidas. En cuanto a las características de las que hablábamos, hay que decir que, en visión global, se puede considerar que en esta **base de datos** hay **pocas identidades y muchas imágenes por identidad**. También hay que tener en cuenta que todas **las imágenes son frontales, algunas de TVC con ligeras variaciones de pose**, a las cuales la técnica de reconocimiento facial *Sparse Representation* dice ser robusta, ya que gracias a la robustez a la oclusión lo permite [10]. También se presentan **variaciones de expresión, sobre todo en JAFFE**, las cuales también quedan solventadas con esta técnica [10]. Las imágenes no presentan variaciones de iluminación, ni ningún tipo de oclusión o corrupción. Si hubieran presentado estas características, esta técnica también hubiera solucionado hipotéticamente estos problemas [10].

Para la técnica de extracción de características **holística** se obtienen los siguientes resultados:

Tabla I: Resultados para TVC utilizando la técnica holística.
Valores de *accuracy*, F_1 *score*, y RR, tanto para el conjunto de imágenes de entrenamiento como de test..

		Imágenes de entrenamiento	Imágenes de test
Validación	*Accuracy*	78,92%	**79,49%**
	F_1 *score*	88,22%	**88,57%**
	Precision	78,92%	79,49%
	Recall	100,00%	100,00%
Reconocimiento	*RR*	42,75%	**37,08%**
Tiempo de ejecución		2 min 5 s	

Como se puede observar, se obtienen valores altos, excepto para el RR, tanto para el conjunto de imágenes de entrenamiento como de test.

También se obtienen las gráficas correspondientes a la curva ROC y la curva PR:

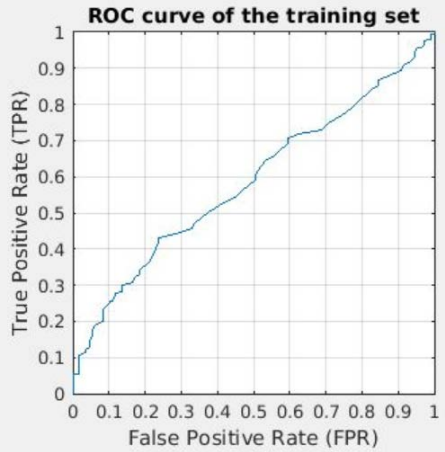

Figura 30: Curva ROC para TVC utilizando la técnica holística.
Gráfica de evaluación realizada para el conjunto de imágenes de entrenamiento de la base de datos.

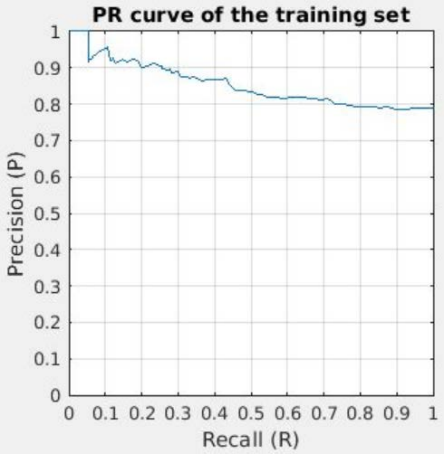

Figura 31: Curva PR para TVC utilizando la técnica holística.
Gráfica de evaluación realizada para el conjunto de imágenes de entrenamiento de la base de datos.

Como se puede apreciar, el AUC de la curva ROC es bajo, por lo que esta gráfica da a entender un mal rendimiento del sistema clasificador binario para este caso en que utilizamos la técnica holística. Sin embargo, no ocurre lo mismo con la curva PR, cuyo AUC es alto. Esto es debido a que los valores de *precision* y *recall* son altos, sea cual sea el valor del umbral τ. En este sentido nuestro clasificador da a entender un buen rendimiento para esta técnica.

Para la técnica de extracción de características ***Random Pixels*** se obtienen los siguientes resultados:

Tabla II: Resultados para TVC utilizando la técnica *Random Pixels*.
Valores de *accuracy*, F_1 *score*, y RR, tanto para el conjunto de imágenes de entrenamiento como de test..

		Imágenes de entrenamiento	Imágenes de test
Validación	*Accuracy*	87,04%	**84,42%**
	F_1 *score*	92,00%	**90,01%**
	Precision	89,74%	91,75%
	Recall	94,36%	88,34%
Reconocimiento	*RR*	86,71%	**89,18%**
Tiempo de ejecución		51 s	

Como se puede observar, se obtienen **valores altos** en todos los aspectos, tanto para el conjunto de imágenes de entrenamiento como de test. Además, el **tiempo de ejecución es inferior a la mitad del tiempo que han llevado los experimentos con las otras dos técnicas de extracción de características**. Esto es debido por la propia técnica, la cual tiene en cuenta un mínimo número de píxeles para caracterizar las caras, en comparación con las otras técnicas. Aun así, vemos que es la que mejores resultados proporciona o, al menos, para esta base de datos.

También se obtienen las gráficas correspondientes a la curva ROC y la curva PR:

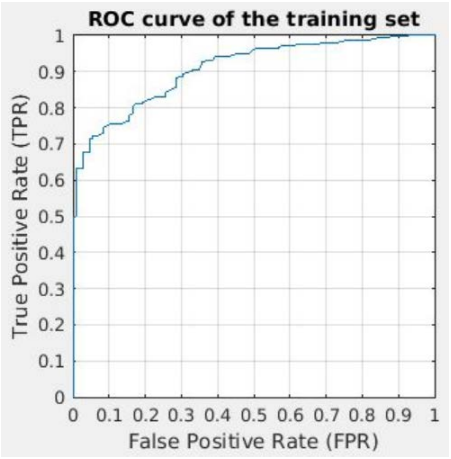

Figura 32: Curva ROC para TVC utilizando la técnica *Random Pixels*.
Gráfica de evaluación realizada para el conjunto de imágenes de entrenamiento de la base de datos.

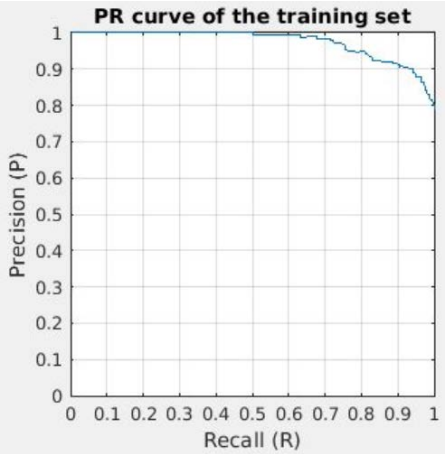

Figura 33: Curva PR para TVC utilizando la técnica *Random Pixels*.
Gráfica de evaluación realizada para el conjunto de imágenes de entrenamiento de la base de datos.

Como se puede apreciar, el **AUC tanto de la curva ROC como de la curva PR son altos**. Así pues, estas gráficas dan a entender un buen rendimiento del sistema clasificador binario para esta técnica *Random Pixels*.

Para la técnica de extracción de características **LBP** se obtienen los siguientes resultados:

Tabla III: Resultados para TVC utilizando la técnica LBP.
Valores de *accuracy*, F_1 *score*, y RR, tanto para el conjunto de imágenes de entrenamiento como de test..

		Imágenes de entrenamiento	Imágenes de test
Validación	*Accuracy*	78,92%	**79,49%**
	F_1 *score*	88,22%	**88,57%**
	Precision	78,92%	79,49%
	Recall	100,00%	100,00%
Reconocimiento	*RR*	16,05%	**19,72%**
Tiempo de ejecución		2 min 3 s	

Como se puede observar, se obtienen **valores altos, excepto para el RR**, tanto para el conjunto de imágenes de entrenamiento como de test. Vemos que casualmente son los mismos resultados que para la técnica holística, excepto el valor de RR, el cual es aún más bajo en este caso. El tiempo de ejecución casualmente es también prácticamente el mismo que para la técnica holística. Son simplemente casualidades.

También se obtienen las gráficas correspondientes a la curva ROC y la curva PR:

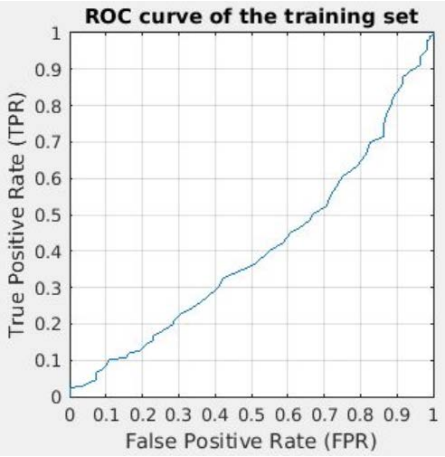

Figura 34: Curva ROC para TVC utilizando la técnica LBP.
Gráfica de evaluación realizada para el conjunto de imágenes de entrenamiento de la base de datos.

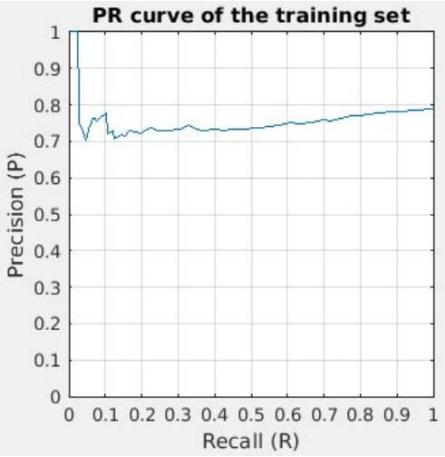

Figura 35: Curva PR para TVC utilizando la técnica LBP.
Gráfica de evaluación realizada para el conjunto de imágenes de entrenamiento de la base de datos.

Como se puede apreciar, el AUC de la curva ROC es bajo, por lo que esta gráfica da a entender un mal rendimiento del sistema clasificador binario para este caso en que utilizamos la técnica LBP. El AUC de la curva ROC es aún más inferior que el obtenido mediante la técnica holística. En cuanto a la curva PR, su AUC es alto. Esto es debido a que los valores de *precision* y *recall* son altos, sea cual sea el valor del umbral τ. En este sentido nuestro clasificador da a entender un buen rendimiento para esta técnica. El AUC de la curva PR es ligeramente inferior que el obtenido mediante la técnica holística.

4.2. Reconocimiento en vídeos

Para vídeos no daremos resultados para ningún conjunto de imágenes de entrenamiento, sino únicamente de test, ni tampoco diferenciaremos entre resultados de validación y resultados de reconocimiento. Para cada vídeo se obtendrán los valores de *accuracy*, F_1 *score*, *precision*, y *recall*. Como valor global de todos los vídeos también se obtendrá el valor de MAP.

Como se ha podido comprobar en el reconocimiento en imágenes, la mejor técnica de extracción de características en este caso ha resultado ser *Random Pixels*. Esta técnica proporciona los mejores valores y, además, en un tiempo mínimo. Al menos, así lo ha sido para la base de datos TVC. Así pues, para el reconocimiento en vídeos fijaremos esta técnica de extracción de características, a la vez que también seguiremos definiendo los parámetros de redimensionamiento a 20 píxeles y el método de cálculo de coeficientes a *Primal-Dual Logarithmic Barrier*, tal y como hemos hecho con imágenes estáticas.

Dicho esto, es necesario definir las características de los vídeos que se han procesado. Como ya se ha explicado en apartados anteriores es necesario definir previamente un conjunto de imágenes de entrenamiento pertenecientes a identidades inválidas. Estas identidades no pertenecen en absoluto con la base de datos que se esté creando para un vídeo en concreto. De este modo, se nos permitirá obtener en cada caso el valor óptimo de umbral τ que permite diferenciar entre identidades de imágenes de test válidas e inválidas. En concreto, se ha asignado la base de datos TVC para definir el conjunto de identidades inválidas. Recordemos que esta base de datos nos ha servido para designar identidades válidas en el reconocimiento en imágenes del apartado anterior. Por cada nuevo *track* incorporado a la base de datos de un vídeo en concreto se toma ½ de *frames* para información de modelos y otro ½ para información de entrenamiento. En cuanto a las identidades inválidas acabadas de comentar, se toman todos los *frames* como información de entrenamiento, ya que no forman parte de los modelos de la base de datos del vídeo en cuestión. Y de cada nuevo *track* a testear se toman todos los *frames* como información de test. En cuanto a las características de las que hablábamos, hay que decir que, en visión global, se puede considerar que en la base de datos de cada vídeo hay muchas identidades y muchas imágenes por identidad. También hay que tener en cuenta que las imágenes se ven sometidas a variaciones de pose, ya que estamos tratando con imágenes en movimiento. En cuanto a las imágenes inválidas de TVC recordemos que éstas tienen únicamente ligeras variaciones de pose. La técnica *Sparse Representation* implementada en principio sólo es robusta a pequeñas variaciones de pose [10]. También se presentan variaciones de expresión, las cuales también quedan solventadas con esta técnica [10]. Las imágenes no presentan variaciones de iluminación, ni ningún tipo de oclusión o corrupción. Si hubieran presentado estas características, esta técnica también hubiera solucionado hipotéticamente estos problemas [10].

A continuación, se muestran los resultados obtenidos para cada uno de los vídeos, así como los resultados globales en conjunto:

Tabla IV: Resultados para los vídeos utilizando la técnica *Random Pixels*.
Valores de *accuracy* y $\mathbf{F_1}$ *score* para cada vídeo, y valor de MAP para el conjunto de vídeos.

	Vídeo #1	Vídeo #2	Vídeo #3	Vídeo #4
Accuracy	73,00%	51,00%	47,00%	75,00%
$\mathbf{F_1}$ *score*	72,16%	63,70%	55,46%	60,32%
Precision	56,45%	48,31%	39,76%	63,33%
Recall	100,00%	93,48%	91,67%	57,58%
Tiempo de ejecución	9 h 21 min	18 h 3 min	11 h 2 min	4 h 32 min
MAP	82,45%			

Como se puede observar en algunos valores de *accuracy* y $\mathbf{F_1}$ *score*, sobre todo en el tercer vídeo, éstos son algo bajos. Esto es debido a la gran cantidad de FP que llegamos a obtener por vídeo. Se ha comprobado además que son FP con el 100% de los *frames* validados, por lo que en realidad todos son erróneamente validados, ya que en verdad se trata de identidades desconocidas (*unknowns*). Éste es un gran error que no podemos controlar, ya que no es problema de que el método esté mal diseñado o implementado. Este hecho perjudica y mucho en los resultados, los cuales en conjunto se ven claramente penalizados en cada vídeo.

Para entender mejor este problema basta con mirar el resultado de MAP obtenido en conjunto para todos los vídeos. Un valor muy bueno de 82,45%, el cual es bastante superior que los resultados obtenidos de *accuracy* y $\mathbf{F_1}$ *score*. Esto es debido a que el MAP recordemos es un promedio de *queries*, donde la *query* de *unknown* —comentada en el párrafo anterior— es una en concreto. Si en conjunto todas las *queries* tienen valores elevados, como es el caso, excepto la de *unknown*, el promedio seguirá siendo elevado. Así pues, podemos considerar que la *query* de *unknown* penaliza bastante para los valores de *accuracy* y $\mathbf{F_1}$ *score*, mientras que apenas influye para el valor de MAP global. Es por este motivo que debemos de presumir y mucho del valor de MAP obtenido, más que de los resultados individuales de *accuracy* y $\mathbf{F_1}$ *score*, ya que es donde de verdad se reflejan mejor los resultados en conjunto.

Cabe comentar también que al final no se han procesado todos los *tracks* que comprenden cada vídeo, sino que se ha parado en el análisis de los 100 primeros *tracks*. Esto es debido a que el tiempo de ejecución se iba incrementando progresivamente en cada *track*. Este tiempo venía del cálculo de coeficientes *sparse* para las imágenes de entrenamiento. Y no es que la función correspondiente tarde mucho en ejecutarse, sino que cada vez hay más imágenes de entrenamiento, y ello supone el cálculo de más coeficientes. En concreto, para cada nuevo *track* añadido a la base de datos de un vídeo específico se toman tantas imágenes como la mitad de *frames* de que consta el *track*. Y este número de imágenes que va creciendo es muy elevado.

CAPÍTULO 5. PRESUPUESTO

Para calcular el presupuesto necesario para poder llevar a cabo este proyecto se deberán de tener en cuenta tanto costes temporales como costes de *hardware* y *software*.

Para los costes temporales diferenciaremos entre los ítems que conforman la elaboración del proyecto: bibliografía, implementación, memoria, y presentación.

La normativa de la Escuela Superior de Ingenierías Industrial, Aeroespacial y Audiovisual de Tarrasa (ESEIAAT) [40] asigna un total de 24 ECTS para la elaboración completa del PFG [41]. Teniendo en cuenta que 1 ECTS equivale a una dedicación de unas 25 horas [42], esto supone un total de

$$24\ \cancel{ECTS} * \frac{25\ h}{1\ \cancel{ECTS}} = 600\ h. \qquad (25)$$

Teniendo en cuenta que un ingeniero junior cobra aproximadamente 10 €/h, se puede extraer el siguiente presupuesto:

Tabla V: Costes de ítems del proyecto.
Estimación del presupuesto relacionado con los costes temporales.

Ítem	Duración	Coste
Bibliografía	50 h	500 €
Implementación	250 h	2500 €
Memoria	250 h	2500 €
Presentación	50 h	500 €
TOTAL	600 h	6000 €

A este precio debemos de incluir también las horas correspondientes a las numerosas reuniones que se han ido manteniendo a lo largo del proyecto. Para ello, deberemos de tener en cuenta que el director del proyecto como ingeniero senior cobra unos 60 €/h. Teniendo en cuenta que cada reunión tiene una duración de 1 h, si se han realizado aproximadamente unas 30 reuniones, el precio total de las reuniones será de

$$30\ reuniones * \frac{60\ €}{1\ h} = 30\ \cancel{h} * \frac{60\ €}{1\ \cancel{h}} = 1800\ €. \qquad (26)$$

Así pues, el presupuesto total referente a costes temporales asciende a

$$costes\ temporales = \\ = costes\ de\ ítems + costes\ de\ reuniones = 6000\ € + 1800\ € = 7800\ €. \qquad (27)$$

Para los **costes de *hardware* y *software*** deberemos de tener en cuenta todas las herramientas necesarias con las que se ha podido llevar a cabo el proyecto:

Tabla VI: Costes de hardware y software del proyecto.
Estimación del presupuesto relacionado con los costes de hardware y software.

Ítem	Tipo	Coste
Ordenador	Hardware	750,00 €
MATLAB	Software	3630,00 €
Microsoft Word y Microsoft PowerPoint	Software	53,24 €
TOTAL	Hardware/Software	**4433,24 €**

Para realizar el proyecto se necesita hacer uso de un ordenador. En este caso se ha sugerido un ordenador de sobremesa, al cual le hemos de añadir todos los componentes necesarios (pantalla, ratón, teclado, CPU, RAM, disco duro, etc.). Se supone que el ordenador escogido cumple con los requisitos para poder llevar a cabo la implementación y ejecución del método implementado. Se propone un **ordenador** de **3000 €** en total (IVA incluido) con todos los componentes incorporados. Para conocer un precio estimado del uso que se habría hecho del ordenador para este proyecto, se realiza el siguiente cálculo:

$$\begin{aligned} &coste\ del\ uso\ de\ ordenador = \\ &= \frac{precio\ del\ ordenador}{vida\ útil\ del\ ordenador} * duración\ del\ proyecto = \\ &= \frac{3000\ €}{3\ años} * 9\ meses = \frac{3000\ €}{36\ \cancel{meses}} * 9\ \cancel{meses} = 750\ €. \end{aligned} \quad (28)$$

Para realizar el proyecto también hemos necesitado conectarnos al servidor del GPI. Para ello, se ha utilizado la aplicación **X2Go Client** [43], la cual es *open-source* (código abierto) y gratuita. Este ítem de *software* nos sale gratis, pero no lo incluiremos como parte del presupuesto del proyecto real, ya que en éste consideramos que toda la elaboración del mismo proyecto se realiza en el ordenador anteriormente propuesto, sin conexión a ningún servidor remoto.

También debemos de considerar el precio de la licencia de la aplicación **MATLAB**. Se propone la **licencia «*Standard*»** [44], la cual es para uso por cuenta propia. No solamente hay que considerar MATLAB sino también las *toolboxes* utilizadas. Las *toolboxes* vienen a ser como *packs* de funciones útiles catalogadas en función del área temática [45]. En este caso se ha hecho uso de ***Image Processing Toolbox*** que, como su propio nombre indica, trata sobre el procesamiento de imágenes. Esta versión de MATLAB cuesta **2000 €**, mientras que la *toolbox* utilizada tiene un valor de **1000 €**, por lo que sumando ambos y aplicando el 21% del IVA sobre este precio obtenemos una cantidad de

$$(2000\ € + 1000\ €) + (2000\ € + 1000\ €) * 0{,}21 = 3630\ €. \quad (29)$$

Y también debemos de considerar el precio de la licencia de las aplicaciones Microsoft Word [46] y Microsoft PowerPoint [47]. La primera aplicación nos ha servido para redactar la memoria, mientras que la segunda para elaborar la presentación. Se propone la licencia «Office 365 Empresa» [48], la cual es para uso empresarial e incluye entre otras las dos aplicaciones acabadas de mencionar. Esta versión de Office tiene la posibilidad de compra de forma mensual, cuyo coste está establecido en 8,80 € por mes. Aproximadamente, para redactar la memoria se han tardado unos 4 meses, mientras que para elaborar la presentación 1 mes. Ello conlleva un pago de 5 meses por el servicio, es decir,

$$\frac{8{,}80\ €}{1\ \cancel{mes}} * 5\ \cancel{meses} = 44\ €. \qquad (30)$$

Aplicando el 21% del IVA sobre este precio obtenemos una cantidad de

$$44\ € + 44\ € * 0{,}21 = 53{,}24\ €. \qquad (31)$$

Finalmente, para conocer el presupuesto total de este proyecto, sumaremos los costes temporales totales con los costes de *hardware* y *software* totales:

$$\begin{gathered} presupuesto\ del\ proyecto = \\ = costes\ temporales + costes\ de\ hardware\ y\ software = \\ = 7800\ € + 4433{,}24\ € = 12\ 233{,}24\ €. \end{gathered} \qquad (32)$$

Así pues, se puede concluir que el presupuesto total que se necesitaría para poder llevar a cabo este trabajo como un proyecto real sería de 12 233,24 €.

CAPÍTULO 6. CONCLUSIÓN

6.1. Conclusiones

En este PFG se ha llevado a cabo una anotación automática de vídeos, en concreto de programas de televisión, mediante un reconocimiento facial no supervisado. La anotación automática ha sido posible gracias a unos datos de *video tracking*, mientras que la no supervisión del reconocimiento facial ha sido posible gracias al nombre de las identidades que van apareciendo en los programas de televisión. La técnica de reconocimiento facial utilizada para reconocer caras ha sido *Sparse Representation*.

En cuanto al tiempo de ejecución que han durado los diversos experimentos realizados podemos considerar que éste es excesivo. Principalmente tanto tiempo viene del cálculo de coeficientes *sparse* para el conjunto de imágenes de entrenamiento. El hecho es que, a medida que se avanza en la ejecución, más *tracks* son procesados y, por tanto, más modelos son creados en la base de datos. Este hecho hace que cada vez se tarde más tiempo en calcular los coeficientes *sparse* para el conjunto de imágenes de entrenamiento propuesto, ya que a medida que hay más imágenes de modelos, más imágenes de entrenamiento aparecen y, por tanto, más vectores de coeficientes son necesarios calcular. Recordemos que se obtiene un vector de coeficientes por cada imagen, que en este caso es de entrenamiento. Y las imágenes de entrenamiento se definen de tal forma que por cada nuevo *track* añadido a la base de datos se toma la mitad de *frames* de dicho *track* como información de entrenamiento. De esta forma el umbral óptimo que discrimina entre identidades válidas e inválidas es calculado satisfactoriamente a medida que se incorporan nuevos *tracks* a la base de datos, ya que para calcularlo correctamente en cada ocasión hace falta información de todas las identidades presentes hasta el momento. Se ha comprobado que el tiempo que se tarda para calcular un vector de coeficientes para una imagen es mínimo. Este cálculo recordemos que es posible gracias a la función externa implementada por terceras personas, la cual se encarga de calcular los coeficientes *sparse* para una imagen de entrada. Sin embargo, si tenemos muchas imágenes de entrenamiento, como es el caso, veremos que el tiempo de cálculo se incrementa demasiado. Al principio con pocos modelos irá rápido, pero a medida que se añadan irá bastante más despacio. Esto es lo que hace ralentizar tanto la tarea de reconocimiento facial.

En cuanto a los resultados obtenidos, para imágenes estáticas vemos que el método funciona muy bien, en cambio para vídeos se obtienen *a priori* peores tasas. Esto no es del todo cierto, ya que en vídeos hemos de considerar el problema que nos hemos encontrado de que a algunas de las identidades desconocidas se les ha identificado como identidades existentes en la base de datos, y además con el 100% de *frames* validados por *track*. Éste no es un problema de programación, ni del método, por lo que en este aspecto no tenemos la culpa nosotros. Recordemos que este hecho se ve claramente contrarrestado con el valor de MAP obtenido, el cual sí es comparable con los resultados obtenidos para imágenes estáticas.

Si nos centramos en los resultados individuales de *tracks*, y no en conjunto, el sistema parece ser más robusto con vídeos, ya que para los *tracks* correctamente reconocidos las tasas de validación y reconocimiento acostumbran a ser por lo general del 100% de los *frames*, mientras que en imágenes no todas son validadas y reconocidas correctamente. No obstante, esto no ocurre para algunos de los *tracks*, en los cuales las tasas de validación y reconocimiento se ven reducidas. El motivo puede ser debido principalmente a las variaciones de pose que presentan los *frames*, ya que al tratarse de vídeos las imágenes no se adquieren en las condiciones ideales que alguien puede esperar como ocurre en el caso de las imágenes estáticas. Y parece ser que la técnica de reconocimiento facial empleada (*Sparse Representation*) no es robusta a grandes variaciones de pose. De hecho, en la bibliografía consultada se comenta que el método propuesto no tiene en cuenta variaciones de pose, aunque sí es robusto a pequeñas variaciones [10].

Hablando de condiciones en que las imágenes son tomadas hemos de decir también que con este método podíamos haber testeado imágenes con variaciones de iluminación, o incluso con algún tipo de oclusión o corrupción en la cara. Recordemos que la técnica de reconocimiento facial utilizada resuelve estas circunstancias [10], pero en este proyecto no se ha demostrado.

6.2. Líneas de futuro

En este PFG se parte de la idea que una identidad es desconocida mientras no aparezca su correspondiente nombre por vez primera. Es por ello que en el *ground truth* de los *tracks* se ha fijado la etiqueta *unknown* para los *tracks* de cada identidad en los que aún no han visto su nombre asociado. Esto supone un falseamiento del sistema, en este caso del *ground truth*, ya que estamos adaptando los resultados a nuestra conveniencia. En este proyecto se van procesando los *tracks* en el orden en que aparecen en el vídeo. Para un futuro trabajo se propone primero crear los modelos, es decir, procesar los *tracks* que tienen nombre asociado y, posteriormente, reconocer las identidades de aquellos *tracks* que no tienen nombre asociado. De este modo, el problema que teníamos acerca de que una identidad es desconocida mientras no aparezca su nombre queda solventado, ya que partiremos en un principio de todos los modelos creados con sus correspondientes nombres y ya podremos reconocer en cada caso la identidad de los *tracks* que mencionábamos. Así, en el *ground truth* pondremos los nombres de las identidades verdaderas a las que pertenecen los *tracks* comentados.

Esta idea, sin embargo, tiene un punto débil, el cual es crucial: el tiempo de ejecución. Si partimos de los modelos creados, empezaremos a reconocer los *tracks* posteriormente ya a partir de muchas imágenes de entrenamiento, ya que se habrán creado muchas identidades en la base de datos. Y ello supone que el tiempo del cálculo de coeficientes para las imágenes de entrenamiento empiece siendo bastante alto. Si ya empezamos con un tiempo de

procesamiento por *track* elevado, a medida que avanza la ejecución éste se irá incrementando progresivamente y veremos que apenas nos da tiempo a procesar *tracks*. Ello promueve, por tanto, buscar también algún método para reducir el tiempo de procesamiento del cálculo de coeficientes en conjunto. Individualmente funciona bien, ya que tarda poco, pero en conjunto, dada la gran cantidad de imágenes que se toman en cuenta, funciona mal, ya que tarda mucho. Es por ello que se propone, por ejemplo, no tomar tantas imágenes de cada modelo.

Otro aspecto a mejorar en nuestro sistema es el caso de cuando en una escena aparece un nombre y, sin embargo, hay más de un *track* solapado con dicho nombre. Nuestro método propone añadir un sufijo numérico a cada uno de estos *tracks*, de modo que hay *tracks* correctamente asociados y *tracks* incorrectamente asociados. Esto se podría solucionar implementando un sistema capaz de reconocer la persona que está hablando, es decir, que gesticula con los labios [49]. De esta manera asignaríamos directamente la identidad de la persona que interactúa con el nombre que se muestra en pantalla, mientras que las otras personas «de fondo» se clasificarían como desconocidas, es decir, como *unknowns*. De esta manera obtendríamos mejores tasas de reconocimiento, ya que el *ground truth* seguiría siendo el mismo, pero el reconocimiento resultaría ahora exitoso, cuando antes no lo era.

También hay que tener en cuenta que en nuestro método se ha propuesto la premisa de que un *track* es válido si al menos el 20% de sus *frames* son validados correctamente. Este umbral recordemos que lo hemos escogido *ad-hoc*, es decir, para un conjunto de *tracks* se ha comprobado que este valor es el que aproximadamente mejor discrimina entre identidades válidas e identidades inválidas (identidades desconocidas, es decir, *unknowns*). Así pues, también se propone para un futuro proyecto el implementar un sistema que determine de forma automática el valor óptimo de umbral, siguiendo la misma metodología de algoritmo que para la obtención del valor óptimo de umbral τ que discriminaba entre identidades válidas o identidades inválidas para imágenes. Sería lo mismo, pero en vez de para imágenes, para tracks.

Por último, también se propone experimentar con imágenes de entrada que presenten variaciones de iluminación, o incluso algún tipo de oclusión o corrupción en la cara, ya que la técnica de reconocimiento facial *Sparse Representation* trata de solucionar también estos casos [10]. Al menos, se plantea testearlo para el caso de imágenes estáticas.

CAPÍTULO 7. BIBLIOGRAFÍA

[1] «Facial recognition system», *Wikipedia*, 2017. [En línea]. Disponible en: https://en.wikipedia.org/wiki/Facial_recognition_system. [Consultado: 20-feb-2017].

[2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, «Face Recognition: A Literature Survey», *ACM Comp. Surv.*, vol. 35, no. 4, pp. 399–459, 2003.

[3] A. S. Tolba, A. H. El-Baz, and A. A. El-Harby, «Face Recognition: A Literature Review», *World Academy Sci., Eng. Technol.*, vol. 2, no. 7, pp. 2556–2571, 2008.

[4] «MediaEval Benchmarking Initiative for Multimedia Evaluation», *MediaEval*, 2017. [En línea]. Disponible en: http://multimediaeval.org/. [Consultado: 07-jun-2017].

[5] C. Vondrick, D. Patterson, and D. Ramanan, «Efficiently Scaling up Crowdsourced Video Annotation», *Int. J. Comput. Vision*, vol. 101, no. 1, pp. 184–205, 2013.

[6] «Video Annotation Tool from Irvine, California», *Inst. Tecnol. Massachusetts*, 2017. [En línea]. Disponible en: http://web.mit.edu/vondrick/vatic. [Consultado: 20-feb-2017].

[7] A. Yilmaz, O. Javed, and M. Shah, «Object Tracking: A Survey», *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, 2006.

[8] B. Raytchev and H. Murase, «Unsupervised Recognition of Multi-View Face Sequences Based on Pairwise Clustering with Attraction and Repulsion», *Comput. Vision Image Understanding*, vol. 91, no. 1-2, pp. 22–52, 2003.

[9] R. Jafri and H. R. Arabnia, «A Survey of Face Recognition Techniques», *J. Inf. Process. Syst.*, vol. 5, no. 2, pp. 41–68, 2009.

[10] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, «Robust Face Recognition via Sparse Representation», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009.

[11] J. Wright *et al.*, «Sparse Representation For Computer Vision and Pattern Recognition», *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1040, 2010.

[12] O. J. Lara, «Tècniques de reconeixement facial», Proyecto Final de Grado, Dpto. Teoría Señal Comun., Univ. Pol. Cataluña, Barcelona, 2014.

[13] «Image Processing Group (GPI)», *Univ. Pol. Cataluña*, 2017. [En línea]. Disponible en: https://imatge.upc.edu/. [Consultado: 04-jun-2017].

[14] «MATLAB», *MathWorks*, 2017. [En línea]. Disponible en: https://mathworks.com/products/matlab/. [Consultado: 20-feb-2017].

[15] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini, «An Interactive Tool for Manual, Semi-Automatic and Automatic Video Annotation», *Comput. Vision Image Understanding*, vol. 131, pp. 88–99, 2015.

[16] J. Poignant, L. Besacier, and G. Quénot, «Unsupervised Speaker Identification in TV Broadcast Based on Written Names», *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 1, pp. 57–68, 2015.

[17] «The 2016 Multimodal Person Discovery in Broadcast TV Task», *MediaEval*, 2016. [En línea]. Disponible en: http://www.multimediaeval.org/mediaeval2016/persondiscovery. [Consultado: 22-feb-2017].

[18] G. Kumar and P. K. Bhatia, «A Detailed Review of Feature Extraction in Image Processing Systems», en *2014 4th Int. Conf. Advanced Comput. Commun. Technol.*, pp. 5–12.

[19] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, «Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About», *Proc. IEEE*, vol. 94, no. 11, pp. 1948–1962, 2006.

[20] Z. Fan, M. Ni, Q. Zhu, and E. Liu, «Weighted Sparse Representation for Face Recognition», *Neurocomput.*, vol. 151, part 1, pp. 304–309, 2015.

[21] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, «Robust Face Recognition via Adaptive Sparse Representation», *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2368–2378, 2014.

[22] J-X Mi and J-X Liu, «Face Recognition Using Sparse Representation-Based Classification on K-Nearest Subspace», *PLOS ONE*, vol. 8, no. 3, pp. 1–11, 2013.

[23] S. Nagendra, R. Baskaran, and S. Abirami, «Video-Based Face Recognition and Face-Tracking using Sparse Representation-Based Categorization», *Procedia Comput. Sci.*, vol. 54, pp. 746–755, 2015.

[24] Y-C Chen, V. M. Patel, S. Shekhar, R. Chellappa, and P. J. Phillips, «Video-Based Face Recognition via Joint Sparse Representation», en *2013 10th IEEE Int. Conf. Workshops Autom. Face and Gesture Recognition*, pp. 1–8.

[25] E. G. Ortiz, A. Wright, and M. Shah, «Face Recognition in Movie Trailers via Mean Sequence Sparse Representation-Based Classification», en *2013 IEEE Conf. Comput. Vision Pattern Recognition*, pp. 3531–3538.

[26] L. C. Paul and A. A. Sumam, «Face Recognition Using Principal Component Analysis Method», *Int. J. Advanced Res. Comput. Eng. Technol.*, vol. 1, no. 9, pp. 135–139, 2012.

[27] T. Ahonen, A. Hadid, and M. Pietikäinen, «Face Description with Local Binary Patterns: Application to Face Recognition», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2051, 2006.

[28] «Local Binary Patterns», *Scholarpedia*, 2016. [En línea]. Disponible en: http://www.scholarpedia.org/article/Local_Binary_Patterns. [Consultado: 05-jun-2016].

[29] «Precision and recall», *Wikipedia*, 2017. [En línea]. Disponible en: https://en.wikipedia.org/wiki/Precision_and_recall. [Consultado: 25-abr-2017].

[30] «Receiver Operating Characteristic», *Wikipedia*, 2017. [En línea]. Disponible en: https://en.wikipedia.org/wiki/Receiver_operating_characteristic. [Consultado: 25-abr-2017].

[31] «What is Precision-Recall (PR) curve?», *Quora*, 2014. [En línea]. Disponible en: https://www.quora.com/What-is-Precision-Recall-PR-curve. [Consultado: 25-abr-2017].

[32] «Curva ROC», *Wikipedia*, 2016. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Curva_ROC. [Consultado: 05-jun-2017].

[33] «Mean Average Precision», *Kaggle*, 2017. [En línea]. Disponible en: https://www.kaggle.com/wiki/MeanAveragePrecision. [Consultado: 24-may-2017].

[34] «Landing», *Grupo Proces. Imagen*, 2017. [En línea]. Disponible en: https://imatge.upc.edu/trac/wiki/LandingNew. [Consultado: 04-jun-2017].

[35] «RGB color model», *Wikipedia*, 2017. [En línea]. Disponible en: https://en.wikipedia.org/wiki/RGB_color_model. [Consultado: 15-abr-2017].

[36] B. Póczos and R. Tibshirani, «Barrier Methods», Univ. Carnegie Mellon, Pittsburgh, Pensilvania, Estados Unidos, 2017. [En línea]. Disponible en: http://www.stat.cmu.edu/~ryantibs/convexopt-F13/lectures/17-BarrierMethods.pdf. [Consultado: 17-abr-2017].

[37] K. Toh and S. Yun, «An Accelerated Proximal Gradient Algorithm for Nuclear Norm Regularized Least Squares Problems», Univ. Nac. Singapur, Singapur, 2017. [En línea]. Disponible en: https://www.robots.ox.ac.uk/~vgg/rg/papers/apg.pdf. [Consultado: 17-abr-2017].

[38] «The Japanese Female Facial Expression (JAFFE) Database», *Facial Expression Home Page*, 2017. [En línea]. Disponible en: http://www.kasrl.org/jaffe.html. [Consultado: 21-may-2017].

[39] «Access Servers», *Grupo Proces. Imagen*, 2017. [En línea]. Disponible en: https://imatge.upc.edu/trac/wiki/DevelopmentPlatform/HardwareResources. [Consultado: 22-may-2017].

[40] «Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa (ESEIAAT)», *Univ. Pol. Cataluña*, 2017. [En línea]. Disponible en: http://eseiaat.upc.edu/. [Consultado: 04-jun-2017].

[41] «Grau en Enginyeria de Sistemes Audiovisuals – Pla d'Estudis», *ESEIAAT*, 2017. [En línea]. Disponible en: http://eseiaat.upc.edu/ca/estudis/estudis-en-enginyeries-de-la-telecomunicacio/grau-en-enginyeria-de-sistemes-audiovisuals/pla-destudis. [Consultado: 15-may-2017].

[42] «European Credit Transfer and Accumulation System», *Wikipedia*, 2016. [En línea]. Disponible en: https://es.wikipedia.org/wiki/European_Credit_Transfer_and_Accumulation_System. [Consultado: 15-may-2017].

[43] «X2Go Client», *X2Go*, 2017. [En línea]. Disponible en: http://wiki.x2go.org/doku.php/doc:installation:x2goclient. [Consultado: 04-jun-2017].

[44] «Pricing and Licensing», *MathWorks*, 2017. [En línea]. Disponible en: https://es.mathworks.com/pricing-licensing.html. [Consultado: 15-may-2017].

[45] «Productos y Servicios», *MathWorks*, 2017. [En línea]. Disponible en: https://es.mathworks.com/products.html. [Consultado: 15-may-2017].

[46] «Microsoft Word», *Office*, 2017. [En línea]. Disponible en: https://products.office.com/es-es/word. [Consultado: 04-jun-2017].

[47] «Microsoft PowerPoint», *Office*, 2017. [En línea]. Disponible en: https://products.office.com/es-es/powerpoint. [Consultado: 04-jun-2017].

[48] «Comparación de todos los productos de Microsoft Office», *Office*, 2017. [En línea]. Disponible en: https://products.office.com/es-ES/compare-all-microsoft-office-products. [Consultado: 15-may-2017].

[49] M. Bendris, D. Charlet, and G. Chollet, «Lip Activity Detection for Talking Faces Classification in TV-Content», en *2010 3rd Int. Conf. Mach. Vision*, pp. 187–191.

«Cualquier necio puede saber. La cuestión es entender».

ALBERT EINSTEIN