Universidad Autónoma de Madrid
Escuela Politécnica Superior

Trabajo de fin de Máster

VERIFICADOR FACIAL EN DISPOSITIVOS MÓVILES, Y ESTRATEGIAS DE FUSIÓN DE BIOMETRÍA FACIAL Y DE VOZ

Máster Universitario en
Ingeniería de Telecomunicación

Máster Universitario en
Investigación e Innovación en las TIC

Ángel Pérez Lemonche
Junio 2017

VERIFICADOR FACIAL EN DISPOSITIVOS MÓVILES, Y ESTRATEGIAS DE FUSIÓN DE BIOMETRÍA FACIAL Y DE VOZ

AUTOR: Ángel Pérez Lemonche
TUTOR: D. Pablo Martín Gila
PONENTE: Dr. Daniel Ramos Castro

Argos Soluciones Globales S.L.
Área de Tratamiento de Voz y Señales (ATVS)
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2017

Resumen

Resumen

Argos Soluciones Globales S.L. y en particular su porfolio de productos biométricos, FaceOn, son especialistas en el desarrollo de soluciones biométricas con tecnología propia. VerifyByFace es un sistema de verificación facial para el control de acceso físico y virtual. Dentro del departamento de investigación y desarrollo, la tarea desempeñada que se expone en este Trabajo de Fin de Máster es el desarrollo y evaluación de la tecnología inicial de biometría facial de FaceOn con el fin de mejorarla y así poder obtener un sistema de verificación facial móvil, consiguiendo combatir las restricciones de tiempo real, tamaño y capacidad de cómputo de un terminal *smartphone*.

Además, debido a la continua evolución tecnológica de FaceOn y de su apuesta por biometría con sensores actualmente disponibles en cualquier dispositivo móvil —ordenador portátil, *smartphone* o *tablet*— el proyecto engloba el diseño y desarrollo de una herramienta que permite evaluar la fusión a nivel de puntuación de las tecnologías de verificación facial y de locutor. En paralelo, y debido al tipo de mercado demandante, diferentes sistemas *antispoofing* fueron diseñados y desarrollados con ánimo de proteger la aplicación de verificación de ataques de suplantación frente al sensor, mediante imágenes o vídeos para el sistema de biometría facial y mediante locuciones pregrabadas o imitaciones para el sistema de biometría de voz.

Palabras Clave

Biometría, Verificación facial, Reconocimiento de locutor, Fusión biométrica, control de acceso móvil, *antispoofing*...

Abstract

Argos Soluciones Globales S.L. and in particular its portfolio of biometric products, FaceOn, are specialists in the development of biometric solutions with proprietary technology. VerifyByFace is a face verification system that provides physical and virtual access control. Within the department of research and development, the task performed in this Master of Science Thesis is the development and evaluation of FaceOn's initial technology in face biometrics in order to improve it and, thereby, obtain a mobile face-verification system, attending time, size and computational cost constraints found in every smartphone device.

In addition, FaceOn is continuously developing its technology usig biometric traits that can be adquiried using sensors that are currently available in any kind of mobile device like laptops, smartphones or tablets. This project includes the design and development of a programming tool that allows evaluate score-level biometric fusion techniques using face and speaker verification technologies. Parallelly, due the demanding market, an antispoofing system was designed and developed in order to protect the verification application against spoofing attacks like images or videos in face-biometrics systems and pre-recorded utterances or voice imitations in voice-biometrics systems.

Key words

Biometrics, Face verification, Speaker recognition, Biometric fusion, Mobile access control, antispoofing...

Agradecimientos

Llegados a este punto en el que casi casi puede considerarse el final de mis estudios (casi), es complicado encontrar a alguien del que no pueda estar agradecido, ya que han sido muchas las personas que me han acompañado a lo largo de este camino, del que podría decir que empecé hace siete añazos.

Con sinceridad, quería agradecer el apoyo a mis compañeros de Argos: Adán, Javier, Ricardo y Adrián, los cuales me enseñaron, me pusieron en situaciones fuera de mi círculo de confianza, y con los que comparto una gran amistad. Particularmente quería agradecer a Pablo el que se ofreciera a dirigirme este Trabajo de Fin de Máster, incluso de manera telemática desde la otra punta de Europa, por su tiempo, dedicación y amistad.

También, y no con menor intensidad, quería dar las gracias a mi ponente Daniel Ramos, por haber querido repetir (de TFG a TFM) y guiarme no solo a la hora de realizar este trabajo, sino cuando no veía las cosas claras a lo largo de la carrera. Es siempre muy aclarador pasar diez minutos charlando con él cuando tienes la mente nublada.

Gracias a mis amigos *Juerveceros* por las risas semanales, a mis colegas de la Cátedra UAM/IBM por su compañerismo, a *éstos* por seguir quedando desde hace ya tantos años que ni me acuerdo y a mi genial Arrieta por estar ahí siempre.

Finalmente, a mis tres chicas (y a la cuadrúpeda) porque sí, porque os quiero mucho; y a mi padre por su sabiduría y su ejemplo.

Índice general

Índice de figuras

Índice de tablas

Preámbulo

Planteamiento y alcance del Trabajo de Fin de Máster

El trabajo expuesto en esta memoria se concibe en el contexto de la asignatura «Trabajo de Fin de Máster» (en adelante, TFM) del plan conjunto *Doble Máster en Ingeniería de Telecomunicación e Investigación e Innovación en Tecnologías de la Información y las Comunicaciones* que ofrece las dos titulaciones de manera conjunta, por lo que se realiza un único Trabajo de Fin de Máster por valor de 24 créditos ECTS.

Esta memoria documenta el desempeño del autor, Ángel Pérez Lemonche, en la empresa Argos Soluciones Globales S.L. desde el mes de septiembre de 2014 hasta el mes de octubre de 2015. Este trabajo se realizó a tiempo parcial durante este periodo, computando un total de 1.056 horas de trabajo.

Todo lo que se incluye en esta memoria se corresponde con trabajo realizado por el autor en la empresa Argos Soluciones Globales S.L., aunque no todo el trabajo desempeñado por el autor ni por la empresa queda aquí reflejado. En la memoria se indicará explícitamente qué tareas han sido realizadas íntegramente por el autor y cuáles ha desarrollado de manera conjunta con el equipo de trabajo, recalcando en este caso el alcance del trabajo realizado por parte del autor.

Este trabajo se hace bajo un acuerdo de confidencialidad, por lo que muchos de los resultados obtenidos se han decidido no hacer públicos o hacerlos solo parcialmente públicos. En cualquier caso, como se verá, los resultados publicados se cree que son suficientes para poder superar el TFM.

El trabajo aquí expuesto describe diversas actividades y proyectos necesarios para cubrir las necesidades de la empresa Argos Soluciones Globales S.L., que en su conjunto responden a un objetivo único concreto.

El objetivo principal de este Trabajo de Fin de Máster es dotar a Argos Soluciones Globales S.L. de un sistema de control de acceso para dispositivos móviles basado en verificación biométrica fusionada facial y de voz.

El cargo asignado y desempeñado por el autor en la empresa era de Ingeniero de Investigación y Desarrollo, por lo que el trabajo realizado tiene tanto una componente profesional, donde se ha diseñado y desarrollado productos *software* de innovación, contribuyendo en diferentes etapas de dicho proceso, así como el diseño y programación de herramientas informáticas para la evaluación de algoritmos biométricos; como una componente de investigación, realizando pruebas de uso de diferentes algoritmos de verificación facial, fusión biométrica a nivel de puntuación (*score*) y *antispoofing*, con el fin de escoger aquellos métodos y algoritmos que ofreciesen los mejores resultados para la aplicación concreta que se estaba llevando a cabo.

Estructura de la memoria

Como se mencionó en el apartado anterior, esta memoria documenta el trabajo realizado de distintos proyectos. Por ello, la memoria está estructurada según la naturaleza del trabajo llevado a cabo.

En esta memoria se diferencian siete secciones:

1. **Introducción**: en esta sección se da a conocer la motivación del TFM, los objetivos específicos de las diferentes partes que componen este trabajo, la metodología y el plan de trabajo llevado a cabo y las competencias transversales adquiridas durante el desarrollo de este trabajo en la empresa Argos Soluciones Globales S.L.

2. **Estado del arte**: se expone el estado de la técnica en sistemas de verificación facial, verificación de locutor, sistemas contra la suplantación frente al sensor o *antispoofing* y fusión de sistemas biométricos.

3. **Bases de datos**: se exponen algunas de las bases de datos utilizadas para el desarrollo de este trabajo, tanto de cara como de voz, como la estructura de directorios utilizada en las herramientas de evaluación de cara, de voz y de cara y voz conjuntamente.

4. **Desarrollo y evaluación de un sistema de verificación facial**: se expone cómo se ha desarrollado una herramienta de evaluación de algoritmos para verificación facial, así como algunas de las pruebas realizadas en la fase de investigación con el fin de maximizar la precisión en la verificación.

5. **Demo comercial y aplicación móvil FaceOnVox**: se explica cómo se ha llevado a cabo la realización de una demo comercial de la solución de verificación facial VerifyByFace, así como una API de verificación facial utilizada entre otras cosas en la aplicación móvil FaceOnVox.

6. **Verificación de locutor y fusión biométrica de cara y voz**: se justifica cómo se ha escogido, integrado y testeado una API de terceros de verificación de locutor, cómo se ha desarrollado una herramienta de evaluación modular de biometría de voz y cómo se ha desarrollado un *software* que evalúa la fusión de ambas tecnologías de verificación facial y de locutor.

7. **Conclusiones y trabajo futuro**: en esta última sección se concluye esta memoria extrayendo diferentes conclusiones de las secciones anteriores y del trabajo desarrollado en general, y se exponen líneas de actuación que quedan fuera del alcance de este TFM.

Anexos

A. **Objetivos específicos**: en este anexo se muestran los objetivos específicos de cada sección presentada en esta memoria del TFM, con una descripción de los mismos y la lista de entregables que se han generado al finalizar cada una de las tareas.

B. **Competencias específicas adquiridas**: en este anexo se presentan las competencias adquiridas en este Trabajo de Fin de Máster en relación con los dos títulos de máster cursados: Máster Universitario en Ingeniería de Telecomunicación y Máster Universitario en Investigación e Innovación en las Tecnologías de la Información y las Comunicaciones.

C. **Justificación de méritos**: en este anexo se justifica el valor excepcional de este trabajo para poder obtener la máxima calificación en las asignaturas Trabajo de Fin de Máster de ambos másteres cursados.

1 Introducción

1.1. Motivación

Como ya se ha mencionado en el Preámbulo, en este Trabajo de Fin de Máster se ha desarrollado y evaluado la tecnología inicial de biometría facial de Argos Soluciones Globales S.L. (Argos Global, en adelante) con el fin de mejorarla y adaptarla a dispositivos móviles. Este trabajo también abarca el diseño y desarrollo de una herramienta informática que permita a Argos Global evaluar la fusión a nivel de puntuación de las tecnologías de verificación facial y de locutor. Además, diferentes sistemas *antispoofing* fueron diseñados y desarrollados para evitar suplantación ante el sensor para ambos sistemas biométricos.

Argos Global es una PYME española que apuesta por ofrecer soluciones biométricas de tecnología propia para control de accesos, tanto físicos como virtuales. Por eso, desde hace más de cuatro años comenzaron a desarrollar FaceOn, su porfolio de soluciones biométricas basadas en biometría facial.

El hecho de que Argos Global decidiera especializarse en tecnología biométrica facial se debe a que en su portfolio tradicional de productos cuenta con un sistema anti-hurto patentado para establecimientos donde se pudiera reconocer al ladrón usando una cámara: *Global Guard*. De ahí que, si el sistema pudiera reconocer al sujeto automáticamente usando biometría facial sería más sencillo alertar al encargado del establecimiento. Por esta razón decidió emprender y empezar a desarrollar tecnología propia de verificación facial.

La biometría facial es un tipo de biometría no intrusiva, de fácil uso, con alta precisión y una estabilidad media a largo plazo [8]. Además, para adquirir los rasgos faciales solo es necesario el uso de una cámara, la cual se encuentra actualmente en prácticamente todos los dispositivos móviles del mercado: *smartphones*, *tablets*, ordenadores portátiles, etcétera. Por estas razones, la biometría facial hace que sea tan idónea para una aplicación de control de acceso móvil.

La utilización de una fusión biométrica permite combinar información de distintos rasgos biométricos con el fin de mejorar la precisión y la robustez en la verificación de un usuario [9]. Argos Global apuesta por una solución que combine la verificación facial con biometría de voz, en particular reconocimiento de locutor, que haga mejorar la precisión en la verificación más que si solo se utilizara un tipo de biometría y que haga que la aplicación sea más robusta ante la suplantación de identidad.

El hecho de que se haya escogido desarrollar tecnología que utilice además verificación de locutor es porque también, en la inmensa mayoría de dispositivos multimedia móviles del mercado, está integrado un micrófono que permite la captación de voz, al contrario que en otras biometrías, como huella dactilar, en la que solo algunos dispositivos de alta gama tienen los sensores necesarios para la captación, o en iris en la que actualmente no existe ningún dispositivo comercial con la capacidad de extraer información del iris con la calidad suficiente para desarrollar un sistema biométrico preciso.

Debido a los problemas que encuentra la biometría ante la suplantación frente al sensor o *spoofing* Argos Global decidió comenzar a investigar en mecanismos que permitiesen distinguir si una cara ante el sensor es real o, en cambio, se tratase de un vídeo o una fotografía. Por ello, para el desarrollo de una aplicación de control de acceso con biometría facial y de voz es necesario incorporar sistemas *antispoofing* que permitan discriminar si frente a los sensores se encuentra una persona física o, por el contrario, algún tipo de imitación.

1.2. Objetivos y enfoque

Este Trabajo de Fin de Máster consiste en un conjunto de actividades y proyectos que están enfocados a realizar un *sistema de control de acceso móvil mediante biometría facial fusionada con voz*.

A continuación, se procede enunciar los objetivos específicos por apartados, correspondientes a las secciones de esta memoria. Hay que tener en cuenta que los objetivos aquí citados se encuentran dentro del alcance de este TFM, como se ha expuesto con anterioridad en el Preámbulo:

1. **Bases de datos**:
   - 1.1. Creación de una base de datos de imágenes de caras para probar algoritmos de biometría facial.
   - 1.2. Creación de una base de datos de voz para probar algoritmos de reconocimiento de locutor.
2. **Diseño y desarrollo de un sistema de evaluación de algoritmos de detección, modelado y verificación facial**:
   - 2.1. Desarrollar un sistema de evaluación de algoritmos de detección, modelado y verificación facial en C/C++.
   - 2.2. Investigar diferentes algoritmos de detección, modelado, verificación y fusión de características de cara con vistas a obtener mejores resultados en la verificación.
   - 2.3. Diseñar un sistema de *antispoofing* facial.
3. **Desarrollo de una demo comercial y diseño de la aplicación móvil FaceOnVox**:
   - 3.1. Desarrollar un sistema todo local y cliente servidor de una demo de VerifyByFace.
   - 3.2. Diseñar una API de verificación facial en C++, VerifyByFace.
   - 3.3. Diseñar una aplicación móvil, FaceOnVox.
4. **Diseño y desarrollo de un sistema de evaluación de una aplicación de voz y de un sistema de verificación fusionando de biometría de cara y de voz**:
   - 4.1. Desarrollar un sistema de evaluación de algoritmos de reconocimiento de locutor en C++.

4.2. Desarrollar un sistema de evaluación de algoritmos de fusión biométrica cara y voz a nivel de puntuación en C++.

En el *Anexo A: Objetivos específicos* se muestran estos objetivos desglosados por secciones de esta memoria, junto con una descripción de los mismos y los entregables que se han generado tras la ejecución de cada actividad.

1.3. Metodología y plan de trabajo

La metodología utilizada en las actividades que se han desarrollado en el contexto del TFM está supeditada a las necesidades y prioridades de la empresa Argos Soluciones Globales S.L. en cada momento, pero generalmente se ha seguido una estructura cíclica similar a la que el *Project Management Institute* define para los proyectos o las fases de proyecto [10], atendiendo a una metodología ágil.

La Figura 1.1 representa la metodología empleada para el desarrollo de los paquetes de trabajo o actividades.

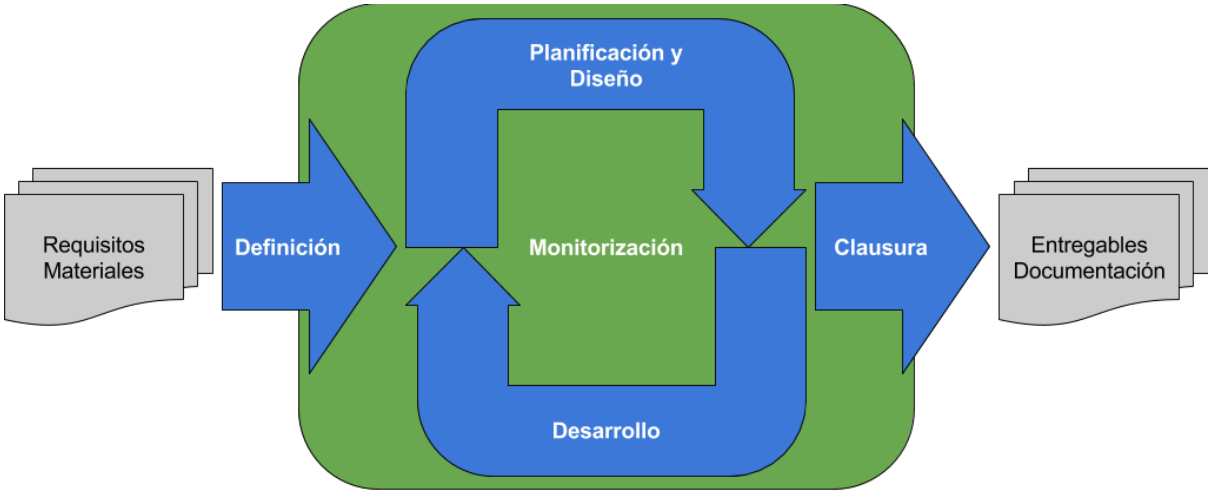

Figura 1.1: Metodología de desarrollo de un paquete de trabajo o actividad.

Utilizando esta metodología, se comienza la definición de la actividad o el paquete de trabajo a realizar a partir de los requisitos necesarios para el cumplimiento de la misma y de los materiales disponibles, sean estos adquiridos o implementados previamente en otras actividades.

Posteriormente, una vez definida la actividad, los entregables que se deben desarrollar y los requisitos que debe cumplir, comienza la fase de planificación y diseño de dicha actividad, seguida posteriormente de la fase propia de desarrollo. La fases de planificación y diseño y la de desarrollo pueden repetirse de forma cíclica debido a los cambios que, seguro, pueden aparecer durante la fase de desarrollo y/o monitorización, que deriven en replanificación, y ésta, a su vez, puede introducir de nuevo cambios en la fase de desarrollo de la actividad. La fase de planificación y diseño y la de desarrollo se ejecutan y en paralelo con la fase de monitorización que sirve, además de para controlar de que el trabajo se realiza siguiendo la definición propuesta, como interfaz entre el diseño y desarrollo de la actividad y cualquier agente externo a la misma.

Finalmente, una vez satisfechos todos los requisitos y generado todos los entregables, durante la fase de clausura se preparan los entregables para servir de entrada a otras actividades o paquetes de trabajo y se documenta el trabajo realizado.

Para el desarrollo de *software* complejo se planifica teniendo en cuenta el algoritmo «divide y vencerás», que consiste en dividir un problema complejo en subproblemas menos complejos y éstos a su vez en rutinas sencillas de implementar, cuyas salidas se combinan para resolver en su conjunto los subproblemas definidos y finalmente el problema complejo que se quería abordar.

También, para el desarrollo de programas que utilizan de la programación orientada a objetos, se ha utilizado una metodología aplicada a este paradigma, diseñando primero los objetos, sus atributos y sus métodos, y la relación entre éstos en el marco de la tarea que deben desempeñar de manera conjunta.

Al igual que la metodología, el plan de trabajo esta altamente influenciado por las necesidades y las prioridades de la empresa en cada momento del desarrollo de este TFM. Por eso, durante la ejecución del trabajo se definían paquetes de trabajo o actividades alineados con el objetivo empresarial y se desarrollaban en los márgenes temporales que se consideraban adecuados para el tipo de tarea que se desempeñaba y la importancia que tenía esta dentro del plan estratégico de la empresa.

1.4. Competencias transversales

Durante el tiempo de estancia en la empresa Argos Soluciones Globales S.L. se han adquirido y desarrollado las siguientes competencias transversales —aplicables por igual a las titulaciones de Máster Universitario en Ingeniería de Telecomunicación y Máster Universitario en Investigación e Innovación en las Tecnologías de la Información y las Comunicaciones— orientadas a un contexto empresarial y de innovación tecnológica. Además, en el *Anexo B: Competencias específicas adquiridas* se presentan las competencias específicas de cada titulación de máster adquiridas con la realización de este trabajo.

CT-1. Desenvolverse de forma activa y profesional en un entorno empresarial.

CT-2. Compresión del emprendimiento y la innovación en empresas de base tecnológica en España.

CT-3. Desenvolverse en un entorno de investigación e innovación tecnológica.

CT-4. Capacidad de comprender y presentar ideas abstractas y conocimiento científico o proyectos a público tanto especializado o como no especializado en un contexto internacional y multidisciplinar.

CT-5. Capacidad de trabajar en equipos físicos como distribuidos en proyectos tecnológicos de investigación en un contexto europeo multidisciplinar.

CT-6. Resolución de problemas complejos y abstractos utilizando una metodología estructurada y científica.

CT-7. Capacidad de tomar decisiones en el contexto empresarial, de hacerlo con un juicio crítico basado en conocimiento e información técnica y capacidad de defender dichas decisiones con autoconfianza y determinación, basándose en hechos objetivos.

CT-8. Capacidad de adaptación y flexibilidad a cambios en el desempeño laboral.

CT-9. Capacidad de adquirir y actualizar habilidades y destrezas de forma autónoma.

CT-10. Capacidad de planear el trabajo a realizar de forma autónoma, estimar tiempos de desempeño y cumplir con los plazos propuestos.

CT-11. Capacidad de comunicación en el entorno laboral, ya sea con los miembros dentro de la oficina como con clientes, socios, colaboradores y proveedores.

CT-12. Capacidad de crear un ambiente agradable de trabajo donde prevalezcan valores éticos y de responsabilidad social.

2
Estado del arte

En esta sección se enuncia el estado del arte de la Biometría, centrándose particularmente en las tecnologías de biometría facial y de voz, que son las relevantes en este Trabajo de Fin de Máster. Además se explicará qué es la fusión biométrica y a qué niveles se aplican las técnicas de fusión, así como diferentes tipos de estrategias utilizadas para evitar la suplantación de identidad ante el sensor o *spoofing*, orientadas a dichos rasgos biométricos.

Lo que se desarrolla a continuación no pretende abarcar todas las tecnologías, métodos, técnicas y aplicaciones de la Biometría, sino solo aquellos que directamente se relacionen de uno u otro modo con este trabajo. Además, se indicará explícitamente cuáles de las técnicas aquí presentadas han sido empleadas.

2.1. Introducción a la Biometría

El término Biometría se deriva de las palabras griegas *bio* que significa «vida» y *metric* que significa «medible». Según [11], la Biometría se define como la ciencia que establece la identidad de un individuo basándose en los atributos tanto físicos, como químicos y de comportamiento de una persona. Un rasgo biométrico es una característica humana que se puede medir con el fin de conocer la identidad del usuario.

Existen tres métodos principales para conocer la identidad de un usuario:

- **Basada en conocimiento**: Algo que el usuario *sabe* o *conoce*, como puede ser una contraseña o un código PIN. El problema que tiene este tipo de técnicas es que el usuario puede olvidar la contraseña con facilidad o, peor aún, establecer una de baja seguridad con el fin de acordarse más fácilmente.
- **Basada en *tokens***: Algo que el usuario *posee* o *tiene*, como puede ser una tarjeta identificativa electrónica o de crédito. Este tipo de técnicas tienen el problema de que pueden extraviarse o ser robadas, perdiendo así el usuario su identidad.
- **Basada en biometría**: Algo que el usuario *es* o *produce*, como puede ser su huella dactilar, o su propia voz. Este tipo de técnicas tienen en la actualidad un elevado coste computacional y, en determinadas aplicaciones, requieren garantizar una alta precisión. En ocasiones pueden ser técnicas intrusivas o con baja aceptación por parte del usuario.

La autentificación más robusta es una combinación del conjunto de las tres técnicas anteriormente descritas.

2.1.1. Propiedades y características

Típicamente se comparan los rasgos biométricos en función de sus propiedades —inherentes al tipo de rasgo biométrico— y las características— dependen del nivel de eficacia de las tecnologías desarrolladas para el tipo de rasgo biométrico concreto—.

Las propiedades de los rasgos biométricos dependen del tipo de rasgo biométrico concreto y son las siguientes cuatro, a saber, universalidad, unicidad, estabilidad y recolección:

- **Universalidad (*universality*)**: toda persona ha de poseer dicho rasgo biométrico. En la práctica, por determinados motivos no siempre es posible, por lo que este rasgo debe poseerlo prácticamente todo el mundo (entrono al 99 % de la población debe tener dicho rasgo). Determinados factores sociales y ambientales pueden afectar en determinadas poblaciones, como es el caso de la huella dactilar para trabajadores que hagan un uso excesivo de trabajos manuales o estén en contacto directo con químicos, pueden tener la huella dactilar deteriorada para utilizar este rasgo biométrico como un método identificativo.
- **Unicidad (*uniqueness*)**: dos individuos diferentes no pueden poseer el mismo rasgo biométrico. En ocasiones, por motivos genotípicos o fenotípicos se dan casos en los que se puede encontrar el mismo rasgo biométrico en diferentes individuos, no necesariamente emparentados, como puede ser el caso de dos personas que se parecen mucho en sus rasgos faciales: sosias o *doppelgangers*.
- **Permanencia (*permanence*)**: el rasgo biométrico ha de ser invariante en el tiempo, es decir, que no se vea degradado por los años, o por lo menos que tenga alta estabilidad a largo plazo. Factores externos pueden deteriorar el rasgo biométrico, como puede ser el consumo del tabaco en la voz de los fumadores, que la va deteriorando paulatinamente.
- **Recolección (*collectability*)**: el rasgo, a la hora de recogerlo, ha de ser medido de forma cuantitativa. También esta propiedad tiene que ver con lo fácil que es adquirir una medición de un determinado rasgo biométrico. Determinados rasgos biométricos pueden ser difíciles de recoger, como el caso del ADN u otros rasgos químicos corporales, donde se requiere una extracción invasiva que puede provocar que determinados usuarios no estén dispuestos a acceder a ella.

Otras características aparecen a la hora de trabajar con determinados rasgos biométricos en función del desarrollo de la tecnologías relacionadas con la adquisición y el procesamiento de los rasgos, como son la precisión, el rendimiento, la usabilidad, la aceptación y la elusión:

- **Precisión (*accuracy*)**: cómo de precisos son los sistemas y las tecnologías que trabajan con este tipo de rasgo en concreto, es decir, cómo de bien se identifican a los usuarios en una aplicación. Los métodos de evaluación de los sistemas biométricos se explicarán en el siguiente apartado.
- **Rendimiento (*performance*)**: cómo de potentes y cómo de bien funcionan los sistemas en función del tipo de rasgo. Algunos requerirán procesos y características más complejas que otros.
- **Usabilidad (*usability*)**: cómo de fáciles de utilizar son estas tecnologías por parte del usuario. Tiene relación con los factores socioculturales y con las aplicaciones de esta tecnología.

- **Aceptación (*user acceptance*)**: cómo está de acuerdo un usuario de que se le adquiera un rasgo biométrico concreto. Tiene que ver mucho con aspectos socioculturales y legislativos, y con lo intrusivas que son las técnicas de adquisición de un determinado rasgo biométrico.

- **Elusión (*circumvention*)**: cómo de fácil es eludir o burlar a un sistema que utilice determinado rasgo biométrico. Tiene que ver con la facilidad que tienen los usuarios impostores para ser aceptados por el sistema y la facilidad que puede existir para falsificar el rasgo.

Algunos tipos de biometría comúnmente utilizados son la huella dactilar, la retina, el iris, la geometría de la mano, el ADN, la voz, la cara, la firma, dinámicas de ratón y teclado, etcétera. En concreto en la Tabla 2.I se muestran en una escala cualitativa las propiedades y características de los dos rasgos biométricos utilizados en este Trabajo de Fin de Máster: la cara y la voz [8, 12, 13].

Tabla 2.I: Características de la biometría facial y la de voz.

Biometría	**Universalidad**	**Unicidad**	**Estabilidad**	**Recolección**
Facial	Alta	Alta	Media	Alta
Voz	Media	Baja	Baja	Media
Precisión	**Rendimiento**	**Usabilidad**	**Aceptación**	**Elusión**
Alta[1]	Baja	Media[1]	Alta	Alta
Alta[1]	Baja	Alta[1]	Alta	Alta

La Tabla 2.I se ha rellenado con los datos extraídos de dos estudios publicados en 2011 [8] y 2014 [13], debido a que no se encontraban todas las características tecnológicas —precisión, rendimiento, usabilidad, aceptación y elusión— en un mismo estudio. Cabe destacar que las tecnologías de reconocimiento facial y de locutor han evolucionado desde la publicación de estos estudios comparativos en diferentes factores, siendo estas más potentes, precisas y difíciles de eludir; además de que la biometría está cada vez más integrada en la sociedad y la aceptación y usabilidad por parte de los usuarios es cada vez mayor. También se puede observar que en determinadas características tecnológicas varían los datos presentados en función del autor del estudio, como por ejemplo, en [8] exponen que la elusión para el rasgo facial es baja, mientras que en [13] indica que es alta.

Lo interesante de la Tabla 2.I respecto al TFM realizado es, sobre todo, la alta precisión y aceptación por parte de los usuarios de las tecnologías de cara y voz, y de que son rasgos altamente universales y fáciles de recoger, en especial con los sensores actualmente disponibles en los dispositivos móviles actuales.

2.1.2. Verificación biométrica

Es preciso incidir, sobre todo en este contexto, en las diferencias que existen entre lo que se refiere a verificación biométrica e identificación biométrica:

- **Verificación** o *autentificación* biométrica responde a la pregunta «¿Soy quien digo ser?». La respuesta del sistema puede ser «sí» o «no», y en ocasiones esta respuesta puede venir acompañada de cierto nivel medible de confianza (como puede ser un porcentaje o una puntuación). En verificación biométrica se enfrenta un modelo de autentificación contra otro modelo previamente enrolado en el sistema. Requiere pues, como mínimo, una comparación de modelos, como se verá más adelante.

[1]Estos datos han sido extraídos de [8]. El resto de datos se han extraído de [13].

- **Identificación** biométrica —en ocasiones también llamada *reconocimiento* biométrico— responde a la pregunta «¿Quién soy?». La respuesta del sistema es una identidad o un *ranking* de identidades, ordenadas por nivel de confianza. La identificación enfrenta un modelo de identificación contra n modelos previamente enrolados en el sistema. Este proceso requiere comparar contra todos los usuarios enrolados en el sistema si no se ha aplicado anteriormente un filtrado previo.

En ocasiones se le denomina *reconocimiento* biométrico al conjunto de técnicas que permiten comparar dos modelos de usuarios: el llamado modelo de *enrolamiento* —previamente incorporado o enrolado al sistema— contra el modelo del usuario que se desea *verificar* o *identificar*, independientemente del número de usuarios enrolados contra los que se realice la comparación de modelos. Por eso mismo, es preciso dejar claro que en este TFM solamente se trabaja técnicas de *verificación* biométrica, aunque en ocasiones se haga alusión en esta memoria a las técnicas utilizadas cómo técnicas de *reconocimiento* biométrico.

En la verificación biométrica se distinguen dos etapas:

1. **Enrolamiento**, *registro* o *entrenamiento*, es la etapa en la que el usuario accede por primera vez al sistema y completa los datos de su perfil con un modelo biométrico. Este paso es fundamental y necesario para poder realizar la todo el proceso de verificación.
2. **Verificación**, *acceso* o *autentificación*, es la etapa en la que el usuario desea autentificarse como quien dice ser, haciendo uso del mismo rasgo biométrico utilizado en la etapa de enrolamiento.

La etapa de enrolamiento debe realizarse al menos una vez, y dependiendo del sistema implementado pueden crearse uno o diferentes modelos de enrolamiento que contemplen variaciones ambientales, de entorno o temporales, haciendo así más robusta la verificación. Es importante que los modelos de entrenamiento se generen siempre en las mejores condiciones posibles (dentro de que estas pueden ser distintas) para conseguir los mejores resultados en la etapa de verificación. El enrolamiento se debe realizar, en la medida de lo posible, bajo condiciones de seguridad, para evitar que una persona con ánimo de intentar suplantar al usuario enrolado genere un modelo de enrolamiento falso, y pueda así suplantarle en la etapa de verificación o corromper su modelo de enrolamiento, no permitiendo que el usuario genuino pueda acceder al sistema.

La etapa de verificación puede realizarse indefinidas veces. Para esta etapa es necesario proveer al sistema de una identidad contra la que verificarse y un modelo biométrico de la persona que se quiere verificar. Según el sistema implementado puede comparase el modelo de verificación con uno o varios modelos de enrolamiento, pero siempre del mismo usuario del que se provee la identidad. Es necesario haber generado previamente un modelo de enrolamiento para poder compararse con éste en la etapa de verificación.

Se puede entiende la verificación como un sistema de clasificación con dos clases: usuarios genuinos e impostores. Los usuarios genuinos son aquellos que acceden al sistema con intención de autentificarse como ellos mismos y los impostores como usuarios que intentan autentificarse como otro usuario distinto a sí mismo.

A la hora de comparar dos modelos se genera una puntuación o *score* que mide el grado de verosimilitud que existe entre el modelo de enrolamiento y el modelo de verificación. Generalmente esta puntuación es tanto más alta cuanto más parecidos sean los dos modelos entre sí. A la hora de tomar una decisión es necesario fijar un umbral donde si la puntuación obtenida al comparar dos modelos es más alta que el umbral, el usuario es aceptado, y si ésta es más baja, el usuario es rechazado. Posteriormente se explicará el compromiso que existe a la hora de determinar el valor del umbral de decisión.

Visto esto, se pueden presentar cuatro situaciones posibles a la hora de realizar una verificación biométrica en función de cuál es la clase del usuario —genuino o impostor— y de la decisión tomada por el sistema verificador dado un umbral de decisión —aceptación o rechazo—, como se muestra en la Tabla 2.II.

Tabla 2.II: Situaciones posibles en la verificación biométrica.

	Aceptación	**Rechazo**
Genuino	**Verdadero Positivo**	**Falso Negativo**
Impostor	**Falso Positivo**	**Verdadero Negativo**

- **Verdadero positivo**: esta situación se produce cuando un usuario genuino se quiere autenticar como sí mismo y el sistema determina que sí que es quien dice ser, aceptándolo. Es una situación deseable, pues obedece al propósito del sistema.
- **Falso negativo**: también llamada *falso rechazo*. Esta situación se da cuando un usuario genuino se quiere autenticar como él mismo y el sistema le rechaza. Es una situación que hay que evitar, pues puede causar insatisfacción por parte del usuario al intentar verificarse correctamente y no poder hacerlo.
- **Verdadero negativo**: esta situación aparece cuando un usuario impostor o no genuino desea autenticarse como alguien que no es y el sistema le rechaza. Esta situación también es deseable, pues evita que se cometa una suplantación de identidad.
- **Falso positivo**: también llamada *falsa aceptación*. Esta situación se da cuando cuando un usuario impostor se desea autenticar como otro usuario que no es él mismo y el sistema le da acceso. Es una situación indeseable que además puede acarrear problemas de seguridad.

Normalmente se trabaja con la tasa de verdaderos positivos —número de usuarios genuinos aceptados entre el total de usuarios genuinos—, la tasa de falso rechazo —número de usuarios genuinos clasificados como rechazados entre el total de usuarios genuinos—, la tasa de verdaderos negativos —número de usuarios impostores rechazados entre el total de usuarios impostores— y la tasa de falsa aceptación —número de usuarios impostores aceptados por el sistema entre el total de usuarios impostores—.

Un sistema de verificación biométrica funciona correctamente cuando posee una alta tasa de verdaderos positivos y verdaderos negativos. Lo ideal es tener un sistema cuya tasa de falsos negativos y positivos sea prácticamente nula y que, al mismo tiempo, el sistema generalice bien en diversas condiciones ambientales y tenga una complejidad sencilla o moderada. Es aquí donde se encuentra el reto de los sistemas de verificación biométrica ya que si el sistema está bien diseñado se establece un compromiso entre generalización y precisión, por lo que existirá tasa de falso rechazo y tasa de falsa aceptación, por pequeña que sea.

Como se ha mencionado anteriormente, es necesario fijar un umbral de decisión para poder, de hecho, hacer que el sistema tome la decisión de aceptar o rechazar a los usuarios, y así obtener el número de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos a partir de los datos de una base de datos utilizada para evaluar el sistema.

En la Figura 2.1 se muestra un ejemplo de lo que podría ser un sistema de verificación. Se pueden observar dos curvas normalizadas de puntuaciones que reflejan en rojo la distribución de las puntuaciones de usuarios impostores usando un sistema de verificación biométrico, y en azul la distribución normalizada de puntuaciones de usuarios genuinos, verificados contra ellos mismos. También se puede apreciar en verde el umbral de decisión que se ha determinado para

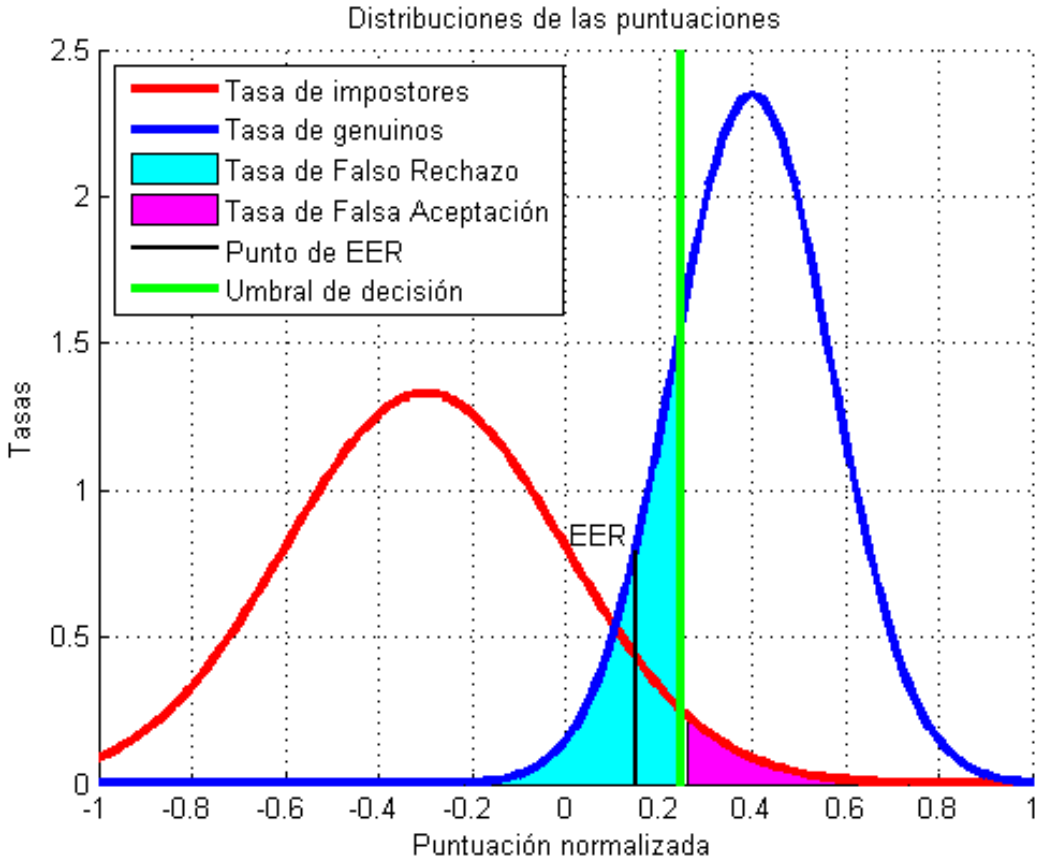

Figura 2.1: Distribuciones de puntuaciones de genuino e impostor.

una aplicación determinada, delimitando dos áreas de especial interés, que corresponden con la tasa de falso rechazo en color aciano y la tasa de falsa aceptación en magenta.

Un punto de especial interés en la Figura 2.1 es el que aparece indicado en negro como «EER», siglas de *Equal Error Rate* o *Tasa de Igual Error* que es donde las tasas de falso rechazo y falso positivo son iguales —si se desplaza el umbral (en verde) a donde se encuentra la línea vertical negra se obtiene que las áreas que representan la tasa de falso rechazo (cian) y tasa de falsa aceptación (magenta) son iguales entre sí, y a la tasa de igual error—. Este punto se suele utilizar para medir la eficiencia de los sistemas clasificadores siendo éste mejor cuanto menor valor tenga este punto, es decir, cuando el punto en el que las áreas sean iguales esté más bajo.

Se pueden encontrar otros puntos de trabajo de interés, como los llamados FNMR, siglas de *False No-Match Rate* o *Tasa de Falsos Rechazados* o FMR, siglas de *False Match Rate* o *Tasa de Falsos Aceptados*. Ésta última medida se suele utilizar para conocer cuál es la tasa de falso rechazo para valores de FMR del 1 %; 0,1 %; 0,001 % o 0 % generalmente (usando esta medida es sencillo obtener el umbral de decisión correspondiente dada una determinada tasa de FMR); y de igual modo se puede evaluar diferentes valores de FNMR, conocer la tasa de falsa aceptación dado un valor FNMR y el correspondiente umbral de decisión asociado. Interesará utilizar más una u otra medida en función del tipo de aplicación.

La decisión de dónde colocar el umbral depende del tipo de aplicación con la que se quiera trabajar: si es una aplicación de seguridad, hay que reducir al mínimo posible la tasa de falsa aceptación, por lo que se subirá el umbral de decisión, mientras que si la aplicación se quiere que

sea amigable o *user-friendly* se desea que la tasa de falso rechazo sea la menor posible, bajando el umbral de decisión. Por lo tanto, en aplicaciones de seguridad interesa decidir el umbral y evaluar la tasa de falsa aceptación para valores de FMR bajos, mientras que en aplicaciones amigables interesa trabajar en valores de FNMR bajos.

En el contexto de este Trabajo de Fin de Máster la aplicación resultante es una aplicación de tipo seguro, luego se evaluará la bondad del clasificadores utilizando medidas del tipo FMR y con umbrales superiores al asociado al EER.

2.2. Biometría facial

La biometría facial es aquella que utiliza la información que se encuentra en el rostro del individuo con el fin de identificarlo de manera automática [11]. El reconocimiento basado en la información de la cara es un método tradicional de identificación de personas y es fácil encontrar una fotografía facial en numerosos documentos administrativos como el DNI o pasaporte, con el fin de que la autoridad pertinente verifique que el portador de dicho documento es quien dice ser.

La biometría facial lleva desarrollándose desde finales del siglo XX, pero hasta principios del siglo XXI no se ha explotado en todo su potencial debido a las limitaciones de las computadoras para poder procesar con facilidad y en poco tiempo imágenes con la resolución suficiente para acometer la tarea de eficientemente.

La biometría facial tiene dos principales vías de investigación: el reconocimiento facial 2D, que utiliza imágenes o secuencias de imágenes en blanco y negro o en color para reconocer al individuo; y, con la incorporación de cámaras 3D, el reconocimiento facial 3D, que además de captar imágenes o secuencias de imágenes con información de color añade información de profundidad. Estos sensores 3D pronto aparecerán de manera integrada en el mercado de los dispositivos móviles de manera masiva[1], con el ánimo de poder explotar la información de profundidad en imágenes para el desarrollo de diferentes aplicaciones de ocio, seguridad y accesibilidad.

La variación intraclase —aquella que ocurre al tomar diferentes muestras (imágenes 2D o 3D, o vídeos de la cara) de un mismo sujeto en diferentes condiciones o momentos— es el principal reto al que se enfrentan los algoritmos de reconocimiento facial. Algunos ejemplos de variaciones que se pueden encontrar a la hora de realizar el reconocimiento utilizando biometría facial, y que dificultan esta tarea son:

- **Expresión**: cambio en la forma de algún elemento de la cara debido a que se realice un gesto o expresión facial, como pueden ser los ojos (cerrados o abiertos) o la boca (cerrada, abierta, sonrisa, mueca...), gestos con las cejas, etcétera.
- **Capilares**: cambio tanto en el peinado del sujeto como en el vello facial.
- **Oclusiones**: algunas partes de la cara están cubiertas o parcialmente cubiertas utilizando gafas graduadas, gafas de sol, bufandas, sombreros, etcétera.
- **Pose**: el ángulo con el que se ha tomado la imagen de la cara presenta variación que incluso puede llegar a ocluir ciertos elementos de la cara, como puede suceder con las fotos tomadas de perfil.
- **Iluminación y calidad**: la presencia de sombras o la falta de iluminación en la escena, así como el uso de cámaras que distorsionan (ojo de pez), con alta granularidad, ruido o

[1] Ver Intel RealSense (Consultado 01/2017).

baja resolución (imágenes de caras por debajo de 60 píxeles de ancho o largo) es un reto para nada despreciable cuando se emplean algoritmos de reconocimiento en imágenes.

- **Distancia**: otro reto al que se enfrenta el reconocimiento facial es diseñar sistemas que sean capaces de identificar personas por su cara en imágenes con baja resolución o baja calidad [14].
- **Edad**: la variación de la edad afecta en el reconocimiento facial de manera muy notoria en la época desde temprana edad hasta llegar a la juventud y a partir de la tercera edad, siendo más o menos estable durante la edad adulta del individuo.
- **Artificiales**: cambios en la cara de un individuo debido al el uso de maquillaje, cirugía estética, tatuajes en la cara, *piercings* o dilataciones en nariz y labios.
- **Genéticas y fenotípicas**: en ocasiones determinadas etnias o diferenciar a dos gemelos con biometría facial puede convertirse en un reto debido a la poca variabilidad interclase —entre distintos sujetos—. También ocurre con individuos físicamente parecidos aunque no compartan lazos familiares, como los denominados sosias o *doppelgangers* (ver el ejemplo mostrado en la Figura 2.2, donde a la izquierda está el actor y humorista Will Farrell y a la derecha el músico y batería Chad Smith).

Figura 2.2: Ejemplo de *doppelgangers*: Will Farrell (izquierda) y Chad Smith (derecha) en el programa de televisión *The Tonight Show Starring Jimmy Fallon.*

El esquema tradicional de verificación facial se ilustra en la Figura 2.3, y es el que ha sido empleado en la realización de este TFM. Estas fases del esquema tradicional de verificación facial pueden variar si bien el propio sistema o bien las tecnologías utilizadas lo requieren. Como se expuso en la *Sección 2.1: Introducción a la Biometría*, la verificación facial —caso particular de la verificación biométrica— consta de dos etapas, una de enrolamiento, donde el usuario crea un perfil biométrico con un modelo de enrolamiento y otra etapa de verificación, donde el usuario, además de generar un modelo de verificación, se compara con el modelo de enrolamiento previamente creado y el sistema decide si el usuario es aceptado o rechazado.

En el ejemplo de la Figura 2.3 se puede ver que Alicia (sujeto femenino) crea un nuevo perfil biométrico con su nombre y con un modelo facial. Posteriormente se verifica y al tratarse de la misma persona el sistema la acepta. En el caso en el que Benito (sujeto masculino) quisiera

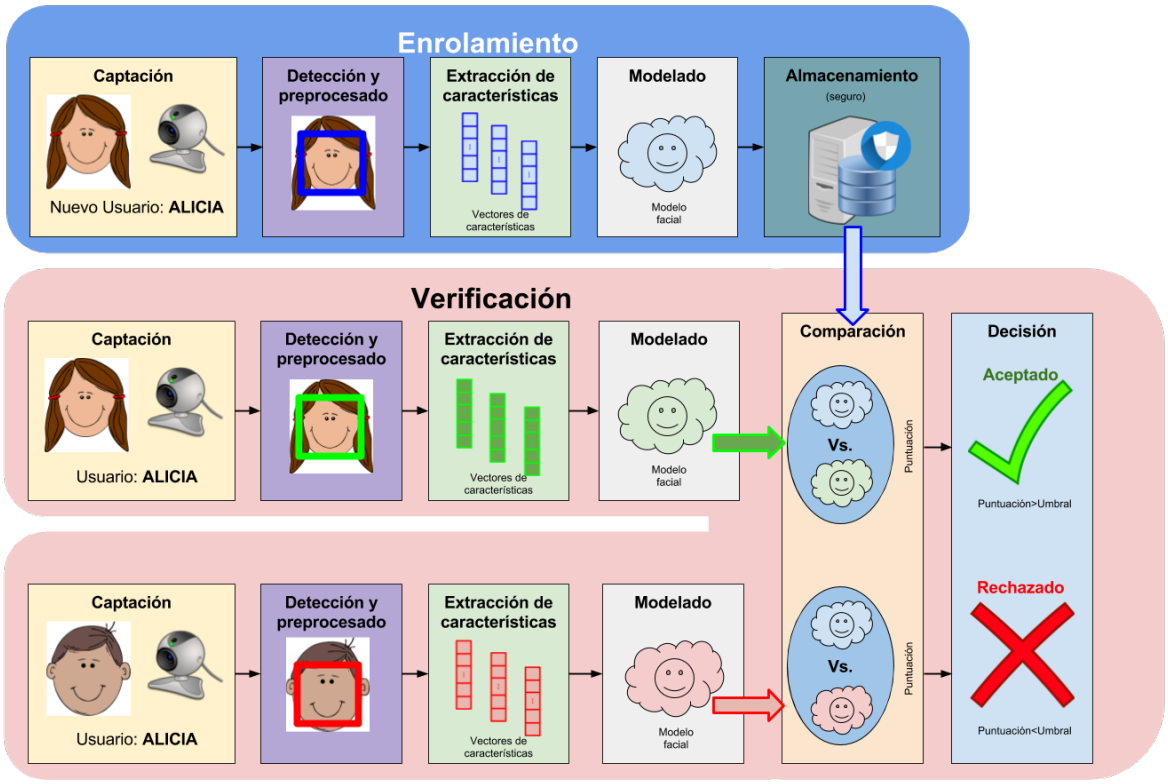

Figura 2.3: Esquema tradicional de verificación facial.

verificarse como Alicia, el sistema lo rechazaría al ser los modelos faciales diferentes (el de enrolamiento de Alicia y el de verificación de Benito). En esta Figura 2.3 se observa que las fases de Captación, Detección y preprocesado, Extracción de características y Modelado son las mismas tanto para la etapa de enrolamiento como para verificación. En la etapa de enrolamiento aparece la fase de Almacenamiento, que guarda el modelo de enrolamiento para poder ser usado en la fase de Comparación, exclusiva de la etapa de verificación, que utiliza ambos modelos, de enrolamiento y de verificación, para posteriormente tomar una Decisión de si el usuario es aceptado o rechazado, en función de las puntuaciones de la fase de Comparación y de un umbral prefijado.

1. **Captación**: se adquiere una muestra facial —sea una imagen en blanco y negro o color, una secuencia de imágenes, o imágenes 3D— usando una cámara y se solicita una identidad, como puede ser un nombre de usuario —excepto en sistemas para un solo usuario, que se asume que la identidad corresponde al propietario—. En la etapa de enrolamiento se asocia la identidad con el modelo de enrolamiento en la fase de Almacenamiento, mientras que en la etapa de verificación se provee una identidad —igualmente no requerida en sistemas de verificación de un solo usuario, ya que se asume— para que durante la fase de Comparación se compare el modelo de verificación computado en esta etapa con el modelo de enrolamiento asociado a dicha identidad.

2. **Detección y preprocesado**: se busca si hay una subimagen de una cara en la imagen o imágenes captadas. En el caso en el que se detecte más de una cara en la imagen captada, se requerirá de un criterio para elegir cuál de ellas es la que se va a utilizar (generalmente se utiliza la de mayor tamaño). En el caso en el que no se detecte ninguna o la que se capture no cumpla determinados criterios de diseño (tamaño, calidad, posición más

o menos fija entre diferentes *frames* de una secuencia de imágenes...) se esperará hasta detectar alguna que satisfaga dichos criterios para continuar con la siguiente fase. Una vez se haya detectado una cara en la imagen o imágenes capturadas se puede realizar diferentes tratamientos de la imagen, desde realce, transformaciones de espacios de color, correcciones, incluidas rotaciones o reconstrucciones, recortes y escalado.

3. **Extracción de características**: tradicionalmente se desean encontrar vectores de características que sean bastante similares para el mismo usuario —como se habló anteriormente, reduciendo la variabilidad intraclase— y lo suficientemente discriminatorias entre distintos usuarios —con alta variablilidad interclase—.

4. **Modelado**: se modelan las características extraídas a partir de la imagen o imágenes del usuario. Los algoritmos utilizados para generar un modelo varían según la tecnología que se desee emplear.

5. **Almacenamiento**: el modelo de enrolamiento se almacena, generalmente de forma segura y/o cifrada, para que pueda ser utilizado en la fase de Comparación de modelos. Organismos como FIDO[2] recomiendan que este modelo se almacene de forma local y trabajan para que los modelos biométricos se puedan guardar en sitios tan seguros como puede ser la tarjeta SIM de los teléfonos móviles.

6. **Comparación**: se compara el modelo de verificación con el de enrolamiento una vez provista una identidad. Al igual que los algoritmos de detección y modelado, los algoritmos de comparación pueden ser distintos según la tecnología que decida utilizar. Tradicionalmente, estos algoritmos se basan en medidas de puntuaciones o *scores*, que tienen un valor tanto más alto cuanto más parecidos son estos modelos. Generalmente se suelen utilizar normalizaciones sobre estas métricas o aplicar transformaciones a otros espacios con mayor resolución (como puede ser el espacio logaritmo).

7. **Decisión**: según la puntuación determinada en la fase de Comparación y un umbral que se determina a la hora de diseñar el sistema, el usuario que se verifica es aceptado si la puntuación supera el umbral y rechazado en el caso contrario. En algunos sistemas se consideran otros factores para tomar la decisión, utilizando información de la calidad de la imagen adquirida, añadiendo un umbral de incertidumbre superior al de decisión que pueda provocar que el usuario tenga que volver a verificarse, o usando una confianza provista por otro sistema, como puede ser un sistema *antispoofing* y/u otro sistema de reconocimiento facial diferente.

A continuación se citan algunos algoritmos clásicos en la detección, el modelado y la comparación de modelos faciales que constituyen el estado del arte en el área del reconocimiento facial. Los algoritmos a continuación presentados están diseñados para trabajar con imágenes o secuencias de imágenes 2D, a no ser que se diga lo contrario.

2.2.1. Detección de caras en imágenes

Como ya se mencionó con anterioridad, la detección facial consiste en encontrar rostros en imágenes donde puede o no haberlos.

Uno de los algoritmos de detección de caras más extendidos y utilizados es el detector facial Viola-Jones [15], basado en características Haar, que son una familia de *wavelets* las cuales se escogen entrenando dichos filtros con imágenes de caras y escogiendo aquellos que mejor detecten

[2]Ver `https://fidoalliance.org/` (Consultado 04/2017).

imágenes de caras. Este algoritmo recorre la imagen con los descritos filtros Haar de diferentes formas y tamaños, para encontrar subimágenes de caras en posiciones más o menos alejadas de la cámara. La eficacia del algoritmo Viola-Jones es que detecta caras a gran velocidad, ya que utiliza los filtros Haar en cascada empleando el algoritmo AdaBoost; es decir, se hace una primera pasada donde se encuentran más subimágenes de caras que las que en verdad existen y progresivamente se van descartando aquellas que sean rechazadas por los subsiguientes filtros más detallados.

En la Figura 2.4 se muestran diferentes filtros Haar, que recorren las imágenes faciales obteniendo un valor máximo en un punto que es en el cual en la imagen aparece el filtro superpuesto a la cara.

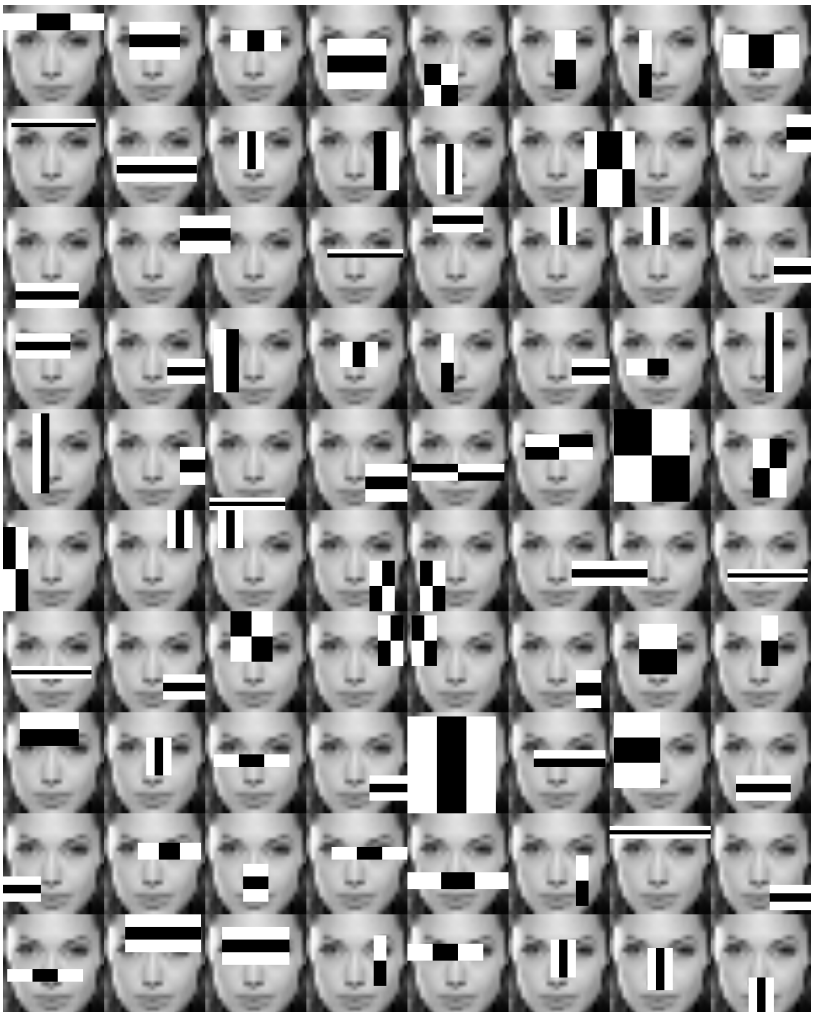

Figura 2.4: Filtros Haar para detección facial [1].

Se pueden entrenar también filtros Haar para encontrar distintos elementos de la cara, los cuales se pueden utilizar de forma conjunta con los filtros entrenados para detectar caras con el fin de ganar robustez a la hora de comprobar que realmente una cara detectada en una imagen contiene los diferentes elementos que la componen, como pueden ser ojos, nariz, labios, cejas, etcétera.

El algoritmo de Viola-Jones funciona bien en imágenes donde las caras se encuentran con pose frontal, sin mucha rotación en el plano de la imagen, haciendo este algoritmo más rápido y eficiente si se tienen conocimientos previos de la imagen donde se quiere encontrar caras: cuántas caras se desean detectar, posición de dónde se podrían encontrar, si existe un ángulo de rotación

de la cara más frecuente que otros, etcétera. El resultado de este algoritmo es un *bounding box*, es decir, un rectángulo que recuadra la cara y que tiene cuatro atributos: las posiciones x e y del píxel de arriba a la izquierda del rectángulo y el ancho y alto del rectángulo en número de píxeles.

Con el desarrollo de algoritmos de Aprendizaje Profundo o *Deep Learning*, una forma más costosa —principalmente a la hora de entrenar el sistema— pero más eficaz de detectar caras en una imagen es utilizando redes neuronales convolucionales multicapa [2], las cuales solucionan algunos problemas en la detección que podía tener el algoritmo de Viola-Jones, como puede ser rotaciones, cambios de poses y oclusiones, a costa de una elevada cantidad de datos de entrenamiento para la red.

La Figura 2.5 muestra un ejemplo de cómo funcionaría una red neuronal convolucional para la detección de caras en una imagen. La red se entrena utilizando imágenes de caras, que es lo que se quiere detectar, y cara capa de la red profunda aprende características de diferentes niveles. Los primeros niveles aprenden características de más bajo nivel como esquinas, bordes o puntos, y los últimos niveles aprenden estructuras más complejas, como caras completas o parciales.

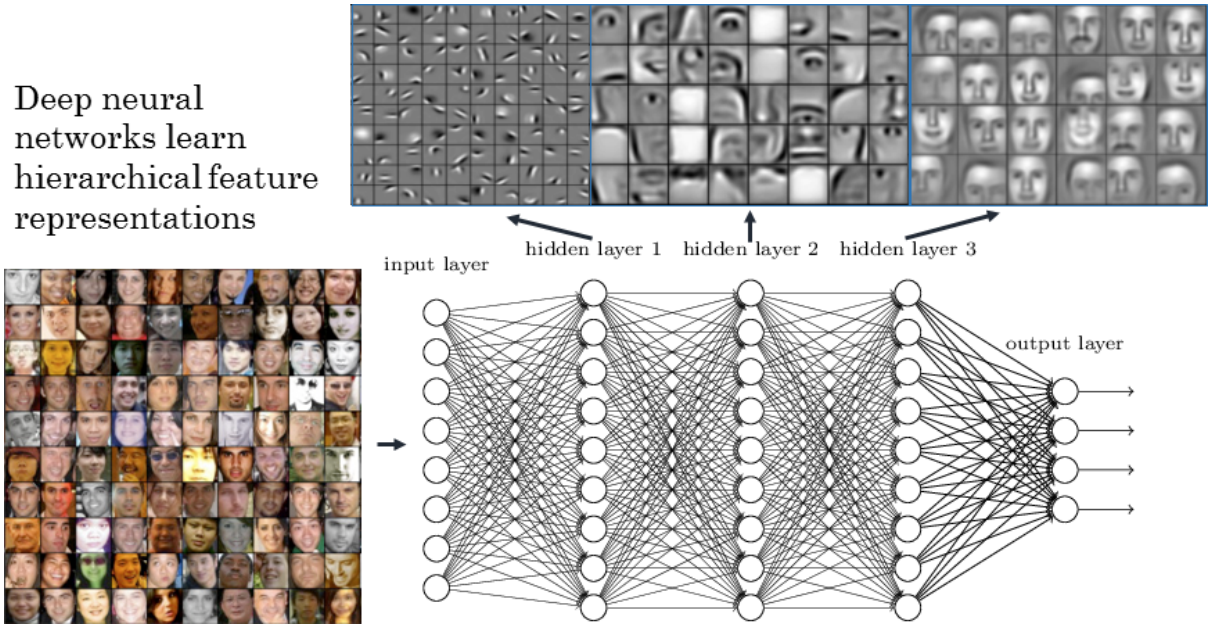

Figura 2.5: Red neuronal convolucional profunda para la detección de caras en imágenes (`http://www.rsipvision.com`).

El resultado de aplicar Aprendizaje Profundo a una imagen para reconocer caras es el que se presenta en la Figura 2.6. A la izquierda se ve la imagen que se quiere evaluar, donde aparecen los *bounding boxes* recuadrando las caras detectada. A la derecha de la imagen se ve una imagen que muestra el grado de confianza que tiene la red para asegurar que en ese subespacio de la imagen hay una cara.

Si se quiere realizar una detección 3D, generalmente se suelen utilizan algoritmos de detección 2D y se aplica el *bounding box* resultante sobre el plano de profundidad para obtener la imagen de cara en 3D, aunque también podrían entrenarse distintos filtros o redes que aprendan a detectar subimágenes de caras utilizando imágenes de profundidad.

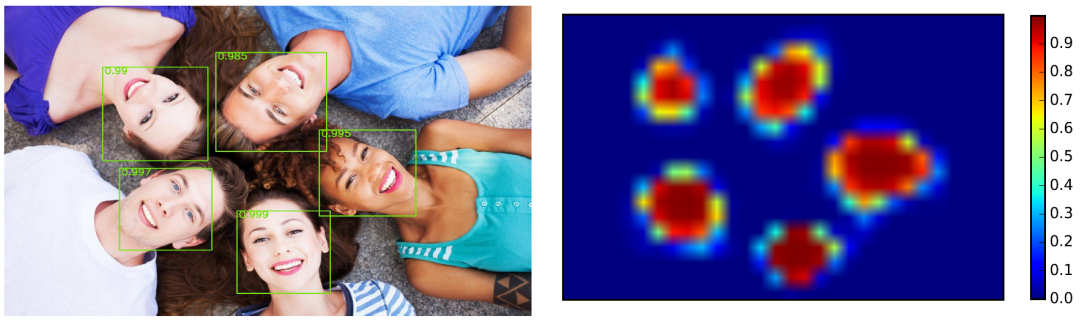

Figura 2.6: Detección de caras en imágenes con Aprendizaje Profundo [2].

2.2.2. Características y modelado facial

Según [16] se pueden clasificar los algoritmos de reconocimiento facial en tres generaciones, en función del tipo de características para modelar seleccionadas, su complejidad y cuándo se empezaron a utilizar.

Primera generación

La primera generación nace entorno a los años 90, y utiliza métodos de modelado básicos a partir de modelos generativos con características simples y lineales, o bien descriptores de imágenes. Requiere conjuntos de datos de entrenamiento del orden de millares de imágenes de caras para que el sistema pueda funcionar correctamente, y una capacidad de procesamiento de M-FLOPS[3]. La precisión de estos algoritmos típicamente alcanza valores de EER entre el 10 y el 20 % [3, 4, 17, 18]. La ventaja de utilizar este tipo de características en la actualidad es que tienen baja dimensionalidad —ocupan poco espacio a la hora de almacenar los modelos—, no son computacionalmente costosas de adquirir y existen publicadas numerosas investigaciones a partir de estos métodos clásicos que mejoran la precisión en la clasificación ya que han sido estudiados durante bastante tiempo.

Las autocaras o *Eigenfaces* [17] consiste en aplicar *Principal Component Analysis* (PCA) —Karl Pearson (1901)— para obtener aquellas características que son las más discriminativas. Este método es un modelo generativo, es decir, cualquier imagen de una cara puede reconstruirse como una combinación lineal de diferentes autocaras, por eso también puede utilizarse como un método de codificación. El espacio de las autocaras tiene una dimensionalidad equivalente al número de píxeles de la imagen —en una imagen de 100x100 píxeles se podrá obtener 10.000 autocaras— pero no todas tienen una información igual de importante; por eso, usando PCA se pueden obtener los autovectores que generen aquellas autocaras que sean las más discriminativas y aporten mayor información. En la Figura 2.7 se observa el conjunto de las 10 principales autocaras para una misma imagen, donde se puede apreciar —sobre todo en las caras número 4 y 5 de la fila de arriba— que la información que representan éstas es la iluminación en la escena.

Este método se utiliza en verificación facial cuando se tiene un conjunto de imágenes de un mismo usuario para realizar el enrolamiento y así extraer los autovectores y autovalores que generan las autocaras para crear el modelo de enrolamiento. Consecuentemente, el conjunto de autovectores de autocaras generadas para el modelo de verificación deben ser tanto más iguales cuanto más se parezcan las imágenes utilizadas en la etapa de enrolamiento y verificación entre sí.

[3]El FLOPS es una medida de rendimiento de una computadora que mide el número de operaciones de coma flotante por segundo que se requieren realizar. M-FLOPS indica que el orden es de millones de FLOPS.

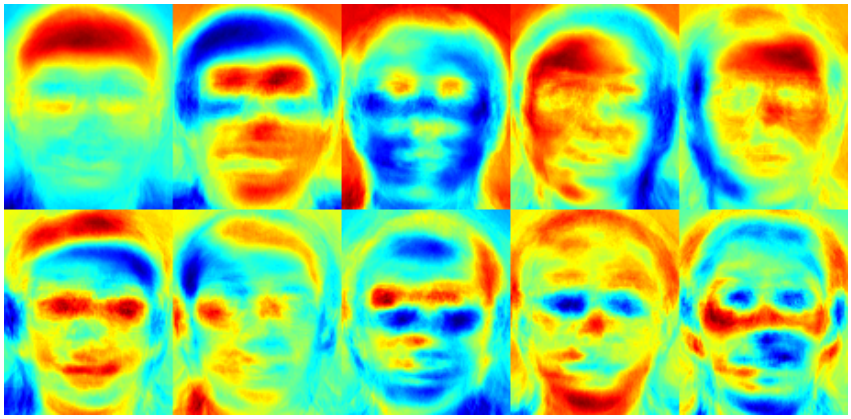

Figura 2.7: Diez principales *Eigenfaces* para una imagen [1].

Las características denominadas *Fisherfaces* [18] resuelven la limitación que puede tener teóricamente las características *Eigenfaces* ya que PCA puede ser óptimo para la reconstrucción o codificación de imágenes de caras, pero puede no ser óptimo para la discriminación entre ellas. Por ello, *Fisherfaces* se basa en *Linear Discriminant Analysis* (LDA) —Ronald Fisher (1936)— buscando un subespacio de características que maximice la relación entre la matriz de dispersión interclase y la intraclase. Una vez computado el subespacio y halladas las matrices de dispersión que maximicen dicho ratio, debido a que la matriz de dispersión intraclase es singular [18] se reduce su dimensión utilizando PCA, con el fin de tomar solo las componentes principales que describen dicha matriz. En la Figura 2.8 se muestran las diez *Fisherfaces* más discriminativas para un usuario concreto.

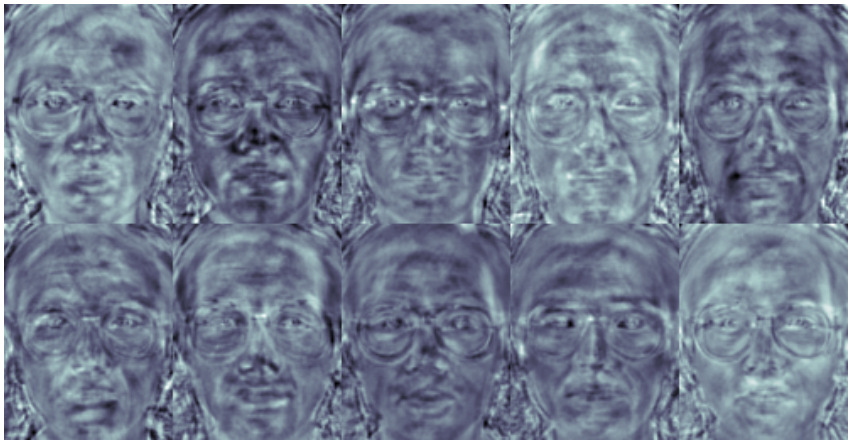

Figura 2.8: Diez principales *Fisherfaces* para una imagen [1].

Fisherfaces mejora con respecto a *Eigenfaces* a la hora de resolver el problema de reconocimiento y es más robusto a cambios de iluminación, si bien como modelo generativo, no es tan potente como *Eigenfaces*. Una desventaja de este método es que puede resultar complicado discriminar caras que se encuentren en el límite entre una clase y otra.

Un descriptor de imágenes apliamente utilizado cuyo objetivo es combatir la variación de la iluminación en la imagen es el de los *Local Binary Patterns* (LBP) [3]. La Figura 2.9 ilustra cómo funciona la técnica para extraer los LBPs: en primer lugar recorre la imagen con un filtro de un tamaño determinado (en la Figura 2.9 se utiliza un filtro de conectividad 8) y se obtienen los valores de intensidad de los píxeles bajo el filtro si la imagen es monocromática. Posteriormente se toma como umbral el valor del píxel en el centro del filtro y se umbralizan los valores del resto de los píxeles del filtro respecto a este valor, valiendo 1 si son mayores o iguales y 0 si son

menores. Finalmente se escoge un orden en el que recorrer los valores binarios de los píxeles umbralizados, resultando un número binario que describe la textura en ese píxel concreto.

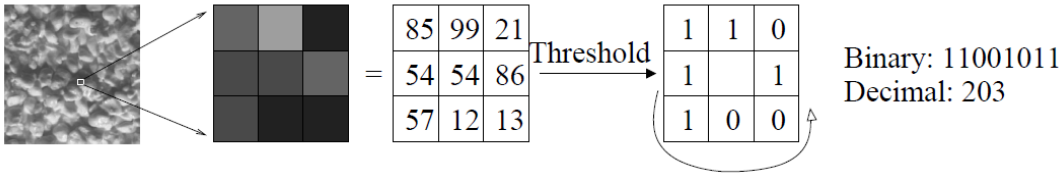

Figura 2.9: Técnica de descripción de una textura utilizando LBP [3].

Se pueden escoger diferentes parámetros del filtro, como puede ser cuántos valores se toman para describir un píxel y cuánto de alejados están los píxeles utilizados para la descripción. Como vector de características se suelen utilizar los histogramas de los LBPs. Para esto es necesario escoger la resolución (número de barras) del histograma y en cuántas regiones espaciales se divide la imagen resultante de aplicar el filtrado LBP, en ancho y alto, para describirlas con los histogramas.

En la Figura 2.10 se puede observar que los descriptores LBPs son prácticamente iguales para las cuatro imágenes originales, donde se han cambiado la intensidad de luz.

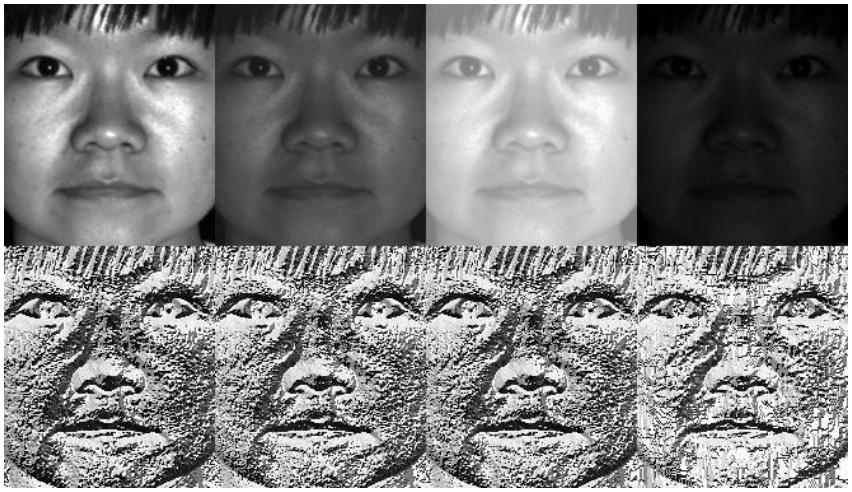

Figura 2.10: Descriptores LBP para imágenes con diferentes intensidades de luz [1].

Pese a ser descubiertos en el año 2011, el descriptor basado en la ley de Weber o *Weber Local Descriptor* (WLD) se le puede considerar de primera generación por ser un método de descripción de imágenes. Los descriptores WLD se utilizan para obtener características invariables a cambios de iluminación, como se puede observar en la Figura 2.11. Estas características se basan en la ley de Ernst Weber (1834) que postula que la relación entre un pequeño cambio perceptual en un estímulo y el nivel del estímulo es constante, es decir, los cambios apreciados son relativos, no absolutos. Esta ley se ha utilizado en diferentes campos de aplicación como clasificación de texturas, compensación de la iluminación y muestreo adaptativo de señal, como se hace mención en [4].

Para calcular el descriptor de las *Weber-faces*, se calcula la diferencia entre el píxel central y los adyacentes, todo ello normalizado respecto al valor del píxel central. Con el ánimo de parametrizar la intensidad entre el ratio anteriormente descrito para un píxel y su vecino, se ponderan por un factor α y se aplica la función arcotangente a modo de filtro, para paliar valores de alta magnitud positivos o negativos, como se muestra en la Ecuación 2.1, donde x_c es el valor del píxel central y x_i es el valor de cada uno de los p píxeles adyacentes.

Figura 2.11: (a) Muestras de la base de datos PIE. (b) Correspondientes *Weber-faces* [4].

$$\xi(x_c) = \arctan\left(\alpha \sum_{i=0}^{p-1} \frac{x_c - x_i}{x_c}\right). \tag{2.1}$$

De las características aquí presentadas se partió de un estudio realizado por Argos Global donde se evaluaban las *Eigenfaces*, las *FisherFaces* y los LBPs, siendo LBPs las mejores características evaluadas entonces y las cuales se explotan en este trabajo. Un posterior estudio de Argos Global tras la finalización de este trabajo demuestra que las *Weber-faces* funcionan mejor que los LBPs para las tareas de verificación facial, como se presenta en la *Sección 4.2: Investigación de diferentes algoritmos de detección, modelado, verificación y fusión de características de cara*.

Segunda generación

La segunda generación se desarrolla entorno la década del 2000, donde se siguen aplicando modelos lineales entrenando las transformaciones a partir de grandes conjuntos de datos de entrenamiento. Estos datos suelen ser del orden de millones de imágenes de caras para que el sistema funcione correctamente, y requieren una capacidad de cómputo de G-FLOPS, y los modelos utilizados son bien generativos, bien discriminativos. La precisión de estos algoritmos típicamente alcanza valores de EER entre el 5 y el 10 % [5, 19, 20, 21].

Los dos métodos comúnmente utilizados son *Sparse Representation* o Representación Dispersa [5] y *Metric Learning* o Aprendizaje Métrico [19]. En este apartado se recalca la idea detrás del método por encima que en la técnica del mismo en sí. Además de estos dos métodos se incluye en qué consisten los *Support-Vector Machine* (SVM) [22], debido a que es en esta generación cuando se empiezan a utilizar vectores de características de elevada dimensionalidad y una técnica ampliamente utilizada consiste en entrenar hiperplanos de alta dimensionalidad para poder delimitar espacios que separen las características de un usuario y las del resto.

La Representación Dispersa o *Sparse Representation* [5] nace con el ánimo de combatir los problemas de oclusión, iluminación, ruido y corrupción para el reconocimiento facial, pero a cambio, requiere de un número de características más elevado para que la representación dispersa obtenga buenos resultados. En la Figura 2.12 se puede ver en qué consiste la idea: una imagen con oclusiones (arriba a la izquierda) o ruidosa (abajo a la izquierda) se puede representar como una combinación lineal entre el conjunto de imágenes de entrenamiento (en medio) ponderada por unos coeficientes, más un ruido como puede ser las áreas ocluidas (arriba a la derecha) o ruido aleatorio (abajo a la derecha). De esta manera, aquél coeficiente de mayor valor se corresponde con la imagen de entrenamiento que proporciona mayor verosimilitud con la imagen con oclusiones o ruidosa.

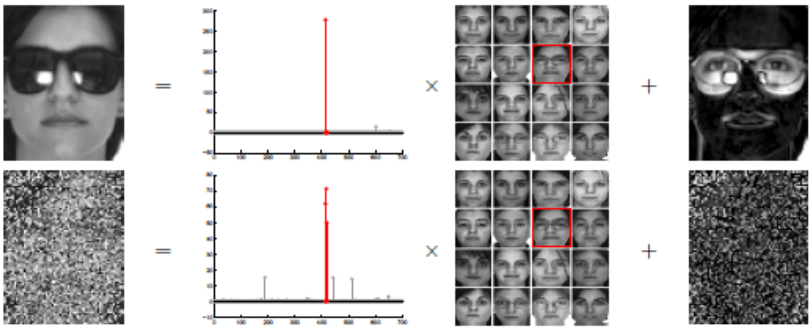

Figura 2.12: Funcionamiento de *Sparse Representation* [5].

Es sencillo visualizar la aplicación de identificación facial utilizando este método. Para aplicarlo en verificación facial, simplemente habría que establecer un umbral de forma que si un coeficiente supera dicho umbral el usuario es aceptado y si ningún coeficiente lo supera, es rechazado.

El Aprendizaje Métrico o *Metric Learning* [19] consiste en optimizar una matriz M de forma que la distancia entre una pareja de características de entrenamiento $\mathbf{x}_e$ y de verificación $\mathbf{x}_v$ sea mínima si pertenecen al mismo sujeto y máxima si pertenecen a sujetos diferentes, como se muestra en la Ecuación 2.2.

$$d(\mathbf{x}_e, \mathbf{x}_v) = (\mathbf{x}_e - \mathbf{x}_v)^T M (\mathbf{x}_e - \mathbf{x}_v). \tag{2.2}$$

Con el fin de entrenar esta matriz se requiere maximizar un criterio de divergencia, de los cuales en la literatura recomiendan o bien el de Log-Determinant (LogDet) [19, 21] o bien el de Kullbach-Leibler (KL) [16]. Para realizar esto y conseguir buenos resultados, se requiere una gran cantidad de datos de entrenamiento y elegir correctamente los vectores de características y la dimensionalidad de la matriz M con el fin de poder alejar espacialmente características de usuarios similares a la vez que se mantienen cercanas características de un mismo usuario en diferentes condiciones ambientales.

Los *Support-Vector Machine* (SVM) [22] es una técnica consistente en generar un hiperplano en un espacio de alta dimensión con el fin de poder clasificar vectores de elevada dimensionalidad reduciendo el error. Se ha decidido incluir en este apartado la técnica de los SVMs por dos motivos: el primero es porque el hiperplano entrenado define en qué parte del espacio se encuentran los vectores de características de usuario genuino e impostor, luego este define un modelo que utiliza aprendizaje automático para su adquisición; y el segundo es porque las técnicas de alta dimensionalidad se empiezan a utilizar más en esta generación que en la primera.

La idea tras este método es que al aumentar drásticamente la dimensionalidad no es necesario crear fronteras de clasificación complejas, simplemente con un hiperplano es suficiente, ya que,

al estar en un espacio de elevada dimensión, es sencillo encontrar con más facilidad una frontera que separe, sin dejar apenas lugar a error, los vectores de características de usuarios genuinos e impostores. Es habitual utilizar SVMs con funciones de kernel a la hora de definir el espacio de alta dimensión. Existen tres problemas principales que aparecen al utilizar esta técnica: el primero es que se requieren un gran número de vectores de características de usuario impostor y aún más de genuino, para definir correctamente la frontera de decisión; el segundo es que las características utilizadas deben ser lo suficientemente discriminativas —es decir, que existan poca correlación entre usuarios distintos— para poder encontrar dicho hiperplano y que sea eficaz; y el tercero es la necesidad de reentrenamiento del sistema completo en el caso en el que se desee incluir más datos tanto de usuarios genuinos o de usuarios impostores.

Los métodos de segunda generación no se implementan en el trabajo, aunque sí que se consideran evaluar su eficacia para resolver problemas de verificación facial en entornos móviles en un futuro próximo, ya que se estima que no requieren de una computación excesiva que pueda presentar problemas de eficiencia en un *smartphone* o *tablet* actual.

Tercera generación

Por último, la tercera generación comienza a desarrollarse en torno al año 2010, y se dejan de lado las características lineales para pasar a métodos no lineales, con modelos discriminativos y características complejas. Se necesitan una inmensa cantidad de datos de entrenamiento, del orden de miles de millones de imágenes faciales para que el sistema de reconocimiento funcione correctamente, y necesitan una capacidad de procesamiento de T-FLOPS. La precisión de estos algoritmos típicamente alcanza valores de EER entre el 1 y el 5 %, llegando en ocasiones a ser incluso menor [2, 23, 24].

Los algoritmos aquí utilizados utilizan técnicas de *Deep Learning* o Aprendizaje Profundo, como redes neuronales convolucionales multicapa donde se transforma el espacio de características de manera no lineal y requiere una enorme cantidad de datos para poder entrenar el modelo. Estos métodos son robustos ante variaciones de la posición de la cara, oclusiones e iluminación.

Estos métodos están dando con precisiones próximas la habilidad humana —que se encuentran en torno al 97 % de precisión—, por eso, ya se comienza a hablar de sistemas de reconocimiento facial que van más allá de la precisión humana [16].

Se ha desestimado utilizar estos métodos en este trabajo en el contexto de aplicaciones de verificación facial móviles con una arquitectura local, debido al elevado coste en el entrenamiento en términos de imágenes necesarias para generar un modelo del usuario genuino de calidad, y de capacidad de proceso para computar el mencionado modelo. No obstante, debido a que es la tecnología de reconocimiento facial con mejores resultados no se ha descartado utilizarla en un futuro sobre una arquitectura distribuida.

2.2.3. Comparación basada en distancias

Como se ha mencionado anteriormente, para poder realizar operaciones de reconocimiento facial, sea tanto verificación como identificación, se pueden utilizar modelos basados en puntuaciones o en distancias (inverso de la puntuación) entre características. En este subapartado se enuncian medidas de distancia típicas usadas sobre vectores de características e histogramas. Como ya se dijo al principio de este apartado, una distancia devuelve un valor más alto cuanto más diferentes sean las características y es lo inverso a una puntuación o *score*.

Distancia Euclídea

La distancia Euclídea, según la definición del NIST es la distancia entre dos puntos unidos por una línea recta. Es una de la distancias más utilizadas ya que mide la distancia más corta entre dos puntos. La Ecuación 2.3 muestra cómo se computa.

$$d_{Euclidean}(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}, \tag{2.3}$$

siendo $\mathbf{p}$ y $\mathbf{q}$ vectores y p_i y q_i, la componente i-ésima de cada vector.

Distancia Coseno

La distancia coseno se diferencia de las anteriores debido a que está normalizada en el rango $[-1, 1]$. Esta distancia mide el coseno del ángulo que forman dos vectores: si estos dos vectores son similares, el ángulo que forman tiende a cero, por lo que su coseno tiende a 1; en cambio, si ambos vectores son totalmente opuestos entre sí forman un ángulo llano, por lo que el valor del coseno es -1. La Ecuación 2.4 muestra la fórmula para calcular la distancia coseno entre dos vectores.

$$d_{Cosine}(\mathbf{p}, \mathbf{q}) = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\sqrt{\sum_{i=1}^{n} q_i^2}}, \tag{2.4}$$

siendo $\mathbf{p}$ y $\mathbf{q}$ vectores y p_i y q_i, la componente i-ésima de cada vector.

Distancia Chi-cuadrado

La distancia Chi-cuadrado o χ^2 es una medida de distancia utilizada en el ámbito de la visión artificial, ya que se suele utilizar para la la medición de similitudes entre histogramas. Esta distancia está basada en el test estadístico con el mismo nombre, el cual evalúa la diferencia entre dos tablas de frecuencias o distribuciones de probabilidad. En la Ecuación 2.5 puede observarse cómo se calcula esta distancia.

$$d_{Chi-Square}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} \frac{(p_i - q_i)^2}{p_i + q_i}, \tag{2.5}$$

siendo $\mathbf{p}$ y $\mathbf{q}$ vectores y p_i y q_i, la componente i-ésima de cada vector.

Distancia Bhattacharyya

La distancia Bhattacharyya, similar a la distancia Chi-cuadrado es bastante popular en el ámbito de la visión por computador ya que mide la similitud entre dos distribuciones de probabilidad continuas o discretas. Esta medida está estrechamente relacionada con el coeficiente de Bhattacharyya, que mide el solape entre dos muestras o poblaciones estadísticas. La Ecuación 2.6 muestra el procedimiento para calcular esta distancia.

$$d_{Bhattacharyya}(\mathbf{p}, \mathbf{q}) = -\ln\left(\sum_{i=1}^{n}\sqrt{p_i q_i}\right), \tag{2.6}$$

siendo $\mathbf{p}$ y $\mathbf{q}$ vectores y p_i y q_i, la componente i-ésima de cada vector.

Distancia Manhattan

La distancia Manhattan, según la definición del NIST[4] es la distancia entre dos puntos medida a lo largo de ejes con ángulos rectos. Se denomina Manhattan porque mide la distancia recorrida por un coche para llegar de un punto a otro en una ciudad de bloques cuadrados, como lo es la ciudad de Manhattan en Nueva York. Es una de las distancias más sencillas de calcular, ya que, como muestra la Ecuación 2.7, no requiere de ninguna operación de complejidad multiplicativa.

$$d_{Manhattan}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} |p_i - q_i|, \tag{2.7}$$

siendo $\mathbf{p}$ y $\mathbf{q}$ vectores y p_i y q_i, la componente i-ésima de cada vector.

Distancia Mahalanobis

La distancia Mahalanobis es una distancia estadística que mide la similitud que existe entre una muestra y una distribución, siendo ésta cero cuando la muestra coincide con la media. Si se considera que los vectores $\mathbf{p}$ y $\mathbf{q}$ de la Ecuación 2.8 tienen la misma distribución, S es la matriz de covarianza de dicha distribución.

$$d_{Mahalanobis}(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{p} - \mathbf{q})^T S^{-1} (\mathbf{p} - \mathbf{q})} = \sqrt{\sum_{i=1}^{n} \frac{(p_i - q_i)^2}{s_i^2}}, \tag{2.8}$$

siendo $\mathbf{p}$ y $\mathbf{q}$ vectores y p_i y q_i, la componente i-ésima de cada vector.

Además de los modelos basados en distancias, con los vectores de características se podían entrenar modelos basados en clasificadores "uno contra todos", teniendo que ser estos modelos actualizables con posteriores enrolamientos de usuarios nuevos en el sistema.

2.3. Biometría de voz

La biometría de voz es aquella que utiliza la información que se encuentra en la señal de voz, producida por un sujeto a diferentes niveles, con el fin de explotarla para distintas aplicaciones [11]. No existen dos sujetos que tengan la misma señal de voz, ya que varía la forma de su tracto vocal, tamaño de la laringe y otras partes de su anatomía [25], por lo que es un buen tipo de rasgo para identificar a los sujetos, como se mencionó en la *Sección 2.1: Introducción a la Biometría.*

La biometría de voz comienza a desarrollarse a finales del siglo XX, siendo la voz un rasgo que el individuo produce. La señal de voz contiene una información muy completa, desde el idioma o el acento de un individuo, su identidad, el mensaje que transmite, el estado de ánimo, hasta posibles patologías[5]. Esta información puede encontrarse a diferentes niveles lingüísticos de la señal de voz, como puede ser:

- **Nivel semántico**: relacionado con el sentido o significado del mensaje emitido.
- **Nivel fonético**: relacionado con los sonidos emitidos o secuencias de sonidos que forman las palabras.

[4] *National Institute of Standards and Technology.*

[5] Ver Eduardo Gonzalez-Moreira. Automatic Prosodic Analysis to Identify Mild Dementia. *BioMed Research International*, 2015.

- **Nivel prosódico**: relacionado con la entonación, las pausas o la intensidad de los sonidos producidos.
- **Nivel espectral**: relacionado con la producción de los sonidos, coarticulaciones, nasalidad de los sonidos.

Debido a la madurez de las tecnologías en este área, así como la información subyacente en la señal de voz, existen tres principales aplicaciones en relación con las Tecnologías del Habla, dependiendo de qué información es la que se quiere extraer de una locución:

- **Reconocimiento de voz**: el sistema reconoce cual es el mensaje producido por un locutor pudiendo transcribirlo a texto o activando una respuesta predeterminada ante determinadas palabras o secuencias de palabras.
- **Reconocimento de locutor**: el sistema reconoce la identidad de la persona que habla. Esta aplicación se relaciona directamente con la biometría, y es la que se explora en los siguientes apartados.
- **Reconocimiento de lenguaje**: el sistema reconoce el idioma o dialecto en el cual está hablando un locutor.

Al igual que se vio en la *Sección 2.2: Biometría facial*, las Tecnologías del Habla también se enfrentan a distintos retos, algunos intrínsecos al productor de voz, y otros relacionados con el entorno de captación. Estos retos pueden caracterizarse en tres grandes grupos:

- **Factores internos intrínsecos**: aquellos relacionados directamente con el productor de voz. Estos factores pueden ser permanentes, como el sexo del locutor, la edad, la sesión en la que se realice las muestras, el tipo y la cantidad de habla recogida; o transitorios, como el estado emocional o si el usuario tiene patologías fonatorias.
- **Factores internos forzados**: aquellos relacionados con el productor de voz, pero que dependen del entorno en el que se encuentra recogiéndose la muestra de habla. Pueden ser el *efecto Lombard*, que se produce cuando el locutor produce voz en un sistema ruidoso y el *efecto Cocktail party*, que se produce cuando el locutor produce voz en un entorno con otras voces de fondo.
- **Factores externos**: dependientes del medio donde se esté recogiendo la voz o del canal de comunicación por el que se transmite la voz, como puede ser ruido acústico o eléctrico, reverberación de la sala, el sensor que recoge la muestra, su respuesta en frecuencia, rango dinámico y distorsión de los micrófonos, distancia entre el locutor y el medio de captación (campo cercano/lejano), ancho de banda de transmisión, etcétera.

Como se ha mencionado anteriormente, la tarea a resolver en este trabajo es la de reconocimiento de locutor. De manera análoga, se define la tarea de *verificación de locutor* dada una locución y una identidad como la de afirmar si una el locutor que ha generado la locución coincide o no a dicha identidad. Al igual que ocurría con la verificación y el reconocimiento facial, en esta memoria se usará indistintamente reconocimiento o verificación de locutor, para referirnos la tarea de decidir si un usuario es quien dice ser.

Existen dos estrategias principales para acometer la tarea de reconocimiento de locutor, llamadas *reconocimiento de locutor dependiente de texto* y *reconocimiento de locutor independiente de texto*:

- **Reconocimiento de locutor dependiente de texto**: el locutor dice o bien una frase la cual ha utilizado para la etapa de registro o enrolamiento (texto fijo), o bien una frase que le provee el sistema (texto variable). El sistema conoce de antemano la frase que el usuario dirá para verificarse y adaptará el modelo de usuario a esta frase, reconociendo no solo lo que se dice, sino quién lo dice, implicando ello una mayor seguridad a la par que una mayor complejidad.

- **Reconocimiento de locutor independiente de texto**: el locutor dice cualquier frase, o bien habla de forma libre, y el sistema ha de reconocerle independientemente del mensaje que esté transmitiendo.

En los siguientes apartados de esta memoria se distinguirán estos diferentes enfoques de reconocimiento de locutor, así como los distintos algoritmos y modelos diseñados para solventar los problemas que ellos acarrean y sus virtudes.

El esquema tradicional de verificación de voz es similar al de verificación facial (mostrado anteriormente en la figura Figura 2.3), siendo las fases para la verificación las mismas, aunque ahora se trabaja con señales de voz, como se muestra en la Figura 2.13.

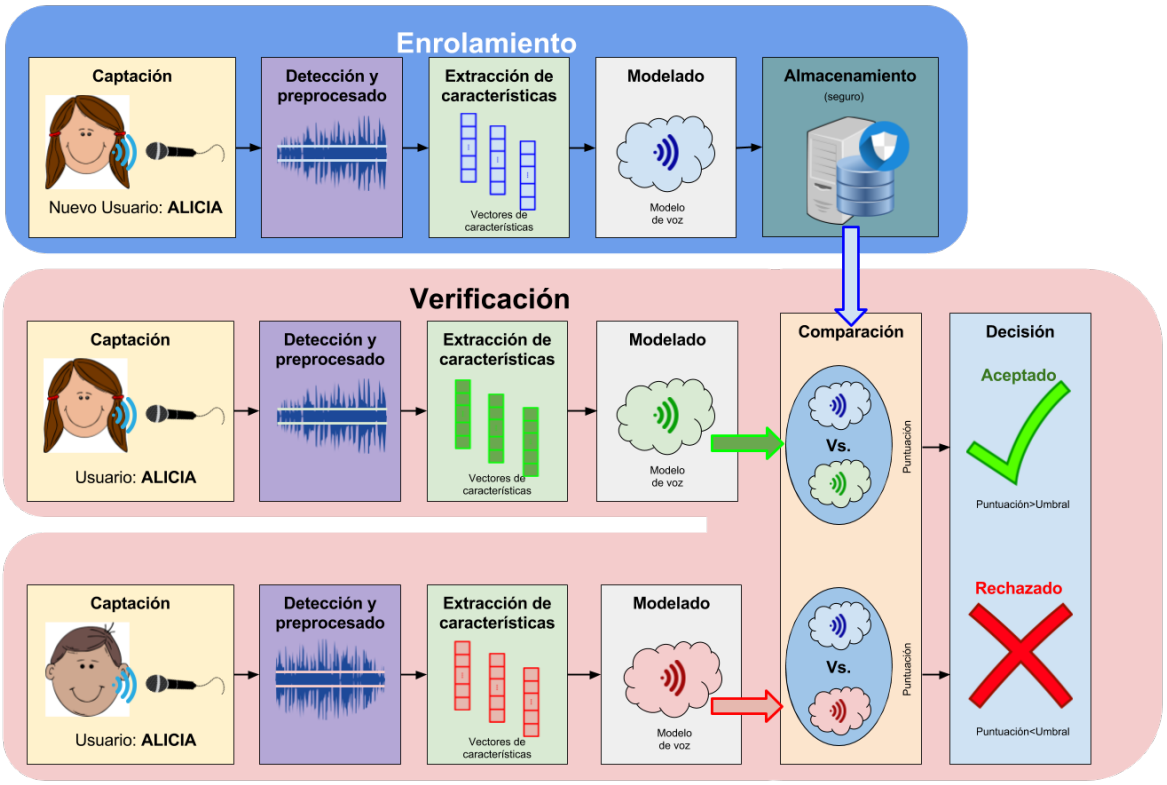

Figura 2.13: Esquema tradicional de verificación de voz.

1. **Captación**: se adquiere una locución de audio a través de un micrófono, y se solicita, nuevamente, una identidad, como puede ser un nombre de usuario. En la etapa de enrolamiento se asocia la identidad con el modelo de enrolamiento en la fase de Almacenamiento, mientras que en la etapa de verificación se provee una identidad para que durante la fase de Comparación se compare el modelo de verificación computado en esta etapa con el modelo de enrolamiento asociado a dicha identidad.

2. **Detección y preprocesado**: a partir de la grabación de audio se realiza una detección de actividad (VAD, *Voice Activity Detection*) para tomar las muestras de voz y no de ruido de fondo. Además, se pueden realizar compensaciones de distorsión de dispositivo de captación, o de canal, para tomar la señal de audio de la manera más fiel posible.

3. **Extracción de características**: al igual que para reconocimiento facial, las características a utilizar serán aquellas que más discriminen a un usuario de otro. Estas características pueden extraerse para intentar modelar distintos niveles de la señal de voz, como se mencionó anteriormente. Las características MFCC (*Mel-Frequency Cepstral Coefficients*) suelen ser las que han dado mejores resultados tradicionalmente en reconocimiento de voz [6], ya que modelan el tracto vocal del locutor, como se explicará más adelante.

4. **Modelado**: se modelan las características extraídas en la fase anterior. Depende de la estrategia que se quiera utilizar para identificarle —reconocimiento de locutor dependiente o independiente de texto— el mejor modelo será diferente.

5. **Almacenamiento**: el modelo de enrolamiento se almacena, generalmente de forma segura y/o cifrada, para que pueda ser utilizado en la fase de Comparación de modelos, como ya se vió en el esquema tradicional de verificación facial.

6. **Comparación**: se compara el modelo de verificación con el de enrolamiento una vez provista una identidad. Según el Modelado realizado, la manera de comparar los modelos será distinta dependiendo de la tecnología utilizada.

7. **Decisión**: según la puntuación determinada en la fase de Comparación y un umbral predeterminado, el usuario que se verifica es aceptado si la puntuación supera el umbral y rechazado en el caso contrario, tal como se explicó en verificación facial. De igual modo, pueden tomarse en consideración otros factores a la hora de tomar la decisión, como la cantidad de ruido en la señal de audio captada, ruido producido por los dispositivos de captación, otros modelos diferentes trabajando en paralelo o sistemas *antispoofing*.

A continuación, se citan métodos de parametrización de voz, y algoritmos de reconocimiento de locutor independientes y dependientes del texto.

2.3.1. Parametrización de la señal de voz

La parametrización de voz depende del nivel donde se encuentra la información a extraer de la señal de voz. Los parámetros de bajo nivel son fáciles de extraer y de modelar, pero son altamente sensibles a variaciones y ruido. Los parámetros de alto nivel son difíciles de extraer y tienen alta complejidad. Estos parámetros solo se pueden adquirir en locuciones de larga duración, como puede ser una llamada telefónica. Ambos tipos de características poseen información complementaria, por lo cual se pueden realizar sistemas fusionados con características de alto y bajo nivel que ofrecen mejores resultados para reconocimiento de locutor. Para el problema a acometer se centrará la memoria en los parámetros a bajo nivel, ya que se desea que la verificación no le lleve al usuario un tiempo excesivo que ponga en riesgo la usabilidad de la aplicación.

La señal de voz es una señal de audio la cual suele estar acotada a los 8 kHz. Esta señal suele estar activa el 50 % del tiempo, aunque depende de la aplicación en la que se esté captando, por lo tanto, se suele utilizar un Detector de Actividad (VAD) con el fin de tomar solo aquellas muestras de voz, aunque con determinados métodos también es común modelar los silencios. A la hora de tratar esta señal a nivel espectral es común enventanar la señal en periodos de entre 10 y 50 ms, que es lo que se tarda en producir un fonema o sonido. Este enventanado suele

hacerse utilizando ventanas rectangulares (por su bajo coste computacional) o de Hamming (para reducir lóbulos secundarios por debajo de los -40 dB). Además, es típico solapar estas ventanas un 50 %, con el ánimo de no perder información debido al enventanado.

El modelo de producción de voz define para señales de tiempo corto cómo puede ser producido un fonema o sonido vocal. En la Figura 2.14 se muestra un diagrama de este modelo.

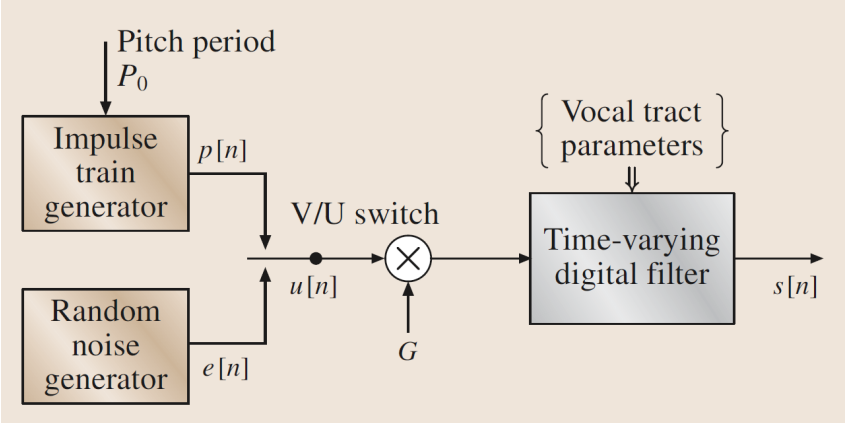

Figura 2.14: Modelo simplificado de producción de voz [6].

A la izquierda de la Figura 2.14 se puede ver que si el sonido es sonoro (*voiced*) se genera un tren de impulsos a una frecuencia fundamental (*pitch period*) determinada. En cambio, si el sonido es sordo (*unvoiced*) se genera un ruido aleatorio. Posteriormente, existe un conmutador que selecciona el tipo de fuente según el tipo de sonido que se produzca, y a esta señal se le aplica una ganancia y se le hace pasar por un filtro que modela el tracto vocal. Es en este filtro donde fundamentalmente se encuentra la información de locutor de la señal de voz, ya que modela las resonancias vocales intrínsecas a éste.

El modelo de predicción lineal parametriza la señal de voz utilizando el modelo de producción de voz. Este modelo utiliza un filtro de tracto vocal todo-polos, de orden entre 4 y 10 [6]. Utiliza los parámetros de frecuencia fundamental, selección de sonido sordo o sonoro, ganancia, los polos del filtro y la señal diferencia como estimador. Este estimador se ha utilizado más en codificación y síntesis de voz que en reconocimiento.

Los coeficientes MFCC o *Mel-Frequency Cepstral Coefficients* son ampliamente utilizados en las tecnologías de habla, como se mencionó anteriormente. Estas características se extraen aplicando la Transformada Discreta de Fourier (DFT, *Discrete Fourier Transform*) a una señal de audio enventanada previamente, y hallando la energía total logarítmica de las distintas bandas del banco de filtros en escala Mel (ver Figura 2.15). Hecho esto, se halla la Transformada Discreta Inversa de Fourier (IDFT) para transformar estas componentes al espacio cepstral. Se puede de esta manera seleccionar el número de características MFCC, aunque se recomienda utilizar entre 4 y 10, ya que se corresponden con las formantes generadas por el tracto vocal.

Los coeficientes delta miden la variación de los vectores de características MFCC dentro de una ventana de vectores de características. Estos coeficientes suelen ser de primer y segundo orden, intentando representar no solo la velocidad sino la aceleración a la que cambian estas características [6].

Otras características de la señal son típicamente usadas, como la energía total de la señal enventanada, tasa de cruces por cero (ZCR, *Zero-Crossing Rate*), autocorrelación de la señal, la frecuencia fundamental o *pitch*, etcétera.

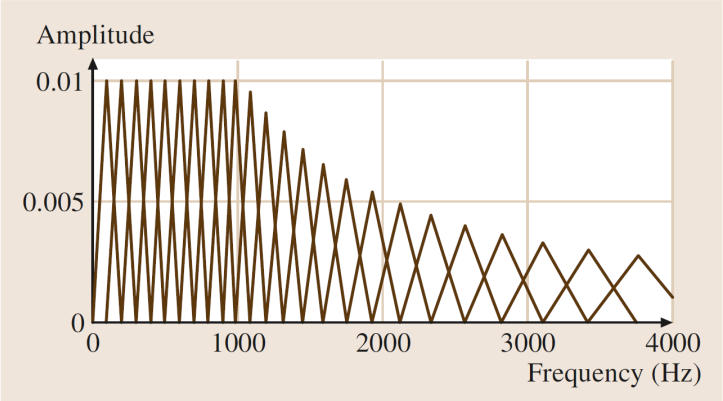

Figura 2.15: Banco de filtros en escala Mel ponderados [6].

2.3.2. Reconocimiento de locutor independiente de texto

El reconocimiento de locutor independiente de texto ofrece soluciones ligeras ya que no requieren de entrenamientos de modelos específicos, como se verá en el reconocimiento de locutor dependiente de texto. Este tipo conjunto de métodos de reconocimiento de voz son los más utilizados y están implementados en aplicaciones como son en *call-centers* y en banca [6]. Debido a que en este TFM no se llega realmente a implementar ninguno de estos modelos, se procede a mencionar los distintos modelos, agrupándolos en cuatro grupos principales, sin entrar en detalles de algoritmia ni de implementación:

- **Basados en codificadores**: en este grupo están los clasificadores que se basan en los trabajos de codificación de la señal de voz, principalmente para comunicaciones. El modelo de Cuantificación Vectorial (VQ, *Vector Quantization*) es un ejemplo de estos modelos, en el cual se realiza *clustering* sobre los datos para posteriormente medir la distancia de las muestras de test a los centroides de los *clusters*. Este tipo de modelos están obsoletos actualmente para realizar reconocimiento de locutor.

- **Probabilísticos**: este grupo está formado básicamente por el Modelo de Mezclas de Gaussianas (GMM, *Gaussian Mixture Model*). Existen dos estrategias fundamentales en el entrenamiento de los hiperparámetros de este modelo: aplicando el criterio de Máxima Verosimilitud (ML, *Maximum Likelihood*) o entrenándolos a partir de un Modelo Universal de Fondo (UBM, *Universal Background Model*), ambas formas bien afianzadas en la literatura [6]. El UBM consiste en entrenar un Modelo de Mezclas de Gaussianas con características extraídas de un conjunto amplio de locutores distintos, y posteriormente ajustar a cada usuario concreto estas gaussianas a sus vectores de características de que modelan su forma de hablar en particular [26]. De esta forma el modelo utiliza datos de habla genérica para estimar las distribuciones en el espacio de características sin necesidad de alta cantidad de habla por parte de los usuarios.

- **Basados en SVM**: con la invención de las Máquinas de Vectores Soporte (SVM), ya mencionadas en la *Sección 2.2.2: Características y modelado facial*, se investigan distintos modelos que hagan uso de su potencia de cálculo. Cabe destacar tres de ellos: el primero es la aplicación directa de SVM sobre los vectores de características, el segundo utiliza los SVM con un kernel GLDS (*Generalized Linear Discriminant Sequence*)[27], y el tercero GSV (*GMM Super-Vector*) [28], que combina los modelos probabilísticos con las máquinas de vectores soporte. Debido a que no se implementan en este trabajo, no se considera necesario más que citarlos en esta sección.

- **Basados en variabilidad total**: estos modelos, también llamados modelos de *i-vectors* [29], se basan en el modelo GSV, pero entrenan un subespacio de variabilidad total aplicando PCA al conjunto de locutores entrenados. Este modelo requiere muchos más datos que los anteriores para ser correctamente entrenado ya que no solo requiere entrenar un UBM, sino también la matriz de variabilidad total; pero aun así, es el que mejores resultados ofrece.

2.3.3. Reconocimiento de locutor dependiente de texto

El reconocimiento de locutor dependiente de texto permite aplicaciones de mayor seguridad a costa de una necesidad mayor de datos, al ser los modelos más complejos. Dos modelos típicos son *Dynamic Time Wrapper* (DTW), que mide la distancia entre dos cadenas de tamaño variable que pueden tener diferente duración, y los Modelos Ocultos de Markov o *Hidden Markov Models* (HMM) que modela probabilísticamente cadenas de gaussianas. Este último modelo es el más utilizado para reconocimiento de voz en la actualidad, y por tanto es el que se expone en los siguientes párrafos.

Según la aplicación se pueden distinguir dos maneras de enrolar y verificar estos modelos:

- **Texto fijo**: el locutor dice una frase para enrolamiento que él conoce y posteriormente, en verificación, dice esa misma frase. Este tipo de aplicación utiliza dos técnicas para conocer la identidad del usuario, como se vio al principio de este apartado, ya que mezcla algo que el usuario conoce —la frase— con algo que el usuario tiene —su voz—. El problema es que burlar el primer sistema es sencillo, ya que un suplantador solo tiene que oír decir la frase para conocerla.

- **Texto variable** o *text-prompted*: el usuario entrena una serie de submodelos (palabras, sílabas, fonemas, trifonemas, etc.) para que después, en tiempo de verificación, se sintetice una frase a partir de consecuciones de submodelos ya entrenados. Este método es más robusto a ataques de suplantación mediante grabaciones, como se verá más adelante, ya que el usuario impostor desconoce con anterioridad la frase que se sintetiza en el momento de realizarse la verificación.

En relación con estas aplicaciones existen cuatro tipo de topologías principales, las cuales se muestran en la Figura 2.16.

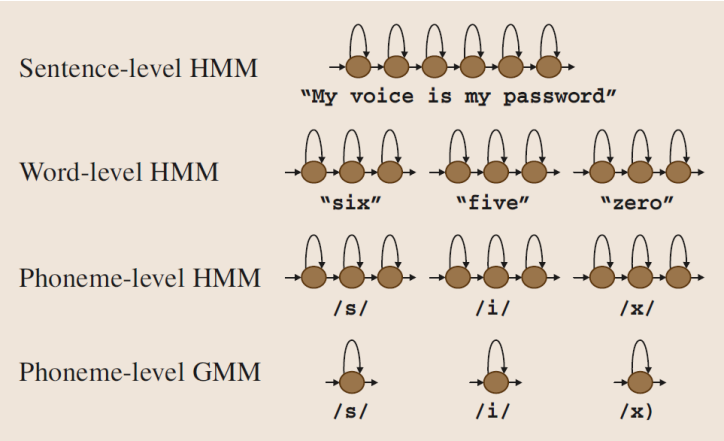

Figura 2.16: Topologías de los HMM [6].

- **Topología a nivel de frase**: relacionada con aplicaciones de texto fijo. El usuario entrena una o varias frases prefijadas y se genera un modelo determinado para cada frase.
- **Topología a nivel de palabras**: el usuario entrena distintas palabras. Este tipo de modelos tiene un entrenamiento sencillo, ya que solo requiere entrenar un conjunto generalmente pequeño —del orden de 10 palabras—. En verificación se solicita al usuario que diga un subconjunto de palabras propuestas de manera aleatoria.
- **Topología a nivel de fonema HMM**: también conocido como trifonemas —ya que generalmente se entrenan tres fonemas: el fonema de transición anterior, el fonema de llegada y el fonema de transición posterior—, estos modelos no solo modelan los fonemas en sí sino las transiciones entre éstos. Estos son los modelos más complicados de entrenar ya que requiere de gran cantidad de datos porque no todos los conjuntos de fonemas son igual de frecuentes en un idioma determinado. En verificación se puede generar cualquier frase para verificar.
- **Topología a nivel de fonema GMM**: similar a los trifonemas, estos modelos solo entrenan un fonema por elemento de la cadena. Este modelo es más sencillo de entrenar que el de trifonemas pero menos robusto. Igualmente que en el caso anterior, en verificación se puede generar cualquier frase para verificar.

Como se puede observar, los modelos más robustos y más flexibles requieren de más datos para entrenarlos. Esto puede ser un problema para determinado tipo de aplicaciones ya que puede que el usuario no utilice la aplicación porque no es *user-friendly*; por ello, muchos modelos utilizan modelos preentrenados con modelos universales de fondo (UBM) y adaptan los datos al nuevo usuario, requiriendo así menos datos de éste.

También, es necesario entrenar de manera equilibrada todos los modelos relacionados con fonemas o trifonemas, es decir, no es una buena práctica entrenar con más datos unos modelos que otros. Por ello, existen *corpora* de frases denominados «fonéticamente *balanceados*» donde con un conjunto de frases se entrenan de manera equitativa los fonemas [30], también disponible el *corpus* en inglés [31][6].

Se puede observar que las topologías basadas en palabras, fonemas y trifonemas permiten realizar aplicaciones de texto variable, y por lo tanto, es un mecanismo para evitar suplantaciones mediante grabaciones, ya que no es tan sencillo adquirir la frase concreta de un usuario genuino en tiempo de ejecución.

2.4. Fusión biométrica

La fusión biométrica consiste en la combinación de información de distintas fuentes biométricas con el ánimo de mejorar la veracidad de la identificación [32]. El concepto que hay tras la fusión es que la combinación de distintas fuentes de información independientes aporta más información al problema y, por lo tanto, puede llegar a ser más sencillo realizar una clasificación altamente fiable. De la misma manera, esta fusión soluciona problemas relacionados con la suplantación de identidad, ya que burlar varios sistemas de manera simultánea se convierte en una tarea razonablemente más complicada que el burlar solamente uno.

Se puede categorizar la fusión biométrica en dos conjuntos principales: la fusión biométrica intra-rasgo e inter-rasgo, en función de la fuente de donde provengan los datos. La fusión biométrica intra-rasgo combina distintas fuentes de información provenientes del mismo rasgo

[6] Ver *Harvard Sentences* (Consultado 04/2017).

biométrico, como por ejemplo, en la *Sección 2.3: Biometría de voz* se distinguieron dos niveles de extracción de información de la señal de voz: alto nivel, donde se analiza el léxico, la semántica y la entonación entre otras características, y bajo nivel, donde se analiza la señal de voz a nivel espectral, fundamentalmente. La combinación de estas dos fuentes de información a partir de un mismo rasgo biométrico es lo que se denomina fusión intra-rasgo, en contraposición con la fusión de distintos rasgos biométricos —fusión inter-rasgo o multibiométrica— donde se combina la información procedente de distintas fuentes de información de distintos rasgos biométricos. En este TFM se investiga la fusión intra-rasgo en biometría facial y se diseña un sistema para la realización de fusión inter-rasgo de biometrías faciales y de voz.

Además de esta primera categorización, se pueden definir distintos tipos de fusión biométrica dependiendo de en qué etapa se realice la fusión, como se observa en la Figura 2.17, donde se observa la fase de verificación de un sistema donde se adquieren dos rasgos biométricos (distinguiendo las etapas en verde y rojo), suponiendo un previo entrenamiento (modelo marcado con la letra «E» en ambos rasgos). Huelga decir que se necesita disponer de la información de distintos rangos biométricos tanto en la fase de enrolamiento como en la fase de verificación. En el caso en el que la fusión biométrica fuera intra-rasgo, podría no ser necesaria la captación de los rasgos de manera independiente. Por otro lado, si la fusión biométrica fuera inter-rasgo sería necesaria la adquisición de los distintos rasgos utilizando los dispositivos de captación requeridos.

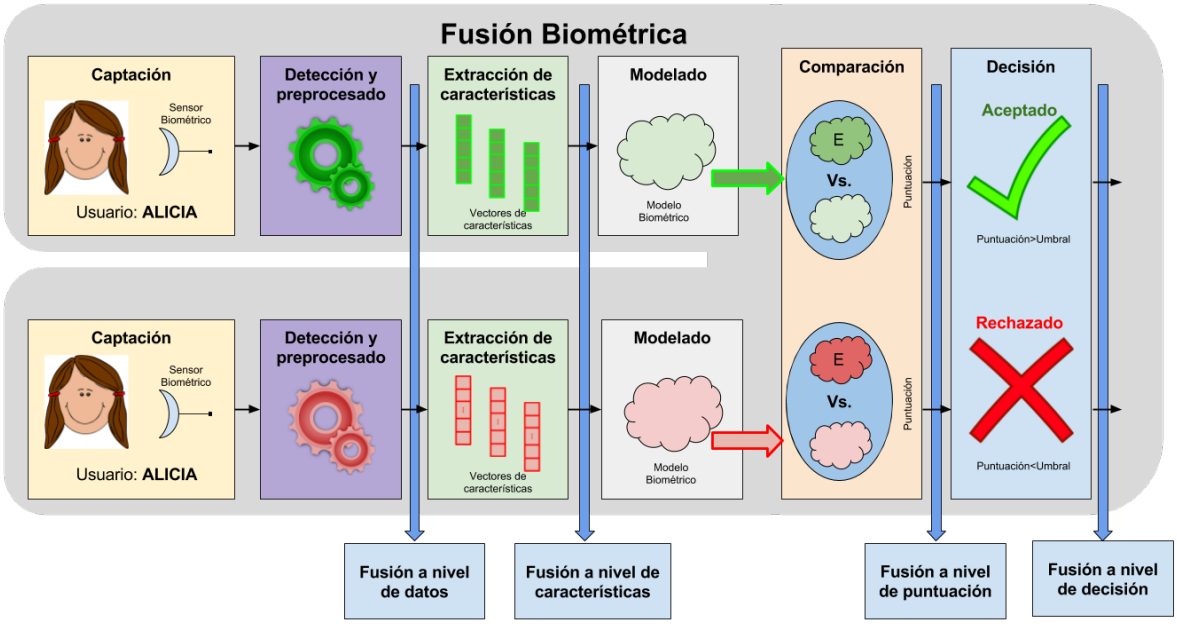

Figura 2.17: Niveles de fusión biométrica.

Tras el preprocesado de la señal se puede utilizar la fusión a nivel de datos, combinando la salida de los datos de múltiples maneras, como se explica más adelante. Posteriormente, tras la extracción de características se podría realizar fusión biométrica a nivel de características y combinar éstas en un único modelo. Otra opción para la fusión es realizarla a nivel de puntuación (o de *score*), aunque cabe destacar la necesidad de realizar una normalización de las puntuaciones con el fin de trabajar en el mismo rango. Por último, una opción sencilla pero menos versátil, como se verá posteriormente, es combinar la decisión de los distintos sistemas biométricos.

A continuación se detallan los diferentes niveles de fusión, así como los métodos existentes en el estado del arte para realizar la fusión biométrica.

2.4.1. Fusión a nivel de datos

La fusión a nivel de datos es la más amplia de definir, debido a que depende de la naturaleza de los rasgos y de los datos para realizar la fusión. Se puede subdividir este tipo de fusión en los siguientes subtipos [33].

- **Multi-sensor**: las muestras están tomadas de distintos sensores comerciales y se combinan la señal tomada por estos sensores con el fin de aumentar más información. Como ejemplo se puede encontrar la combinación de imágenes de una cámara en color con una de profundidad para reconocimiento facial.
- **Multi-algoritmo**: para procesar la señal se extrae la información con distintos algoritmos y se fusionan los datos. Este puede ser el caso de la fusión de señales de voz a alto y bajo nivel.
- **Multi-instancia**: en rasgos biométricos donde el rasgo a capturar está repetido una (iris, retina, palma de la mano) o múltiples veces (huella del dedo) se puede fusionar la información de estos rasgos que aparecen multiples veces para mejorar el desempeño.
- **Multi-muestra**: similar a multi-instancia, la fusión multi-muestra toma distintas tomas de un mismo rasgo biométrico con el fin de combinarlas. Como ejemplo, se pueden encontrar distintas poses en una cara (frente, perfil izquierdo, perfil derecho) o distintas posiciones de una huella dactilar (de frente y rodada).
- **Multi-modal**: combinación de distintos rasgos biométricos que pueden ser tratados de manera similar, como puede ser la segregación de una cámara de alta calidad en reconocimiento facial y de iris al mismo tiempo.

De este nivel de fusión no se darán más detalles ya que no se implementa en este trabajo.

2.4.2. Fusión a nivel de características

La fusión a nivel de características consiste en extraer características de distintos sistemas biométricos con el fin de generar un único vector de características para modelar los usuarios [32]. Para trabajar correctamente con estas características es preciso realizar una normalización de las mismas, pudiendo utilizar métodos de normalización típicos como la normalización MinMax (Ecuación 2.9) donde los datos de entrada (x en el rango $[min_x, max_x]$) se normalizan a un rango de salida deseado ($[min_n, max_n]$), o la normalización Z-Score (Ecuación 2.10), donde se normalizan en función de su media y desviación estándar (recomendable si se conoce que la distribución de los datos es gaussiana).

$$Norm_{MinMax}(x, [min_x, max_x], [min_n, max_n]) = \frac{x - min_x}{max_x - min_x}(max_n - min_n) + min_n. \quad (2.9)$$

$$Norm_{Z-Score}(x, [\mu_x, \sigma_x]) = \frac{x - \mu_x}{\sigma_x}. \quad (2.10)$$

Posteriormente, se suelen emplear métodos de selección (ANOVA F-Value, Chi-square o *Mutual Information*) o transformación de características (LDA, PCA) con el ánimo de reducir la dimensionalidad de los vectores de características, los cuales quedan fuera del alcance de esta memoria.

2.4.3. Fusión a nivel de puntuación

La fusión a nivel de puntuación combina la puntuación o *score* obtenida a partir de la confianza de clasificación (que puede o no ser probabilística) entre distintos clasificadores.

Esta puntuación normalmente se normaliza [34] utilizando MinMax (Ecuación 2.9) o Z-Score (Ecuación 2.10) entre otras [35], aunque de igual modo, se puede entrenar un algoritmo que asigne pesos a las distintas puntuaciones según su importancia, o directamente un clasificador que utilice estas puntuaciones como características.

Reglas típicas que se utilizan para combinar las salidas de las puntuaciones normalizadas son la regla de la suma, consistente en sumar la salida de los distintos clasificadores o la regla del producto, que consiste en multiplicar las salidas de los diferentes clasificadores, además de otras funciones sencillas como el mínimo, o el máximo [35, 36]. Además, las puntuaciones obtenidas se pueden introducir como nuevos vectores de características para otros clasificadores más sencillos (como por ejemplo, basados en árboles) que permitan tomar la decisión utilizando una clasificación más compleja y más datos para poder entrenar los sistemas. Otras formas de combinación se pueden explorar en [37, 38], aunque no se implementan en este trabajo.

La principal ventaja de realizar la fusión a nivel de puntuación en vez de hacerlo en el resto de niveles es que en este punto se obtiene una información más manipulable que tras haber tomado una decisión, ya que el ajuste de los umbrales de manera individual no es algo trivial, y que en sistemas fusionados puede afectar a los resultados de la clasificación (algunos sistemas pueden tener una tasa de falso positivo o falsa aceptación muy sesgada con respecto a otros), y por otro lado, se tienen sistemas especialistas en resolver problemas concretos, como es el caso de los sistemas de fusión inter-rasgo, donde cada una de los rasgos biométricos se adquiere y procesa de manera independiente, utilizando técnicas y algoritmos en los que se ha investigado y desarrollados desde hace décadas, cosa que en la fusión de características no es posible.

Una técnica aplicada en este trabajo es el tratamiento diferenciado de distintos subrasgos biométricos dentro de la misma imagen facial, aplicando un tratamiento diferente con cada subrasgo y luego realizar una fusión de los mismos a nivel de puntuación.

2.4.4. Fusión a nivel de decisión

La fusión a nivel de decisión combina el valor de decisión tomado por distintos clasificadores. Esta decisión en el ámbito de la biometría generalmente es binaria (aceptado o rechazado) aunque podrían existir distintos niveles de decisión según cómo esté implementado el sistema.

Existen tres métodos principales para realizar una fusión a nivel de decisión. El primero es tomando una decisión fuerte (*hard decision*) que consiste en aplicar la función binaria *AND* sobre los resultados binarios de los clasificadores, esto es, cuando todos dicen que un usuario es aceptado se considera aceptado. Este método proporciona alta seguridad, a costa de que puede que el falso rechazo aumente.

El segundo se denomina decisión débil (*soft decision*) consiste en aplicar la función binaria *OR* sobre los resultados binarios de los clasificadores, por lo que si alguno de ellos considera a un usuario, éste es aceptado, independientemente del resultado del resto. Este tipo de fusión permite aplicaciones más amigables para el usuario a costa de incrementar la tasa de falsa aceptación.

Por último se encuentra la decisión ponderada (*weighted decision*) que es una solución intermedia entre ambas, ya que la decisión se toma ponderando la salida de distintos sistemas. Estos pesos pueden seleccionarse *a priori*, o se puede entrenar un algoritmo que asigne los mejores pesos para las distintas salidas de los clasificadores.

2.5. Antispoofing

Los sistemas en los que se requiera una autentificación por parte del usuario son susceptibles de sufrir ataques que vulneren la seguridad de la información de los mismos, en concreto usando técnicas del ámbito de la suplantación de identidad informática o *phishing*. En un sistema biométrico, estos ataques de suplantación de identidad pueden ocurrir en diferentes etapas, como se muestra en la Figura 2.18.

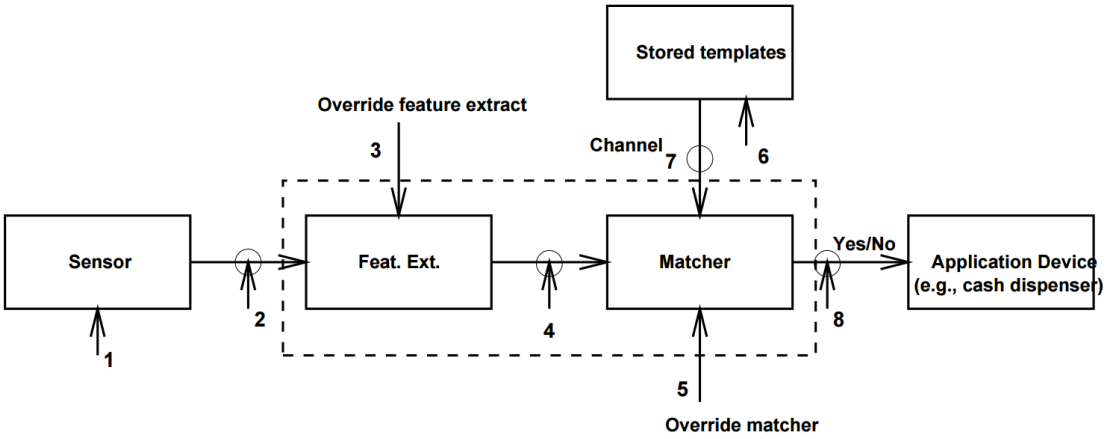

Figura 2.18: Posibles puntos de ataque en un sistema biométrico genérico [7].

Los puntos de ataque que se muestran en la Figura 2.18 son los siguientes [7]:

1. **Nivel de presentación**: se le presenta al sensor un rasgo biométrico falso, como puede ser un dedo de plástico, una máscara, una copia de una firma, una locución grabada, etcétera. Este tipo de ataques son los llamados de *spoofing* y son los que se van a exponer a lo largo de esta sección.

2. **Nivel de señal**: en este tipo de ataque se reproduce una señal grabada tras el sensor. Este tipo de ataque también se le reconoce como ataques de *replay*, pero no hay que confundirlo con los ataques de *spoofing* por *replay* (presentados más adelante): este ataque se realiza interviniendo el dispositivo de captación.

3. **Nivel de extracción de características**: el módulo de extracción de características puede ser atacado con un virus informático de tal manera que produzca el conjunto de características elegidas por el atacante.

4. **Nivel de representación de características**: una vez extraídas las características, estas son reemplazadas por otro conjunto sintético (se asume que la representación de características es conocida). Este tipo de ataques en sistemas biométricos locales son complicados de realizar, pero este problema se vuelve real cuando la extracción de características se realiza en el cliente y se mandan por una red externa como puede ser Internet, donde los paquetes pueden ser alterados, o guardados para posteriores usos. Por ello es importante que el canal de comunicaciones esté cifrado si el sistema es presumible de ser atacado en este punto.

5. **Nivel de comparador**: el comparador se puede atacar para producir siempre de manera directa una puntuación alta (aumenta la falsa aceptación) o baja (aumenta el falso rechazo).

6. **Nivel de base de datos**: los registros de enrolamiento se pueden encontrar almacenados de forma remota o local, y de manera centralizada o distribuida. Este tipo de ataques a la base de datos puede bien modificar los modelos de enrolamiento almacenados provocando autorizaciones fraudulentas o denegación de servicio a los usuarios afectados por una corrupción del modelo.

7. **Nivel de canal**: similar al nivel de representación de características, el canal de comunicación entre la base de datos y el comparador puede verse afectado por alteraciones sobre los modelos.

8. **Nivel de decisión**: si la decisión final del sistema biométrico puede ser modificado por un atacante, el resultado puede ser altamente peligroso, ya que incluso si el sistema de reconocimiento biométrico tiene unas características de alto rendimiento, el resultado es inútil modificando simplemente el resultado del sistema.

Se conoce como *antispoofing* en un entorno de seguridad que requiera autentificación biométrica al conjunto de técnicas que evitan ataques de suplantación de identidad frente el sensor (nivel de presentación) [39]. Se pueden clasificar a los métodos de *antispoofing* en tres categorías: colaborativos, no colaborativos o semicolaborativos, en función de la necesidad de participación por parte del usuario para que este sistema pueda distinguir si ante el sensor hay una persona real o se intenta eludir el sistema con algún tipo de fraude o *spoof*:

- **Métodos de *antispoofing* colaborativos**: requieren colaboración o participación por parte del usuario que se está identificando para detectar que ante el sensor hay un usuario real. Ejemplos de este tipo de métodos destacan los enfoques reto-respuesta, donde se le solicita al usuario que realice una acción no estrictamente necesaria para la identificación del usuario, pero necesaria para la detección de *spoofs*.

- **Métodos de *antispoofing* no colaborativos**: la información necesaria para el método de *antispoofing* está presente en la señal capturada, siendo esta procesada por el sistema *antispoofing* en paralelo al sistema de identificación. Ejemplos de estos tipos de sistemas pueden ser la detección de reflejos y texturas en reconocimiento facial y la caracterización del canal en reconocimiento de voz.

- **Métodos de *antispoofing* semicolaborativos**: estos métodos requieren de la participación por parte del usuario, pero la información que se genera es también útil para la identificación de éste. Un ejemplo de este método para detectar ataques de suplantación ante el sensor en un sistema de verificación por voz sería la generación de una frase aleatoria en el momento de la verificación: el sistema *antispoofing* detecta que la frase que se dice en el momento de la verificación es la misma que la generada, esta frase sirve igualmente para la verificación y el usuario desconoce que está participando en el proceso de detección de ataques mediante *spoofs*.

Debido a los diferentes sensores existentes para capturar cada rasgo biométrico, y también según la capacidad de elusión inherente a la naturaleza de cada rasgo biométrico, tanto los ataques como las técnicas de protección frente a dichos ataques son diferentes. En esta memoria se enuncia el estado del arte en estas técnicas, centrándose de manera específica en los diferentes tipos de ataques de suplantación de identidad ante el sensor y contramedidas ante estos ataques en las biometrías de cara y de voz.

2.5.1. Antispoofing facial

Los métodos *antispoofing* para biometría facial, también llamados *detectores de vida* intentan discernir si lo que hay ante el sensor es una persona, o por el contrario, un fraude.

Según [40] se distinguen tres tipos principales de *spoofs*, o tipos de suplantaciones ante el sensor, ordenadas según la dificultad de detección de los mismos. Se puede ver un ejemplo de estos *spoofs* en la Figura 2.19.

Figura 2.19: Ataques frente al sensor en biometría facial (`http://www.comp.hkbu.edu.hk`).

1. **Representación estática en 2D**, o **fotografía**. Mediante una fotografía de un sujeto se puede realizar un ataque de suplantación de identidad de la manera más barata y sencilla, ya que puede ser fácil extraerla sin costes realizando una fotografía o descargándola de Internet. Mecanismos clásicos para la detección de este tipo de fraude incluyen la detección de vida (gestos faciales, parpadeo, etc.) o el uso de información de profundidad (3D).

2. **Representación dinámica en 2D**, o **vídeo**. Un vídeo incluye variaciones temporales de la cara del sujeto, y puede que los mecanismos *antispoofing* basados en detección de vida, como los gestos faciales, se vean vulnerados. Sin embargo, esta información es más complicada de extraer de sujetos anónimos, ya que conseguir la información de vídeo de un sujeto en las condiciones adecuadas para ser posteriormente verificado no es una tarea fácilmente accesible. Los sistemas de reconocimiento en 3D detectarían que en un vídeo 2D no existe profundidad.

3. **Representación en 3D**, o **máscara 3D**. Las máscaras hiperrealistas en 3D son difíciles de adquirir, debido a la información necesaria del usuario, además de que carecen de movilidad para realizar gestos faciales, aunque podría vulnerar tanto sistemas 2D como 3D.

En [39, 41] se pueden encontrar diferentes métodos de *antispoofing* facial, de entre los cuales se destacan los siguientes:

- **Detección de vida**: como ya se mencionó, la detección de vida tiene en cuenta signos fisiológicos como el pestañeo, o cambios de expresión en la cara o en la boca. El pestañeo es el gesto facial más utilizado con este tipo de técnicas *antispoofing* [40], aunque se realizó una prueba con la demo comercial de reconocimiento facial con *antispoofing* de la compañía KeyLemon[7] resultando fácilmente eludible con vídeos donde el sujeto a identificarse pestañeaba. Es posible necesitar colaboración por parte del usuario para detectar estos cambios, depende de la implementación del sistema.

[7] Ver `https://www.keylemon.com/` (Consultado 04/2017).

- **Análisis de movimiento**: los objetos en 2D como las fotografías y las pantallas con vídeo tienen un movimiento sobre un plano, no en 3D. Esto puede ayudar a detectar presentaciones sobre un plano haciendo uso únicamente de una cámara 2D. Un ejemplo para entender esto sería detectar cambios de pose de la cara del usuario, tanto en dirección vertical, como lateral. Al hacer esto, debido al relieve de la cara, el sujeto ocluiría ciertas partes de la cara con la nariz que de otra forma, con una imagen en 2D no podría conseguir. Es posible que para detectar con seguridad esto se requiera de colaboración por parte del usuario.

- **Reflejos y texturas**: otro método es entrenar modelos con diferentes texturas, como pueden ser papel, pantallas de plasma, pantallas de teléfonos móviles o de *tablets* con el fin de detectar una cara real de caras con texturas. Es posible observar que dependiendo del material podrían provocar unos reflejos que dependen de la superficie donde se esté proyectando la imagen facial, siendo inexistentes si la imagen fuera real. También esta técnica se puede utilizar entrenando texturas de máscaras faciales frente a modelos humanos. Este tipo de sistema *antispoofing* podría no requerir ningún tipo de colaboración por parte del usuario.

- **Información contextual**: diferente información contextual puede ser utilizada para detectar actividad de *spoofing*. Por ejemplo, en entornos controlados donde la verificación es siempre en el mismo sitio (un banco, o una cámara de seguridad fija), el sistema puede tener información de la iluminación y de distintos elementos de fondo que aparecen en la escena. También se puede analizar si aparecen dedos sujetando el contorno de la imagen o del vídeo que sujeten el elemento de presentación o formas cuadradas en los contornos de la imagen o del vídeo, delimitando los bordes del elemento de presentación. Otro tipo de información contextual externa puede ser el uso de otra cámara situada en otro sitio o detectores de presencia, para conocer simplemente la distancia entre el dispositivo de presentación y la cámara, pudiendo determinar que si está demasiado cerca, posiblemente haya intención de fraude, ya que una imagen presentada en un dispositivo móvil es más pequeña que una cara real. Este tipo de sistema podría no requerir de colaboración por parte del usuario de la aplicación.

- **Enfoque reto-respuesta** (*Challenge-Response*): a diferencia de los métodos anteriores, en los cuales podía o no requerirse de la participación por parte del usuario según estuviera implementada el sistema *antispoofing*, este método es colaborativo por excelencia. En este tipo de métodos se solicita al usuario que cambie la pose de la cara (por ejemplo a perfil derecho), que haga un gesto, el cual luego se reconozca. Otro método relacionado con el enfoque reto-respuesta que es altamente combinable con la biometría fusionada de cara y voz es la solicitud de una frase a decir por el usuario, de forma que se reconozca la lectura de los labios.

2.5.2. Antispoofing de voz

Como ya se mencionó anteriormente en la *Sección 2.3: Biometría de voz*, la biometría de voz tiene dos enfoques claramente muy diferenciados: dependiente e independiente de texto. En sistemas biométricos dependientes de texto, generalmente se puede utilizar como método *antispoofing* la generación de una frase aleatoria a decir por el usuario que se quiere autentificar generada en el momento de la verificación (*text prompting*). La verificación de voz dependiente de texto es más segura en este aspecto, en contraposición con la verificación de voz independiente de texto, ya que puede realizar verificaciones con cualquier frase del locutor.

A diferencia de la biometría facial, biometría de voz suele funcionar en entornos distribuidos sin supervisión humana (como por ejemplo en llamadas telefónicas), lo cual hace que la señal

de voz sea más presumible de ser manipulada que otros tipos de biometrías [39].

Se pueden diferenciar cuatro tipos diferentes de *spoofs* en biometría de voz [39]:

- **Imitación**: se conoce como ataques de imitación cuando el sujeto atacante intenta imitar la forma de hablar, entonación y sonidos que realiza el sujeto genuino. Es una forma de *spoofing* natural, y que depende de las habilidades naturales del atacante para realizar la imitación. Según [39], debido a los limitados estudios al respecto y a los pequeños conjuntos de las bases de datos no es sencillo diseñar contramedidas para este tipo de ataque, si bien requiere ciertas habilidades para realizar una imitación buena, depende también de la voz del sujeto que se desea imitar. Este tipo de ataques también hace vulnerables los sistemas biométricos de voz dependientes de texto.

- **Ataques por reproducción (*replay*)**: los ataques por reproducción, también conocidos como ataques de *spoofing* por *replay* (no confundir con suplantación mediante intervención a nivel de señal), consisten en presentar ante el sistema biométrico locuciones previamente grabadas del sujeto genuino. Este tipo de ataques no requieren alta tecnología ni tampoco elevado conocimiento por parte del atacante. Los sistemas *antispoofing* que tratan de evitar este tipo de ataques se basan en caracterizar el canal, ya que al añadir el elemento de grabación y reproducción de voz, las señal se ve afectada por los filtros que modelan los micrófonos y los altavoces que están reproduciendo el sonido. Este tipo de *antispoofing* no requiere explícitamente la colaboración por parte del usuario.

- **Síntesis del habla**: la síntesis de habla generalmente consiste en transformar texto a habla, siendo esta una técnica para generar una voz artificial que suene de la manera más natural posible. Requiere de alta cantidad de habla del sujeto genuino para poder realizar este tipo de ataque, ya que se trabaja con características a nivel de fonema. Actualmente, este tipo de ataques constituyen una amenaza contra los sistemas biométricos. Las contramedidas contra este tipo de ataques se centran en las diferencias acústicas entre los *vocoders* o productores sintéticos de voz y la voz natural, sobre todo en la fase de la señal de voz, así como en los saltos de la frecuencia fundamental de la señal, que caracteriza la prosodia. Al igual que la imitación, estos ataques también vulneran los sistemas dependientes de texto. Por contra, este tipo de ataques requieren gran cantidad de audio del usuario a atacar que es poco probable que pueda realizarse con éxito en usuarios anónimos.

- **Conversión de voz**: a diferencia que la síntesis del habla la conversión de voz genera voz sintética del usuario genuino a partir de voz del usuario atacante, en vez de utilizar texto como entrada, minimizando una función que transforma la señal de habla del usuario atacante a la señal de habla del usuario genuino, generalmente modificando el filtro del tracto vocal y la entonación, entre otros parámetros. Estudios [42] determinan que clasificadores utilizando SVM son naturalmente robustos a este tipo de ataques ya que se detectan las diferencias de naturalidad y variabilidad dinámica características del habla natural. Otras formas de detección es el análisis de la magnitud y la fase de la señal.

Los métodos basados en la sugestión de texto (*text prompting*) hace que los sistemas más comunes de *spoofing* como son los ataques por reproducción no sean efectivos a la hora de burlar el sistema. Como se vio en la *Sección 2.3: Biometría de voz*, con sistemas de reconocimiento de locutor dependientes de texto se podría reconocer una frase o secuencia de palabras concreta. Si el sistema es dependiente de texto, el sistema *antispoofing* se podría considerar semicolaborativo, ya que la voz recogida se utiliza indistintamente para identificar al usuario como para evitar ataques de suplantación por reproducción.

Debido a que los sistemas de reconocimiento de locutor más comunes son los sistemas de reconocimiento independientes de texto, se pueden combinar el sistema de reconocimiento de

locutor independiente de texto con un sistema de reconocimiento de voz (o de mensaje) con el fin de, con el primero, identificar al locutor, y con el segundo, asegurar que la frase que éste dice es la frase sugerida. Al hacer esto el sistema *antispoofing* se vuelve colaborativo, ya que el usuario participa diciendo un texto no necesario explícitamente para la identificación.

3

Bases de datos

En esta sección se presentan las características de las bases de datos utilizadas en este Trabajo de Fin de Máster. Como el objetivo de este trabajo es la implementación de un sistema de verificación facial y de voz en un entorno móvil, se ha querido utilizar y adaptar las bases de datos públicas ya existentes a los requerimientos de la aplicación real. Por ello, se ha debido definir qué se considera una sesión para realizar el entrenamiento de los modelos de cara y de voz, siendo para los modelos biométricos faciales una secuencia de imágenes donde aparezca una cara tomada de manera consecutiva y para voz una locución corta (entre 5 y 10 segundos).

La estructura de la base de datos que se diseñó para este proyecto —juntando los diseños individuales que se hicieron tanto para cara como para voz— es la que se muestra en la Figura 3.1, donde se diferencian los siguientes niveles diseñados para que tengan una estructura similar la base de datos facial y la de voz:

1. **Nivel raíz**: este nivel contiene el directorio donde se almacenan todas las bases de datos.
2. **Nivel de rasgo biométrico**: en este nivel se diferencian las bases de datos por rasgo biométrico. En nuestro caso, solo se tienen dos rasgos biométricos: la cara y la voz.
3. **Nivel de base de datos**: este nivel tiene diferentes bases de datos, públicas o privadas, con información de cada rasgo biométrico. Cada base de datos puede tener un número variable de usuarios.
4. **Nivel de usuario**: cada usuario es único dentro de una base de datos y se procura que sea también único entre bases de datos, evitando repeticiones de usuarios debido a ampliaciones de éstas o que tengan el mismo origen.
5. **Nivel de sesión**: cada usuario puede tener diferentes sesiones. Las sesiones contienen información recogida en un mismo día o en unas mismas condiciones ambientales (de luz para imágenes o de ruido para locuciones). Dependiendo del tipo de biometría, se requerirá también que cada sesión contenga elementos con las características más similares a la situaciones reales. Estas sesiones pueden funcionar como sesiones de entrenamiento y de test, de manera diferente en la biometría facial que en la de voz, como se verá más adelante. Finalmente, estas sesiones contienen los elementos utilizados para entrenar los verificadores, siendo en el caso de reconocimiento facial imágenes de caras y en el caso de verificación de voz locuciones.

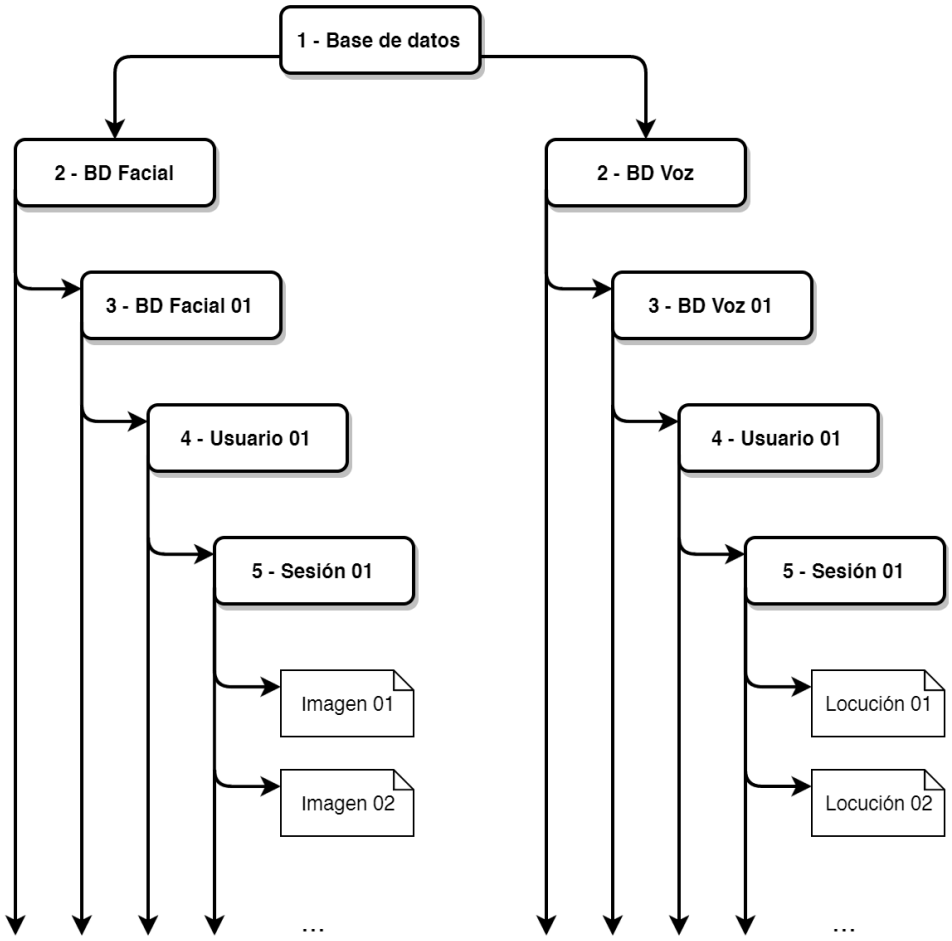

Figura 3.1: Estructura de la base de datos.

En esta sección se dividen las actividades realizadas en dos apartados, según los objetivos específicos descritos en el *Anexo A: Objetivos específicos*:

- **Creación de una base de datos de imágenes de caras para probar algoritmos de verificación facial (Base de datos facial)**: en este apartado se exponen las características que requiere nuestro sistema para evaluar los algoritmos de verificación facial desarrollados, y las bases de datos públicas y privadas consideradas para acometer dicha tarea.
- **Creación de una base de datos de voz para probar algoritmos de reconocimiento de locutor (Base de datos de voz)**: en este apartado se presentan las características de las locuciones que componen la base de datos de voz, prestando especial atención a la elección de frases fonéticamente balanceadas para aplicaciones comerciales en entornos móviles.

3.1. Base de datos facial

La aplicación que se desea desarrollar consiste en un sistema de verificación biométrico en entorno móvil de cara y voz, por ello, como se dijo anteriormente, se han de buscar y adaptar las bases de datos públicas a los requerimientos de la aplicación. Junto con estas bases de datos

públicas, Argos Global disponía de una base de datos privada en entornos móviles, con la cual también se ha trabajado haciendo uso de ella.

Los requisitos que se decidieron para la selección de las bases de datos públicas a utilizar en esta etapa de desarrollo de la aplicación de verificación facial fueron los que se muestran en la Tabla 3.I.

Tabla 3.I: Características de la base de datos facial.

Característica	**Propiedades**
Imágenes por usuario	5+
Entorno e iluminación	Controlado | Semicontrolado
Sesiones	2+
Variación de gesto (s)	Mínimo
Variación de iluminación (s)	Leve | Media | (Alta)
Variación de detalles (s)	Leve
Oclusiones	No
Pose	Frontal
Tamaño de las imágenes	150x150 +
Color	Gris | Color
Etnia	Cualquiera
Edad	Adulta
Sexo	Cualquiera

Nota: (s) quiere decir «entre sesiones».

Se puede ver que los requisitos que se solicitan para la selección de la base de datos de cara (ver Tabla 3.I) son los siguientes:

- **Imágenes por usuario**: el número de imágenes mínimo dentro de la misma sesión y misma secuencia de captura debe ser de 5. La frecuencia de muestreo entre imágenes puede estar comprendida entre 10 y 30 imágenes por segundo.
- **Entorno e iluminación**: el entorno donde se realizan las capturas se refiere al lugar donde se realizan. Esto incluye que solo aparezca una cara predominante en la secuencia de imágenes, que la iluminación sea suficiente, que la imagen aparezca centrada en el marco y no cortada. En cuanto a iluminación, ha de ser interior o exterior, suficientemente iluminada, evitando sombras marcadas o grandes fuentes de luz (luz solar directa).
- **Sesiones**: el número de sesiones ha de ser de dos o superior. En el caso en el que haya suficientes imágenes por usuario, bases de datos con una única sesión se podrá dividir en dos sesiones. Al menos una de las sesiones debe estar en condiciones de entorno controladas, para realizar el entrenamiento del modelo.
- **Variación de gesto**: las variaciones de gesto entre sesiones deben ser mínimas. Esto quiere decir que en algunas bases de datos públicas diseñadas para reconocimiento de gesto se tendrá que evaluar qué gestos pueden ser utilizados para introducir en la base de datos diseñada para esta tarea.
- **Variación de iluminación**: la variación de la iluminación entre sesiones es preferible que sea leve o media. Adicionalmente se añadió un nivel alto de variabilidad para ver cómo se comportaban algoritmos invariantes a la luz, pero no se utilizaron en este trabajo.
- **Variación de detalles**: la variación de detalles entre sesiones se refiere a la incorporación o substracción de objetos o alteraciones sobre la cara del individuo entre sesiones. Se

admite el uso de gafas con cristales transparentes o maquillaje. No se admite, por ejemplo elementos oclusivos, tales como gafas de sol, bufandas que cubran la boca y/o nariz u oclusiones de la imagen con objetos.

- **Oclusiones**: como ya se ha mencionado, no se admiten elementos oclusivos, parcialmente oclusivos, o semitransparentes en la imagen facial.
- **Pose**: la pose de la imagen ha de ser frontal, y en la medida de lo posible enderezada con los marcos de la misma. Se admite un máximo de 5º de rotación tanto sobre el plano de la imagen, como en la pose (vertical y horizontal) con respecto al dispositivo de captación.
- **Tamaño de las imágenes**: el tamaño de las imágenes ha de ser superior a 150x150, o en ella se ha de encontrar, al menos, una imagen con una cara de ese mismo tamaño. En esta base de datos no se utilizan imágenes de caras donde únicamente aparezca la cara, ya que se pretende evaluar también la efectividad del detector de características.
- **Color**: debido a que en esta fase del proyecto se trabaja con imágenes en blanco y negro, las imágenes de la base de datos pueden ser en escala de grises o en color. En un futuro se requerirá que todas las imágenes sean en color.
- **Etnia**: los sujetos de la base de datos pueden ser de cualquier etnia.
- **Edad**: los sujetos de la base de datos deberán comprenderse en una franja de edad adulta, de manera general. Excepcionalmente, pueden ser jóvenes (+15 años) o de avanzada edad.
- **Sexo**: los sujetos de la base de datos pueden ser de cualquier sexo, masculino o femenino.

Definidas estas características de la base de datos, se seleccionaron las siguientes bases de datos públicas para evaluar los modelos de verificación facial: FERET, Cohn-Kanade, Cohn-Kanade+, YaleB, NIST, yalefaces y ND2006.

Adicionalmente se utilizó una base de datos privada de Argos Global, denominada FOnFaces que cuenta con 67 usuarios, con 15 imágenes por usuario, entornos controlados y semicontrolados, 5 sesiones por usuario (variaciones de luz, diferentes dispositivos de captación: móvil, *tablet*, *webcam*), variación de gesto entre sesiones mínimo, variación de iluminación entre sesiones leve y media, variación de detalles leve, sin oclusiones, pose frontal, imagen a color, etnias caucásica ($\sim 70\,\%$), latina ($\sim 20\,\%$) y afro-americana ($\sim 10\,\%$), de edad adulta y con ambos sexos en una proporción próxima al $50\,\%$.

La base de datos al completo está formada por el conjunto de estas bases de datos.

Junto con esta base de datos, se comenzó a diseñar una base de datos de caras para *antispoofing* con ataques por reproducción de fotografía en papel, fotografía revelada, pantalla reflejante y pantalla mate, la cual no se dará más detalles en esta memoria al no realizarse evaluaciones del sistema *antispoofing* en este TFM.

3.2. Base de datos de voz

A diferencia del reconocimiento facial, la verificación de voz tiene un diseño diferente. Para modelar la voz del individuo en el enrolamiento es necesaria más información que a la hora de verificarlo —tal y como está diseñado el sistema de verificación que se mostrará más adelante—, y por tanto la base de datos ha de contemplar estas variaciones en la cantidad de datos para entrenar y para testear.

La creación de la base de datos de voz fue de las últimas tareas que se llevaron a cabo en este TFM, por lo tanto no es tan completa como la base de datos facial anteriormente descrita. Para realizar esto se decidió que esta base de datos fuera en castellano, y para los modelos de entrenamiento se utilizara un *corpus* fonéticamente balanceado (proveniente de la palabra inglesa *balanced*, equilibrado), es decir, donde todos los sonidos estén aproximadamente igual de representados.

En el estado del arte se pueden diferenciar tres tipos de *corpora* fonéticamente balanceados según la longitud del texto: texto fonéticamente balanceado, frases fonéticamente balanceadas y palabras fonéticamente balanceadas.

El texto fonéticamente balanceado consiste en un fragmento de texto, de entre 30 y 40 segundos de duración donde la totalidad de éste hace un conjunto fonéticamente balanceado. Como ejemplo de este *corpus* en español se encontró el siguiente texto[12]:

> El joyero Federico Vanero ha sido condenado por la audiencia de Santander a ocho meses de arresto mayor y cincuenta mil pesetas de multa por un delito de compra de objetos robados. La vista oral se celebró el miércoles pasado y, durante ella, uno de los fiscales, Carlos Valcárcel, pidió para el joyero tres años de prisión menor y una multa de cincuenta mil pesetas. Gracias a las revelaciones de Vanero de hace dos años y medio se llegó a descubrir la existencia de una sospechosa mafia policial en España, parte de la cual se vio envuelta en el llamado "caso el Nani".

Las frases fonéticamente balanceadas son locuciones de entre 5 y 7 segundos, siendo como ejemplo de estas en español el ya mencionado *Sharvard Corpus* [30], formado por 70 conjuntos de 10 frases cada uno, siendo cada conjunto fonéticamente balanceado:

1. Hay gemas de gran valor en la tienda.
2. Tienen un nuevo puerto para botar barcos.
3. Para hacer hielo, echa más agua.
4. Mandó la postal en un sobre de papel grueso.
5. Se sintió feliz cuando vio llegar el tren.
6. Plantan un árbol en el bosque todos los años.
7. El casco viejo de la ciudad es muy amplio.
8. Cosió la capa rota con mucha fuerza.
9. No lleves jersey en un día como éste.
10. El reloj de la plaza marcó las cuatro de la tarde.

Finalmente, los conjuntos de palabras fonéticamente balanceadas consisten en conjuntos de palabras, bisílabas o trisílabas que forman un conjunto fonéticamente balanceado. En el momento de la realización de este trabajo no se encontró un *corpus* en español castellano con estas frases, aunque sí se encontró uno en dialecto argentino[3].

Con vistas a la aplicación a diseñar se descartó utilizar texto fonéticamente balanceado en español por dos motivos: el primero la carencia de este tipo de textos largos en la literatura con respecto a, por ejemplo, las frases fonéticamente balanceadas, y el segundo, debido a que

[1]Bruyninckx, M., Harmegnies, B., Llisterri, J. y Poch, D. (1994). Language-Induced voice quality variability in bilinguals. Journal of Phonetics, 22(1), 19-31.

[2]Más textos disponibles en `http://liceu.uab.es/~joaquim/phonetics/fon_esp/Textos_equilibrio_fonetico_espanol.html` (Consultado 04/2017).

[3]Ver Julieta D'Onofrio, Alejandro E. Navarro. Listas de palabras fonéticamente balanceadas del Dr. Tato y cols. vs. Listas de palabras P.I.P.-C 25. *Mutualidad Argentina de Hipoacúsicos*, 2012.

la adquisición de un texto tan largo podría ser poco práctico a la hora de ser adquirido en una aplicación móvil, donde puede ser cortado a causa de interrupciones o puede que el usuario no esté dispuesto a grabarse recitando un texto de larga dimensión. Para el uso de palabras fonéticamente balanceadas se valoraron dos enfoques: el primero grabar individualmente cada palabra, lo cual era de nuevo impracticable, puesto que los *corpus* consistían en aproximadamente 25 palabras, y el segundo grabarlo de manera ininterrumpida, lo cual, de nuevo, era poco práctico para la aplicación dada. Finalmente se escogió el uso de las frases fonéticamente balanceadas, que además de tener un *corpus* lo suficientemente extenso para poder trabajar con él, se podía diseñar la aplicación de manera sencilla para poder adquirir, por parte del usuario, la información de voz necesaria usando estas frases.

Existen otras bases de datos, como la de BioSec [43] descartada al recogerse dígitos en vez de frases fonéticamente balanceadas. Se valoró utilizar también utilizar BioSecurID [44] basada en el *corpus* fonéticamente balanceado AHUMADA [45], pero finalmente, para la fase de pruebas preliminares (antes de evaluación de los sistemas de verificación de locutor y de fusión) se decidió diseñar una base de datos propietaria denominada FOnVoice por los motivos que se exponen más adelante.

Esta base de datos está formada por 17 usuarios (adultos, 9 hombres y 8 mujeres de habla nativa española), donde se recogieron 8 sesiones tomadas en 3 días distintos repartidos en un mes: dos sesiones con 10 frases fonéticamente balanceadas distintas (Listas 1 y 2 del *Sharvard Corpus* [30]), recogidas con dos dispositivos de captación distintos (micrófono capacitivo omnidireccional de un terminal móvil situado a 30 cm del usuario, y utilizando un *headset* de la marca *Sennheiser* posicionado a menos de 5 cm de la boca del locutor); y seis sesiones con 10 frases cada una, no fonéticamente balanceadas: dos sesiones recogidas en buenas condiciones de grabación (sala con poca reverberación, silenciosa), otras dos sesiones en condiciones de grabación aceptables (parque, a 10 metros de una fuente) y dos sesiones recogidas en entornos de grabación ruidoso (cafetería en hora punta); siendo una sesión de cada entorno grabada con el micrófono omnidireccional del dispositivo móvil, y otra con el *headset*. Estas frases no fonéticamente balanceadas tenían entorno a 5 segundos de duración y su contenido era de carácter arbitrario (citas, frases célebres, eslóganes, etcétera). La Tabla 3.II muestra de manera más gráfica la composición de las sesiones de la base de datos FOnVoice.

Tabla 3.II: Composición de las sesiones de la base de datos FOnVoice.

Sesión	Frases	Micrófono	Entorno
1	10 - Balanceadas Sharvard - Lista 1	Omnidir. Día 1	Bueno
2	10 - Balanceadas Sharvard - Lista 2	Headset Día 3	Bueno
3	10 - Aleatorias Citas - Lista 1	Omnidir. Día 2	Bueno
4	10 - Aleatorias Citas - Lista 1	Headset Día 1	Bueno
5	10 - Aleatorias Citas - Lista 2	Omnidir. Día 1	Levemente Ruidoso
6	10 - Aleatorias Citas - Lista 2	Headset Día 2	Levemente Ruidoso
7	10 - Aleatorias Citas - Lista 3	Omnidir. Día 3	Ruidoso
8	10 - Aleatorias Citas - Lista 3	Headset Día 1	Ruidoso

El uso de esta base de datos se justifica debido a que es una base de datos recogida por Argos Global durante la realización de los sistemas de evaluación de algoritmos de reconocimiento de locutor y de fusión biométrica que se presentan en la *Sección 6: Verificación de locutor y fusión biométrica de cara y voz*, correspondiente a la última etapa de este trabajo, por lo tanto para verificar que estos sistemas funcionaban, este número de usuarios era suficiente, ya que en ese momento primaba la elaboración del *software* a la evaluación de los diferentes algoritmos. En la actualidad, Argos Global está aumentando el número de usuarios de la base de datos FOnVoice, además de ampliar este conjunto con otras bases de datos de voz en español y en inglés (británico y americano) que cumplan con los requisitos mínimos de tener un conjunto de frases fonéticamente balanceadas para entrenar (entorno a 10, de ~ 5 segundos de duración) y otro conjunto de frases no necesariamente fonéticamente balanceadas para medir el rendimiento (mínimo 10, también de ~ 5 segundos de duración). Requisitos adicionales para la inclusión de una nueva base de datos de voz es que se presenten sesiones con variabilidad de canal y de ruido en el entorno.

4

Desarrollo y evaluación de un sistema de verificación facial

Esta sección cubre los contenidos relacionados con el diseño y desarrollo de una herramienta informática que permite medir el rendimiento de los algoritmos de detección, modelado y verificación facial implementados en la empresa Argos Global. La motivación por la cual se decidió llevar a cabo este sistema de evaluación es poder realizar un ajuste de parámetros en las distintas etapas del sistema de verificación facial (ver Figura 2.3 para más detalles) con el fin de obtener aquella combinación de parámetros que permita obtener los mejores resultados, pudiendo hacer uso de la base de datos descrita en la *Sección 3.1: Base de datos facial*.

En esta sección se dividen las actividades realizadas en tres apartados, según los objetivos específicos descritos en el *Anexo A: Objetivos específicos*:

- **Desarrollar un sistema de evaluación de algoritmos de detección, modelado y verificación facial en C/C++**: en este apartado se explica el diseño de la herramienta informática desarrollada por el autor del TFM.
- **Investigar diferentes algoritmos de detección, modelado, verificación y fusión de características de cara con vistas a obtener mejores resultados en la verificación**: en este apartado se exponen las principales actividades del ámbito de la investigación presentadas en este TFM.
- **Diseñar un sistema de antispoofing facial**: en este apartado se explica el diseño de un sistema de antispoofing facial basado en texturas, donde el autor ha contribuido en el diseño.

4.1. Desarrollo de un sistema de evaluación de algoritmos de detección, modelado y verificación facial en C/C++

El sistema de evaluación de algoritmos es un programa que hace uso de la base de datos facial recogida para evaluar la bondad de los distintos algoritmos utilizados de verificación facial (*Software* de evaluación de algoritmos de verificación facial en C/C++[1]). Este sistema

[1]Entregable E-4.1.1. Ver *Anexo A: Objetivos específicos*.

es diseñado e implementado desde cero por el autor, no habiendo referencias anteriores de este tipo de sistemas en Argos Global.

Este sistema se ha decidido programar en el lenguaje C/C++ por tres motivos: el primero es porque ya existía en Argos Global trabajo desarrollado en estos lenguajes, el segundo es para poder hacer uso de la librería de Visión por Computador en C++ OpenCV y el tercero es porque tanto C como C++ son lenguajes de alto nivel muy potentes, permitiendo ejecuciones rápidas y pudiendo ser fácilmente paralelizables.

Como se explicará en la *Sección 4.2: Investigación de diferentes algoritmos de detección, modelado, verificación y fusión de características de cara*, en un principio se parte de un sistema implementado en Argos Global, del cual se desea conocer su rendimiento sobre la base de datos adquirida. Este sistema inicial no calcula las puntuaciones o *scores*, únicamente decía si un usuario estaba aceptado o rechazado en verificación, por lo que se ha de evaluar los resultados en diferentes puntos de trabajo, variando el umbral de decisión. A partir de este sistema inicial, se empiezan a construir sistemas que devuelven las puntuaciones, por lo que permiten medir el rendimiento en distintos puntos de trabajo, los cuales son desarrollados en su totalidad por el autor del trabajo. Por este motivo, en este sistema se desarrollan dos versiones principales del sistema de evaluación de algoritmos: en la primera se permite la inserción del un umbral de decisión, y en la segunda se devuelven las puntuaciones del comparador.

Internamente, el sistema de evaluación se divide en cinco funciones principales, para poder realizar la tarea de comparación:

1. **Iniciación**: en esta función se inicia el sistema, de donde se lee un fichero de configuración.
2. **Detección**: se detectan caras en imágenes según los parámetros escogidos en el fichero de configuración.
3. **Generación de modelos**: se generan los modelos utilizando el procedimiento escogido en el fichero de configuración.
4. **Comparación de modelos**: se comparan los modelos haciendo uso de una medida de comparación escogida en el fichero de configuración.
5. **Resultados**: genera un fichero con un resumen de los resultados.

Este sistema permite guardar los modelos creados en el apartado de generación de modelos, con el fin de no tener que volver a computarlos en el caso en el que se desee probar distintas medidas en la parte de comparación de modelos.

El fichero de configuración es un archivo de texto plano que permite configurar el sistema de evaluación de algoritmos para que se ejecuten distintas configuraciones de parámetros. En este fichero se incluyen parámetros como el nombre de la simulación, el fichero de base de datos (descrito en el siguiente párrafo), el directorio de trabajo, si existen modelos previos, la versión del sistema de evaluación, tipo de detección, tipo de generación de modelo, tipo de comparación, número de imágenes utilizadas para la generación de los modelos, el umbral de decisión (implementado únicamente para la versión 1 del sistema) y una descripción de la prueba de evaluación a realizar. Este fichero se lee como un diccionario y si alguno de los datos está incompleto se toman valores por defecto.

El fichero de base de datos se genera de manera externa y depende de la base de datos que se desea utilizar. En él se determinan qué sesiones de los distintos usuarios de la base de datos se utilizan para entrenar el modelo y cuáles se utilizan para test. Se determinó que para cada usuario al que se ha generado un modelo de entrenamiento se compararía con tantas sesiones

de test del mismo usuario (genuinos) como sesiones útiles[2] tuviera (a parte de la utilizada para entrenar el modelo), y con el mismo número de sesiones de test de otros usuarios (impostores). De esta forma, el número de comparaciones de los modelos enrolados con usuarios genuinos e impostores es el mismo. Tanto la elección de cuál es el modelo de entrenamiento, como la de cuáles son las sesiones de impostores de test se realiza de forma pseudo-aleatoria, de forma que puedan reproducirse los resultados.

El fichero de resultados que genera el sistema muestra simplemente métricas utilizadas para evaluar de manera preliminar qué tan bien funciona el conjunto de parámetros seleccionados. En este fichero, además de un resumen de los parámetros utilizados, se presentan las métricas EER, FMR al 0 %, 0,01 %, 0,1 % y 1 % y el Área Bajo la Curva (AUC, *Area Under the Curve*). Además de estos resultados se genera también un fichero de puntuaciones, donde se guarda la puntuación (o *score*) de las distintas comparaciones.

La Figura 4.1 muestra cómo se realizó la implementación de este sistema. En azul se pueden observar las cinco funciones implementadas ya mencionadas: Iniciación, Detección, Generación de modelos, Comparación de modelos y Resultados. También se observa que la comunicación con el sistema se realiza mediante el fichero de configuración y el fichero de base de datos. Como respuesta se obtienen los ficheros de resultados y de puntuaciones. Además, los resultados de las etapas previas (imágenes de detecciones de caras y modelos faciales) se almacenan con el fin de no tener que computarlos en siguientes ejecuciones del sistema de evaluación, si así se indica en el fichero de configuración.

En los siguientes apartados se enuncian los diferentes tipos de detección implementados, las características empleadas para generar los modelos y cómo se han realizado las comparaciones en éstos utilizando la herramienta de evaluación descrita.

4.1.1. Detección facial y preprocesado

Para la detección facial se utiliza el algoritmo Viola-Jones, descrito en la *Sección 2.2.1: Detección de caras en imágenes*, modificándolo para nuestras necesidades. La motivación de utilizar este algoritmo en lugar de algoritmos más avanzados (como redes neuronales profundas) viene dada por dos factores: el primero es que este algoritmo es fácilmente implementable y rápido en ejecutarse, funcionando bien en entornos semicontrolados, y el segundo es que las subimágenes de cara que se desea encontrar aparecen en condiciones de pose y oclusión que son deseables para el verificador, de tal forma que nos sirve como un primer filtro para rechazar imágenes en unas condiciones de calidad no aptas para la aplicación de verificación en situaciones de seguridad.

Los filtros Haar se han entrenado utilizando una base de datos pública destinada para tal propósito, diferente a las bases de datos empleadas para evaluar la verificación. Se han entrenado filtros Haar para detectar caras, ojo derecho, ojo izquierdo, nariz y boca de manera independiente.

La detección de imágenes se realiza utilizando los filtros desde el centro de la imagen hacia los extremos, buscando una única imagen de la cara, la de mayor tamaño en la imagen, que es la que el sistema usará para generar el modelo. Este método permite reducir el tiempo de búsqueda, ya que se sabe que generalmente la cara del individuo a verificarse se encuentra en el centro de la imagen. Una vez encontrada la cara principal en la imagen se procede a buscar los distintos elementos de la cara (ojos, nariz y boca) en la subimagen de la cara utilizando también filtros Haar. La búsqueda de estos elementos de la cara se hace también haciendo

[2]Se denominan sesiones útiles a aquellas que cumplen los requesitos determinados por el fichero de configuración. Por ejemplo, el número de imágenes para generar un modelo es superior al número de imágenes presentes en una sesión, esta sesión no es útil, y por tanto se descarta.

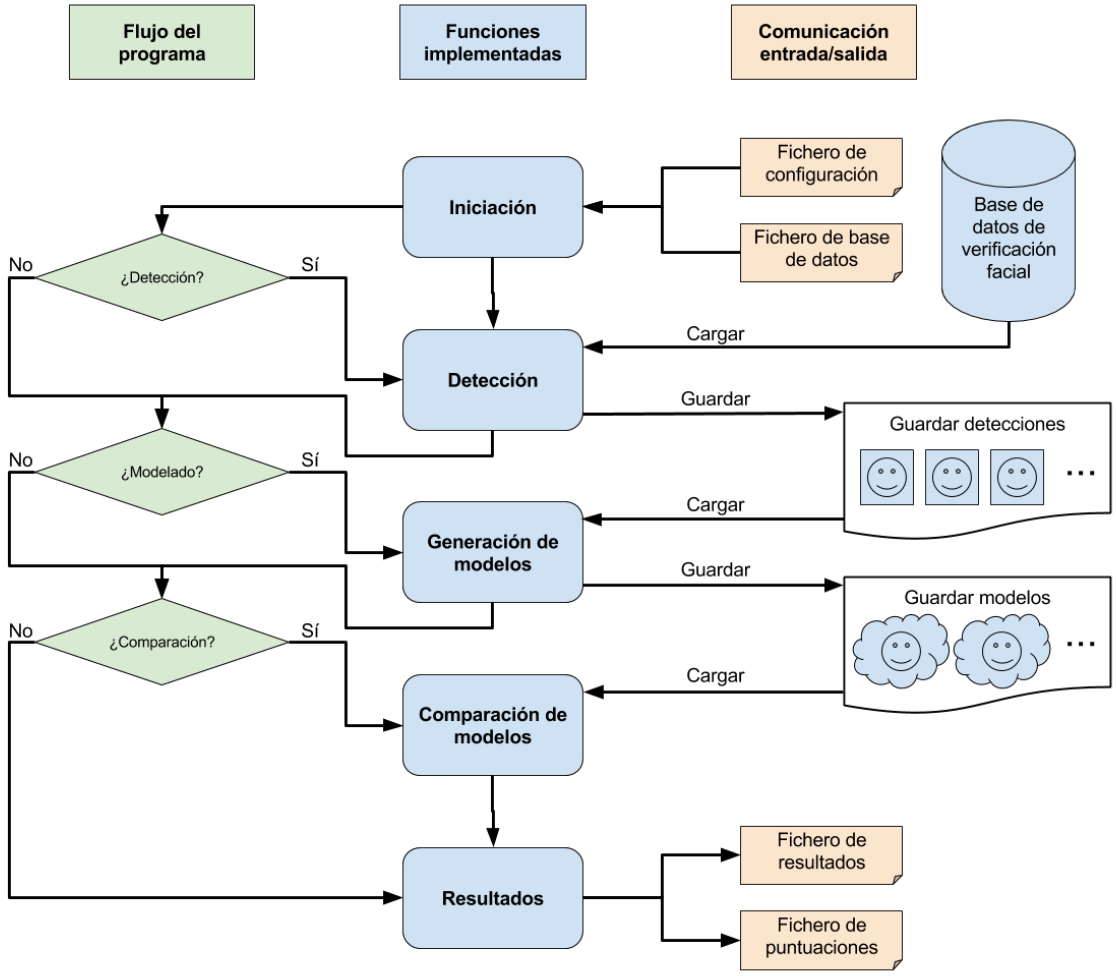

Figura 4.1: Sistema de evaluación de algoritmos de verificación facial.

uso de información de dónde es más probable encontrar estos elementos en la cara, utilizando la información de un estudio realizado previamente en Argos Global. Esta detección se realiza sobre las imágenes en escala de grises, tiene una duración de decenas de milisegundos por imagen (influye la localización de la cara en la imagen, la resolución de la imagen y elementos como la pose o la iluminación) y se realiza de forma paralela para las distintas imágenes.

Además de esta detección se realiza un preprocesado de las imágenes de cara consistente en los siguientes pasos:

- **Normalización del tamaño de la imagen**: una vez tomada la imagen de la cara, esta se convierte en una imagen cuadrada con el fin de que todas las imágenes capturadas tengan las mismas proporciones. Los mecanismos que se evalúan para hacer esto es bien un recorte de la imagen para conseguir que sea cuadrada o bien una deformación de la imagen.
- **Reescalado de la imagen**: la imagen de la cara se reescala a un menor tamaño, el mismo para todos los modelos, con el fin de reducir el tiempo de proceso de la generación y la comparación de los modelos.
- **Contraste adaptativo**: se puede utilizar el contraste adaptativo o CLAHE (*Contrast*

Limited Adaptive Histogram Equalization) con el fin de ayudar a combatir los problemas que vienen dados por la iluminación.

4.1.2. Generación del modelo

Para la generación del modelo se han utilizado variaciones sobre dos características principales descritas en la *Sección 2.2.2: Características y modelado facial*: *Local Binary Patterns* (LBP, en adelante) y *Weber Local Descriptor* (WDA, en adelante). La utilización de características LBP se escogió en un estudio de Argos Global previo a la realización de este trabajo, al dar mejores resultados que *Fisherfaces* y *Eigenfaces* para acometer la tarea de verificación.

Como ya se dijo anteriormente, estas características de primera generación pueden ser aplicables de manera eficaz en entornos móviles de manera local, ya que su extracción se realiza sin el elevado coste temporal que puede llevar entrenar modelos de generaciones más avanzadas. Estas características se extraen y se combinan de manera diferente utilizando tanto las distintas imágenes de caras extraídas por sesión, como los elementos de la imagen de cara extraídos adicionalmente por cada imagen.

Aunque la evaluación de las características WDA se realizó utilizando el *software* de evaluación de algoritmos aquí descritos, el proceso de evaluación no fue realizado por el autor del TFM.

4.1.3. Comparación de modelos

Para la comparación de modelos se ha entrenado un sistema basado en distancias, de nuevo, debido al bajo coste computacional que esto implica y su adecuación para utilizarlo en un entorno móvil. Las distancias que han sido utilizadas en esta apartado están descritas en la *Sección 2.2.3: Comparación basada en distancias*, a saber, distancia Euclídea, distancia Coseno, distnacia Chicuadrado, Distancia Bhattacharyya, distancia Manhattan y distancia Mahalanobis.

Adicionalmente, se realiza una fusión de características utilizando los elementos de la cara para comprobar si la combinaciones de los elementos ofrecen o no mejores resultados, la cual se explica en la siguiente sección.

4.2. Investigación de diferentes algoritmos de detección, modelado, verificación y fusión de características de cara

En esta sección se explican algunas de las pruebas realizadas en la empresa Argos Global para seleccionar los parámetros que forman el mejor sistema de verificación facial. Esta parte de la memoria es la que está más relacionada con la parte de investigación, junto con la exploración de las tecnologías presentadas en la *Sección 2: Estado del arte*.

Esta sección se divide en cuatro apartados según la naturaleza de las pruebas realizadas: evaluación de preprocesado, evaluación de comparaciones, evaluación de fusión intra-rasgo y otras evaluaciones. Para las evaluaciones de los algoritmos, se tomará como medidas de rendimiento el punto de Igual Tasa de Error (EER), el punto de *False Match Ratio* (FMR) al 1 % y el Área Bajo la Curva (AUC). Además, para algunos experimentos se mostrarán las curvas *Detetion Error Tradeoff* (DET) donde se muestra la tasa de falso negativo en función de la tasa de falso positivo, en escala logarítmica.

4.2.1. Evaluación de preprocesado

Al principio, cuando se empezó a trabajar en este proyecto donde se requería una aplicación de verificación facial móvil, se disponía de un sistema de verificación facial inicial desarrollado por Argos Global, de primera generación, que sirvió de base para comenzar a realizar experimentos con el ánimo de mejorarlo. Se decidió partir de un sistema de primera generación debido a que requiere de pocos datos para entrenarlo, su procesamiento es lo suficientemente ligero como para poder implementarlo en dispositivos móviles y los modelos ocupan poco espacio, lo cual es una ventaja a la hora de trabajar con dispositivos móviles.

Este sistema inicial estaba construido con tecnología propietaria de Argos Global y consistía de una serie de funciones en C/C++ para realizar verificaciones faciales. Para seleccionar el punto de trabajo de esta aplicación (situación del umbral de decisión) existía un parámetro de umbral que variaba entre los valores 0 a 10, siendo 0 un punto de trabajo con con alta falsa aceptación y bajo falso rechazo (sistema de fácil usabilidad) y 10 un punto de trabajo con alto falso rechazo y baja falsa aceptación (sistema de alta seguridad). Además, mediante el uso de este código no se podía disponer de las puntuaciones resultantes, por lo que se tuvo que estimar de manera aproximada diferentes puntos de trabajo de la aplicación variando el parámetro de umbral de decisión, pudiendo así calcular los valores de EER y FMR@1 % de partida del sistema, los cuales se muestran en la Tabla 4.I.

Tabla 4.I: EER y FMR@1 % del sistema inicial.

EER	**FMR@1 %**
13,3 %	48,3 %

Para la estimación de estos valores se utilizó de manera conjunta base de datos FOnFaces, junto con la FERET y la Cohn-Kanade de manera conjunta, utilizando el sistema de evaluación de algoritmos de verificación facial descrito en la sección anterior, donde se utilizan 5 imágenes para entrenamiento y verificación, modelo basado en LBPs, reescalado de la imagen a 90x90 píxeles, realizando un recorte de la imagen detectada en los casos necesarios (el proceso completo de este algoritmo de detección se explica más adelante).

Debido a que este conjunto de funciones utilizaba una métrica derivada de la distancia Chi-cuadrado, resultante de un estudio realizado por Argos Global el cual no se disponía en el momento de trabajar con estas funciones, se tomó la decisión de utilizar como primera aproximación el uso de la distancia Euclídea para los siguientes experimentos, con el ánimo de poder evaluar las puntuaciones haciendo uso de las puntuaciones o *scores* generados por el sistema de evaluación de algoritmos, obteniendo los resultados mostrados en la Tabla 4.II.

Tabla 4.II: EER y FMR@1 % del sistema inicial utilizando la distancia Euclídea.

EER	**FMR@1 %**
14,9 %	57,3 %

A partir de este resultado de partida, se comenzó a realizar mejoras en el sistema de detección de imágenes de caras. Para ello se definió los siguientes experimentos de detección facial:

- ***Detección_0* - Detección del sistema inicial (S.I.)**: este sistema detecta caras en imágenes utilizando la adaptación del algoritmo Viola-Jones realizado por Argos Global (descrito en la sección anterior). Además detecta distintos elementos de la cara (ojos, nariz y boca) y utiliza la posición del centro del *bounding box* de cada uno de los ojos para encontrar el ángulo de rotación de la cara sobre el plano de la imagen (no se realizan transformaciones de cambio de pose). Se rota la imagen original utilizando esta transformación

y se adquieren la imagen de la cara enderezada y sus elementos, utilizando las posiciones de los *bounding boxes* originales y teniendo en cuenta la transformación aplicada. La rotación de la imagen original se realiza desde el origen de coordenadas de la imagen (píxel de arriba a la izquierda). La imagen de la cara primero se reescala al ancho de la imagen (en principio a 90x90 píxeles) y se recorta para que sea cuadrada en el caso que no lo sea. Este recorte se realiza eliminando información de arriba y abajo de la imagen de la cara por igual.

- ***Detección_1* - Detección del sistema inicial con deformación de la imagen**: este algoritmo utiliza la detección del S.I., pero en vez de recortar la imagen deforma esta para que sea cuadrada, es decir, realizando un reescalado tanto a lo ancho como a lo alto.

- ***Detección_2* - Detección mejorada (D.M.)**: los primeros pasos de este algoritmo de detección son iguales a los del S.I., hasta que se adquiere el ángulo de rotación de la cara. En este algoritmo, la rotación se realiza desde el centro de la imagen, en vez de en el origen de coordenadas de la imagen, y tras la rotación se vuelve aplicar el algoritmo de Viola-Jones adaptado por Argos Global, para la detección de caras y elementos de la cara. Este algoritmo es más lento que el del S.I., ya que aplica Viola-Jones dos veces, a costa de esperar una mejora en los resultados. Se realiza un reescalado y recorte de la cara al igual que en el algoritmo del S.I.

- ***Detección_3* - D.M. con deformación de la imagen**: el algoritmo empleado aquí es igual que el de la D.M., deformando la imagen para que sea cuadrada en lugar de recortándola, como ya se dijo, reescalándola tanto a lo ancho como a lo alto.

- ***Detección_4* - S.I. con contraste adaptativo**: en este algoritmo se introduce una etapa de preprocesamiento utilizando contraste adaptativo (CLAHE) sobre la imagen, con un filtro de tamaño 8x8 y un *clip* de valor 2,0 (el parámetro *clip* regula la cantidad de contraste realizado). El hecho de utilizar contraste adaptativo pretende analizar si se mejora la verificación ante variaciones fuertes y direccionales de iluminación, adicionalmente al uso de LBP. El hecho de usar el valor de *clip* 2,0 se debe a que al variar el tamaño del *clip* entre 0,5 y 4,0 en intervalos de 0,5 se aprecia un mínimo local en ese valor, tanto en EER como en FMR@1 %, como se muestra en la Figura 4.2 (utilizando las bases de datos faciales FOnFaces y FERET). El valor de *clip* 0,0 en esta Figura equivale a no utilizar CLAHE.

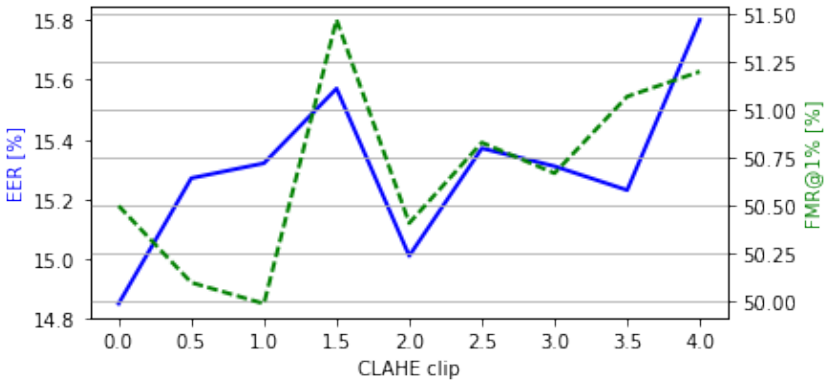

Figura 4.2: EER y FMR@1 % variando el parámetro *clip* del CLAHE.

- ***Detección_5* - S.I. con deformación y contraste adaptativo**: combinación del sistema de detección inicial, con deformación de la imagen incluyendo un preprocesado de contraste adaptativo de tamaño de filtro 8x8 y *clip* 2,0.

- ***Detección_6* - D.M. con contraste adaptativo**: se aplica la detección mejorada con un preprocesado de contraste adaptativo con tamaño de filtro 8x8 y tamaño de *clip* de 2,0.

- ***Detección_7* - D.M. con deformación y contraste adaptativo**: se utiliza la detección mejorada, realizando una deformación de la imagen para hacerla cuadrada en lugar de recortando la imagen, y aplicando CLAHE con un filtro 8x8 y tamaño de *clip* 2,0.

Estos algoritmos de detección de la imagen facial se evalúan haciendo uso del *software* de evaluación de algoritmos de detección facial y utilizando las bases de datos FOnFaces, FERET y Cohn-Kanade y haciendo uso de las puntuaciones resultantes se extrae la siguiente curva DET, mostrada en la Figura 4.3.

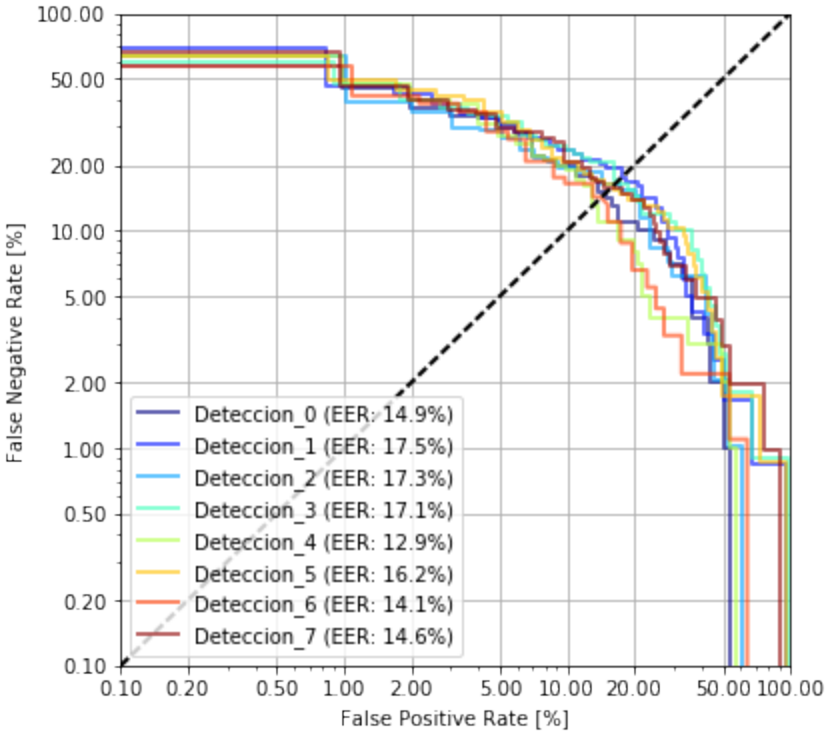

Figura 4.3: Curva DET de algoritmos de detección y preprocesado.

Haciendo uso de la Figura 4.3 junto con la Tabla 4.III se puede ver que los algoritmos de detección con mayor AUC son el 4 (94,2 %) y el 6 (94,4 %), teniendo también el menor EER (12,9 % y 14,1 % respectivamente). Ambos algoritmos tienen en común el uso de contraste adaptativo y el recorte de la imagen en lugar de la deformación. Se puede ver también de manera comparativa, los algoritmos que tienen deformación de la imagen funcionan peor que aquellos que recortan la imagen, como sucede comparando las pruebas de detección 0 y 1, 4 y 5, y 6 y 7. Por otro lado, también se observa que al aplicar contraste adaptativo se consiguió mejorar el acierto del sistema, comparando los algoritmos de detección 0 y 4, 1 y 5, 2 y 6, y 3 y 7.

Tabla 4.III: EER, FMR@1 % y AUC de algoritmos de detección y preprocesado facial.

Detección	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
EER	14,9 %	17,5 %	17,3 %	17,1 %	12,9 %	16,2 %	14,1 %	14,6 %
FMR@1 %	57,0 %	68,9 %	63,9 %	59,1 %	64,4 %	63,9 %	57,2 %	65,7 %
AUC	93,5 %	91,8 %	93,1 %	91,3 %	94,2 %	91,7 %	94,4 %	92,0 %

Debido a que utilizando solamente estas bases de datos no se garantiza la robustez estadística de los resultados que ofrece el FMR@1 % (ver la poca resolución de las curvas entorno a la tasa de falso positivo en 1 %) no se puede realizar ninguna conclusión utilizando esta métrica.

Se puede observar que cambiando las características de preprocesado se supera el sistema de evaluación inicial en términos de EER. Otras evaluaciones relacionadas con el preprocesado se presentan al final de esta sección, ya que se realizaron tras las evaluaciones de comparaciones y de fusión intra-rasgo.

4.2.2. Evaluación de comparaciones

En este apartado se toma como punto de partida los mejores algoritmos implementados en el apartado anterior, denominados en la memoria como *Detección_4* y *Detección_6*, y se prueban diferentes medidas de distancia entre los vectores histogramas resultantes de extraer en el modelado las características LBP. La formulación de las distancias utilizadas son las que se encuentran en la *Sección 2.2.3: Comparación basada en distancias*, y son la distancia Euclídea, Coseno, Chi-cuadrado, Bhattacharyya, Manhattan y Mahalanobis.

Utilizando la *Detección_4* y las imágenes de las bases de datos FOnFaces, FERET y Cohn-Kanade conjuntamente se obtienen los resultados de EER, FMR@1 % y AUC que se presentan en la Tabla 4.IV y la curva DET mostrada en la Figura 4.4.

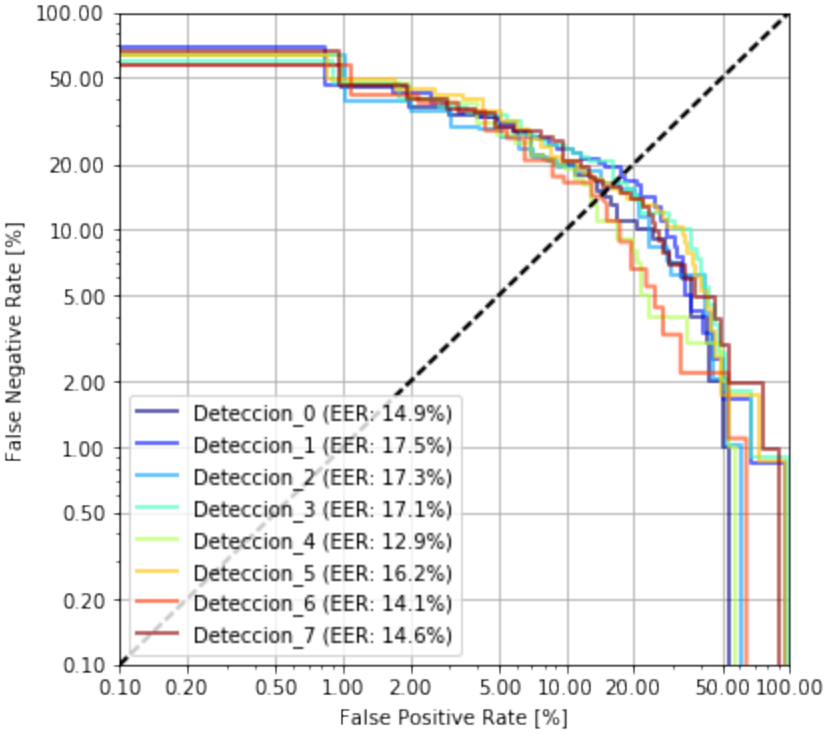

Figura 4.4: Curva DET de algoritmos de comparación por distancias, utilizando *Detección_4*.

En este experimento se puede observar que utilizando la distancia Coseno y Manhattan se mejora el sistema con un AUC del 97,0 % o superior, y reduciendo el EER por debajo del 8 %.

En la Figura 4.4 se aprecia claramente que las dos peores distancias para comparar histogramas son la Mahalanobis, consiguiendo peores resultados que un sistema aleatorio, y la Euclídea en puntos próximos al EER. La manera que se ha encontrado para explicar por qué el uso de la distancia Mahalanobis ofrezca resultados tan catastróficos puede ser debido a que al realizar la

Tabla 4.IV: EER, FMR@1 % y AUC de algoritmos de comparación por distancias, utilizando *Detección_4*.

Distancia	Euclídea	Coseno	Chi-cuadrado	Bhattacharyya	Manhattan	Mahalanobis
EER	12,9 %	7,9 %	9,9 %	9,9 %	7,9 %	50,5 %
FMR@1 %	64,4 %	87,4 %	94,1 %	96,6 %	87,2 %	100,0 %
AUC	94,2 %	97,1 %	96,5 %	95,8 %	97,0 %	47,5 %

normalización de la distancia entre los vectores de características por la matriz de covarianza de dichos vectores hace que los vectores se pase a un espacio donde no es fácil distinguir usuarios genuinos e impostores, llegando a obtener un EER que muestra un comportamiento prácticamente aleatorio.

De igual manera, pero utilizando la *Detección_6* y, de nuevo, las bases de datos de caras FOnFaces, FERET y Cohn-Kanade de manera conjunta se obtienen los resultados de EER, FMR@1 % y AUC mostrados en la Tabla 4.V y las curvas DET en la Figura 4.5.

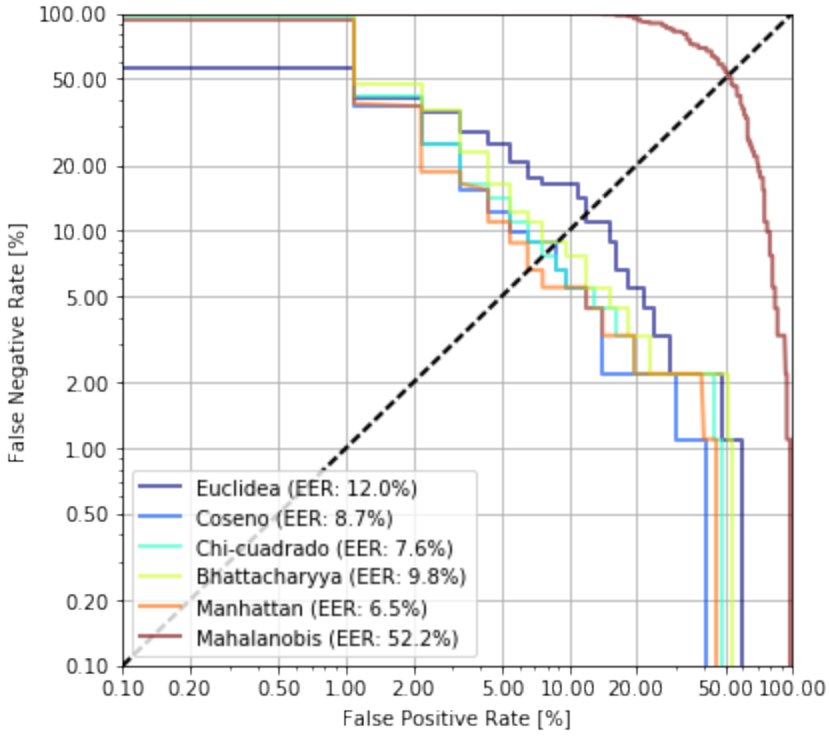

Figura 4.5: Curva DET de algoritmos de comparación por distancias, utilizando *Detección_6*.

Tabla 4.V: EER, FMR@1 % y AUC de algoritmos de comparación por distancias, utilizando *Detección_6*.

Distancia	Euclídea	Coseno	Chi-cuadrado	Bhattacharyya	Manhattan	Mahalanobis
EER	14,1 %	8,7 %	7,6 %	9,8 %	6,5 %	52,2 %
FMR@1 %	57,1 %	93,4 %	96,7 %	98,9 %	93,4 %	100,0 %
AUC	94,4 %	96,9 %	96,5 %	95,9 %	96,9 %	48,1 %

En la Tabla 4.V se muestra que las mejores medidas de distancia según AUC son las distancia Coseno y Manhattan (96,9 % en ambos casos) y según EER son la Manhattan (6,5 %) y Chi-

cuadrado (7,6 %). De nuevo, las peores medidas de distancia son la Mahalanobis, funcionando como un sistema aleatorio y la Euclídea, en otros puntos de trabajo más próximos al EER, como se observa en la Figura 4.5.

De manera comparativa entre los anteriores experimentos, se puede observar que las distancias que consiguen menor EER en ambos experimentos son la Manhattan y la máxima AUC con la distancia Coseno. De nuevo, se desestima utilizar en este caso la información de FMR@1 % debido a la poca robustez estadística que ofrece esta métrica utilizando únicamente las bases de datos con los que se han realizado estos experimentos.

Se observa también que en general los resultados son mejores utilizando la *Detección_4*, destacando sobre este las distancias Coseno, Chi-cuadrado y Manhattan, que ofrecen mayores AUC y menores EER. Con el ánimo de poder comparar de manera más general estas medidas de distancia, se tomó la decisión de ampliar la base de datos con las que se iba a realizar las pruebas, evaluando estas tres distancias utilizando la *Detección_4*, y así poder discernir mejor su rendimiento al utilizar una mayor cantidad de datos.

Para este nuevo experimento se utilizan de manera conjunta las bases de datos FOnFaces, FERET, Cohn-Kanade, Cohn-Kanade+, YaleB, NIST, yalefaces y ND2006, aunque se toma una muestra del 25 % del total para hacer esta prueba con el fin de incluir variabilidad y más datos al problema al usar más bases de datos, pero por otro lado reduciendo el tiempo que llevaría el experimento utilizando todos los datos. La selección de los usuarios se ha realizado de manera pseudo-aleatoria entre las distintas bases de datos, manteniendo la proporción de datos.

El resultado de este experimento se muestra en la Tabla 4.VI, de donde se concluye que el mejor sistema de comparación basado en distancias es el basado en la distancia Manhattan, tanto en EER, FMR@1 % y AUC. La curva DET asociada a este experimento es la que se muestra en la Figura 4.6, donde se puede verificar que el sistema que utiliza la distancia Manhattan es mejor que al utilizar la distancia Coseno y la distancia Chi-cuadrado en todos los puntos de trabajo.

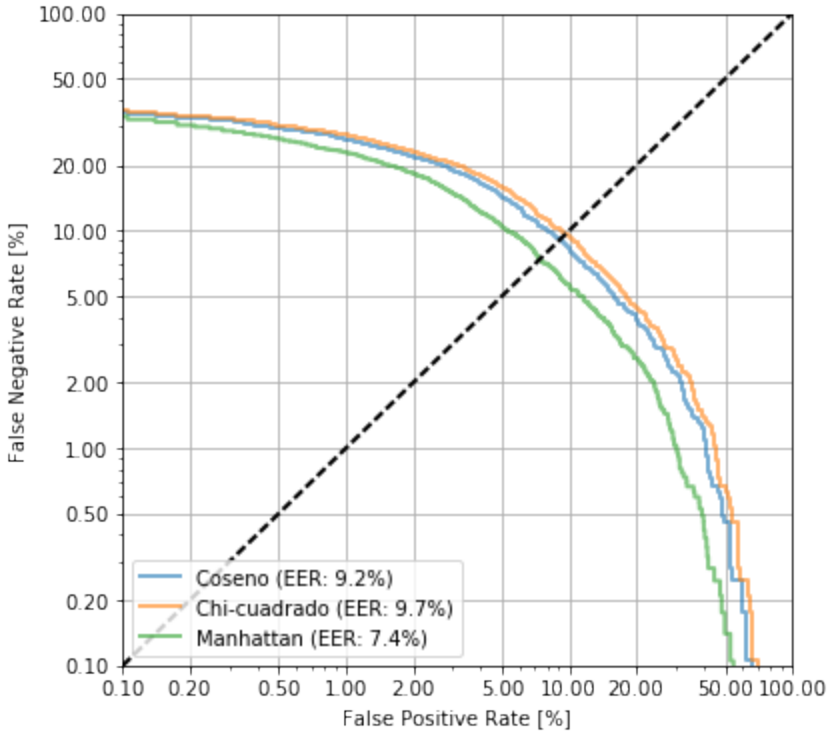

Figura 4.6: Curva DET de algoritmos de comparación por distancias, utilizando *Detección_6*.

Tabla 4.VI: EER, FMR@1 % y AUC de los mejores algoritmos de comparación por distancias, utilizando *Detección_4*.

Distancia	**Coseno**	**Chi-cuadrado**	**Manhattan**
EER	9,2 %	9,7 %	7,4 %
FMR@1 %	26,6 %	27,7 %	22,9 %
AUC	97,2 %	96,9 %	98,1 %

Con estas pruebas se ha conseguido mejorar el sistema inicial reduciendo el EER en cinco puntos utilizando las mismas base de datos. Además, utilizando el nuevo conjunto de base de datos se observa que el FMR@1 % está por debajo del 25 % y que al usar estas bases de datos se obtiene la robustez estadística suficiente para tenerlo en cuenta en los próximos experimentos. Este último modelo, utilizando *Detección_4* y distancia Manhattan para comparar con el nuevo conjunto de base de datos será el utilizado en la siguiente evaluación.

4.2.3. Evaluación de fusión intra-rasgo

En esta evaluación se pretende encontrar una respuesta a las siguientes dos preguntas: la primera es conocer qué componentes de la cara ofrecen más información para poder realizar correctamente una verificación, y la segunda es saber de qué manera se puede combinar esta información a nivel de puntuación con el fin de obtener mejores resultados en la verificación.

Para hacer esto se entrenan modelos basados en LBPs tanto para la cara, como se ha estado haciendo en estas pruebas, como para los elementos de ésta: una imagen combinación de los ojos derecho e izquierdo, nariz y boca; y se utiliza la distancia Manhattan como medida de distancia entre modelos. El inverso de esta distancia es la puntuación, sobre la cual se realizará la fusión de elementos intra-rasgo, como se explicó en la *Sección 2.4: Fusión biométrica*, utilizando elementos dentro del propio rasgo para fusionar.

Se puede observar en la Figura 4.7 las curvas DET de los elementos de la cara por separado —cara, ojos, nariz y boca— y cuatro métodos de fusión a nivel de puntuación —suma de las puntuaciones de cara, ojos, nariz y boca, producto de las puntuaciones de cara, ojos, nariz y boca, suma de las puntuaciones de cara y ojos y producto de las puntuaciones de cara y ojos—.

Con el ánimo de contestar a la primera pregunta planteada en este apartado, observando la Figura 4.7 se puede apreciar que los ojos y la cara en su conjunto ofrecen la mayor información para la verificación facial utilizando el método descrito. Se puede ver que la información de los ojos (curva azul oscuro próxima a las curvas naranja y roja) hace que el sistema disminuya su falso negativo en tasas de falso positivo bajas, es decir, funciona mejor en situaciones de seguridad. Por otro lado, la curva de puntuaciones obtenida al realizar la verificación con la cara (azul intermedio), se cruza con la de las puntuaciones de verificación con los ojos en un punto cercano al EER, siendo esta mínima en puntos donde el falso negativo es más bajo, esto es, funciona mejor en situaciones de alta detectabilidad. Respecto al resto de elementos de la cara, se ve que la nariz (curva en azul claro) es el siguiente elemento que más información provee, mientras que la boca (curva verde turquesa) es el que menos de todas las curvas presentadas, posiblemente debido a la alta variabilidad de ésta si los usuarios cambian el gesto o la expresión.

En cuanto a la segunda pregunta, se observa que la fusión suma (curva verde claro) o producto (curva amarilla) de todos los elementos no ofrece resultados mejores que los elementos de manera independiente; en cambio, la fusión suma y producto de las puntuaciones de cara y ojos (curva naranja y rojo respectivamente) sí que ofrecen resultados ligeramente mejores a los que ofrece únicamente los ojos. En la Tabla 4.VII se pueden ver los resultados de EER, FMR@1 % y AUC de los sistemas, donde queda claro que el mejor sistema resultante de todas

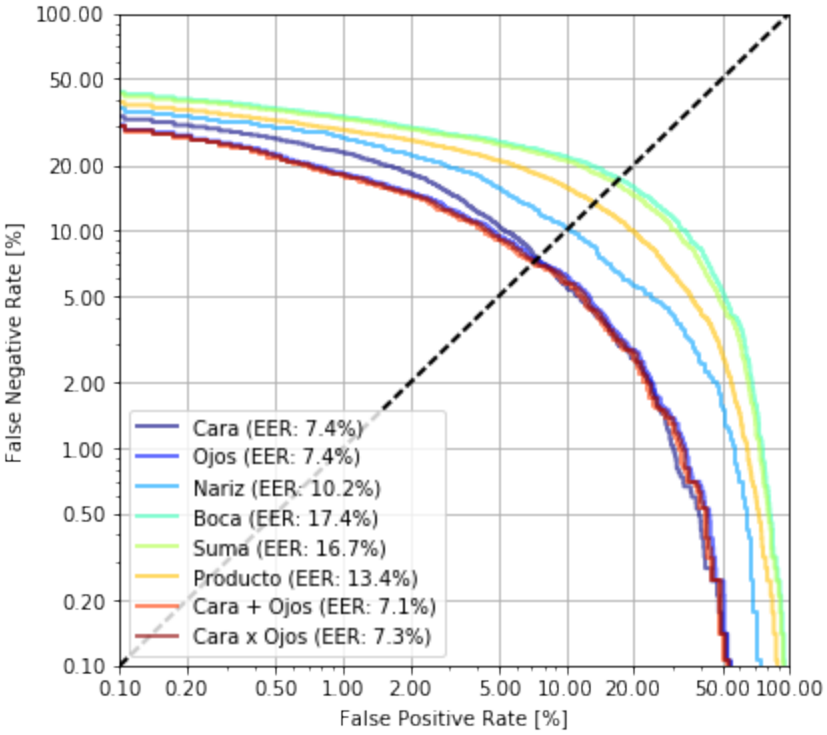

Figura 4.7: Curva DET de algoritmos de fusión intra-rasgo, utilizando *Detección_4* y distancia Manhattan.

las métricas es el de fusión suma entre cara y ojos, obteniendo un EER del 7,3 %, FMR@1 % del 18,2 % y AUC del 98,2 %.

Tabla 4.VII: EER, FMR@1 % y AUC de algoritmos de fusion intra-rasgo, utilizando *Detección_4* y distancia Manhattan.

Elemento	**Cara**	**Ojos**	**Nariz**	**Boca**	**Suma**	**Prod.**	**Cara + Ojos**	**Cara x Ojos**
EER	7,4 %	7,4 %	10,2 %	17,4 %	16,7 %	13,4 %	7,3 %	7,4 %
FMR1 %	22,9 %	18,4 %	26,7 %	33,5 %	32,7 %	29,2 %	18,2 %	18,3 %
AUC	98,1 %	98,1 %	96,3 %	91,6 %	92,2 %	94,5 %	98,2 %	98,1 %

Tras este experimento en el cual se han utilizado distintas bases de datos y se han conseguido resultados positivos nos planteamos que distintos efectos que son comunes en una misma base de datos, como pueden ser condiciones de iluminación o alteraciones, ajustes y defectos de la cámara pueden afectar de igual manera a usuarios dentro de una misma base de datos, de tal manera que a la hora de comparar usuarios entre distintas bases de datos estos efectos provoquen que sea sencillo discriminarlos entre sí. Este efecto nos hace reflexionar sobre el funcionamiento de estos sistemas en entornos donde la verificación se realice en múltiples dispositivos, pudiendo aumentar el falso rechazo.

4.2.4. Otras evaluaciones

Otras evaluaciones que se llevaron a cabo que quedan fuera del alcance de este trabajo están relacionadas con la etapa de preprocesamiento y modelado. En cuanto a preprocesamiento se evaluó con el tamaño de las imágenes y el número de imágenes a utilizar para generar tanto los modelos de entrenamiento como los de verificación. La selección de ambas variables

requiere tomar una decisión de compromiso: imágenes más grandes permiten una mejora en la verificación, al presentarse más información, pero a cambio el preprocesamiento y el modelado se hace más costoso, sobre todo teniendo en cuenta que la aplicación de verificación facial se quiere realizar en entornos móviles.

En cuanto al tamaño de las imágenes se estudió que un bueno compromiso entre rendimiento y espacio y tiempo de procesado sería trabajar con imágenes de caras de 120x120 en vez de 90x90 que fueron los tamaños de imágenes con los que inicialmente que realizaron las pruebas descritas en los apartados anteriores. Por otro lado, en cuanto a número de imágenes a utilizar para generar el modelo el compromiso venía dado también por la diferencia entre las capturas de la imágenes, y por tanto el tiempo necesario para la generación de los modelos de entrenamiento y verificación. En este caso, se realizó un experimento utilizando imágenes tomadas en intervalos de un segundo entre 1 y 15 y se observó que aparecía un mínimo en EER utilizando 12 imágenes en lugar de 5 para generar los modelos (que es lo que se utilizó en las pruebas anteriores), pero esto es inviable en la práctica ya que se tiende a utilizar el menor número de imágenes posibles ya que se quiere reducir el tiempo en el que se lleva a cabo las verificaciones, por lo que se siguió trabajando con 5 imágenes.

En cuanto a la generación de modelos, se hicieron distintas pruebas relacionadas con los parámetros del modelo LBP, tales como el tamaño y la conectividad del filtro, el número de muestras del histograma de los LBPs, además de cómo combinar las características de las imágenes capturadas. También, aunque ajeno al trabajo del autor, se realizó un estudio empleando características WDA utilizando el sistema de evaluación de características descrito en la sección anterior, ofreciendo éstas resultados mejores (6,2 % de EER utilizando la *Detección_4*, distancia Manhattan y la fusión suma de cara y ojos) que los que se consiguieron con LBP.

4.3. Diseño de un sistema de antispoofing facial

Externamente al sistema de evaluación de algoritmos de verificación facial se trabajó en el diseño de un sistema de *antispoofing* facial, aunque la realización de la implementación de este sistema no fue realizada por el autor de este TFM, quedando esta última parte fuera del alcance del trabajo. El objetivo de este sistema *antispoofing* es ejecutarlo de forma paralela al sistema de verificación a partir de las imágenes de cara detectadas en la imagen fusionando la salida de ambos sistemas a nivel de decisión mediante la operación lógica AND, de tal manera que si alguno de los dos sistemas rechaza a un usuario, el usuario no puede acceder al sistema.

El sistema se diseña con el fin de ser robusto ante el mayor número de ataques, incluyendo ataques mediante fotografía y vídeo, y siendo extensible en un futuro ante ataques por máscara 3D. Tras nuestra experiencia, observamos que sistemas comerciales como el de la compañía KeyLemon que tienen un *antispoofing* basado en detección de vida mediante pestañeo son fácilmente vulnerables ante ataques de reproducción de vídeo, en los que el sujeto pestañee. Por esta razón, se decidió crear nuestro sistema *antispoofing* basado en reflejos y texturas.

Este sistema se entrena con cinco tipos de modelos diferentes, representando las texturas que se desea detectar.

- **Modelo de fotografía en papel**: este modelo se entrena con fotografías impresas en papel, tanto satinado como no, con diferentes niveles de calidad.
- **Modelo de fotografía revelada**: este modelo se entrena con fotografías reveladas en papel fotográfico de alta calidad, donde a diferencia del modelo anterior se pueden percibir reflejos, debido al material del papel.

- **Modelo de fotografía/vídeo en móvil**: este modelo se entrena con fotografías presentadas en un dispositivo móvil, bien un *smartphone* o una *tablet*. A diferencia de los modelos anteriores, este modelo emite la imagen, no la refleja, además de poder tener otras peculiaridades, como marcas de dedos sobre la pantalla, que puede hacer detectable el *spoof*.
- **Modelo de fotografía/vídeo en plasma**: este modelo, similar al anterior, permite la detección de ataques presentados en pantallas de plasma (como las pantallas de ordenador o los televisores). A diferencia del modelo anterior, estas pantallas no tienen tantos reflejos, por lo que se consideró generar otro modelo independiente al anterior.
- **Modelo de cara real**: este modelo permite detectar caras reales, en contraposición a las imágenes de *spoofing*.

Los datos utilizados para entrenar estos modelos corresponden a una base de datos creada para tal propósito por Argos Global, contando con 25 personas, y tres sesiones por tipo de *spoof*. Esta base de datos pretende extenderse próximamente para generalizar aún más los modelos de *spoofing*.

El sistema consiste en entrenar un clasificador con dos clases por cada *spoof*: el modelo de cara real y el modelo de ataque concreto. Estos cuatro clasificadores se ejecutarán de forma paralela, y la puntuación de todos los clasificadores se utiliza para realizar fusión a nivel de puntuación, utilizando otro clasificador que aprende los diferentes tipos de puntuaciones. Los clasificadores entrenados se basaron en LDA, con el objetivo de encontrar una proyección que maximizara la varianza interclase, minimizando la intraclase, siendo las clases a analizar «imagen real» e «imagen de *spoofing*».

El sistema *antispoofing* inicial se diseñó utilizando también características LBP, similar a la verificación, ya que es una manera extendida en el estado del arte para discriminar distintos tipos de texturas, aunque los parámetros de los LBPs es distinto al empleado en verificación.

Además de este sistema *antispoofing* basado en texturas se diseñó un sistema *antispoofing* facial 3D altamente eficaz contra ataques de representación en 2D, utilizando una cámara de profundidad. Este sistema consiste en detectar la imagen de la cara utilizando la imagen en blanco y negro, asignar la correspondiente imagen en 3D de la cara y comprobar que las variaciones de la imagen de profundidad en la nariz respecto a los ojos superaba determinado umbral. Se observó que en escenarios reales, pese a la simplicidad de este sistema, los resultados eran altamente eficaces, a costa de la necesidad de una cámara 3D para la adquisición de estas imágenes.

5

Demo comercial y aplicación móvil FaceOnVox

Esta sección cubre los contenidos relacionados con el desarrollo de un *software* de demostración (demo) en arquitectura local y cliente servidor, el diseño de una API de verificación facial en C++, y el diseño de una aplicación móvil de verificación facial aplicada a la protección de aplicaciones (*App Locker*). El autor del trabajo participó en el desarrollo del *software* de las demos y en el diseño de la API y de la aplicación móvil, sin llegar a programar de manera explícita las dos últimas aplicaciones, aunque el trabajo de investigación y de desarrollo implementado en la *Sección 4: Desarrollo y evaluación de un sistema de verificación facial* sí que ha sido utilizado tanto como para reutilizar las funciones de detección, modelado y comparación implementadas en el *software* de evaluación de algoritmos de verificación facial desarrolladas por el autor del trabajo, como para la elección de los valores por defecto que tendrá la API de verificación facial.

En esta sección se dividen las actividades realizadas en tres apartados, según los objetivos específicos descritos en el *Anexo A: Objetivos específicos*:

- **Desarrollar un sistema todo local y cliente servidor de una demo de VerifyByFace**: en este apartado se explica en qué consistía el *software* de demostración implementado en arquitecturas todo local y cliente servidor.
- **Diseñar una API de verificación facial en C++, VerifyByFace**: en este apartado se explica de qué manera se diseñó la API de la aplicación de verificación facial de VerifyByFace.
- **Diseñar una aplicación móvil, FaceOnVox**: en este apartado se explica el diseño llevado a cabo para la aplicación de bloqueador de aplicaciones (*App Locker*) FaceOnVox.

5.1. Desarrollo de un sistema todo local y cliente servidor para una demo de VerifyByFace

Debido a requisitos comerciales, el equipo técnico de Argos Global tuvo que desarrollar un *software* de demostración de su producto de verificación facial denominado VerifyByFace. Este *software* consistía en una interfaz de usuario que hiciera uso de las, entonces, funciones

en C/C++ que permitían detectar caras en imágenes, generar modelos de verificación facial y compararlos entre sí en un entorno de verificación biométrica. La aplicación ha ido evolucionando teniendo diferentes versiones, donde se han añadido mejoras gracias a la investigación y al desarrollo de algoritmos de verificación facial y elementos para detectar el *spoofing* con tecnologías basadas en imágenes 2D y 3D.

Para la realización de esta tarea existía un desarrollo de partida, el cual se adaptó para los requisitos necesarios por el departamento comercial de Argos Global y para incluir las nuevas mejoras que se estaban desarrollando desde el equipo técnico. Este *software* también permitía evaluaciones de los algoritmos que se estaban desarrollando en escenarios reales, donde las condiciones podían no ser tan idóneas como en las bases de datos utilizadas, y también se ha utilizado para recoger la base de datos FOnFaces de Argos Global. El diseño original constaba de dos arquitecturas: una todo local donde el *software* consistía en un ejecutable de Windows, y otra cliente servidor, donde el sistema cliente generaba el modelo, lo enviaba por la red y el sistema servidor lo almacenaba si era de entrenamiento, y en caso contrario lo utilizaba para realizar una verificación.

5.1.1. Demo con arquitectura local

La herramienta de demostración o demo con arquitectura local (*Software* de demostración de VerifyByFace con arqitectura local[1]) se desarrolló en un principio con el fin de poder mostrar un ejemplo de uso de la aplicación a futuros clientes, *partners* e inversores, aunque se han implementado diferentes aplicaciones a partir del código generado. Esta sencilla herramienta tiene una interfaz simple desarrollada en C++ utilizable dispositivos con sistema operativo Windows (XP, 7, 8 y 10, y Surface Pro 2) que dispongan de *webcam*. La interfaz muestra por un lado la imagen capturada de la cámara, y por otro lado las opciones de registrar un nuevo usuario y verificar un usuario, como se muestra en la Figura 5.1.

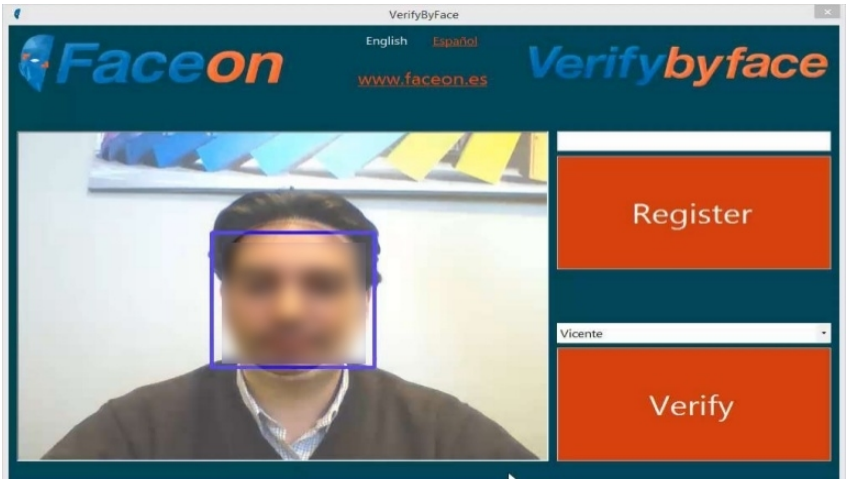

Figura 5.1: Interfaz del *software* de demostración.

A la derecha de la interfaz aparecen dos apartados, cada uno con un cuadro de texto y un botón: el cuadro de texto sobre el botón de registro (*Register*) solicita la identidad de la persona frente al sensor para generar un modelo (nombre de usuario) y el botón de registro captura genera y guarda un modelo del usuario frente al sensor y asocia ese modelo al nombre introducido en el cuadro de texto que hay en la parte superior a este, mientras el cuadro de texto sobre el botón de verificación (*Verify*) muestra un desplegable de los usuarios registrados

[1]Entregable E-5.1.1. Ver *Anexo A: Objetivos específicos.*

y el botón de verificación captura y genera un nuevo modelo de verificación del usuario que esté en ese momento frente el sensor y lo compara con aquel asociado a identidad provista. Finalmente se muestra en una ventana desplegable si el usuario que ha intentado verificarse ha sido aceptado o rechazado.

Esta aplicación se distribuye utilizando un asistente de instalación, y los modelos se guardan en el sistema de forma cifrada con clave simétrica, utilizando la librería para C/C++ OpenSSL. El sistema de compresión y cifrado de los modelos de registro fue implementado por el autor de la memoria. Esta aplicación se distribuye con una licencia temporal, que caduca a los 30 días de la instalación. También están limitados el número de usuarios de registro a cinco, pudiéndose borrar usuarios anteriores una vez se llega a ese límite.

La aplicación en cuestión se ha mejorado tanto gráficamente como en funcionalidad, añadiendo algoritmos que mejoran el rendimiento de la aplicación, utilizando la API de verificación facial descrita en la siguiente sección y permitiendo aplicar un filtro *antispoofing* 2D y 3D, además de posibilitar la variación del umbral de trabajo para distintos tipos de aplicación. Este sistema ha sido utilizado en la feria profesional *eShow* de 2014 en Madrid —en la que el autor participó como expositor presentado los productos biométricos para comercio electrónico de Argos Global— y estuvo expuesto en el *Global Sports Innovation Center* de Microsoft en Madrid.

También, a partir del desarrollo de esta demo se trabajó en una aplicación de salvapantallas para Windows 7, 8 y 10, donde si el ordenador del usuario entra en modo suspensión o es inutilizado durante cierto periodo de tiempo, el programa lanza un proceso cuando se vuelva a adquirir el control del ordenador en el que se adquieren imágenes del usuario que hay frente a la cámara del ordenador, las procesa y compara este usuario con el o los que están previamente enrolados en el sistema. Esta aplicación tiene una pantalla de configuración más avanzada y en la versión de pago tiene incorporada la funcionalidad *antispoofing*, y está disponible en la página web de FaceOn, en el apartado de descargas[2].

Con vistas a continuar el desarrollo de esta aplicación quedó pendiente añadirle la funcionalidad de reconocimiento de locutor utilizando la herramienta *software* de demostración de reconocimiento de locutor en C++, descrita en la *Sección 6.1: Desarrollo de un sistema de evaluación de algoritmos de reconocimiento de locutor en C++.*

5.1.2. Demo con arquitectura cliente servidor

La herramienta de demostración con arquitectura cliente servidor (*Software* de demostración de VerifyByFace con arqitectura cliente servidor[3]) se desarrolló con el fin de poder utilizar los algoritmos de verificación facial de segunda y tercera generación que dispone Argos Global[4].

La interfaz de esta demo distribuida (parte del cliente) está basada en la desarrollada en la arquitectura local, mostrada en la Figura 5.1. Hasta la fecha había dos modalidades de uso de esta interfaz: en la primera versión tanto la detección como la generación del modelo es computada por el sistema cliente, evitando así que las imágenes salgan del dispositivo el cual se han capturado; mientras que en una segunda versión, las imágenes se envían de forma cifrada para realizar la detección y la generación del modelo en el servidor[5].

Una vez se ha generado el modelo, éste se comprime y se cifra, y se envía de manera segura a

[2] `http://faceon.es/descarga` (Consultado 05/2017).

[3] Entregable E-5.1.2. Ver *Anexo A: Objetivos específicos.*

[4] Como se mencionó en secciones anteriores, estos algoritmos se descartaron integrar en aplicaciones móviles bajo una arquitectura local debido a su coste en entrenamiento y el tamaño de los modelos, que contradicen una funcionalidad dinámica deseada de la aplicación móvil.

[5] Ninguna de las dos versiones de este modelo llegaron a distribuirse, en su lugar se utilizó el *software* de demostración con arquitectura local.

un servidor de Argos Global junto con la identidad de la persona codificada de manera diferente para cada instalación de la demo. Esto permite que cada usuario de la demo distribuida realice consultas únicamente sobre los usuarios que tiene registrados asociados a su instalación de la demo, y no sobre otros usuarios registrados en la base de datos. La conexión entre cliente y servidor se realiza de manera segura y solo es accesible cuando se autentica desde la aplicación. El servidor almacena la información de cada registro y verificación en una base de datos SQL protegida contra ataques de *Cross-Site Scripting* (XSS), en particular de inyección de código SQL, y envía las respuestas de aceptación o rechazo de una verificación al cliente de manera cifrada, como se muestra en la Figura 5.2, siguiendo los siguientes cuatro pasos:

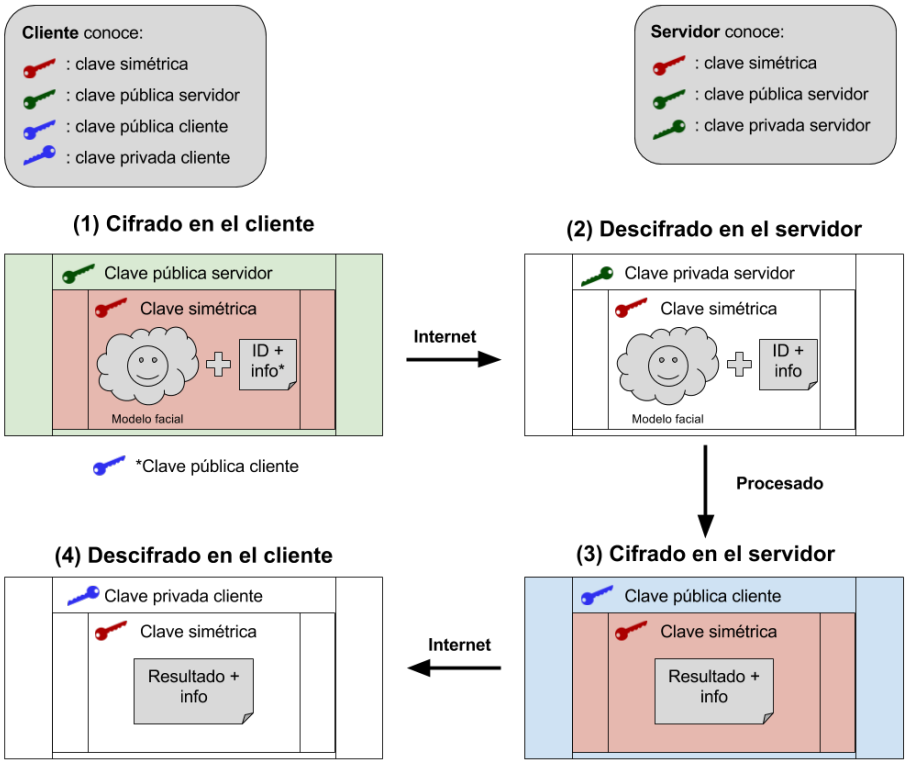

Figura 5.2: Esquema de comunicación cifrada entre cliente y servidor (primera versión).

1. **Cifrado en el cliente**: en esta fase el modelo facial comprimido y el ID del usuario, junto con información relacionada con la versión de la demo, información de sesión, el tipo de modelo del usuario (registro o verificación) y la clave pública del cliente se cifra con la clave simétrica —conocida tanto por el cliente como el servidor—, y el resultado se cifra con la clave pública del servidor, la cual conoce el sistema cliente, y eso se envía por la red de forma segura.

2. **Descifrado en el servidor**: el servidor descifra primero utilizando la clave privada del servidor y posteriormente la clave simétrica que ambos sistemas comparten. El servidor utiliza el ID y la información asociada para añadir un registro en la base de datos asociada a ese cliente y procesa el modelo, bien guardándolo si es de enrolamiento o bien comparándolo contra el modelo de entrenamiento asociado con la ID aportada si es de verificación.

3. **Cifrado en el servidor**: el servidor envía el resultado de la petición —*Registrado* o *Error* en el caso en el que el modelo sea de registro y *Aceptado*, *Rechazado* o *Error* en el caso de

que el modelo sea de verificación— junto con otra información relacionada con la sesión de la petición (a comprobar por el cliente). Esta información se cifra primero con la clave simétrica y luego con la clave pública del cliente, la cual proveyó el cliente durante su petición, y se envía de nuevo al cliente.

4. **Descifrado en el cliente**: el sistema cliente descifra el mensaje usando primero la clave privada del cliente y después la clave simétrica que ambos sistemas comparten y utiliza internamente el contenido del fichero de resultados para continuar con su rutina.

Partiendo de este sistema, quedó pendiente la implementación de un servicio web en el que hiciera uso de este mecanismo de comunicación para poder detectar caras, generar y comparar modelos, y detectar ataques de *spoofing* en imágenes utilizando nuestra tecnología, todo ello bajo una arquitectura distribuida y con un sistema de licencias de pago por uso.

5.2. Diseño de una API de verificación facial en C++, VerifyByFace

En esta sección se explica el diseño de la API de verificación facial en C++ denominada VerifyByFace. VerifyByFace es el producto de verificación facial desarrollado íntegramente por Argos Global. Como ya se mencionó en la *Sección 4: Sistema de evaluación de algoritmos de verificación facial*, este producto parte de un sistema original inicial implementado por Argos Global anteriormente a la llegada del autor de este trabajo a la empresa en forma de funciones en C/C++ que realizan las tareas necesarias para realizar una verificación facial. Gracias a la herramienta de evaluación de algoritmos de verificación facial desarrollada íntegramente por el autor, este sistema inicial ha podido evolucionar de manera consecutiva desde entonces. Debido a la necesidad de acuerdos comerciales, se decidió desarrollar una API que pueda ser integrada en aplicaciones de verificación facial, de tal manera que se pudiera distribuir licencias de ésta para colaboradores o clientes evitando el mal uso, copia o manipulación del código fuente de la aplicación.

Esta API utiliza el trabajo de investigación desarrollado en la *Sección 4.2: Investigación de diferentes algoritmos de detección, modelado, verificación y fusión de características de cara* para establecer los mejores valores por defecto. Además, se han realizado dos versiones de esta API, una de ellas con vistas a poder ser distribuida y otra con el objetivo de que se puedan realizar actividades de investigación y desarrollo sobre ella. La versión comercial de la API requiere activar una licencia limitada bien en tiempo o bien en uso.

El autor de este trabajo participó en el proceso de diseño de esta API, describiendo las clases, métodos y atributos (públicos y privados) correspondientes. Además, el sistema de compresión y cifrado de los modelos (mencionados en el apartado anterior), y los métodos relativos a la detección, modelado y comparación están basados en el desarrollo realizado en la *Sección 4.2: Investigación de diferentes algoritmos de detección, modelado, verificación y fusión de características de cara* por el autor, aunque la programación explícita de la API no ha sido realizada por éste. Para la programación de la API se hizo uso de la referencia [46].

La API en cuestión cuenta con tres clases principales denominadas *Caputre*, *Face* y *Model*:

- **Clase *Capture***: esta clase ofrece una serie de funciones utilizando como base una imagen en la cual se desea detectar una cara. Esta imagen puede ser a color, monocromática o 3D. El método público `DetectFaces` permite detectar las caras que hay en una imagen según el algoritmo de detección deseado, ordenándolas en un vector de objetos de la clase

Face según su tamaño. Para minimizar el tiempo de trabajo, se puede indicar que solo se desea encontrar una única cara en la imagen: la de mayor tamaño, para aplicaciones estrictamente dedicadas a la verificación facial. Adicionalmente, esta clase implementa un algoritmo de *tracking* de caras sencillo, además de ofrecer una máscara que se superpone y ajusta a donde se encuentra la cara en la imagen con objetivo de implementarlo en desarrollos *front-end*, como se muestra en la Figura 5.3.

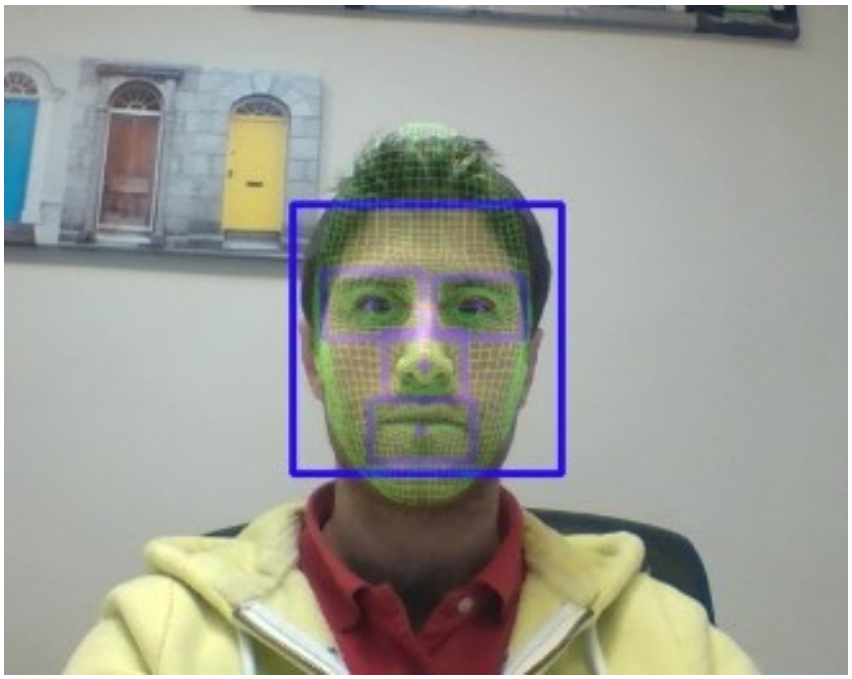

Figura 5.3: Ejemplo de máscara superpuesta en la detección.

- **Clase *Face***: esta clase ofrece una serie de funciones utilizando como elemento central una imagen de una cara. Los objetos de esta clase pueden tienen tener asociada la correspondiente imagen de profundidad si la imagen original era también en 3D. Esta clase permite utilizar el sistema *antispoofing* implementado en la *Sección 4.3: Diseño de un sistema de antispoofing facial* para saber si la cara detectada es real o una suplantación.

- **Clase *Model***: esta clase ofrece una serie de funciones para los modelos biométricos creados a partir de un conjunto de imágenes de cara de un usuario. Utilizando un vector con el número de caras detectadas en diferentes imágenes, se puede crear un modelo utilizando el método público `Create`, escogiendo el tipo de modelado que se quiere realizar, pudiéndolo guardar de manera cifrada dado un nombre de fichero de salida. También es posible realizar una comparación entre diferentes objetos de la clase *Model* utilizando el método público `Compare`, devolviendo una puntuación de la comparación, y una verificación dado un modelo a comparar y un umbral de decisión utilizando el método público *Verify*. La versión de investigación de esta API permite escoger el método de comparación, mientras que la versión comercial viene por defecto el mejor desarrollado que, dependiendo de la versión de la API, puede variar según los estudios e investigaciones realizados por Argos Global.

En la Figura 5.4 se muestra un ejemplo de uso de la API de VerifyByFace para una realizar una verificación facial con *antispoofing* 2D en diez pasos. Cada recuadro con un color distinto señala los métodos asociados a cada clase concreta.

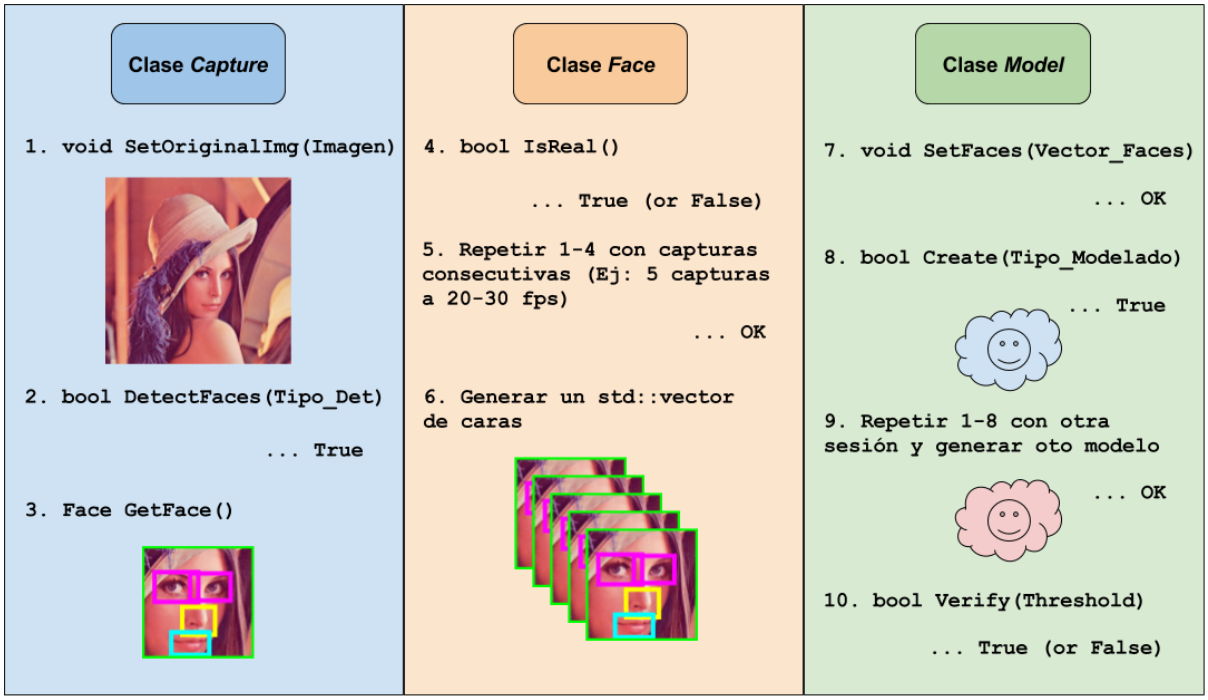

Figura 5.4: Ejemplo de uso de las clases de la API para verificación.

1. **Carga de una imagen**: usando la clase *Capture* se carga una imagen del usuario que quiere generar un modelo utilizando el método `SetOriginalImg`.
2. **Detección de caras**: se detectan las caras que hay en la imagen. En este caso se asume que solo se quiere detectar una cara, que es la más grande en toda la imagen cargada. Se espera la respuesta y se obtiene que se han detectado caras, porque la imagen tiene una cara (`True`).
3. **Adquisición de la cara**: el método `GetFace` devuelve una objeto de la clase *Face*.
4. **Antispoofing**: con el método *IsReal* de la clase *Face* se detecta si la imagen es real o por el contrario es un fraude o suplantación ante el sensor. Se asume en este ejemplo que las imágenes detectadas son de caras reales. Si no lo fueran es decisión del integrador de la API decidir qué hacer.
5. **Obtención de más imágenes de caras**: se repiten los pasos 1-4 por ejemplo 5 veces en total con capturas tomadas entre 20 y 30 veces por segundo para dar consistencia al modelo.
6. **Generar un vector de caras**: se genera un vector de la clase `std:vector` con el conjunto de objetos de la clase *Face*.
7. **Cargar las caras al modelo**: se carga ese vector de caras a un objeto de la clase *Model* usando el método `SetFaces`.
8. **Crear un modelo facial**: se genera un modelo facial utilizando el método `Create` de la clase *Model*.
9. **Crear un modelo de verificación**: se repiten los pasos 1-8 con otra sesión del mismo usuario o de otro diferente cargándolo en otro objeto de clase *Model* para poder verificarlo con el modelo que ya se ha creado en el paso 8.

10. **Verificar el nuevo modelo**: se realiza una verificación con el método `Verify` de la clase *Model*, pasándole como atributos el modelo de verificación creado en el paso 9 y un umbral. El resultado será aceptado `True` si ambos usuarios son el mismo y rechazado `False` si son usuarios diferentes.

Esta API, en su versión comercial, se ha utilizado en la aplicación móvil FaceOnVox, descrita en el siguiente apartado, al igual que en las últimas versiones de las demos descritas en el apartado anterior, y en su versión de desarrollo en el verificador de algoritmos de fusión biométrica de cara y voz descrita en la *Sección 6.2: Desarrollo de un sistema de evaluación de algoritmos de fusión biométrica cara y voz a nivel de puntuación en C++*.

5.3. Diseño de una aplicación móvil, FaceOnVox

En esta sección se presenta la aplicación móvil desarrollada por Argos Global, en la que el autor del trabajo participó directamente en el diseño, además de haber contribuido en parte del desarrollo. Esta aplicación denominada FaceOnVox es un *App Locker* con biometría facial, esto es una aplicación de seguridad que permite bloquear otras aplicaciones y para desbloquearlas es necesario realizar una verificación facial. FaceOnVox utiliza internamente la API de VerifyByFace presentada en la sección anterior.

El objetivo de lanzar esta aplicación es tener conocimiento de la aceptación por parte de los usuarios y del mercado a utilizar tecnología biométrica en los dispositivos móviles, además de tener el primer contacto de cómo funciona la tecnología desarrollada por el equipo técnico de Argos Global en aplicaciones reales, y tener *feedback* por parte de los usuarios de la aplicación para saber qué aspectos se han de mejorar en la aplicación móvil y cuales gustan en relación con otros desarrolladores. La aplicación se publicó en la PlayStore de Android en el mes de octubre de 2015 con la intención de lanzar posteriormente una versión pro que tuviera *antispoofing* facial incorporado.

Existen tres etapas que definen el funcionamiento de la aplicación: generación del perfil, bloqueo de aplicaciones y acceso a aplicaciones:

1. **Generación del perfil**: en esta etapa el usuario genera un perfil biométrico. Una vez abierta la aplicación (ver Figura 5.5, fase 1) el usuario selecciona el botón central e introduce en la siguiente ventana su nombre y un código PIN cuya misión es que sea utilizado en el caso en el que, por ejemplo, la cámara frontal del dispositivo no funcione correctamente.

   Una vez hecho esto, se pasa a la fase 2 (ver Figura 5.5), donde se solicita al usuario que realice una serie de capturas de imágenes para crear el perfil. El usuario debe presionar el botón de *Comenzar* para empezar a realizar las capturas de imágenes. Estas capturas se realizan rellenando una barra de progreso, como se observa en la fase 3 de la Figura 5.5. Si no se puede reconocer ninguna cara durante la captura, bien porque el usuario no esté colaborando en este proceso, bien porque las condiciones de luz no sean adecuadas dado el sensor (mucha oscuridad) salta un temporizador a los 30 segundos volviendo a la fase 1, sin haber completado la generación del perfil.

   Una vez realizado el registro se genera un modelo biométrico de la cara del usuario y se guarda cifrado en el dispositivo. Finalmente, se pasaría a la fase 4 de la Figura 5.5, donde el usuario ha creado su perfil y puede pasar a la siguiente etapa consistente en el bloqueo de aplicaciones.

2. **Bloqueo de aplicaciones**: en esta etapa el usuario escoge las aplicaciones que desea bloquear para que puedan ser abiertas aplicando antes una verificación facial. Para ello,

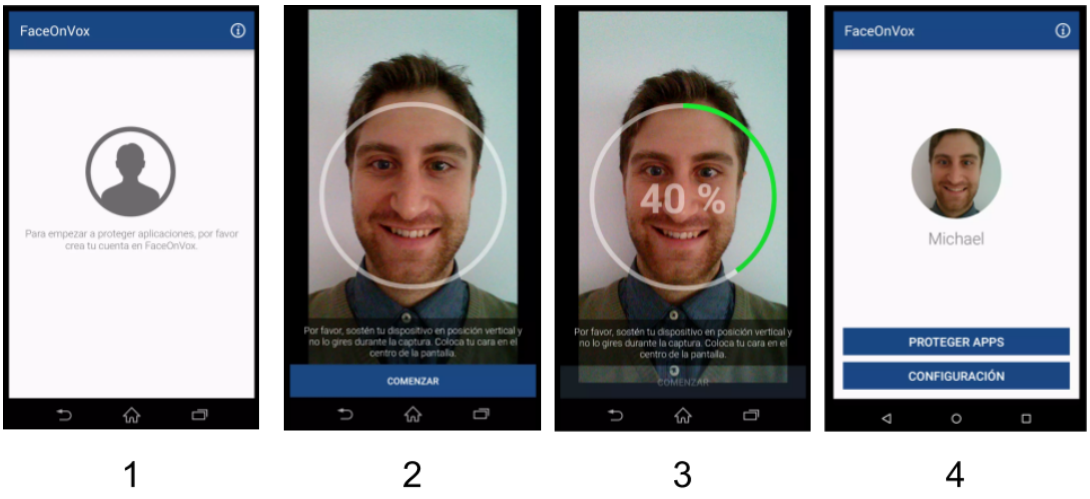

Figura 5.5: Secuencia de generación del perfil en FaceOnVox.

se parte de la fase 4 (ver Figura 5.6) y se selecciona el apartado de *Proteger Apps*. Seleccionando esta opción, se pasa a la fase 5 de la Figura 5.6, donde se observa la lista de aplicaciones que tiene el usuario descargadas en su terminal móvil. Seleccionando estas aplicaciones, aparece a la derecha de las mismas una figura de un candado, indicando que estarían seleccionadas para ser bloqueadas por FaceOnVox. Si se selecciona una aplicación previamente, se puede cancelar la selección volviendo a presionar sobre la aplicación. Estos cambios no se guardan hasta que no se presione el botón *Guardar*, y si esto se hace, se confirma este proceso mediante una verificación facial, donde se comprueba que el usuario que ha seleccionado las aplicaciones es el mismo que el que se ha registrado.

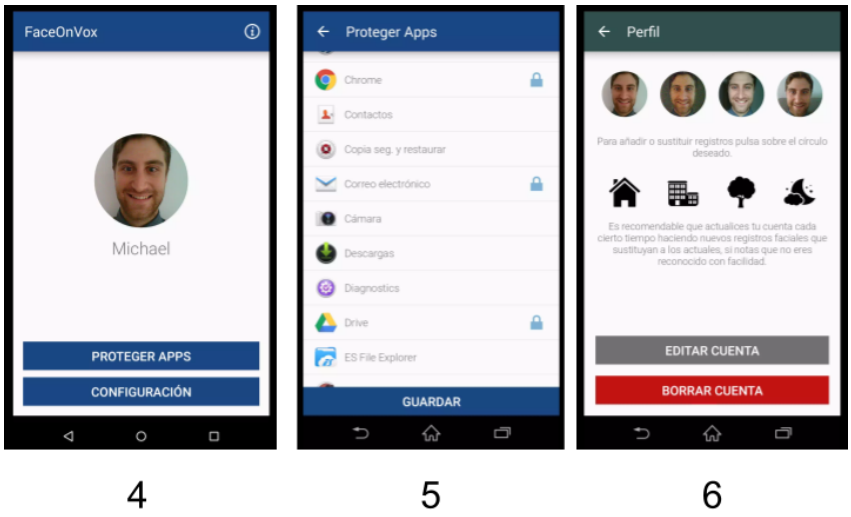

Figura 5.6: Bloqueo de aplicaciones y mejora del perfil.

La verificación se realiza de manera similar al registro: se toman imágenes de la cara del usuario parecido a como se muestra en la Figura 5.5 en la fase 3, permitiendo la opción de desbloqueo por PIN. Con las imágenes adquiridas se genera un modelo de verificación y se compara con el modelo de registro.

Partiendo de la fase 4 (ver Figura 5.6) se pueden cambiar los colores del tema de la aplicación entre otros ajustes presionando *Configuración*. Si, en cambio, se presiona en la imagen del usuario se accede al perfil de éste (fase 6, Figura 5.6). En el perfil se pueden

añadir hasta cuatro modelos faciales distintos del usuario, preferiblemente en diferentes localizaciones típicas, para hacer más sencilla la verificación facial. Cabe destacar que estos modelos nuevos se hacen en condiciones de seguridad, para evitar que otros usuarios impostores generen un modelo bajo un perfil que no es el suyo y así corromper el modelo. Si se realizan más de cuatro modelos faciales de registro en el perfil se borrarán los más antiguos. También se ofrecen otras opciones en esta ventana, como *Editar Cuenta* que permite cambiar la configuración o el PIN de seguridad, o borrar la cuenta, desapareciendo toda información asociada al perfil creado y liberando las aplicaciones protegidas.

3. **Acceso a aplicaciones**: la última etapa es el acceso a las aplicaciones. Cuando se abre una aplicación la cual se ha bloqueado con FaceOnVox, antes de abrirse la aplicación aparece una ventana solicitando una verificación. Si esta es correcta, se puede acceder a la aplicación móvil solicitada; mientras si es incorrecta debido a que el usuario que la está abriendo no corresponde con el usuario registrado, muestra un mensaje de error. Si no se detecta ninguna cara durante el tiempo de captura en un plazo de 30 segundos, no se abre la aplicación.

Esta aplicación móvil ha sido un primer paso para acercar a Argos Global al desarrollo de tecnología de verificación facial en entornos móviles. Esta aplicación está permitiendo medir el rendimiento de la API de VerifyByFace en diferentes dispositivos móviles existentes en el mercado. Además, gracias al desarrollo de esta aplicación, Argos Global ha presentado un proyecto europeo para el programa H2020 con el ánimo de instaurar tecnología de verificación facial y de voz para pagos seguros, incluyendo la capacidad de evitar ataques de suplantación ante el sensor. En la redacción y presentación de este proyecto ha trabajado el autor de este trabajo, donde además tuvo reuniones con otras empresas internacionales colaboradoras en el proyecto.

6

Verificación de locutor y fusión biométrica de cara y voz

Esta sección cubre los contenidos relacionados con el diseño y desarrollo de dos herramientas *software*: la primera de ellas, para evaluar el rendimiento de una aplicación de reconocimiento de locutor, y la segunda para evaluar distintos tipos de fusión a nivel de puntuación de sistemas de reconocimiento facial y de voz.

De manera similar a la verificación facial, se ha seguido el esquema de verificación de voz (ver Figura 2.13) para desarrollar ambos, la herramienta de evaluación de algoritmos de voz y la herramienta de evaluación de fusión biométrica a nivel de puntuación de cara y de voz. Cabe destacar que debido a que estas herramientas se realizaron en la última etapa del TFM solo se recoge el diseño y el desarrollo de estas herramientas, sin poder presentar los resultados de evaluación de voz ni de fusión biométrica. Por ello, para comprobar el funcionamiento de estas herramientas se utilizó la base de datos descrita en la *Sección 3.2: Base de datos de voz*, que aunque sea limitada para evaluar los sistemas de verificación de locutor y de fusión multibiométrica, es suficiente para comprobar el correcto funcionamiento de las aplicaciones.

En esta sección se dividen las actividades realizadas en dos apartados, según los objetivos específicos descritos en el *Anexo A: Objetivos específicos*:

- **Desarrollar un sistema de evaluación de algoritmos de reconocimiento de locutor en C++**: en este apartado se explica el diseño desarrollado para la herramienta informática que evalúa el rendimiento de una API de reconocimiento de voz.
- **Desarrollar un sistema de evaluación de algoritmos de fusión biométrica cara y voz a nivel de puntuación en C++**: en este apartado se implementa una herramienta de evaluación que permite la evaluación de algoritmos de cara y de voz por separado y de manera conjunta, añadiendo la funcionalidad de evaluar diferentes algoritmos de fusión de características a nivel de puntuación.

Para la realización de ambos apartados se tomó la decisión de integrar tecnologías de terceros, adquiriendo así una API de verificación de voz. Es preciso recalcar que el único producto adquirido para la realización de esta herramienta de evaluación ha sido el *software* de verificación, mientras que ambas herramientas de evaluación, tanto la de algoritmos de voz, como la de fusión biométrica cara-voz, han sido desarrolladas de manera íntegra por el autor del TFM.

Para la selección del proveedor de la tecnología de verificación de voz se hizo un estudio de distintas empresas españolas que tuviesen disponibles una API integrable en código C/C++ para la implementación de un sistema de verificación biométrico fusionado con cara y voz, siendo dos de estas empresas las que permitían cumplir este objetivo. El autor de este trabajo participó en reuniones con ambas empresas como responsable del área técnica de voz, junto con el CEO y el CTO de Argos Global, con el fin de recoger la información necesaria para saber si la herramienta con la que estaban trabajando era o no de utilidad a medio y largo plazo para Argos Global. Debido a que la identidad de las empresas se prefiere mantener oculta, de acuerdo con la política de Argos Global, se hará alusión a estas empresas a lo largo de esta memoria como Proveedor A y Proveedor B.

El Proveedor A presentó su aplicación de verificación de voz con las siguientes características: se trataba de un sistema de reconocimiento de voz dependiente de texto, con una topología a nivel de frase (es decir, solo se puede entrenar una frase fijada de antemano, la misma para entrenamiento como para verificación), disponible en dos idiomas: inglés —*My voice is my password*— y en español —«Mi voz es mi clave»—. El sistema requería verificación distribuída: la voz se captura en el dispositivo móvil mediante un programa cliente permitiendo realizar la verificación en un sistema servidor, donde se encuentran los modelos almacenados. El modelo que tenían implementado era un modelo patentado que utilizaba *i-vectors* como características entrenados a partir de un UBM siendo utilizados en una topología HMM a nivel de frase.

En el caso en el que se quisiese cambiar la frase a utilizar, el Proveedor A ofrecía dos alternativas: la primera es que ellos se encargaban de generar la nueva base de datos de entrenamiento para crear un UBM, teniendo que comprarles por separado la constitución de la base de datos, y la segunda, dándoles a ellos la base de datos con la que se quisiera entrenar este modelo de fondo, aunque en este caso no aseguraban la eficiencia de la verificación, ya que la base de datos no la recogen ellos en las condiciones que estiman oportunas. En cualquiera de los dos casos, el entrenamiento del UBM lo realizan de manera *offline* en sus oficinas, no teniendo la libertad de poder entrenarlo por quien adquiere el producto, aunque hubiera adquirido el sistema servidor.

Además de esto, el sistema de verificación podía contar con un sistema *antispoofing* con tres niveles de seguridad: aceptado, indeciso y denegado; de tal manera que si este sistema detectaba de manera indecisa una verificación, haría uso de otras estrategias para poder resolverla. La API del sistema cliente y sevidor estaba escrita en lenguaje de programación Java, disponible para Windows, Macintosh y Linux.

Por otro lado, el Proveedor B ofrecía un sistema de verificación de voz independiente de texto, lo que permitía entrenar y verificarse con cualquier frase. La verificación se podía realizar de manera local en dispositivos móviles, sin la necesidad de tener un servidor (se recuerda que FIDO recomienda que los modelos biométricos no salgan del dispositivo donde se realiza la captura por cuestiones de seguridad, como se mencionó en la *Sección 2: Estado del arte*). En cuanto al modelo implementado, el Proveedor B no da detalles de la tecnología aplicada, pero aseguran que tiene compensación de ruido de canal, ganancia adaptativa, y que los modelos se entrenan a partir de un UBM (disponible en español, inglés de Reino Unido e inglés de Estados Unidos en el momento en el que se hizo la reunión, a principios del año 2015), por lo que puede ser o bien un modelo probabilístico o bien basado en SVM, como GLDS o GSV. Esta tecnología está implementada y puesta en funcionamiento en empresas de *call centers* de España y América Latina.

Este Proveedor B no tiene un sistema de *antispoofing* implícito (sí que estaban trabajando en uno de categorización de canal), pero disponen de otro producto desarrollado por ellos que es un reconocedor de voz, siendo posible utilizarlo como *antispoofing* para evitar ataques suplantación por grabación y reproducción. Ambas APIs están disponibles en lenguaje de programación

C++, y para los sistemas operativos Windows, Macintosh y Linux.

Tras tener reuniones con ambas empresas, el equipo técnico, entre los que se encontraba el autor de este TFM, decidió cuál de las dos tecnologías se adaptaba más a las necesidades de Argos Global, para la implementación de un sistema de verificación biométrica móvil fusionando cara con voz, resultando escogida la aplicación del Proveedor B, debido a su versatilidad a la hora de permitir la verificación con cualquier frase (no únicamente con una única frase pre-entrenada en el servidor) y la facilidad a la hora de realizar la integración con la tecnología actual de Argos Global, entre otras cuestiones comerciales no concernientes a este trabajo.

6.1. Desarrollo de un sistema de evaluación de algoritmos de reconocimiento de locutor en C++

Antes de la implentación del *software* de evaluación de la aplicación de verificación de voz, se decide desarrollar un programa en C++ para comprobar el funcionamiento de la API de verificación de voz adquirida al Proveedor B y familiarizarse con las funciones de la interfaz. Para ello, el autor del trabajo aprendió el lenguaje de programación C++, haciendo uso de la referencia [47]; y se desarrolló esta aplicación de demostración (*Software* de demostración de reconocimiento de locutor en C++[1]), consistente en un programa que utiliza los valores por defecto de la API (recomendados por el Proveedor B) y que permite realizar dos opciones: iniciar un enrolamiento, recogiendo un mínimo de cinco frases fonéticamente balanceadas, y realizar una verificación, mostrando si la verificación ha sido aceptada o rechazada.

Una vez habiendo experimentado con esta aplicación, se decide desarrollar el sistema de evaluación de esta aplicación de verificación de voz (*Software* de evaluación de algoritmos de verificación por voz en C++[2]), de manera similar al sistema de evaluación de algoritmos de verificación facial descrito en la *Sección 4.1: Desarrollo de un sistema de evaluación de algoritmos de detección, modelado y verificación facial en C/C++*, pero en este caso en lenguaje C++ puro. Este sistema se diseña con el ánimo de usar el *software* de verificación de voz del Proveedor B como un modelo más, ya que el objetivo de Argos Global ha sido poder disponer de su propia tecnología de voz; de tal modo que esta herramienta tiene como propósito servir en su momento para evaluar algoritmos de verificación de voz implementados por Argos Global.

El hecho de programar esta aplicación en C++ puro viene determinado por los siguientes tres motivos: primero, la migración de lenguaje de C/C++ a C++ puro del *software* implementado por Argos Global, segundo, las ventajas que ofrece C++ en aplicaciones que requieran alto rendimiento, y tercero, porque se quería que la herramienta que permitiese evaluar la fusión biométrica (la cual se basa en este *software* de evaluación) quería que estuviese implementada en el mismo lenguaje que el de desarrollo.

Internamente, el sistema de evaluación se divide en cuatro funciones principales (métodos de una clase denominada *Evaluation*) a diferencia del sistema de evaluación de algoritmos de verificación facial, ya que el preprocesado del *software* de verificación de voz del Proveedor B se realiza de forma interna tanto a la hora de generar los modelos, como a la hora de verificar:

1. **Iniciación**: en esta función se inicia el sistema, de donde se lee un fichero de configuración.
2. **Generación de modelo de locutor**: se generan los modelos de entrenamiento (indistintamente llamados en la memoria «modelos de voz») utilizando los parámetros escogidos en el fichero de configuración.

[1] Entregable E-6.1.1. Ver *Anexo A: Objetivos específicos.*
[2] Entregable E-6.1.2. Ver *Anexo A: Objetivos específicos.*

3. **Verificación de voz**: utilizando una locución de verificación se compara con diferentes modelos de locutor.

4. **Resultados**: genera un fichero con un resumen de los resultados.

Cabe destacar las diferencias de este sistema de evaluación de voz con respecto al sistema de evaluación facial desarrollado: a diferencia de los algoritmos de verificación facial basados en distancias implementados, donde se utilizan una secuencia de imágenes tanto para entrenamiento como para verificación, se extraen características y se mide la distancia entre ellas de manera simétrica —los datos y la extracción de características utilizados en entrenamiento del modelo facial y en verificación facial pueden ser usados para una etapa u otra de manera indistinta—, para el sistema de reconocimiento de voz se necesita considerablemente más datos para ajustar el UBM al locutor durante la etapa de entrenamiento, y únicamente una locución (de unos 5 segundos aproximadamente) para verificar; aunque el preprocesamiento de las locuciones empleadas para entrenamiento y verificación ha de ser el mismo.

Por esta razón, se ha tenido que modificar tanto los ficheros de configuración para adaptarlo a esta nueva estructura y a la forma en la que se genera el fichero de base de datos, en el que se determinan qué sesiones de los distintos usuarios de la base de datos se utilizan para entrenar el modelo y qué ficheros de audio se utilizan para test. De igual manera a como se hizo en el sistema de evaluación de algoritmos de verificación facial, el número de comparaciones de los modelos enrolados con usuarios genuinos e impostores es el mismo, y el sistema permite guardar los modelos biométricos de enrolamiento, con el fin de no tener la necesidad de generarlos cada vez que se desee realizar distintas verificaciones.

Al igual que en el sistema de evaluación de algoritmos facial, el fichero de resultados que genera el sistema muestra simplemente métricas utilizadas para evaluar de manera preliminar qué tan bien funciona el conjunto de parámetros seleccionados. En este fichero, además de un resumen de los parámetros utilizados, se presentan las métricas EER, FMR al 0 %, 0,01 %, 0,1 % y 1 % y el Área Bajo la Curva (AUC, *Area Under the Curve*). Además de estos resultados se genera también un fichero de puntuaciones, donde se guarda la puntuación (o *score*) de las distintas comparaciones.

La Figura 6.1 muestra cómo se realizó la implementación de este sistema. En azul se pueden observar las cuatro funciones implementadas ya mencionadas: Iniciación, Generación de modelo de locutor, Verificación de voz y Resultados. También se observa que para comunicarse con el sistema se utilizan los ficheros de configuración y de base de datos. Como se dijo, el fichero de base de datos determina qué locuciones generan el modelo de entrenamiento y las dirige al método de Generación del modelo de locutor, y cuales son locuciones de test y las envía al método de Verificación de voz. Como respuesta, una vez realizadas todas las verificaciones, se obtienen los ficheros de resultados y de puntuaciones. Al igual que se hacía con el sistema de evaluación de algoritmos de verificación facial, es posible almacenar los modelos de voz con el fin de no tener que calcularlas si se desean utilizar otras locuciones de verificación u otros parámetros relacionados con las locuciones de verificación.

En los siguientes apartados se explica en qué consisten las fases de generación del modelo de locutor y la verificación de voz.

6.1.1. Generación del modelo de locutor

En este apartado se genera un modelo de entrenamiento utilizando 10 locuciones de aproximadamente 5 segundos cada una. Se recomienda que estas locuciones sean fonéticamente balanceadas (como se vio en la *Sección 3.2: Base de datos de voz*) con el fin de poder generar

un modelo que contemple pruebas del mayor número de sonidos posibles producidos por el usuario de entrenamiento. Este modelo queda almacenada de manera cifrada para cada usuario. Como parámetros a modificar, la API de reconocimiento de locutor integrada tiene un parámetro que varía la manera de realizar el preprocesamiento relacionado con la energía de la señal del Detector de Actividad de Voz (VAD), la cual se puede modificar variando los parámetros del fichero de configuración.

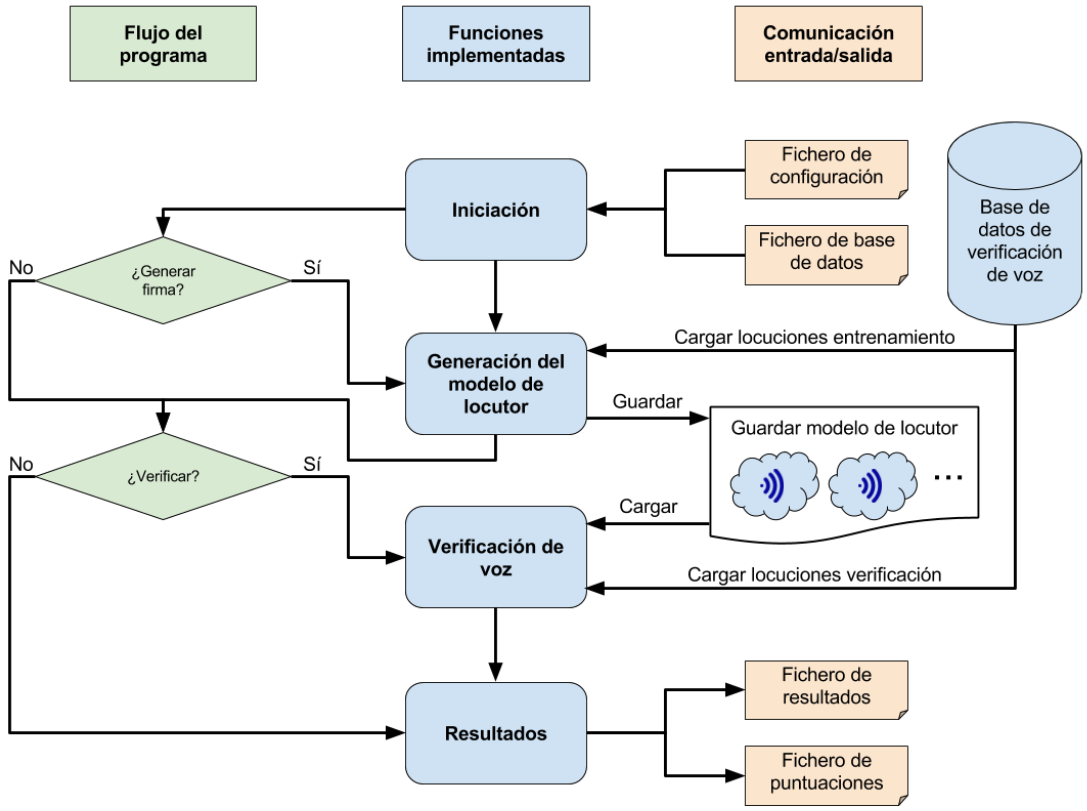

Figura 6.1: Sistema de evaluación de algoritmos de verificación por voz.

6.1.2. Verificación de voz

Para la parte de evaluación de los modelos de entrenamiento se utilizan, como se dijo anteriormente, locuciones de entorno a 5 segundos de duración cada una. El fichero de base de datos generado permite que por cada usuario se escojan pseudo-aleatoriamente tantas locuciones de verificación de usuarios impostores como de genuinos, con el fin de que ambas puntuaciones estén equilibradas. La API de reconocimiento de locutor integrada ofrece, además de devolver un valor de puntuación normalizada por cada verificación, la posibilidad de utilizar una variable umbral con diferentes niveles de seguridad prefijados, además de poder modificar el parámetro relacionado con el Detector de Actividad de Voz (VAD), el cual siempre se considerará el mismo que el que se utilice en entrenamiento, para que el procesamiento realizado en los datos utilizados en entrenamiento del modelo y en los de test sea el mismo.

6.2. Desarrollo de un sistema de evaluación de algoritmos de fusión biométrica cara y voz a nivel de puntuación en C++

Siguiendo el esquema de desarrollo implementado en los *softwares* de evaluación de algoritmos de cara y de voz descritos anteriormente en esta memoria, se decide realizar un sistema que pueda integrar ambos sistemas y que además permita evaluar distintos algoritmos de fusión de puntuaciones de ambos sistemas biométricos, es decir, fusión multibiométrica (*Software* de evaluación de algoritmos de verificación facial y de locutor[3]).

Para acometer este problema se utiliza el código íntegro de *software* de evaluación de algoritmos de verificación por voz en C++ presentado en la sección anterior y se adapta el *software* de evaluación de algoritmos de verificación facial en C/C++ presentado en la *Sección 4.1: Desarrollo de un sistema de evaluación de algoritmos de detección, modelado y verificación facial en C/C++* en dos aspectos: el primero para que esté escrito íntegramente en C++, y el segundo para que haga uso de la API de verificación facial en C++ de Argos Global presentada en la *Sección 5.2: Diseño de una API de verificación facial en C++, VerifyByFace.*

Este *software* permite la ejecución en paralelo de los verificadores de cara y voz. Internamente, este sistema se divide en siete funciones principales (métodos de una clase denominada *Evaluation*).

1. **Iniciación**: en esta función se inicia el sistema, de donde se lee un fichero de configuración.
2. **Detección y generación del modelo facial**: detecta las imágenes en la base de datos facial y genera los modelos faciales.
3. **Verificación facial**: realiza las comparaciones entre los modelos de enrolamiento y verificación.
4. **Generación del modelo de locutor**: se generan los modelos de entrenamiento utilizando los parámetros escogidos en el fichero de configuración.
5. **Verificación de voz**: utilizando una locución de verificación se compara con diferentes modelos de locutor.
6. **Fusión biométrica**: utilizando las puntuaciones de la verificación de cara y de voz se pueden evaluar diferentes tipos de algoritmos de fusión biométrica.
7. **Resultados**: genera un fichero con un resumen de los resultados.

Este programa permite realizar ejecuciones parciales e independientes, permitiendo por ejemplo utilizarlo con un único tipo de biometría en concreto, utilizando modelos guardados y utilizar las puntuaciones de ejecuciones anteriores para realizar la fusión biométrica. Esto se controla mediante el fichero de configuración, indicando si existen modelos o puntuaciones ya extraídas, permitiendo alta versatilidad para las pruebas de evaluación a realizar, evitando recalcular modelos o puntuaciones previamente computados.

También el fichero de base de datos se modifica para poder realizar la fusión biométrica, de tal manera que el número de usuarios a evaluar es el mínimo entre usuarios de la base de datos de biometría facial y de voz. A los usuarios se les asigna un ID virtual que enlaza un usuario de la base de datos facial con otro de la base de datos de voz, maximizando el número de sesiones disponibles entre todos los usuarios: este programa externo que genera el fichero de base de datos se ejecuta en dos tiempos, primero analizando el número de sesiones útiles y luego

[3]Entregable E-6.2.1. Ver *Anexo A: Objetivos específicos.*

asignando un ID virtual a estas sesiones maximizando el número de sesiones por usuario. No se tiene en cuenta, por ejemplo, que un usuario varón de la base de datos facial se le asocie con un usuario mujer de la base de datos de voz, ya que el objetivo de esta evaluación es que ambos usuarios de ambas bases de datos estén relacionados unívocamente. Finalmente se determinan las sesiones de entrenamiento y test para que cada usuario tenga las mismas sesiones de genuino y de impostor, seleccionando estas de manera pseudo-aleatoria[4].

La Figura 6.2 muestra cómo se realizó la implementación de este sistema. En azul se pueden observar los siete métodos implementados: Iniciación, Detección y generación del modelo facial, Verificación facial, Generación del modelo de locutor, Verificación de voz, Fusión biométrica y Resultados. De igual modo que en los sistemas de evaluación anteriormente presentados, la comunicación con el sistema se observa que para comunicarse con el sistema se utilizan los ficheros de configuración y de base de datos. Las verificaciones de cara y de voz se pueden ejecutar en paralelo hasta la parte de fusión, donde se requiere que ambas partes hayan acabado para fusionar las puntuaciones. Como respuesta, una vez realizadas todas las verificaciones, se obtienen los ficheros de resultados y de puntuaciones. Se puede utilizar este sistema para evaluar algoritmos de verificación facial, de voz o de fusión, de manera independiente al resto, siendo en este último caso necesarias las puntuaciones de ambas anteriores. Igual que se vio en los otros *softwares* de evaluación, es posible guardar y cargar los resultados de pasos intermedios, como detecciones de caras, modelos faciales y modelos de locutor, además de las puntuaciones de cara y de voz para realzar distintas evaluaciones de fusión.

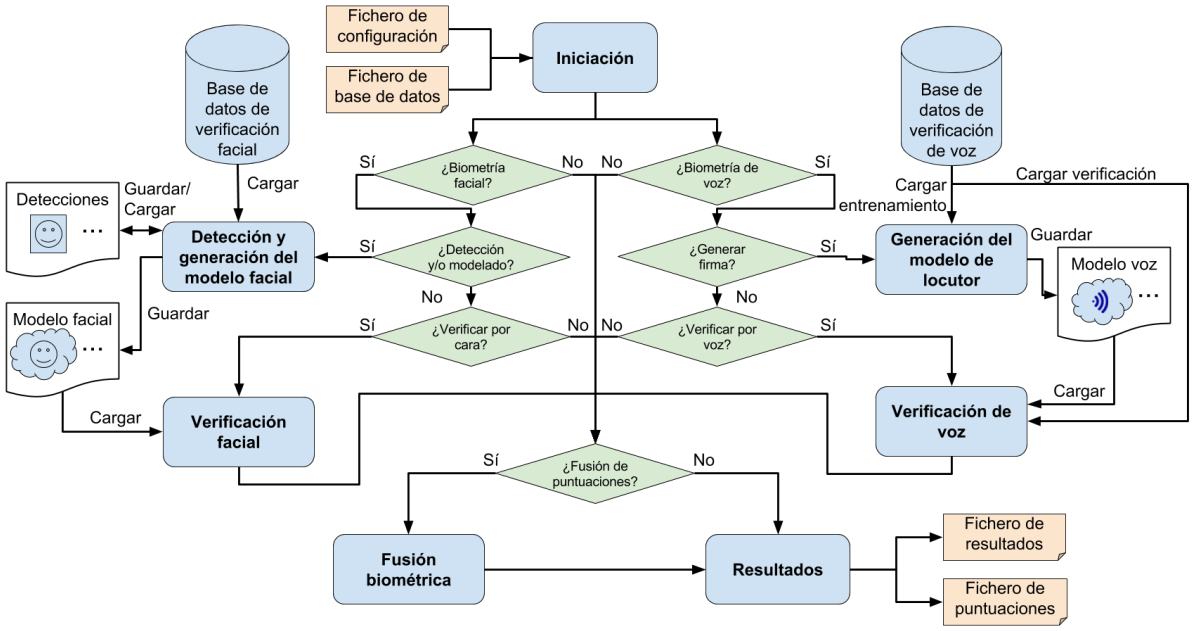

Figura 6.2: Sistema de evaluación de algoritmos de verificación por cara, voz y fusión biométrica.

A continuación, se explica en qué consiste el apartado de la fusión biométrica ya que los otros apartados fueron explicados en secciones anteriores.

[4]Las sesiones de entrenamiento para la verificación por voz se restringen a que sean siempre de las sesiones de frases fonéticamente balanceadas.

6.2.1. Fusión biométrica

En esta función se hace uso de las puntuaciones de ambas evaluaciones de verificación con el ánimo de fusionarlas. En el momento de desarrollo de esta función solo se pudo implementar la función suma y la función producto, pero el programa está diseñado para poder realizar otro tipo de fusión biométrica mencionados en la *Sección 2.4: Fusión biométrica* como son el uso de la función máximo, mínimo, ponderaciones o algoritmos de aprendizaje automático [35, 36].

Esta función además calcula diferentes métricas de evaluación de sistemas como el FMR 0 %, 0,01 %, 0,1 % y 1 %, el EER y el Área Bajo la Curva. Junto a esto, la función puede computar y guardar las curvas DET de las puntuaciones de fusión y de cada rasgo independiente y se pueden calcular diferentes fusiones a la vez.

Como ya se dijo al principio de esta sección este sistema de evaluación se hizo al final del Trabajo de Fin de Máster, por lo que no se han podido presentar evaluaciones de fusión entre ambos sistemas. Sin embargo, este sistema está siendo utilizado por el equipo técnico de Argos Global en la actualidad y se está haciendo uso de él para realizar las evaluaciones de biometría facial, utilizando nuevas versiones de la API de VerifyByFace y de fusión multibiométrica cara y voz.

7

Conclusiones y trabajo futuro

En esta sección se exponen las conclusiones que se han extraído del trabajo realizado, así como tareas que quedaron pendientes y que se plantea continuar desarrollando, o ya se están llevando a cabo en Argos Global, las cuales permiten continuar y mejorar el trabajo que aquí se ha expuesto. Debido a que los objetivos de este trabajo han sido diversos, se expondrán tanto las conclusiones como el trabajo futuro por secciones, al igual que en la memoria, con el fin de poder centrar estas conclusiones a las diferentes partes del trabajo.

7.1. Conclusiones

Gracias al trabajo realizado por el autor de este trabajo se han conseguido satisfacer las necesidades que tenía Argos Global para conseguir implementar un verificador facial en dispositivos móviles y disponer de una herramienta para poder evaluar diferentes algoritmos de verificación tanto faciales, de voz, como de fusión biométrica a nivel de puntuación de ambos rasgos biométricos.

7.1.1. Bases de datos

La estructuración que se ha realizado de las bases de datos ha permitido escoger los usuarios y las sesiones para utilizarlas en los sistemas de evaluación desarrollados en el TFM de manera sencilla y uniforme para cualquier tipo de base de datos que se desee integrar en el sistema. También cabe recalcar que la estructura que se ha diseñado permite añadir bases de datos nuevas tanto de los rasgos biométricos tratados en este trabajo como de otros rasgos biométricos los cuales podrían incluirse en un futuro.

La base de datos facial es suficientemente amplia para comenzar a validar los algoritmos, ofreciendo variabilidad de iluminación y gestos. En cambio, la base de datos de voz es necesario completarla bien aumentando el número de usuarios o bien incluyendo otras bases de datos públicas, para poder realizar evaluaciones de fusión entre biometrías facial y de voz. Además, es conveniente incluir en esta base de datos diferentes tipos de idiomas, para poder entrenar los modelos de voz utilizando locuciones de usuarios con otros sonidos distintos a los que se producen en castellano.

7.1.2. Desarrollo y evaluación de un sistema de verificación facial

La herramienta de evaluación de algoritmos aquí presentada ha servido tanto para realizar las evaluaciones de algoritmos expuestos en esta memoria, como de base para las subsiguientes herramientas de evaluación implementadas en este trabajo, tales como son la herramienta de evaluación de algoritmos de voz y la de algoritmos de fusión, presentadas en la *Sección 6: Verificación de locutor y fusión biométrica de cara y voz*. Esta herramienta es fácilmente adaptable a cualquier tipo de evaluación simplemente realizando cambios sobre el fichero de configuración, el cual se requiere como parámetro de entrada del ejecutable desarrollado. Esta herramienta, como ya se ha mencionado, ha sido utilizada en posteriores investigaciones de Argos Global por otros miembros del equipo para la evaluación de sus algoritmos.

En cuanto a las evaluaciones realizadas de verificación facial se ha mejorado el sistema de partida utilizando la fusión suma entre las puntuaciones obtenidas a partir de las imágenes de la cara y de los ojos del usuario, usando como características el histograma LBP y la distancia Manhattan como medida de comparación, realizando un preprocesado a la imagen facial mediante el uso de contraste adaptativo, consiguiendo resultados de EER del 7,3 %. También se ha comprobado que estos resultados mejoran cuanto mayor es la resolución de la imagen y cuantas más imágenes utilicemos en el modelo, siendo un compromiso aceptable para la verificación en entornos móviles 120x120 píxeles y 5 imágenes. La inclusión de las características WDA mejoran en un punto los resultados del EER que con las características evaluadas en este trabajo.

Respecto a sistemas *antispoofing* se ha comprobado que es sencillo implementar un sistema *antispoofing* facial 3D para evitar ataques de suplantación mediante reproducciones de vídeo y fotografías midiendo si hay diferencias suficientemente pronunciadas entre los valores de los píxeles de la nariz y los ojos en la imagen de profundidad. Por otro lado, respecto a sistemas *antispoofing* facial 2D, se ha visto que sistemas comerciales que utilizan detector de vida por parpadeo son vulnerables a ataques de reproducción por vídeo, además de que hay que esperar a que el usuario parpadee para poder verificarlo, haciendo que la aplicación funcione con mayor latencia. El sistema *antispoofing* 2D en el que el autor participó en la etapa de diseño ha sido implementado en herramientas como la versión *pro* de la aplicación móvil FaceOnVox y la versión *pro* del salvapantallas que utiliza tecnología de VerifyByFace, así como la demo comercial en arquitectura local.

7.1.3. Demo comercial y aplicación móvil FaceOnVox

En esta sección se vieron distintas aplicaciones desarrolladas con la tecnología de verificación facial desarrollada íntegramente por Argos Global denominada VerifyByFace. Estas aplicaciones han permitido acuerdos comerciales con clientes, proveedores, *partners* e inversores al poder mostrar en acción el desarrollo tecnológico de Argos Global en escenarios reales de aplicación, y no únicamente en sesiones de laboratorio.

El *software* de demostración en arquitectura local permite hacer tangible una sencilla aplicación de verificación facial bien para mostrar en las reuniones por el equipo comercial de Argos Global, como para poder ser distribuida bajo una licencia comercial. Además, esta herramienta ha sido utilizada también en entornos comerciales como es la feria eShow 2014 o el *Global Sports Innovation Center*, ambas en Madrid.

Por otro lado, la aplicación de salvapantallas desarrollada a partir de la demo en arquitectura local permite dar una aplicación concreta al sistema de verificación, además de ser utilizada por Argos Global para darse a conocer no solo en círculos comerciales, sino también al público general.

El *software* de demostración en arquitectura cliente servidor no ha llegado a ser tan utilizado

como el de arquitectura todo local, pero se estima que pueda ser desarrollado en mayor profundidad en un futuro. Por otro lado, la investigación y el desarrollo de sistemas de compresión y cifrado de modelos y de mensajes entre cliente y servidor realizado en esta parte ha sido útil para la empresa en diferentes áreas relacionadas con trabajo aquí presentado.

La API de VerifyByFace diseñada en esta sección ha sido utilizada finalmente en todas las herramientas de Argos Global relacionadas con la verificación facial, y ha hecho uso de código desarrollado por el autor del TFM. Su diseño simple ha permitido a empresas afiliadas con Argos Global disponer de la tecnología de la empresa durante el tiempo designado por la licencia adquirida, permitiendo así a Argos Global acuerdos comerciales y de colaboración.

La aplicación móvil FaceOnVox ha permitido acercar a los usuarios la tecnología de Argos Global, mediante el acercamiento de esta aplicación al escenario real de aplicación. El *feedback* recibido por parte de los usuarios sirve como guía para realizar mejoras del sistema con los usuarios finales de futuras aplicaciones de verificación facial en dispositivos móviles que Argos Global lance en un futuro.

7.1.4. Verificación de locutor y fusión biométrica de cara y voz

En esta sección se realiza una primera aproximación al problema de la fusión biométrica con sensores integrados en dispositivos móviles: Argos Global realiza el primer contacto con tecnología de verificación de locutor, a través de un *partner*, el cual fue seleccionado por el grupo técnico y comercial de la empresa, en el que se encontraba el autor del trabajo.

La aplicación de verificación de voz desarrollada para probar la tecnología de este *partner* ha servido para conocer cómo funcionaba su API, y en un futuro será utilizada en una herramienta de verificación multibiométrica con la demo en arquitectura local de verificación facial para poder mostrar el rendimiento de la tecnología de fusión biométrica que está desarrollando Argos Global.

La herramienta de evaluación de algoritmos de voz ha permitido valorar el rendimiento de la API de verificación de voz adquirida. Se prevé que a corto/medio plazo esta herramienta implementada sirva para evaluar los algoritmos de verificación de voz desarrollados con tecnología propia de Argos Global.

Finalmente, el diseño de la herramienta de evaluación de algoritmos de cara, voz y fusión biométrica ha permitido a Argos Global empezar a evaluar diferentes algoritmos de fusión biométrica, lo que encabezará la segunda etapa tecnológica de la empresa al incorporar tecnología de fusión biométrica en sistemas de verificación, utilizando como dispositivos de soporte de estas aplicaciones los dispositivos móviles disponibles en el mercado.

7.2. Trabajo futuro

Este trabajo se encuentra en el contexto de un proyecto dentro de una empresa, con más personas involucradas y de mayor duración que lo que se expone en este trabajo. Por ello, quedan aún tareas que realizar para mejorar el estado actual de la técnica, los algoritmos y las aplicaciones presentadas.

7.2.1. Bases de datos

El uso de más datos siempre permite generalizar mejor los algoritmos, pudiendo garantizar su funcionamiento en diferentes entornos. Respecto a esto, Argos Global tiene pensado incluir otras

bases de datos, tanto públicas como privadas para evaluar sus algoritmos de verificación facial y de voz; además de ampliar las bases de datos facial —FOnFaces— y de voz —FOnVoice— de la empresa.

7.2.2. Desarrollo y evaluación de un sistema de verificación facial

Respecto a los algoritmos de verificación facial es donde más trabajo hay para Argos Global: la implementación de nuevos algoritmos que permitan una verificación más rápida y más fiable, una evaluación en mayor profundidad de los parámetros utilizados en los algoritmos presentados, o el uso de tecnología más compleja (árboles de decisión o métodos probabilísticos), como la exploración de tecnologías de reconocimiento facial de segunda generación que permitan la verificación facial en dispositivos móviles son algunas de las tareas que está llevando a cabo en la actualidad en la empresa.

Además, se está desarrollando actualmente en Argos Global tecnología de verificación en remoto (ya que en local los modelos son altamente pesados) utilizando *DeepLearning* tanto para la detección de caras en imágenes como en el modelado. Para ello, también se ha propuesto mejorar la captación de imágenes para generar el modelo utilizando toda la información que puede ofrecer un vídeo de 30 segundos, pidiendo al usuario colaboración para poder modelar no solo la cara de frente, sino también distintas poses, incluyendo los perfiles del mismo.

Otro sistema a desarrollar en mayor profundidad es el sistema *antispoofing*, el cual se quiere mejorar utilizando una mayor cantidad de datos y modificando el diseño aquí presentado, para hacerlo más rápido y con menor falso rechazo.

7.2.3. Demo comercial y aplicación móvil FaceOnVox

Respecto a las aplicaciones de demostración, se desea mejorar tanto la interfaz, haciéndola más atractiva, como la usabilidad de ésta. En cuanto a la demo en arquitectura local, como se dijo, se espera realizar un sistema que utilice la fusión biométrica cara con voz para poder presentarlo en reuniones que tenga el equipo comercial de Argos Global. Por otra parte, se prevé utilizar el código de la demo con arquitectura cliente servidor con el fin de crear un servicio web que detecte caras, ataques de *spoofing*, e incluso realice verificaciones en remoto de manera segura utilizando tecnología más avanzada, como se ha mencionado en el apartado anterior.

La API de VerifyByFace está en continuo desarrollo, y se va actualizando con los nuevos algoritmos que hacen mejorar la verificación facial del sistema. Además se espera que esta API sea aún más rápida, disminuyendo los tiempos de ejecución de algoritmos explotando más el lenguaje en la que está programada.

FaceOnVox, la aplicación móvil que funciona como *App Locker* está en desarrollo desde que salió al mercado de Android. Se espera que próximamente incluya mejoras en la verificación, reducción de tiempo en la verificación y la adquisición de imágenes, y solvente problemas relacionados con la propia aplicación. Además de esto, también se está desarrollando la versión pro de la aplicación, que incluya el *antispoofing* facial 2D que se está desarrollando.

7.2.4. Verificación de locutor y fusión biométrica de cara y voz

El trabajo futuro próximo de Argos Global es poder desarrollar su tecnología propia de verificación por voz para realizar fusión biométrica junto con algoritmos de verificación facial, siendo el trabajo aquí presentado el primer paso para empezar a trabajar en esta temática. Teniendo las herramientas de evaluación de algoritmos de fusión, la siguiente etapa del proyecto

es partir de una base de datos suficientemente representativa (sobre todo respecto a la voz) para poder realizar las evaluaciones necesarias.

Según avancen los sensores biométricos en los dispositivos móviles se contemplará incluir rasgos biométricos diferentes, como pueden ser el de huella dactilar o el de iris.

En paralelo a esto, también se quiere desarrollar un sistema de *antispoofing* de voz, que genere frases diferentes en tiempo de ejecución y que reconozca que la frase presentada corresponde con la frase que se ha dicho.

Finalmente, se espera que todo este trabajo permita realizar una aplicación de verificación facial y de voz móvil que incluya *antispoofing* de ambas tecnologías para realizar pagos *online*, disminuyendo así el fraude y permitiendo una verificación más segura y rápida, siendo este trabajo el principal paso hacia este objetivo.

Bibliografía

[1] Face Recognition with OpenCV — OpenCV 2.4.13.2 documentation.

[2] Sachin Sudhakar Farfade, Mohammad J. Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 643–650. ACM, 2015.

[3] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006.

[4] Biao Wang, Weifeng Li, Wenming Yang, and Qingmin Liao. Illumination normalization based on weber's law with application to face recognition. *IEEE Signal Processing Letters*, 18(8):462–465, 2011.

[5] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

[6] Jacob Benesty, M. Mohan Sondhi, and Yiteng Huang. *Springer handbook of speech processing*. Springer Science & Business Media, 2007.

[7] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. An Analysis of Minutiae Matching Strength. In *Audio- and Video-Based Biometric Person Authentication*, pages 223–228. Springer, Berlin, Heidelberg, June 2001.

[8] K. P. Tripathi. A comparative study of biometric technologies with reference to human interface. *International Journal of Computer Applications*, 14(5):10–15, 2011.

[9] Austin Hicklin, Brad Ulery, and Craig Watson. A brief introduction to biometric fusion. *National institute of standards and technology*, 2006.

[10] Kenneth H. Rose. A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition. *Project management journal*, 44(3):e1–e1, 2013.

[11] Anil Jain, Patrick Flynn, and Arun A. Ross. *Handbook of biometrics*. Springer Science & Business Media, 2007.

[12] Anil K. Jain, Arun Ross, and Sharath Pankanti. Biometrics: a tool for information security. *IEEE transactions on information forensics and security*, 1(2):125–143, 2006.

[13] Rupinder Saini and Narinder Rana. Comparison of various biometric methods. *International Journal of Advances in Science and Technology (IJAST)*, 2(1):2, 2014.

[14] R. Vera-Rodriguez, J. Fierrez, P. Tome, and J. Ortega-Garcia. Face Recognition at a Distance: Scenario Analysis and Applications. In Andre Ponce de Leon F. de Carvalho, Sara Rodríguez-González, Juan F. De Paz Santana, and Juan M. Corchado Rodríguez, editors, *Distributed Computing and Artificial Intelligence*, number 79 in Advances in Intelligent

and Soft Computing, pages 341–348. Springer Berlin Heidelberg, 2010. DOI: 10.1007/978-3-642-14883-5_44.

[15] Paul Viola and Michael J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[16] Hitoshi Imaoka. Face Recognition: Beyond the Limit of Accuracy. In *International Joint Conference on Biometrics*. International Joint Conference on Biometrics, 2014.

[17] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[18] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[19] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[20] Yaniv Taigman, Lior Wolf, Tal Hassner, and others. Multiple One-Shots for Utilizing Class Label Information. In *BMVC*, volume 2, pages 1–12, 2009.

[21] Chang Huang, Shenghuo Zhu, and Kai Yu. Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. *arXiv preprint arXiv:1212.6094*, 2012.

[22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[23] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[24] Yaniv Taigman, Ming Yang, MarcÁurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[25] Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech communication*, 52(1):12–40, 2010.

[26] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1):19–41, January 2000.

[27] William M. Campbell. Generalized linear discriminant sequence kernels for speaker recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 1, pages I–161. IEEE, 2002.

[28] William M. Campbell, Douglas E. Sturim, and Douglas A. Reynolds. Support vector machines using GMM supervectors for speaker verification. *IEEE signal processing letters*, 13(5):308–311, 2006.

[29] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[30] Vincent Aubanel, Maria Luisa García Lecumberri, and Martin Cooke. The Sharvard Corpus: A phonemically-balanced Spanish sentence resource for audiology. *International journal of audiology*, 53(9):633–638, 2014.

[31] IEEE Recommended Practice for Speech Quality Measurements. *IEEE No 297-1969*, pages 1–24, June 1969.

[32] Arun A. Ross, Karthik Nandakumar, and Anil Jain. *Handbook of Multibiometrics*. Springer Science & Business Media, August 2006. Google-Books-ID: JpUdlJnuE2MC.

[33] Anil Jain, Arun A. Ross, and Karthik Nandakumar. *Introduction to Biometrics*. Springer Science & Business Media, November 2011. Google-Books-ID: ZPt2xrZFtzkC.

[34] J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Fusion strategies in multimodal biometric verification. In *2003 International Conference on Multimedia and Expo, 2003. ICME '03. Proceedings*, volume 3, pages III–5–8 vol.3, July 2003.

[35] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12):2270–2285, December 2005.

[36] Arun Ross and Anil Jain. Information fusion in biometrics. *Pattern Recognition Letters*, 24(13):2115–2125, September 2003.

[37] J. Fierrez-Aguilar, J. Ortega-Garcia, D. Garcia-Romero, and J. Gonzalez-Rodriguez. A Comparative Evaluation of Fusion Strategies for Multimodal Biometric Verification. In *Audio- and Video-Based Biometric Person Authentication*, pages 830–837. Springer, Berlin, Heidelberg, June 2003.

[38] Julian Fierrez-Aguilar, Javier Ortega-Garcia, Joaquin Gonzalez-Rodriguez, and Josef Bigun. Discriminative multimodal biometric authentication based on quality measures. *Pattern Recognition*, 38(5):777–779, May 2005.

[39] *Handbook of Biometric Anti-Spoofing - Trusted Biometrics | Sébastien Marcel | Springer*. 2014.

[40] Gang Pan, Zhaohui Wu, and Lin Sun. Liveness detection for face recognition. In *Recent advances in face recognition*. InTech, 2008.

[41] J. Galbally, S. Marcel, and J. Fierrez. Biometric Antispoofing Methods: A Survey in Face Recognition. *IEEE Access*, 2:1530–1552, 2014.

[42] Federico Alegre, Ravichander Vipperla, and Nicholas Evans. Spoofing countermeasures for the protection of automatic speaker recognition systems against attacks with artificial signals. September 2012.

[43] Julian Fierrez, Javier Ortega-Garcia, Doroteo Torre Toledano, and Joaquin Gonzalez-Rodriguez. Biosec baseline corpus: A multimodal biometric database. *Pattern Recognition*, 40(4):1389–1392, April 2007.

[44] J. Fierrez, J. Galbally, J. Ortega-Garcia, M. R. Freire, F. Alonso-Fernandez, D. Ramos, D. T. Toledano, J. Gonzalez-Rodriguez, J. A. Siguenza, J. Garrido-Salas, E. Anguiano, G. Gonzalez-de Rivera, R. Ribalda, M. Faundez-Zanuy, J. A. Ortega, V. Cardeñoso-Payo, A. Viloria, C. E. Vivaracho, Q. I. Moro, J. J. Igarza, J. Sanchez, I. Hernaez, C. Orrite-Uruñuela, F. Martinez-Contreras, and J. J. Gracia-Roche. BiosecurID: a multimodal biometric database. *Pattern Anal Applic*, 13(2):235–246, May 2010.

[45] Javier Ortega-Garcia, Joaquin Gonzalez-Rodriguez, and Victoria Marrero-Aguiar. AHUMADA: A large speech corpus in Spanish for speaker characterization and identification. *Speech Communication*, 31(2–3):255–264, June 2000.

[46] Martin Reddy. *API Design for C++*. Elsevier, March 2011. Google-Books-ID: IY29LylT85wC.

[47] Stephen Prata. *C++ primer plus*. Sams Publishing, 2002.

Definiciones

- **Antispoofing**: soluciones tecnológicas aplicadas a evitar la suplantación de identidad en un sistema biométrico a nivel del sensor.
- **App Locker**: aplicación móvil cuya misión es bloquear otras aplicaciones en un dispositivo, solicitando algún método de autenticación.
- **Argos Soluciones Globales S.L.**: PYME de base tecnológica donde se realizó el Trabajo de Fin de Máster aquí presentado.
- **FOnFaces**: base de datos facial recogida y constituida por Argos Soluciones Globales S.L.
- **FOnVoice**: base de datos de voz recogida y constituida por Argos Soluciones Globales S.L.
- **FaceOn**: portfolio de soluciones biométricas de la empresa Argos Soluciones Globales S.L.
- **FaceOnVox**: aplicación móvil de Argos Soluciones Globales S.L. que utiliza biometría facial (VerifyByFace) para el bloqueo de otras aplicaciones del dispositivo móvil.
- **Global Sports Innovation Center**: centro de innovación tecnológica para empresas de Microsoft que ofrecen soluciones innovadoras en el ámbito deportivo.
- **KeyLemon**: empresa de referencia con soluciones de verificación facial y *antispoofing*.
- **OpenCV**: librería de programación en C++ con funciones dedicadas a la visión por computador y al tratamiento de imágenes.
- **OpenSSL**: librería de programación en C con funciones dedicadas a la encriptación.
- **Spoof**: cualquier mecanismo o tipo de fraude que pretenda suplantar la identidad de un usuario burlando el sistema biométrico frente al sensor.
- **Tecnologías del Habla**: área del tratamiento de señales acústicas en la que se recogen las tecnologías relacionadas con el tratamiento de señales de voz.
- **VerifyByFace**: solución tecnológica de la empresa Argos Soluciones Globales S.L. de verificación facial.
- **eShow**: feria profesional de comercio electrónico y márketing digital.

Glosario de acrónimos

- **API**: *Application Programming Interface*
- **AUC**: *Area Under the Curve*
- **CLAHE**: *Contrast Limited Adaptive Histogram Equalization*
- **DET**: *Detection Error Tradeoff*
- **DFT**: *Discrete Fourier Transform*
- **DTW**: *Dynamic Time Wrapper*
- **EER**: *Equal Error Rate*
- **FIDO**: *Fast Identification Online*
- **FLOPS**: *Float-Point Operations per Second*
- **FMR**: *False Match Ratio*
- **FNMR**: *False No-Match Ratio*
- **GLDS**: *Generalized Linear Discriminant Sequence*
- **GMM**: *Gaussian Mixture Model*
- **GSV**: *GMM Supervector*
- **HMM**: *Hidden Markov Model*
- **IDFT**: *Inverse Discrete Fourier Transform*
- **LBP**: *Local Binary Patterns*
- **LDA**: *Linear Discriminant Analysis*
- **MFCC**: *Mel-Frequency Cepstral Coefficients*
- **ML**: *Maximum Likelihood*
- **NIST**: *National Institute of Standards and Technology*
- **PCA**: *Principal Component Analysis*
- **SVM**: *Support-Vector Machine*
- **TFM**: Trabajo de Fin de Máster
- **UBM**: *Universal Background Model*
- **VAD**: *Voice Activity Detection*

- **VQ**: *Vector Quantization*
- **WLD**: *Weber Local Descriptor*
- **XSS**: *Cross-Site Scripting*
- **ZCR**: *Zero-Crossing Rate*

Anexos

A

Objetivos específicos

En este anexo se detallan los objetivos de cada sección de la memoria, haciendo una breve descripción de cada uno de ellos y los entregables que se han generado al finalizar las actividades. Como se puede observar, las tablas que a continuación se presentan muestran los objetivos específicos de las secciones relacionadas con el trabajo descrito en esta memoria, a saber, la *Sección 3: Bases de datos*, la *Sección 4: Desarrollo y evaluación de un sistema de verificación facial*, la *Sección 5: Demo comercial y aplicación móvil FaceOnVox* y la *Sección 6: Verificación de locutor y fusión biométrica de cara y voz*.

Tanto los objetivos como los entregables definidos a continuación tienen un nombre y un código único para que estos sean inequívocos en el contexto de este Trabajo de Fin de Máster y, así, fácilmente identificables a lo largo de la memoria.

Sección 3	**Bases de datos**	
Objetivo 3.1	Creación de una base de datos de imágenes de caras para probar algoritmos de verificación facial	O-3.1
Descripción	Con vistas a comparar algoritmos de verificación facial, se desea diseñar y constituir una base de datos de imágenes de caras, con distintas sesiones para los diferentes usuarios. Esta base de datos se constituirá utilizando bases de datos públicas de caras anexionando también bases de datos privadas, con el fin de poder poner a prueba diferentes situaciones ambientales y de luminosidad que se deseen evaluar. Además, ésta se estructurará de forma que sea fácilmente escalable y de utilización sencilla para el *software* evaluador de algoritmos de verificación facial.	
Entregables	Base de datos de biometría facial (pública y privada).	E-3.1.1
	Estructuración de la base de datos facial para evaluación de algoritmos de verificación facial.	E-3.1.2
	Documentación de bases de datos de biometría facial.	E-3.1.3
Objetivo 3.2	Creación de una base de datos de voz para probar algoritmos de reconocimiento de locutor	O-3.2
Descripción	Con el objetivo de realizar pruebas de fusión biométrica a partir de las puntuaciones de los algoritmos de verificación facial y de voz, se decide diseñar y constituir una base de datos de locuciones de voz, la cual se formará utilizando bases de datos públicas de locuciones, anexionando también bases de datos privadas donde se puedan poner a prueba diferentes situaciones ambientales y de ruido, usando distintos dispositivos de captación.	
Entregables	Base de datos de biometría de voz (pública y privada).	E-3.2.1
	Estructuración de la base de datos de voz para evaluación.	E-3.2.2
	Estructuración de la base de datos facial y de voz para evaluación de algoritmos de fusión biométrica.	E-3.2.3
	Documentación de bases de datos de biometría de voz y de fusión biométrica cara y voz.	E-3.2.4

Tabla A.I: Tabla de objetivos específicos, Sección 3.

Sección 4	**Desarrollo y evaluación de un sistema de verificación facial**	
Objetivo 4.1	Desarrollar un sistema de evaluación de algoritmos de detección, modelado y verificación facial en C/C++	O-4.1
Descripción	Con el fin de evaluar diferentes tipos de algoritmos y escoger aquellos que mejores resultados ofrezcan es necesario diseñar un *software* que utilice una base de datos con imágenes de caras y evalúe diferentes algoritmos de detección, modelado y verificación. Este *software* se ha programado utilizando los lenguajes C y C++.	
Entregables	*Software* de evaluación de algoritmos de verificación facial en C/C++.	E-4.1.1
	Documentación del *software* de evaluación de algoritmos de verificación facial.	E-4.1.2
Objetivo 4.2	Investigar diferentes algoritmos de detección, modelado, verificación y fusión de características de cara, con vistas a obtener mejores resultados en la verificación	O-4.2
Descripción	Es necesario conocer el estado de la técnica inicial e ir mejorando paulatinamente diferentes parámetros en los algoritmos de verificación facial, tales como el tiempo de ejecución de los algoritmos, la capacidad de detección de caras en imágenes, las diferentes técnicas utilizadas para la generación de modelos faciales y los mecanismos de verificación.	
Entregables	Informe de evaluación de la técnica inicial.	E-4.2.1
	Informe de evaluación de nuevas técnicas de detección.	E-4.2.2
	Informe de evaluación de nuevas técnicas de modelado.	E-4.2.3
	Informe de evaluación de nuevas técnicas de comparación por distancias.	E-4.2.4
	Informe de evaluación de nuevas técnicas de fusión intra-rasgo facial.	E-4.2.5
Objetivo 4.3	Diseñar un sistema de *antispoofing* facial	O-4.3
Descripción	Con vistas a proteger la aplicación de verificación facial de ataques de suplantación de identidad frente al sensor, se dispone a investigar y desarrollar herramientas de *antispoofing* facial, que permitan discriminar si un sujeto frente a la cámara es real o en cambio es una imposición, como una fotografía, un vídeo o una máscara.	
Entregables	Estudio de sistemas de *antispoofing* facial comerciales y de investigación.	E-4.3.1
	Diseño de un sistema de *antispoofing* facial.	E-4.3.2

Tabla A.II: Tabla de objetivos específicos, Sección 4.

Sección 5	**Demo comercial y aplicación móvil FaceOnVox**	
Objetivo 5.1	Desarrollar un sistema todo local y cliente servidor de una demo de VerifyByFace	O-5.1
Descripción	Debido a la importancia comercial de la aplicación VerifyByFace, se desea implementar la tecnología en una demo comercial, con fines de *marketing* y negocios. El *software* de demostración se ha diseñado en dos arquitecturas: cliente servidor y local.	
Entregables	*Software* de demostración de VerifyByFace con arquitectura local.	E-5.1.1
	Software de demostración de VerifyByFace con arquitectura cliente servidor.	E-5.1.2
	Documentación del *software* de demostración de VerifyByFace.	E-5.1.3
Objetivo 5.2	Diseñar una API de verificación facial en C++, VerifyByFace	O-5.2
Descripción	De cara a distribuir licencias de la solución de verificación facial de Argos Soluciones Globales S.L., VerifyByFace, se desea a realizar el desarrollo de una API en C++ y que pueda ser integrable en aplicaciones Android y en otras aplicaciones que utilicen la librería en C++ de OpenCV.	
Entregables	Diseño de la API VerifyByFace en C++.	E-5.2.1
Objetivo 5.3	Diseñar una aplicación móvil, FaceOnVox	O-5.3
Descripción	Con vistas a ganar visibilidad de los servicios de FaceOn en el mercado, se decide diseñar y desarrollar una aplicación móvil para Android, FaceOnVox, que consiste en un bloqueador de aplicaciones (*AppLocker*) que funciona con una tecnología estable y testeada de VerifyByFace. Esta aplicación servirá para el equipo de desarrollo para mejorar los tiempos de procesamiento y precisión en la verificación en diferentes terminales móviles del mercado.	
Entregables	Diseño de la aplicación móvil FaceOnVox.	E-5.3.1

Tabla A.III: Tabla de objetivos específicos, Sección 5.

Sección 6	**Verificación de locutor y fusión biométrica de cara y voz**	
Objetivo 6.1	Desarrollar un sistema de evaluación de algoritmos de reconocimiento de locutor en C++	O-6.1
Descripción	De manera similar que con la biometría facial, se desea implementar un *software* de evaluación de algoritmos de reconocimiento de locutor. Debido al método de enrolamiento y verificación de la biometría de voz, que funciona mediante locuciones y no usando conjunto de imágenes, el diseño del *software* de evaluación de algoritmos varía con respecto al de biometría facial, pero se pretende diseñar de forma que las interfaces sean lo más similares posibles. Por motivos comerciales se decide adquirir externamente una API comercial para probar el *software* evaluador. Esta API se desea además integrarla en una demo para realizar pruebas de forma conjunta con la demo de VerifyByFace. El lenguaje de programación utilizado para el diseño y desarrollo de estos programas, el evaluador de algoritmos de verificación por voz y la demo de la API de reconocimiento de locutor, es C++.	
Entregables	*Software* de demostración de reconocimiento de locutor en C++.	E-6.1.1
	Software de evaluación de algoritmos de verificación por voz en C++.	E-6.1.2
	Documentación del *software* de demostración de biometría de reconocimiento de locutor.	E-6.1.3
	Documentación del *software* de evaluación de algoritmos de verificación de locutor.	E-6.1.4
Objetivo 6.2	Desarrollar un sistema de evaluación de algoritmos de fusión biométrica cara y voz a nivel de puntuación en C++	O-6.2
Descripción	Con el propósito de probar distintos algoritmos de fusión a nivel de puntuación o de *scores* resultado de las tecnologías de verificación de cara y de voz, se decide implementar una herramienta *software* que permita evaluar ambas tecnologías de forma independiente y además de manera conjunta utilizando diferentes estrategias de fusión de puntuación.	
Entregables	*Software* de evaluación de algoritmos de verificación facial y de locutor.	E-6.2.1
	Documentación del *software* de evaluación de algoritmos de verificación facial y de locutor.	E-6.2.2

Tabla A.IV: Tabla de objetivos específicos, Sección 6.

B

Competencias específicas adquiridas

En este anexo se presentan las competencias adquiridas durante el Trabajo de Fin de Máster. Debido a que el Trabajo de Fin de Master se presenta para dos titulaciones de máster: Máster Universitario en Ingeniería de Telecomunicación y Máster Universitario en Investigación e Innovación en Tecnologías de la Información y las Comunicaciones, las competencias que se exponen a continuación son diferentes para cada máster.

B.1. Competencias adquiridas del Máster Universitario Ingeniería de Telecomunicación

CE-1. Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la ingeniería de telecomunicación.

CE-2. Capacidad para la elaboración, planificación estratégica, dirección, coordinación y gestión técnica y económica de proyectos y recursos humanos en todos los ámbitos de la Ingeniería de Telecomunicación siguiendo criterios de calidad y medioambientales.

CE-3. Capacidad para la dirección general, dirección técnica y dirección de proyectos de investigación, desarrollo e innovación, en empresas y centros tecnológicos.

CE-4. Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.

CE-5. Que los estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades

CE-6. Capacidad para seleccionar la metodología adecuada para formular juicios a partir de información incompleta o limitada incluyendo, cuando sea preciso y pertinente, una reflexión sobre la responsabilidad social o ética ligada a la solución que se proponga en cada caso.

CE-7. Capacidad para transmitir de un modo claro y sin ambigüedades a un público especializado o no, resultados procedentes de la investigación científica y tecnológica o del ámbito de la innovación más avanzada, así como los fundamentos más relevantes sobre los que se sustentan. Capacidad para argumentar y justificar lógicamente dichas decisiones de un modo claro y sin ambigüedades, sin dejar de considerar puntos de vista alternativos o complementarios.

CE-8. Capacidad para trabajar en equipos o proyectos tecnológicos o de investigación en un contexto internacional y multidisciplinar.

B.2. Competencias adquiridas del Máster en Investigación en Innovación en Tecnologías de la Información y las Comunicaciones

CE-1. Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.

CE-2. Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.

CE-3. Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios.

CE-4. Que los estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades.

CE-5. Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

CE-6. Capacidad para identificar y analizar problemas, diseñar, implementar, y verificar soluciones que impliquen el uso de técnicas de computación avanzadas orientadas al desarrollo de aplicaciones, servicios y sistemas basados en sistemas distribuidos e inteligentes.

CE-7. Capacidad para aplicar los conocimientos adquiridos y resolver problemas en entornos nuevos, integrando tecnologías, aplicaciones servicios y sistemas propios de las tecnologías de la información y las comunicaciones en contextos más amplios y pluridisciplinares.

CE-8. Capacidad para tomar decisiones basadas en criterios objetivos (datos experimentales, científicos o de simulación disponibles). Capacidad para argumentar y justificar lógicamente dichas decisiones de un modo claro y sin ambigüedades, sin dejar de considerar puntos de vista alternativos o complementarios.

CE-9. Capacidad para actualizar conocimientos habilidades y destrezas de forma autónoma, realizando un análisis crítico, análisis y síntesis de ideas nuevas y complejas abarcando niveles más integradores y pluridisciplinares.

CE-10. Capacidad para seleccionar la metodología adecuada para formular juicios a partir de información incompleta o limitada incluyendo, cuando sea preciso y pertinente, una reflexión sobre la responsabilidad social o ética ligada a la solución que se proponga en cada caso.

CE-11. Capacidad para participar en proyectos de investigación y colaboraciones científicas o tecnológicas en un contexto internacional y multidisciplinar con una alta componente de transferencia del conocimiento.

CE-12. Capacidad para asumir la responsabilidad de su propio desarrollo profesional y de su especialización en el campo de las tecnologías de la información y las comunicaciones.

C

Justificación de méritos

En este anexo se justifican los méritos excepcionales derivados de este Trabajo de Fin de Máster con el objetivo de poder obtener la máxima calificación en las asignaturas Trabajo de Fin de Máster de ambos másteres cursados: Máster Universitario en Ingeniería de Telecomunicación y Máster Universitario en Investigación e Innovación en las Tecnologías de la Información y las Comunicaciones.

Javier Espinosa Mejías, con D.N.I. 1.185.428P, con cargo de CEO y representando en este documento a la empresa Argos Soluciones Globales S.L. certifica que Ángel Pérez Lemonche, ha estado trabajando en dicha empresa entre el 1 de septiembre de 2014 y el 16 de octubre de 2015, desempeñando el puesto de *Research & Development Engineer* (Ingeniero de Investigación y Desarrollo), en el que ha llevado a cabo diversos proyectos relacionados con la biometría facial y de voz. Este trabajo se corresponde con el Trabajo Final del Máster del Programa de Doble Máster en Ingeniería de Telecomunicación e Investigación e Innovación en las Tecnologías de la Información y las Comunicaciones de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid, presentado por el estudiante Ángel Pérez Lemonche con el título "Verificación facial en dispositivos móviles, y estrategias de fusión de biometría facial y de voz".

Durante este tiempo con nosotros, Ángel ha trabajado eficazmente y con un alto grado de calidad en el diseño y el desarrollo de los productos VerifyByFace (verificador facial), SmartFox (segmentador facial) y FaceOnVox (aplicación móvil de control de accesos), pertenecientes al porfolio de productos biométricos que ofrece Argos Soluciones Globales S.L. denominado FaceOn.

Este trabajo ha servido para dotar a la empresa Argos Soluciones Globales S.L. de tecnología utilizada para obtener un Producto Mínimo Viable previo a su paso a producción, de manera que permite a nuestra empresa establecer un punto de partida demostrable a la hora de entablar relaciones comerciales y la búsqueda de inversores. La actividad desarrollada por Ángel en la empresa ha favorecido sensiblemente la situación de mercado y tecnológica de ésta, impulsando nuevas relaciones con partners y clientes y mejorando la tecnología propietaria de Argos Soluciones Globales S.L. en el ámbito del reconocimiento facial y de voz.

En razón de la calidad tecnológica del trabajo realizado por el alumno y su alta aportación de valor a nuestra empresa, me permito proponer su candidatura a obtener la más alta calificación en su Trabajo Fin de Máster.

Sin otro particular, aprovecho para darles un cordial saludo.

ARGOS SOLUCIONES
GLOBALES S.L.
B84335558
C/ Gran Vía 74 3º Dcha, 28013
902 929 744

Fdo. Javier Espinosa Mejías