Reconocimiento facial en un sitema IoT

Alberto Rubio Pérez

Resumen—En este proyecto se desarrolla un sistema de reconocimiento facial con aprendizaje profundo en una NVIDIA Jetson Nano, la cual obtiene imágenes a través de una cámara y las procesa en tiempo real, para enviar una salida al identificar a un individuo. El programa ubica un rostro en el plano con una red basada en *Single-Shot Detector* (SSD) y utiliza *FaceNet* para transformarlo en un vector de características y determinar su similitud con las caras almacenadas. El sistema proporciona robustez frente a cambios en el aspecto de las personas y distorsiones en las imágenes, y una precisión igual o superior a 0.95 en los experimentos realizados con una base de datos creada durante el transcurso del proyecto. El funcionamiento está basado en *One-Shot Learning*, técnica que permite reconocer a cada persona con una única imagen. Los procedimientos utilizados se seleccionan mediante la previa investigación, desarrollo y comparación de distintos métodos. El trabajo se complementa con un sistema de reconocimiento por voz, basado en una red neuronal que predice una identidad a través de las características de un audio en tiempo real. Se emplea la iluminación de LEDs para el envío de salidas cuando se produce un reconocimiento.

Palabras clave— Aprendizaje profundo, Detección facial, *FaceNet*, Internet de las cosas, NVIDIA Jetson Nano, *One-Shot Learning*, Reconocimiento facial, Reconocimiento por voz, *Single-Shot Detector* (SSD).

Abstract—This project develops a deep-learning face recognition system on an NVIDIA Jetson Nano, which acquires images from a camera and processes them in real time to send an output when an individual is identified. The software locates a face in the plane with a *Single-Shot Detector* (SSD)-based network and uses *FaceNet* to transform it into a feature vector and determine its similarity to stored faces. The system provides robustness against changes in people's appearance and distortions in the images, and an accuracy equal to or better than 0.95 in the experiments carried out with a dataset created during the course of the project. The operation is based on One-Shot Learning, a technique that allows each person to be recognised with a single image. The procedures used are selected through prior research, development and comparison of different methods. The work is complemented by a speaker recognition system, based on a neural network that predicts an identity through the characteristics of audio in real time. LED illumination is used to send outputs when recognition occurs.

Index Terms—Deep Learning, Face detection, *FaceNet*, Internet of Things, NVIDIA Jetson Nano, *One-Shot Learning*, Face recognition, Speaker recognition, *Single-Shot Detector* (SSD).

1 Introducción

El reconocimiento facial está en primer plano de la revolución de la percepción algorítmica. Las aplicaciones de este sistema están cada vez más extendidas en el ámbito social, siendo utilizadas para desbloquear el teléfono móvil, vigilancia y seguridad en aeropuertos, como forma de pago y en investigaciones policiales, entre otros. Una de sus aplicaciones más recientes ha sido su incorporación en el aeropuerto Josep Tarradellas (Barcelona – El Prat), en el cual Aena (junto con Vueling) ha puesto en marcha este sistema, que sustituye todas las veces que se debe mostrar el DNI o la tarjeta de embarque por cámaras, agilizando el recorrido.

Otro concepto que ha revolucionado la vida cotidiana de las personas en los últimos años es *Internet of Things*, un planteamiento caracterizado por la interconexión digital de objetos inteligentes conectados a Internet, redes en expansión y grandes cantidades de datos. El éxito de estos dispositivos ha provocado su integración en las viviendas, seguridad y privacidad, vehículos, agricultura y hospitales, entre muchos otros ámbitos.

En este proyecto se pretende fusionar estas dos ideas innovadoras para crear un dispositivo reconocedor de caras, gestionado por un mecanismo que, después de validar la identidad de una persona almacenada en una base de datos, interconexiona con objetos inteligentes para realizar acciones como la apertura de una puerta automática.

Un posible emplazamiento del sistema podría ser en el Centro de Visión por Computador (CVC), situado en la Universitat Autònoma de Barcelona (UAB). Actualmente, los trabajadores del centro acceden al recinto tecleando un código personal para ser identificados. Con esta modificación, se mejoraría la comodidad de los empleados para la entrada, además de evitar contactos por el COVID-19.

Para complementar el experimento, se plantea crear un sistema de verificación única con otro rasgo descriptivo del individuo; la huella dactilar, las orejas o el iris del ojo son algunos de los más utilizados. En este caso, se utilizará la voz para identificar a los sujetos y añadir una funcionalidad más al proyecto.

- *E-mail de contacto: albertorp2000@hotmail.com*
- *Mención realizada: Computación*
- *Trabajo tutoritzado por: Coen Jacobus Antens (Centro de Visión por Computador)*
- *Curso 2021/22*

2 OBJETIVOS

El objetivo principal del proyecto consiste en la implementación de un algoritmo de reconocimiento facial y vocal en tiempo real, el cual identifique de forma inequívoca a la persona analizada para, posteriormente, enviar una salida.

El proyecto sigue la arquitectura *Device-Edge-Cloud* (Figura 1). El *Edge* es el dispositivo encargado de comunicarse con los *Devices*, de los cuales obtiene información u ordena realizar una determinada acción. En este caso, el *Edge* obtiene la imagen de la cámara y efectúa la detección, el procesado y el reconocimiento del rostro, para, después, enviar una salida a otro *Device*. El *Cloud* se utilizaría para almacenar las fotografías y hacer el procesamiento; en este caso, el *Edge* tiene capacidad de cómputo y almacenamiento para poder prescindir de una plataforma que actúe de *Cloud*. En el caso extremo de llenar la memoria, podría añadirse esta funcionalidad, aunque podría interferir visiblemente en la rapidez del sistema.

La cámara utilizada es una Logitech HD Pro C920, la cual graba en Full HD (1080p a 30 fps) y tiene dos micrófonos estéreo integrados. Se conecta al *Edge* mediante USB-A. A través de ella se obtendrán las imágenes y los audios.

El *Edge* es una NVIDIA Jetson Nano, una microcomputadora con un procesador de 4 núcleos a 1,4 GHz y una GPU con 128 núcleos para ejecutar modelos de inteligencia artificial, la cual permite ejecutar las cargas requeridas en el experimento.

Para posibilitar el envío de una salida se harán uso de dos LEDs, los cuales sirven para simular el uso de dispositivos de mayores dimensiones.

A continuación, se desglosan las fases del proyecto para detallar qué metas se pretenden conseguir.

2.1 Detección facial

Esta fase es el primer paso que debe implementarse para realizar el reconocimiento. Para el desarrollo de sistemas de identificación de rostros en imágenes digitales, las aplicaciones emplean algoritmos con aprendizaje automático. Para su desarrollo, se llevará a cabo una investigación de los diferentes métodos existentes hasta la fecha. De esta manera, se profundizará en diversas técnicas, obteniendo la información y los resultados necesarios para evaluar las ventajas y desventajas de cada una aplicadas al proyecto.

2.2 Reconocimiento facial

Alineación de rostros y extracción de características

La alineación es la técnica de visión por computador que identifica la estructura geométrica de los rostros en las imágenes digitales, e intenta obtener una alineación basada en la traslación, la escala y la rotación. Entre todos los rasgos faciales, la localización de los ojos es la más importante, a partir de la cual se identifican todos los demás.

La extracción de características es el proceso posterior a la alineación, en el cual, una vez localizados los rasgos faciales, se obtienen los puntos más relevantes y diferenciales del rostro. Estas propiedades son utilizadas para reconocer la identidad de la persona analizada.

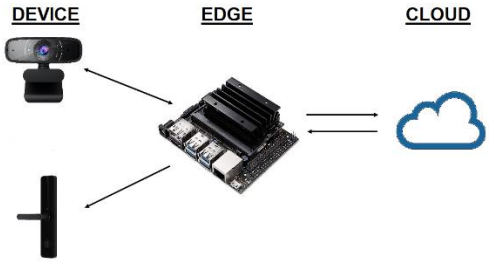

Figura 1. Arquitectura Device-Edge-Cloud

Reconocimiento facial

El reconocimiento facial es la tecnología capaz de identificar o verificar a una persona a través de una imagen. Para su identificación, el sujeto debe haber registrado previamente un perfil en la base de datos del sistema.

Hoy en día, los algoritmos de visión por computador más avanzados utilizan *Deep Learning*. La clasificación estándar requiere una enorme cantidad de datos para predecir con buena precisión, ya que el modelo tiene que ser entrenado con un gran número de imágenes etiquetadas y en un gran número de épocas. Este método puede no ser adecuado, porque cada vez que se modifica el *dataset* hay que volver a entrenar el modelo.

Para una primera toma de contacto, se profundizará en modelos estándares, de los más clásicos de reconocimiento facial. Posteriormente, se indagará en sistemas más modernos y se comparará el rendimiento de todas las técnicas utilizadas para estudiar qué tipo de reconocedor se adapta más a las necesidades del trabajo.

2.3 Reconocimiento por voz

Igual que en el facial, la verificación por voz es un problema de reconocimiento de patrones, ya que el propósito de ambos es extraer información que permita establecer propiedades para categorizar la identidad de la persona.

El objetivo es implementar un modelo capaz de aportar un criterio fiable para la verificación de los individuos. Su puesta en marcha sirve para tener más de una posibilidad de autenticación, complementando el reconocimiento facial, que es el principal fin del proyecto. Además, también cumple con la finalidad de evitar los contactos.

2.4 Internet de las cosas

Los aparatos inteligentes realizan sus funciones pertinentes una vez que el reconocimiento ha identificado al sujeto. Todos los instrumentos estarán comunicados con la NVIDIA Jetson Nano, que se encarga de dirigir las órdenes.

En el proyecto se empleará la iluminación de LEDs para verificar el reconocimiento de una persona, con el objetivo de facilitar la implementación.

3 ESTADO DEL ARTE

En esta sección, se reúnen las tecnologías y modelos más importantes hasta la fecha, las cuales sirven de referencia para el progreso del proyecto.

3.1 Detección facial

La detección facial ha progresado desde técnicas rudimentarias de visión por computador hasta avances en aprendizaje automático, pasando por redes neuronales artificiales cada vez más sofisticadas y tecnologías relacionadas; el resultado ha sido una mejora contínua del rendimiento. Los métodos clásicos más importantes han sido **Viola-Jones** [1] y **HOG** [2]. Aunque estas técnicas siguen siendo populares en aplicaciones en tiempo real, tienen limitaciones. Por ejemplo, si una cara está cubierta con una máscara o bufanda o no está orientada correctamente, es posible que los algoritmos no puedan encontrarla. Para mejorar la detección de rostros se han desarrollado otros algoritmos, como *Max-Margin Object Detection* (**MMOD CNN**) [3], *Multi-Task Cascaded* (**MTCNN**) [4] y *Single-Shot Detector* (**SSD**) [5]. Las tres opciones proporcionan resultados mejores que *Viola-Jones* y HOG, pero consumen más recursos por la complejidad de sus modelos, lo cual puede ser determinante para la fluidez del sistema.

3.2 Reconocimiento facial

Igual que en la detección, en el reconocimiento facial se ha producido una importante mejora del rendimiento de los modelos, gracias al uso de arquitecturas de *Deep Learning* y las *Convolutional Neural Networks* (CNN).

Eigenfaces [6], *Fisherfaces* [7], y *Local Binary Pattern Histogram* (LBPH) [8] son tres de los métodos más importantes, y, actualmente, están en desuso, a pesar de ser los más efectivos antes de que se implementasen técnicas más avanzadas. Los mejores modelos de reconocimiento facial de última generación son los siguientes:

- **DeepFace:** es un modelo de red neuronal profunda desarrollado por investigadores de Facebook. Sigue el flujo de detectar, alinear, representar y clasificar. Adopta un enfoque un poco más avanzado que otros, al añadir la transformación 3D y la transformación afín a trozos en el procedimiento. El algoritmo está capacitado para ofrecer resultados de identificación casi tan precisos como un humano medio [9].
- **FaceNet:** fue introducido por los investigadores de Google al integrar el aprendizaje automático en el reconocimiento facial. *FaceNet* entrena directamente el rostro, convirtiéndolo en un vector de características y utilizando el espacio euclidiano, donde la distancia consiste en las similitudes entre las caras [10].
- **OpenFace:** es un modelo de reconocimiento facial de aprendizaje profundo basado en el modelo *FaceNet*. Una ventaja significativa de *OpenFace* es que se ha desarrollado centrándose en el reconocimiento en tiempo real y puede funcionar sin problemas en dispositivos móviles. Por lo tanto, se puede entrenar un modelo con alta precisión con muy pocos datos [11].
- **VGGFace:** Consta de 38 capas y se ha entrenado con 2,6 millones de imágenes de más de 2.600 personas. Contiene trece capas convolucionales, cada una de las cuales tiene un conjunto de parámetros híbridos [12].

3.3 Reconocimiento por voz

Actualmente, no existe un método de referencia fijo y muy extendido en el reconocimiento por voz, a diferencia de la detección y el reconocimiento facial, pero las tecnologías basadas en redes neuronales están empezando a aparecer en los mejores sistemas publicados.

En los últimos años, los *i-vectors* han sido la técnica más avanzada. Un *i-vector* es una representación de baja dimensión, utilizado para describir una señal de audio a partir de la extracción de sus características idiosincrásicas [13].

En "Front-End Factor Analysis For Speaker Verification" [14] se presentan dos sistemas con *i-vectors*, uno basado en *Support Vector Machine*, que utiliza el núcleo del coseno para estimar la similitud entre los datos de entrada, y otro que utiliza directamente la similitud del coseno como puntuación final de la decisión.

4 Viabilidad técnica

En este apartado se mencionan las plataformas y tecnologías involucradas en el desarrollo el proyecto, junto a la argumentación de las selecciones.

4.1 Metodología

La metodología utilizada para el trabajo es Trello, un software de administración de proyectos con interfaz web basado en la nube. Ha sido la plataforma escogida gracias a su diseño intuitivo, con el cual se pueden organizar y modificar todas las fases del proyecto con sus respectivas fechas y necesidades, y clasificarlas según su estado.

El proyecto se orienta en un desarrollo *Agile*, el cual precisa de rapidez y flexibilidad y permite adaptar la forma del trabajo a las condiciones del proyecto. Con este enfoque, el diseño se divide en pequeñas partes y sigue un proceso iterativo, en el cual se entregarán *sprints* que irán evolucionando con el tiempo.

Para acordar las labores a realizar en cada *sprint* y valorar el trabajo hecho hasta el momento, se efectúan reuniones semanales mediante Microsoft Teams con el tutor.

4.2 Alternativas y selección de implementación

Las tareas principales para implementar el sistema de reconocimiento facial son la detección y el reconocimiento del rostro, que engloba el procesamiento de la imagen y la verificación de su identidad.

En la detección facial se realizarán experimentos con *Viola Jones* y HOG, con tal de probar su funcionamiento y encontrar los principales puntos a mejorar con el uso de otro más actual. También se han barajado las alternativas MMOD CNN, MTCNN y SSD. *Single-Shot Detector* (SSD) ha sido el algoritmo seleccionado, puesto que es el que mejor funciona con la oclusión, los movimientos rápidos de la cabeza y también identifica las caras laterales en ángulos que el resto no; además, también tiene el valor más alto de *frames* por segundo (fps). A pesar de las ventajas del método, las tres alternativas proporcionan buenos resultados, pero el consumo de recursos de SSD también es favorable.

El reconocimiento facial se implementará, primeramente, con los métodos más clásicos, que son *Eigenfaces*, *Fisherfaces* y LBPH. El uso de estos algoritmos permite localizar las ventajas y desventajas de cada uno y comparar su rendimiento con uno más actual. *FaceNet* ha sido el sistema seleccionado para desenvolver el reconocedor a

través de redes neuronales, pero no ha sido la única opción considerada. Primeramente, se barajó la utilización de *DeepFace*. Las desventajas principales de *DeepFace* son que requiere un *dataset* muy grande y el modelado 3D que emplea es muy complicado. Por otro lado, *FaceNet* tiene un modelo demasiado profundo, con un conjunto de datos demasiado grande y difícil de manejar; esta desventaja no supone un problema, puesto que no es necesario entrenar el modelo desde cero con la utilización de uno pre-entrenado. *DeepFace* fue la red neuronal que aportó mejor rendimiento en el año de su lanzamiento (2014), pero *FaceNet* surgió un año más tarde, proporcionando mejores resultados. *FaceNet* también presenta mejor rendimiento que *VGG-Face* y *OpenFace*, demostrado en un experimento de verificación facial con *Labeled Faces in the Wild* (LFW) [15].

El reconocimiento por voz se realizará con la creación de una red neuronal propia. Igual que los *i-vectors*, se extraerán los *Mel Frequency Cepstral Coefficients* (MFCCs) de los audios para obtener las características más descriptivas y reducir la dimensión de los archivos.

5 Planificación

En esta sección se presentan los estados en los que se divide el trabajo, con la finalidad de cumplir con todos los objetivos en los plazos de entrega correspondientes.

5.1 Fases y tareas

A continuación, se desglosan las fases y las tareas definidas para el desarrollo del proyecto.

Fase 1: Inicio y planificación del proyecto.
- **Tarea 1.1:** Consignar una propuesta detallada del TFG.
- **Tarea 1.2:** Definición de los objetivos.
- **Tarea 1.3:** Selección de metodologías.
- **Tarea 1.4:** Planificación de fases y fechas de entrega.

Fase 2: Implementación del reconocimiento facial.
- **Tarea 2.1:** Selección de *datasets* de imágenes.
- **Tarea 2.2:** Desarrollo con los métodos de detección y reconocimiento estándar.
- **Tarea 2.3:** Primeras pruebas con las bases de datos públicas e identificación de los puntos a mejorar.
- **Tarea 2.4:** Desarrollo de la detección y el reconocimiento facial con sistemas actuales.
- **Tarea 2.5:** Pruebas y comparaciones.

Fase 3: Implantación en NVIDIA Jetson Nano.
- **Tarea 3.1:** Configuración de la microcomputadora.
- **Tarea 3.2:** Adaptación del programa al nuevo entorno.
- **Tarea 3.3:** Primeras pruebas con cámara para la captura de imágenes en tiempo real.

Fase 4: Creación de un dataset propio.
- **Tarea 4.1:** Elaboración del programa para la grabación de contenido audiovisual.
- **Tarea 4.2:** Información e impresión de un documento para respetar la protección de datos.
- **Tarea 4.3:** Visita al CVC para realizar grabaciones.

Fase 5: Implementación de la identificación por voz.
- **Tarea 5.1:** Búsqueda de información y desarrollo del programa.
- **Tarea 5.2:** Entrenamiento del modelo y pruebas.
- **Tarea 5.3:** Implantación en la microcomputadora y adaptación del programa para predecir en tiempo real.

Fase 6: Experimentos y resultados.

Fase 7: Activación de una salida con la Jetson Nano.

Fase 8: Cierre del proyecto.
- **Tarea 8.1:** Elaboración del dosier.
- **Tarea 8.2:** Entrega de informe final.
- **Tarea 8.3:** Presentación.
- **Tarea 8.4:** Póster.

En el apéndice A1 se muestra el diagrama de Gantt de las fases en las que se particiona el proyecto, con sus respectivas dependencias y la duración de cada tarea.

5.2 Sprints

La entrega de los *sprints* (marcadas por las fechas pre-establecidas en la planificación del TFG) se han agregado al tablero de *Trello*. En el apéndice A2 se especifican las descripciones y los objetivos de cada *sprint* de forma detallada.

5.3 Seguimiento

El seguimiento de la planificación de los dos primeros *sprints* ha cumplido con las previsiones de fechas establecidas en todas las tareas asignadas. En la tercera entrega también se finalizaron todas las tareas, excepto la identificación por voz, la cual sí que se ha desarrollado a tiempo, pero no se ha podido adaptar para predecir en tiempo real antes del tercer *sprint*. Este inconveniente, sumado a nuevas ideas que surgieron para los experimentos del reconocimiento facial, retrasaron la Fase 7, que consiste en la activación de una salida al reconocer un sujeto. Al disponer de dos semanas más, se han podido completar todas las fases.

En la elaboración de los objetivos, se estableció la identificación por voz como un apartado a realizar adicionalmente para complementar el reconocimiento facial, en el caso de cumplir con unos plazos de entrega optimistas. Finalmente, se ha podido acabar el proyecto en su totalidad, tal y como se había diseñado en la planificación inicial.

6 Desarrollo

Para desarrollar el proyecto, se comienza investigando y desenvolviendo las técnicas más clásicas sobre el reconocimiento facial. De esta manera, se interiorizan los conocimientos básicos y se obtienen los primeros resultados, identificando así los puntos a mejorar en la construcción de modelos más complejos. Los experimentos se desarrollan en el lenguaje Python. Para entrenar y probar los programas implementados se utilizan dos *datasets*.

UTK Face [16] es un conjunto de datos de rostros a gran escala con un amplio rango de edad (entre 0 y 116 años). Consta de más de 20.000 imágenes faciales con anotaciones de edad, sexo y origen étnico. Las imágenes cubren una gran variación en cuanto a pose, expresión facial, iluminación, oclusión y resolución. Las caras fotografiadas de UTK Face son de personas anónimas, y el objetivo es hacer pruebas con la gran diversidad de opciones que se dispone, para encontrar los límites del detector de rostros.

Labeled Faces in the Wild (LFW) [17] es una base de datos de fotografías de celebridades, diseñada para estudiar el problema del reconocimiento facial sin restricciones. El

conjunto de datos contiene más de 13.000 imágenes de caras recogidas de la web y cada rostro está etiquetado con el nombre de la persona. LFW abarca cuatro grupos diferentes, y se ha escogido LFW-a, en el cual se aplica un método de alineación a las imágenes.

6.1 Detección facial

La detección facial ha sido la primera tarea investigada e implementada del proyecto, a través de los métodos *Viola-Jones*, *Histogram of Oriented Gradients* (HOG) y *Single-Shot-Detector* (SSD).

Viola-Jones y HOG

La principal ventaja que se ha experimentado con *Viola-Jones* ha sido su extremada rapidez para realizar la detección y su invariablidad a escala y localización. Por contra, se ha mostrado poco eficiente en imágenes en las que el rostro no era frontal, sensible a cambios de iluminación, propenso a los falsos positivos y, en ocasiones, obtenía varias detecciones de un mismo individuo.

Por otra parte, las pruebas efectuadas con HOG han proporcionado más precisión que *Viola-Jones*, el cual detectaba, frecuentemente, rostros donde no los había; también se ha mostrado más robusto a cambios de iluminación. Sin embargo, este método es computacionalmente más lento, lo cual supone un inconveniente en el proyecto, ya que la rapidez del algoritmo es uno de los asuntos más cruciales.

Single-Shot Detector

Tras probar las anteriores técnicas e investigar varias alternativas, se ha decidido utilizar una *Deep Neural Network* para construir la detección facial del proyecto. La red se utiliza a través de *OpenCV*, que incluye un módulo DNN que permite cargar redes neuronales pre-entrenadas; esto mejora increíblemente la velocidad, reduce la necesidad de dependencias y la mayoría de los modelos tienen un tamaño muy ligero. El marco de aprendizaje empleado ha sido *Tensorflow*. La red neuronal utilizada se basa en *Single-Shot Detector* (SSD) con una red base *ResNet* [18].

Single-Shot Detector (SSD) es un algoritmo que detecta el objeto (rostro) en una sola pasada sobre la imagen de entrada, a diferencia de otros modelos que la recorren más de una vez. SSD se basa en el uso de redes convolucionales que producen múltiples *bounding box* y puntúan la presencia del objeto en esas cajas, seguido de un paso de supresión para producir las detecciones finales.

En la implementación del código, se lee el modelo preentrenado a través de dos ficheros y las imágenes de prueba se introducen en la red, la cual retorna las detecciones encontradas. Para una mayor precisión, se recorren las detecciones y se compara su valor de confianza asociado con un *threshold*, para así eliminar las menos fiables. Los resultados obtenidos han sido muy buenos, puesto que se consigue un algoritmo que mejora mucho las técnicas probadas, mostrándose preciso en las detecciones de rostros con diferentes ángulos de visión y condiciones de iluminación y oclusión (Figura 2).

6.2 Reconocimiento facial

El reconocimiento facial se comienza a tratar a través de las

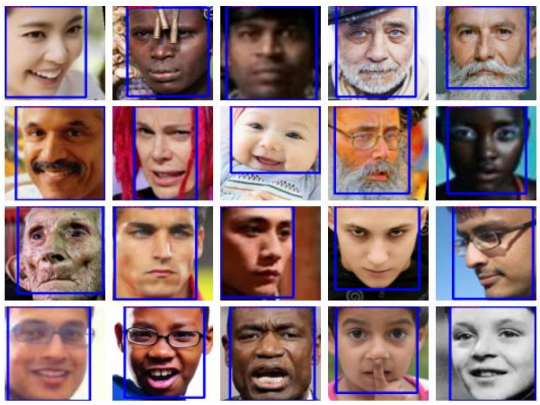

Figura 2. Ejemplos de detecciones de imágenes de UTK Face mediante SSD.

técnicas *Eigenfaces*, *Fisherfaces* y LBPH. Después de obtener y comparar sus resultados, se ha utilizado un modelo de *FaceNet*, método que emplea una red neuronal.

Eigenfaces, Fisherfaces y LBPH

En los experimentos realizados con estos métodos se ha utilizado la librería *OpenCV*. Se ha creado un modelo para cada uno de los tres procedimientos, a partir de rostros guardados previamente. Para la cara detectada, se genera una predicción del individuo que considera más parecido del conjunto de entrenamiento, asociada a un factor de confianza. Este factor es distinto para cada procedimiento, así que se ha utilizado un *threshold* para catalogar una predicción como desconocida en el caso de ser sobrepasado, siendo 4500 en *Eigenfaces*, 450 en *Fisherfaces* y 70 en LBPH.

El método con mejores métricas de rendimiento es LBPH, aunque *Fisherfaces* proporciona una ejecución más rápida [19]. Aún así, las tres opciones, coinciden en errores frecuentes de reconocimiento en imágenes con un entorno e iluminación desigual a las del conjunto de entrenamiento. Además, son necesarias numerosas muestras de cada persona para entrenar los patrones, de manera que ocupa demasiada memoria y el modelo tarda demasiado en crearse. Por lo tanto, estas técnicas no proporcionan la fiabilidad y rapidez necesarias para el trabajo.

FaceNet

El reconocimiento facial del proyecto se hace mediante *FaceNet*, un sistema que usa una *Deep Convolutional Network*. La red está entrenada a través de una función de *triplet loss*, que fomenta que los vectores de la misma persona se vuelvan más similares (menor distancia) y los de diferentes menos similares (mayor distancia). Su utilización en el proyecto se desglosa en dos apartados diferenciados.

Alineación y extracción de características

La alineación es el primer proceso que se aplica a las imágenes entradas para realizar el reconocimiento. *FaceNet* contiene una capa de normalización en la que se detectan los puntos de referencia faciales más descriptivos del rostro, para después extraer las características.

La siguiente capa toma la imagen de la cara normalizada y retorna un vector de 128 números de valor real, que representan las propiedades más importantes del rostro. El vector se denomina *embedding*, y su creación permite la transformación de los datos de alta dimensión (imagen) en

Figura 3. Triplet loss minimiza la distancia entre el ancla y el ejemplo positivo y la maximiza entre el ancla y el negativo.

datos de baja dimensión (números), lo cual posibilita trabajar de forma más sencilla y óptima. Las imágenes de rostros de la misma persona tienen *embeddings* similares, y al contrario para individuos diferentes.

Clasificación del rostro

El método que se usa para llevar a cabo la verificación de identidad es *One-Shot Learning* [20], un problema de categorización en visión por computador, que consiste en el aprendizaje de información sobre los objetos a partir de una imagen y sin necesidad de volver a entrenar el modelo.

Previamente, la red ha sido preparada con millones de imágenes mediante *triplet loss*. Esta técnica genera valores aleatorios para cada una, y todas se sitúan al azar en un espacio X-dimensional. El entrenamiento consiste en la selección aleatoria de una imagen ancla, una de la misma persona al anclaje (ejemplo positivo) y una de un individuo diferente (ejemplo negativo); la red ajusta los parámetros y acerca en el espacio a la imagen positiva, mientras aleja la negativa. Este proceso se repite iterativamente hasta que todas las muestras parecidas están agrupadas en espacios cercanos, y alejadas de las que más difieren en sus características (Figura 3).

La implementación de la técnica se hace mediante un modelo pre-entrenado de *FaceNet*. En la ejecución del programa, primeramente, se detectan los rostros con SSD y se calculan las representaciones vectoriales de las imágenes del *dataset*. Para una imagen entrada, se efectúa la detección a la cara y el cálculo de su *embedding*, junto a las distancias a las imágenes de personas conocidas. El cómputo de la distancia entre las imágenes está ligado a *Siamese Neural Network* [20], un sistema que consta de dos redes idénticas, las cuales crean representaciones vectoriales de las entradas para, después, calcular la diferencia entre ambas salidas y determinar si son el mismo objeto. En este caso, todas las imágenes se han procesado en la misma red, y se calcula la diferencia del *embedding* del rostro desconocido con todos los almacenados mediante un bucle. Si la diferencia entre el desconocido y uno conocido es menor que un *threshold* predefinido, se afirma que el rostro detectado y el almacenado son el mismo sujeto (Figura 4).

Los resultados obtenidos han sido muy positivos. El sistema reconoce correctamente con una sola imagen de entrenamiento en la base de datos, sin mostrar dificultades en representaciones con distintas condiciones lumínicas (Figura 5). El programa funciona con mucha más rapidez que los probados anteriormente, puesto que se emplea un modelo pre-entrenado; además, no es necesario una gran cantidad de muestras para cada persona. Por lo tanto, se han solventado las principales carencias encontradas en los métodos previos, los cuales mostraban lentitud para generar el modelo y fallaban al haber cambios en la escena.

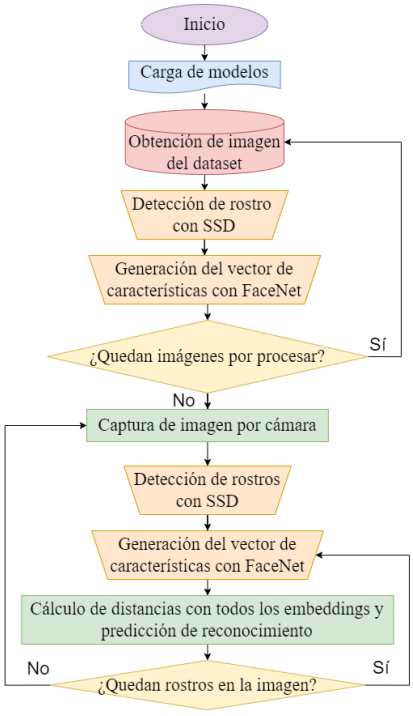

Figura 4. Diagrama de flujo del sistema.

6.3 Creación del dataset

Después de comprobar el comportamiento de los algoritmos con bases de datos públicas, se ha procedido a la creación de un *dataset* propio, a fin de orientar el proyecto al funcionamiento en el mundo real. Se ha establecido un día para llevar el sistema al CVC y obtener registros con cámara y micrófono de los trabajadores que se han prestado a colaborar.

Previamente, se ha elaborado un programa para obtener las muestras. El archivo usa la librería *Cv2* para registrar los vídeos y la librería *PyAudio* para las voces. La ejecución se realiza mediante el uso de dos hilos paralelos para guardar ambas en una misma realización.

Finalmente, se obtuvieron datos de 25 personas diferentes. A cada una de ellas, se le grabó una primera vez con mascarilla y una segunda sin mascarilla, mientras recitaban una frase. De esta manera, es posible testear el programa de reconocimiento facial desde diferentes perspectivas, para comprobar la robustez del sistema frente a un cambio de imagen muy habitual en época COVID-19.

Según la normativa, la autorización para el uso de imágenes es necesaria para todas aquellas representaciones gráficas que tengan derechos de autor, a partir del consentimiento informado de la persona grabada. Por lo tanto, para cumplir la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD) y el Reglamento General de Protección de Datos (RGPD), es necesaria la firma de un documento en el que las personas dan consentimiento para la grabación de dichas imágenes.

6.4 Reconocimiento por voz

El reconocimiento por voz se implementa con *Librosa*, un paquete de Python para análisis de audio y música que proporciona los componentes básicos necesarios para crear

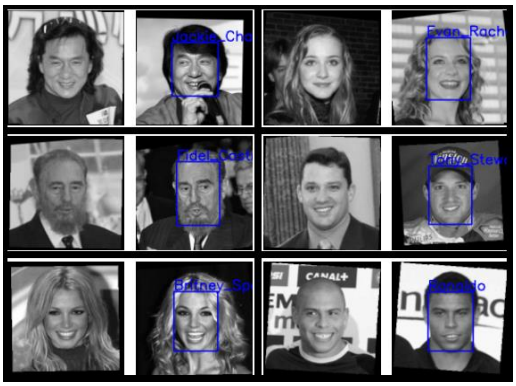

Figura 5. Resultados del reconocimiento facial con imágenes de lfw2-a. La imagen izquierda de cada pareja es el entrenamiento y la derecha es la de test.

sistemas de recuperación de información auditiva. Las características que se extraen de cada audio son:

- MFCCs: coeficientes para la representación del habla. Extraen características de las componentes de una señal de audio que son adecuadas para la identificación de contenido relevante y obvian las que posean información que entorpezca el proceso de reconocimiento.
- Cromagrama: herramienta que permite categorizar el tono de las voces. Hay doce clases de tono diferentes.
- Espectrograma mel: el espectrograma es el resultado del cálculo de varios espectros en segmentos de ventana superpuestos de la señal, a través de la *Short-time Fourier Transform (stft)*. Las frecuencias de este espectrograma se convierten a la escala mel, una escala musical perceptual de tonos.
- Contraste espectral: es la energía media en el cuantil superior con la del cuantil inferior de cada subbanda en la que está dividida el espectrograma.
- Centroide tonal: es una representación que proyecta características cromáticas en 6 dimensiones.

Las características extraídas se procesan con *Sklearn* y serán la entrada de la red neuronal creada con *Keras*. Se usa la función de activación *SoftMax* para categorizar los audios. El modelo se compila con entropía cruzada categórica como función de pérdida y el algoritmo de descenso del gradiente *adam* como optimizador.

Para la obtención de audio en tiempo real se usa la librería *PyAudio*, con la cual se escuchan fragmentos de 32.000 fps y se realiza una predicción cada siete segundos.

6.5 Salida con la NVIDIA Jetson Nano

La microcomputadora empleada permite procesar todo el programa y enviar una salida cuando se efectúa un reconocimiento. La principal funcionalidad aplicable al proyecto es la apertura de una puerta automática, pero se utilizará el iluminado de dos LEDs para simular el comportamiento. Para su puesta en marcha, se ha utilizado una *protoboard*, dos resistencias de 220 *Ohm* y dos LEDs (verde y rojo), unidos a la NVIDIA Jetson Nano mediante cables conectados a sus pines (Figura 6).

Para el iluminado se ha utilizado la librería *Jetson.GPIO*, la cual proporciona todas las funciones necesarias. Los LEDs se inicializan al ejecutar el programa, y se ilumina el rojo mientras no se produzcan reconocimientos y el verde en caso de identificar a un individuo, que simula el envío de la señal a la puerta automática.

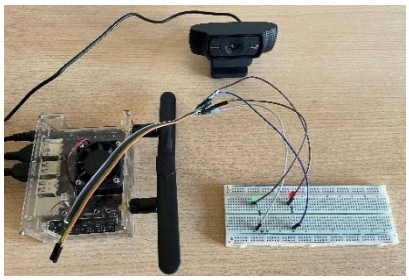

Figura 6. Conexiones hardware con la NVIDIA Jetson Nano.

7 Experimentos y resultados

En esta sección, se explican y se interpretan los resultados obtenidos al aplicar las funcionalidades de reconocimiento argumentadas en el desarrollo.

7.1 Reconocimiento facial

Debido a que *One-Shot Learning* es el método utilizado para el reconocimiento, solo es necesaria una captura de cada persona para que pueda ser reconocida. En la base de datos se utilizarán imágenes con la cara visible y con mascarilla. En este caso, como los participantes han sido trabajadores del CVC, se utilizan las fotografías de sus perfiles de la página web. Para guardar imágenes con mascarilla, se ha creado un programa para capturar caras de los vídeos grabados, a través del proceso de detección (para elegir qué frame del vídeo es mejor, se guarda la detección que proporciona el valor de confianza más alto). En la Figura 7 se muestran 4 ejemplos guardados en la base de datos.

Accuracy

Para valorar el rendimiento del reconocedor, se ha testeado de tres maneras diferentes con el material adquirido de los trabajadores del CVC. En primer lugar, se ejecutará, solamente, con imágenes de entrenamiento en la que los participantes lleven puesta la mascarilla. Después, se hará, únicamente, con imágenes con la cara al descubierto. Finalmente, se ejecutará con una imagen con mascarilla y otra sin de cada persona. En las tres pruebas se testea el programa con todos los vídeos, es decir, 25 con y 25 sin mascarilla; para tener ejemplos negativos a la ejecución, 5 de los trabajadores (10 vídeos) no se incluyen en el *dataset*, y el modelo no debería identificarlos.

En el primer experimento, en el cual se usan imágenes con mascarilla, el modelo reconoce 20 caras, identificando correctamente todos los vídeos con mascarilla y catalogando como desconocido a las personas con el rostro al descubierto, cuando realmente son las mismas; también hay un falso positivo, que es un vídeo de un anónimo con mascarilla (*Accuracy*=0.58). En el segundo caso, en el que solo se utilizan imágenes con la cara visible, se repiten los resultados anteriores, pero a la inversa; el programa reconoce a todos, pero los desconoce si llevan la máscara puesta, con la diferencia de que no hay ningún falso positivo (*Accuracy*=0.6). Por último, con una imagen de cada en la base de datos, el programa funciona casi a la perfección, identificando y desconociendo ciertamente en casi todos los vídeos; el único fallo que comete es el reconocimiento erróneo de una persona desconocida, la misma que en el

Figura 7. Ejemplares de dos individuos del dataset

primer experimento (*Accuracy*=0.98).

Cambio de imagen

El cambio de imagen es una situación habitual en las personas, ya sea, por ejemplo, con diferentes cortes o colores de pelo, la presencia de vello facial o la utilización de accesorios. Debido a la frecuencia en que suceden estas variaciones, el sistema debe mostrarse capaz de reconocerlas frente a ciertas alteraciones físicas. Para probar el programa, se han utilizado dos imágenes sin mascarilla para aplicar tres modificaciones en distintas zonas de la cara:

- Cabello: inserción de un gorro y coloración del pelo, uno en cada caso.
- Ojos: inserción de gafas, en ambos casos, unas tintadas y otras transparentes.
- Barba: adición de vello facial, en ambos casos.

Se han establecido tres niveles de dificultad para comprobar los límites del algoritmo. En cada nivel se incrementa la diferencia entre la imagen almacenada y la de test, a partir de la fusión entre las modificaciones realizadas. Primeramente, se testea el programa con un solo cambio en la imagen, después con dos de ellos y, finalmente, con los tres a la vez.

En la Figura 8a se muestran los resultados con una única modificación en los rostros. El algoritmo reconoce correctamente a los dos individuos, en los seis casos.

Seguidamente, se realiza el mismo experimento con dos cambios en cada imagen (Figura 8b). En este caso, el algoritmo falla en un reconocimiento y acierta en los otros cinco. El falso positivo se atribuye a un sujeto de la base de datos que, igual que en la imagen de test errada, luce lentes transparentes y barba. Si se elimina esa imagen del *dataset*, el sistema sí que reconoce ciertamente a la persona. Por lo tanto, significa que su vector de características es lo suficientemente parecido al de la imagen original como para relacionarlo a la misma persona, pero hay otro en el *dataset* que es aún más cercano y provoca el error.

Finalmente, se prueba con la unión de las tres ediciones en una misma imagen y se muestran los resultados en la Figura 8c. El sistema no reconoce a ninguna de las dos personas. En el primer caso, no se relaciona la entrada con ningún individuo, debido a los grandes cambios integrados en la imagen original. En la otra representación, vuelve a obtenerse un falso positivo, arrastrado del nivel anterior.

Ruido y compresión

El ruido puede entenderse como una distorsión visual identificable, como un efecto de granulado o decoloración que suele reducir el impacto de una imagen, oscurece los detalles y, cuando se presenta en niveles altos, puede arruinar por completo una fotografía. Estas alteraciones podrían influir en el reconocimiento facial, puesto que una

Figura 8a: Resultados con un cambio en las imágenes.

Figura 8b. Resultados con dos cambios en las imágenes.

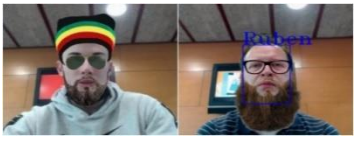

Figura 8c. Resultados con tres cambios en las imágenes.

imagen con ruido puede confundir al reconocedor, si la imagen de entrenamiento presenta más nitidez. Por lo tanto, introducir ruido es una buena opción para comprobar la robustez del sistema. Para ello, se han seleccionado los métodos de ruido Gaussiano, *Localvar*, *Poisson*, *Salt*, *Pepper*, *Salt & Pepper* y *Speckle*, los cuales distorsionan la imagen de forma distinta y se muestran, respectivamente, en la Figura 9. El sistema identifica correctamente a la misma persona en los siete casos. La mayor distancia entre la imagen de entrenamiento y las muestras con ruido ha sido 0.69, la cual pertenece a la alteración con *Salt & Pepper* y es menor al *threshold* predefinido (0.8).

Por otra parte, la compresión de imagen es la reducción de los datos redundantes e irrelevantes con la menor pérdida posible, para permitir su almacenamiento o transmisión de forma eficiente. Mediante la compresión de las muestras se puede comprobar los límites del algoritmo, lo cual puede ser útil en el caso extremo de llenar el almacenamiento de la microcomputadora y requerir de más espacio para introducir más personas en la base de datos. Las imágenes originales (formato JPG) miden 400x300 píxeles y, para realizar la prueba, se ha comprimido en un 50%, 75%, 84% y 87.5% respecto a la principal. En la Figura 10 se aprecia la correcta identificación del sujeto en los tres primeros casos y la errónea catalogación como desconocido en el último, debido a la considerable pérdida que ha sufrido la imagen. Por lo tanto, las imágenes podrían comprimirse, en caso de ser necesario, sin perder la seguridad de que el sistema siga funcionando correctamente.

Conclusiones

Con los resultados de *accuracy* obtenidos, se puede afirmar que la mejor opción es usar dos capturas por persona. Por otra parte, el falso positivo indica que el sistema no es 100%

Figura 9. Reconocimiento facial con ruido en las imágenes.

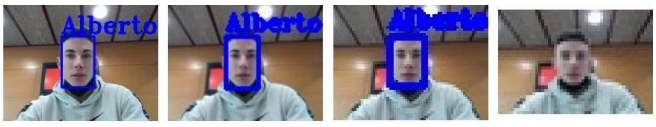

Figura 10. Reconocimiento facial con distintos niveles de compresión.

seguro y puede equivocarse. El fallo se debe a que la mascarilla tapa gran parte de la cara, y el modelo puede asociar una cara desconocida a una conocida (ambas con mascarilla) debido a las pocas características faciales que extrae. Esta carencia puede mejorarse disminuyendo el umbral de verificación, el cual indica la diferencia máxima entre dos vectores de características para determinar un reconocimiento, pero también puede repercutir negativamente en los verdaderos positivos. Para obtener un sistema de seguridad (a priori) sin fallos, la mejor opción es utilizar, solamente, imágenes sin mascarilla, ya que el programa conoce la cara completa de todos y no tiene opción a relacionar dos rostros como iguales por el simple hecho de llevarla tapada; esto se ha evidenciado en el segundo experimento, mostrando un 100% de precisión en los positivos. Además, los buenos resultados reafirman la flexibilidad de su implantación en un entorno real, puesto que solo es necesaria la inclusión de una fotografía en la base de datos para permitir la verificación de identidad de un nuevo trabajador.

En la Figura 11 se muestra la visualización de dos gráficas generadas con T-SNE, un método estadístico para visualizar datos de alta dimensión. En este caso, se exponen las ubicaciones de los vectores de 128 valores reales de los rostros en un espacio de 2D; en ambas se han usado numerosas muestras de cada sujeto. En la Figura 11a se sitúan los *embeddings* de 650 imágenes, correspondientes a 25 personas sin mascarilla, en la cual se observan 25 agrupaciones diferenciadas debido a la similitud de sus características. Después, se añaden otras 650 imágenes de las mismas personas con mascarilla y se muestran en la Figura 11b, visualizando 50 agrupamientos. Ambas gráficas evidencian el comportamiento del sistema, puesto que todos los vectores de la misma persona se juntan entre ellas, pero, si se utilizan fotografías con mascarilla, sus características son tan distintas que se consideran personas diferentes y se agrupan en otro *clúster*, por lo que utilizar dos fotos por persona no es una opción viable.

Actualmente, se han desarrollado experimentos que eliminan la mascarilla en imágenes, los cuales pueden utilizarse para identificar al sujeto con la cara tapada, sin necesidad de tener almacenada una imagen de entrenamiento con mascarilla; "A Novel GAN-Based Network for Unmasking of Masked Face" [21] es uno de ellos.

Por otra parte, se ha comprobado que el sistema es robusto a cambios en el rostro, siempre que la alteración de

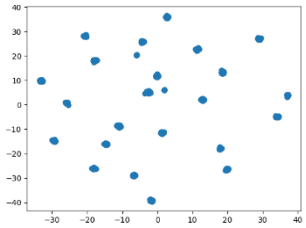

Figura 11a. T-SNE de 25 personas sin mascarilla.

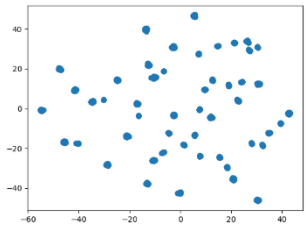

Figura 11b. T-SNE de 25 personas con y sin mascarilla

la imagen no sea muy desproporcionada. La observación más importante que se ha obtenido con el experimento ha sido el falso positivo, provocado por la confusión con un sujeto del *dataset* con características similares. El error se produce a causa de las gafas y la barba, las cuales están presentes en la imagen almacenada del otro individuo, y no en la del testeado. Al guardar una representación de una persona con lentes y vello facial, el algoritmo asocia su rostro a esas dos características, lo cual provoca que tenga vectores más semejantes con todo aquel que cumpla con las mismas particularidades, a pesar de que sus caras difieran significativamente. Por lo tanto, para garantizar un extra de fiabilidad, es conveniente que, al guardar la imagen de un nuevo individuo, este lo haga sin accesorios en el rostro, con tal de mostrar la cara más visible. El hecho de lucir gafas diariamente no supondría un problema, puesto que el algoritmo es capaz de identificar correctamente a personas con cambios en la imagen.

Finalmente, la comprobación de los reconocimientos en imágenes con ruido garantiza un plus de seguridad en el sistema para el uso diario, ya que no ha mostrado dificultades para identificar a las personas en capturas con ruido, el cual podría experimentarse puntualmente por factores externos como, por ejemplo, la luminosidad.

7.2 Reconocimiento por voz

Los resultados del reconocimiento por voz se han obtenido con el uso de las grabaciones de los trabajadores del CVC. Como se dispone de dos grabaciones de cinco segundos por persona, se ha utilizado una para el entrenamiento y otra para la validación.

La red se ha probado con diferentes capas en la red neuronal, para comprobar cual aporta mejores predicciones. La estructura que mejor ha funcionado ha sido con tres capas ocultas. La primera capa es de 193 nodos, y la segunda y tercera de 150, con tasas de abandono incrementales (0.1, 0.25 y 0.5, respectivamente). Se ha entrenado utilizando *Early Stopping*, para evitar *overfitting*. En la Figura 12 se muestra una gráfica del *accuracy* obtenido en el entrenamiento y en la validación durante las 100 épocas de ejecución del programa.

Los resultados, a parte de no ser positivos, son muy

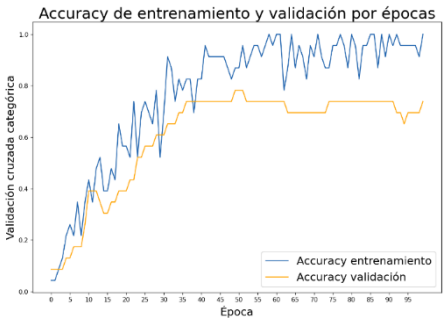

Figura 12. Accuracy entrenamiento y validación en el reconocimiento por voz.

inestables, lo que provoca la obtención de predicciones diferentes cada vez que se ejecuta el modelo. La inconsistencia se debe a que se dispone de muy pocos datos para entrenar una red neuronal clasificadora.

Conclusiones

El reconocimiento por voz no ha aportado resultados del todo satisfactorios, ya que la fiabilidad del algoritmo no es la suficiente como para implantarla en un sistema de seguridad. Estos resultados están condicionados a la corta duración y la baja calidad del material auditivo, debido a que ninguno supera los cinco segundos y algunos contienen ruido. Por otra parte, el uso de audios muy largos retrasaría la construcción de la red, puesto que el proceso de extracción de características sería más costoso. Si se dispusiese de grabaciones más extendidas, lo más adecuado sería la partición de estas en fragmentos, a los cuales se les extraería las características y cada uno sería una muestra de entrenamiento de la red.

Para este experimento, habría sido más adecuado utilizar otro método, como, por ejemplo, un algoritmo *KNN* o *Support-Vector Machines*, dado que no se dispone de los datos suficientes para entrenar una red neuronal. Un trabajo futuro de este proyecto podría ser la construcción del modelo de red neuronal con una base de datos más amplia y su comparación con los resultados de este informe.

Todo el código desarrollado en el proyecto se encuentra en el repositorio de GitHub "*https://github.com/rubio21/Face-Recognition-with-FaceNet-on-NVIDIA-Jetson-Nano*"

AGRADECIMIENTOS

En primer lugar, quiero agradecer a mi tutor Coen Jacobus Antens, quien, con sus conocimientos y apoyo, me guió a través de cada una de las etapas de este proyecto para alcanzar los resultados que buscaba.

También quiero agradecer al Centro de Visión por Computador y a sus trabajadores por brindarme todos los recursos y herramientas que fueron necesarios para llevar a cabo el proceso de investigación.

Por último, quiero agradecer a mi familia, pareja y amigos, por apoyarme a lo largo de toda mi carrera y, en especial, durante el transcurso del proyecto, dándome fuerzas y motivación cuando la presión aumentaba.

A todos ellos, mil gracias.

BIBLIOGRAFÍA

[1] P. Viola y M. Jones, "Rapid object detection using a boosted cascade of simple features," *Procede de 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.*

[2] N. Dalal y B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.

[3] Davis E. King, "Max-margin object detection," *ArXiv, vol. abs/1502.00046*, 2015.

[4] K. Zhang, Z. Zhang, Z. Li y Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, 2016.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C.Y. Fu y A.C. Berg, "SSD: Single Shot Multibox Detector", Procede de *arXiv/1512.02325*, 2015.

[6] M. A. Turk y A. P. Pentland, "Face recognition using eigenfaces," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991.

[7] P. N. Belhumeur, J. P. Hespanha y D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[8] M. S. Karis, N. R. A. Razif, N. M. Ali, M. A. Rosli, M. S. M. Aras y M. M. Ghazaly, "Local Binary Pattern (LBP) with application to variant object detection: A survey and method," *IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA)*, 2016.

[9] Y. Taigman, M. Yang, M. Ranzato y L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[10] F. Schroff, D. Kalenichenko y J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] T. Baltrušaitis, P. Robinson y L. Morency, "OpenFace: An open-source facial behavior analysis toolkit," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016

[12] O. M. Parkhi, A. Vedaldi y A. Zisserman, "Deep Face Recognition", *British Machine Vision Conference*, 2015.

[13] N. Dehak y S. Shum, "Low-dimensional speech representation based on Factor Analysis and its applications", *Spoken Language System Group MIT Computer Science and Artificial Intelligence Laboratory*, 2011.

[14] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel y P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[15] "Face Verification on Labeled Faces in the Wild", *Papers with code*

[16] Z. Zhang, Y. Song and H. Qi, "Age Progression/Regression by Conditional Adversarial Autoencoder ", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] Gary B. Huang, Marwan Mattar, Honglak Lee y Erik Learned-Miller, "*Learning to Align from Scratch*", *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[18] K. He, X. Zhang, S. Ren y J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] M.Ahsan, Y. Li, J. Zhang, T. Ahad y K. Gupta, "Evaluating the Performance of Eigenface, Fisherface, and Local Binary Pattern Histogram-Based Facial Recognition Methods under Various Weather Conditions", *Technologies. 9. 10.3390*, 2021.

[20] G.R. Koch "Siamese Neural Networks for One-Shot Image Recognition," 2015.

[21] N. Ud Din, K. Javed, S. Bae y J. Yi, "A Novel GAN-Based Network for Unmasking of Masked Face," *IEEE Access*, 2020.

APÉNDICE

A1. DIAGRAMA DE GANTT

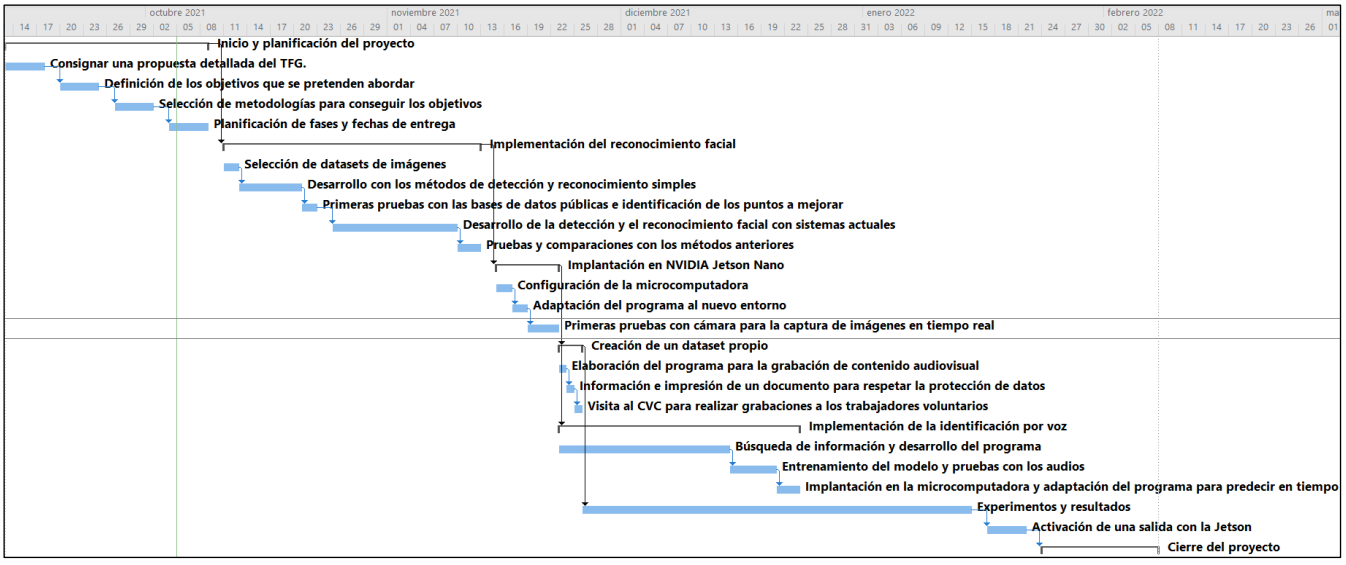

Figura 13. Diagrama de Gantt

A2. TABLA DE LAS ENTREGAS DEL PROYECTO

Entregas	Descripción	Fecha
Informe inicial	En este *sprint* se abordarán todas las tareas de la Fase 1. El informe debe incluir: o Información preliminar sobre el problema que se pretende resolver. o Una propuesta del objetivo del trabajo y hasta dónde se quiere llegar en el desarrollo. o Explicación general de la metodología que se va a seguir para conseguir los objetivos propuestos. o Planificación de los pasos a seguir para el desarrollo del proyecto propuesto. o Bibliografía.	10/10/2021
Informe de progreso I	A la segunda entrega se le asigna la elaboración de la Fase 2 al completo, la cual incluye las tareas de búsqueda de *datasets*, elaboración de detección y reconocimiento facial con los métodos clásicos, tesis sobre los puntos a mejorar, implementación de detección y reconocimiento facial con sistemas actuales y viables en el proyecto y, finalmente, comparación de las ventajas frente a los métodos anteriores. El informe debe incluir: o Indicación del seguimiento de la planificación prevista del desarrollo del TFG y actualización de posibles cambios en los objetivos o metodología del informe anterior. o Explicación general del desarrollo que se está haciendo para conseguir los objetivos. o Bibliografía actualizada.	14/11/2021
Informe de progreso II	Durante este *sprint* se pretende completar la implantación del reconocimiento facial en la microcomputadora NVIDIA Jetson Nano, la creación de un *dataset* propio (con la ayuda de los trabajadores del CVC) y la implementación de la identificación por voz; estos objetivos corresponden a las Fases 3, 4 y 5. Además, se aspira a efectuar una parte de la Fase 6, la cual corresponde a experimentos y resultados con las técnicas utilizadas. El informe debe incluir: o Indicación del seguimiento de la planificación prevista del desarrollo del TFG y de los ajustes efectuados. o Exposición de los resultados en los experimentos. o Conclusiones provisionales. o Bibliografía actualizada.	19/12/2021

Propuesta de informe final	En este *sprint* se requiere, principalmente, que el informe final tenga el formato de artículo, con los siguientes apartados diferenciados: objetivos, estado del arte, metodología, resultados, conclusiones y bibliografía. Para llegar con más margen a esta entrega, se elaborará la memoria del proyecto en el formato requerido desde el principio. De esta forma, se dispondrá de más tiempo para trabajar las Fases 6 y 7, correspondientes a la finalización de los experimentos y resultados y a la activación de una salida con la NVIDIA Jetson Nano.	23/01/2022
Entrega final	En las dos semanas restantes del TFG, se podrán completar todos los puntos que se hayan podido arrastrar de *sprints* anteriores y se cerrará el proyecto con la Fase 8, la cual engloba el repositorio en GitHub, el informe final, la presentación y el póster.	07/02/2022

Tabla 1. Decripción de sprints.