Universidad de la República

Facultad de Ingeniería

Reconocimiento de rostros aplicado a robots de servicio

Informe de Proyecto de Grado presentado por

Manuel Eirea y Federico Peña

como requisito de graduación de la carrera de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de la República

Supervisores

Federico Andrade
Mercedes Marzoa
Gonzalo Tejera

Montevideo, 13 de diciembre de 2022

Resumen

En los últimos años, se ha ido incorporando cada vez más tecnología en los hogares. Sin embargo, pocos de ellos cuentan con un robot de servicio que interactúe con los integrantes del hogar y sea capaz de llevar a cabo tareas domésticas. Para que la interacción sea lo más amena posible, resulta importante que el robot posea un sistema de reconocimiento de rostros, que le permita relacionarse de forma personalizada con las personas que se encuentran en el hogar.

Este proyecto de grado busca, en el marco de la competencia de robots de servicio RoboCup@Home, construir un módulo de ROS a ser utilizado en dicha competencia por AlfredZ, un robot que está siendo desarrollado por el grupo MINA. Para desarrollar este componente, se analizan las pruebas de la competencia, y se evalúan distintos modelos de reconocimiento en búsqueda de aquel que más se adecúe a ella. Esta evaluación se realiza en dos etapas. En la primera, se evalúan los modelos sobre conjuntos de imágenes estándar como LFW y WIDER Face. En la segunda se evalúan los modelos a tiempo real, utilizando un conjunto de imágenes confeccionado en este proyecto para la generación de representaciones. Adicionalmente, se evalúan modelos de predicción de género, edad y raza, que pueden ser útiles en los objetivos opcionales de la competencia.

El componente que se desarrolla como resultado de esta investigación permite principalmente realizar reconocimiento facial utilizando un modelo de detección basado en MTCNN, y otro de extracción de características basado en FaceNet-512, ambos construidos con redes neuronales. A su vez, el componente ofrece de forma secundaria modelos de predicción de edad y género.

Índice general

1. Introducción **1**
- 1.1. Presentación del problema y contexto . . . 1
- 1.2. Estructura del documento . . . 3

2. Estado del arte **5**
- 2.1. RoboCup@Home . . . 5
  - 2.1.1. Tendencias . . . 5
  - 2.1.2. Métodos utilizados por equipos destacados . . . 6
- 2.2. Detección de rostros . . . 6
  - 2.2.1. Trabajos fundacionales . . . 6
  - 2.2.2. Multi-task Cascaded Convolutional Networks (MTCNN) . 8
- 2.3. Reconocimiento de rostros . . . 8
  - 2.3.1. Trabajos fundacionales . . . 9
  - 2.3.2. DeepFace . . . 9
  - 2.3.3. FaceNet . . . 10
  - 2.3.4. ArcFace . . . 11
  - 2.3.5. Circle Loss . . . 12
- 2.4. Resumen y resultados . . . 13
  - 2.4.1. Resultados de los métodos presentados . . . 14

3. Marco teórico **17**
- 3.1. Reconocimiento de rostros . . . 17
- 3.2. Redes neuronales . . . 18
  - 3.2.1. Redes convolucionales . . . 20
  - 3.2.2. ResNet . . . 21
- 3.3. Robot Operating System (ROS) . . . 23

4. Solución propuesta **25**
- 4.1. Problema a resolver . . . 25
  - 4.1.1. RoboCup@Home . . . 26
- 4.2. Solución propuesta . . . 28
  - 4.2.1. Arquitectura de la solución . . . 28
  - 4.2.2. Multi-task Cascaded Convolutional Networks (MTCNN) . 31
  - 4.2.3. FaceNet . . . 32

4.2.4. Parámetros de configuración 33

5. Experimentación **35**
5.1. Métodos a evaluar 37
5.2. Medidas de desempeño 38
5.3. Pruebas sobre conjuntos de datos 40
5.3.1. Conjuntos de datos utilizados 40
5.3.2. Pruebas de detección de rostros 44
5.3.3. Prueba de verificación de rostros 49
5.3.4. Pruebas de predicción de características 52
5.4. Pruebas en escenarios reales 56
5.4.1. Requisitos 57
5.4.2. Escenario I 59
5.4.3. Escenario II 73
5.4.4. Compilación de resultados 76
5.5. Principales decisiones tomadas 78

6. Conclusiones y Trabajo Futuro **79**
6.1. Conclusiones 79
6.2. Trabajo futuro 80
Referencias 82

A. Pruebas de la RoboCup@Home **93**
A.1. Despedida 93
A.2. Encuentra a mis amigos 94
A.3. Recepcionista 94
A.4. Sirviendo las bebidas 95
A.5. Restaurante 96
A.6. Generalidades 96

B. Predicción de características **97**
B.1. Proporciones de los subconjuntos de UTKFace 97
B.2. Resultados de predicción de características 99
B.2.1. Predicción de edad 99
B.2.2. Predicción de género 99
B.2.3. Predicción de raza 100
B.3. Experimentos de predicción de características 100
B.3.1. Experimento 1.4 101
B.3.2. Experimento 2.2 101

C. Experimento 1.1 con variaciones **103**

Capítulo 1

Introducción

En este capítulo, se presenta una introducción al proyecto de reconocimiento de rostros aplicado a robots de servicio. Luego, se presenta el problema a resolver y su contexto, así como las preguntas que motivan el trabajo desarrollado. Finalmente, se brinda un breve resumen de la estructura del documento.

1.1. Presentación del problema y contexto

Los robots de servicio han tenido grandes avances en las últimas décadas. Este tipo de robot tiene como objetivo contribuir al bienestar de los seres humanos, de manera parcial o totalmente autónoma, realizando tareas que pueden ser repetitivas, como las tareas domésticas (Mettler, Sprenger, y Winter, 2017). Hoy en día existen en el mercado robots aspiradores, fregadores de piso, y cortadores de pasto entre otros. Sin embargo, es menos común encontrar en hogares robots multipropósito, que abarquen varias tareas o tareas más complejas. Para comunicar tareas a un robot de este tipo, sería de utilidad que tuviera la capacidad de interactuar con las personas de forma más amena que con un control remoto o botones, siendo esto lo habitual en la industria. Si bien ya existen algunos robots sociales como Jibo (jibo, 2021) y Buddy (Milliez, 2018), estos están enfocados en la interacción emocional, por lo que no llevan a cabo tareas básicas del hogar que involucren el manejo de objetos, como puede ser poner la mesa o lavar los platos. La capacidad de comunicarse y realizar tareas de este tipo es esencial para que en un futuro los robots de servicio entren a los hogares y puedan no solo hacer más fácil la vida de las personas, sino también ayudar a quienes lo necesitan, como son las personas ancianas o aquellas con capacidades diferentes.

Es deseable entonces que los robots de servicio reconozcan a las personas con las que interactúan, tanto para brindar una experiencia más amigable como para poder llevar a cabo funcionalidades específicas que requieran conocer la identidad del usuario. Una de las formas de identificar a las personas es utilizando un sistema de reconocimiento de rostros. Estos sistemas permiten reconocer rostros

a partir de imágenes o videos, e implica tres tareas: detección, extracción de características y reconocimiento de los rostros. La primera consiste en distinguir los rostros de otros elementos en la imagen, retornando los rostros encontrados. Luego se efectúa la extracción de características, la cual permite representar los rostros utilizando un vector de características. Finalmente, se lleva a cabo el reconocimiento de rostros, que se divide en las tareas de identificación y verificación, según el objetivo que se quiera lograr. La primera tarea busca determinar la identidad de un rostro, comparando su representación contra un conjunto de representaciones de las cuales se conoce su identidad. La segunda consiste en comparar la representación de un rostro con otra representación en particular, de la cual se conoce su identidad, para verificar si ambas representaciones se corresponden con la misma identidad (Kortli, Jridi, Al Falou, y Atri, 2020). En la figura 1.1 se puede observar un ejemplo de reconocimiento, denotando las tareas mencionadas.

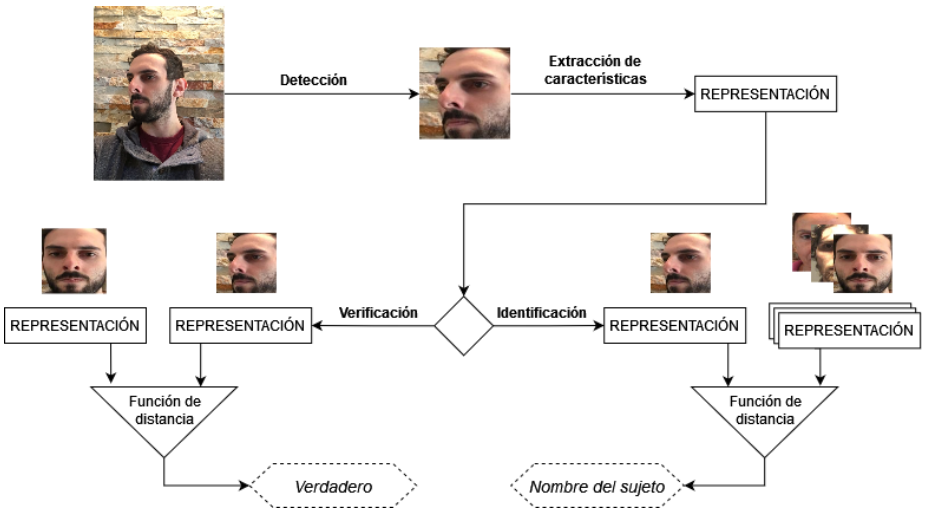


Figura 1.1: Diagrama que representa el flujo de reconocimiento y sus distintas etapas para una imagen.

La presente tesis de grado se realiza en el contexto de la competencia internacional RoboCup@Home[1]. Esta competencia tiene como propósito fomentar el desarrollo de robots de servicio autónomos a partir de pruebas que evalúen sus habilidades, y que impulsen a los participantes a innovar en estas tecnologías. En particular, el sistema de reconocimiento facial desarrollado en el proyecto tiene como objetivo ser incorporado en el robot AlfredZ[2], el cual está siendo desarrollado por el grupo MINA de la Universidad de la República, y se espera

[1] http://www.robocupathome.org/
[2] https://gitlab.fing.edu.uy/superrobot/alfred/-/wikis/AlfredZ-::-Robocup@Home

que participe en la competencia en años próximos. RoboCup@Home se divide en tres ligas: *Domestic Standard Platform League* (DSPL), *Social Standard Platform League* (SSPL), y *Open Platform League* (OPL). AlfredZ participará en esta última liga, dado que las dos primeras tienen un robot predefinido a utilizar.

Varias pruebas de la competencia requieren de reconocimiento de rostros, por lo que el proyecto aportará un componente importante a AlfredZ. Para ejemplificar, en la edición del 2019 de la competencia se presentó una prueba titulada *Sirviendo las bebidas*, la cual requería que los robots tomaran pedidos y entregaran bebidas a las personas correspondientes (Matamoros, Rascon, y cols., 2019). Para poder realizarlo es necesario que el robot reconozca a la persona que hizo el pedido, y recuerde la asociación entre ella y su bebida para luego entregársela.

Este proyecto busca entonces desarrollar un módulo que le permita a un robot llevar a cabo las pruebas de la RoboCup@Home que involucran el reconocimiento de rostros, y a su vez hacer un aporte a la comunidad desarrollando un componente que sea útil en el futuro de la robótica de servicios.

Para desarrollar dicho componente, en este proyecto se evalúan distintos modelos de reconocimiento de rostros con el objetivo de determinar cuál de ellos da mejores resultados, y se empaquetan los dos que presentan un mejor desempeño en un componente de ROS (Stanford Artificial Intelligence Laboratory et al., s.f.). La evaluación realizada sobre estos modelos comprende pruebas sobre conjuntos de datos (del inglés *datasets*) públicos y pruebas en escenarios reales. Sobre estas últimas, algunos aspectos a evaluar son la capacidad de los modelos de reconocer a una persona ya presente en la base de datos, agregar a una persona nueva a la base de datos y luego reconocerla, y reconocer múltiples personas en el mismo campo de visión.

Este trabajo busca responder las siguientes preguntas de investigación:

- *¿Cómo se desempeñan los modelos de reconocimiento de rostros del estado del arte?*
- *¿Cuál es el modelo de reconocimiento de rostros más adecuado para las pruebas de la RoboCup@Home?*

Además, resulta interesante responder la siguiente pregunta auxiliar, que si bien no representa los objetivos principales de este trabajo, puede servir como una guía para su desarrollo:

- *¿Cómo son y en que medida requieren reconocimiento de rostros las pruebas de la RoboCup@Home?*

1.2. Estructura del documento

Este documento está compuesto por seis capítulos, donde el segundo y el tercero reflejan lo estudiado en la etapa de investigación, mientras que el cuarto

y quinto están enfocados en la etapa de implementación y experimentación del proyecto.

El capítulo dos presenta el estado del arte, dónde se describen los principales modelos de reconocimiento de rostros que se utilizan hoy en día y los pioneros en la materia. En el tercer capítulo se expone el marco teórico del estudio realizado. Este capítulo permite comprender los conceptos principales sobre los que se desarrolla el trabajo, y a su vez posibilita un mejor entendimiento de lo presentado en el capítulo anterior.

El capítulo cuarto presenta la solución propuesta, describiendo su arquitectura, los modelos empleados y sus parámetros de configuración. En el capítulo cinco se especifican las pruebas de evaluación realizadas sobre los distintos modelos, y se presentan y analizan los resultados obtenidos.

Finalmente, en el capítulo seis se realiza una síntesis de los resultados obtenidos y se presentan las conclusiones del trabajo. A su vez, se trazan líneas de investigación que surgen del desarrollo de este, en las que sería fructífero profundizar.

Capítulo 2

Estado del arte

El objetivo de este capítulo es brindar al lector una visión del estado del arte en las áreas de detección y reconocimiento, siendo el capítulo un resumen del documento de estado del arte asociado a este proyecto (Eirea y Peña, 2022).

Primero, se realiza un análisis de las tendencias y principales métodos utilizados por equipos que se destacaron en las tres últimas ediciones de la RoboCup@Home (*RoboCup@Home*, 2022). Este análisis permite identificar potenciales métodos de reconocimiento a utilizar, considerando sus buenos resultados en ediciones anteriores de la competencia.

Luego, se tienen dos secciones donde se presentan los trabajos fundacionales en detección y reconocimiento de rostros, así como los trabajos que conforman el estado del arte de cada tarea. Estos últimos trabajos están principalmente basados en redes neuronales.

Finalmente, se presenta un breve resumen de los principales aportes de cada trabajo y los resultados presentados cuando corresponde. Este análisis permite conocer el desempeño de los trabajos del estado del arte, y entender como se comparan entre ellos y con los trabajos fundacionales. Esto último visibiliza la evolución que han tenido la detección y el reconocimiento de rostros a lo largo de los años, y el impacto que tienen las redes neuronales en dichas áreas.

2.1. RoboCup@Home

2.1.1. Tendencias

En 2018, Matamoros (Matamoros, Viktor, y Paulus, 2019) elaboró un artículo presentando las tendencias, los desafíos y las estrategias utilizadas en la competencia, donde se consideraron los participantes hasta el 2017. El autor resalta que entre las técnicas de detección facial más utilizadas se encuentran OpenFace (Amos, Ludwiczuk, Satyanarayanan, y cols, 2016), el algoritmo Viola-Jones (Viola y Jones, 2001) y otros algoritmos basados en características de Haar. Para el reconocimiento de rostros se cuenta con una gran variedad de méto-

dos, incluyendo histogramas de gradientes orientados (HOG) con clasificadores Support Vector Machine (SVM) (Liu, Zhang, Tang, y Chen, 2017), el algoritmo Viola-Jones con Eigenfaces (Turk y Pentland, 1991), redes convolucionales que a su vez son redes siamesas y Haar Cascades con Eigenfaces o redes neuronales profundas. Por último, todos los equipos usaron al menos un sensor RGB-D, siendo la tecnología Microsoft Kinect v2 la más utilizada debido a su buena resolución y a sus sensores de tiempo de vuelo, que miden la distancia hacia un objeto proyectando múltiples rayos de luz infrarroja y registrando el tiempo entre que la luz es emitida, y recibida en los sensores luego de ser reflejada en la superficie (Terpstra, Owens, Hashemian, y Voitel, 2021).

2.1.2. Métodos utilizados por equipos destacados

Se decidió analizar los métodos utilizados por ciertos equipos destacados, que se encuentran en el podio en las ediciones más recientes de la RoboCup@Home. Desafortunadamente, el equipo con mejores resultados (Memmesheimer y cols., 2019) sólo presenta en su artículo el método que utiliza para detectar personas, no rostros. Para realizar esta tarea utiliza un modelo entrenado en base a redes neuronales convolucionales y *boostrapping*. Otros equipos destacados de la competencia (Savage y cols., 2019)(van der Burgh y cols., 2019)(Fabre y cols., 2019) utilizan como métodos de detección Haar Cascades (Viola y Jones, 2001), OpenFace (Amos y cols., 2016), y una MTCNN(K. Zhang, Zhang, Li, y Qiao, 2016). Además, para el reconocimiento facial utilizan FaceNet (Schroff, Kalenichenko, y Philbin, 2015), OpenFace(Amos y cols., 2016) y una implementación propia de una red siamesa. Se decide entonces detallar a continuación, con mayor profundidad, los trabajos fundacionales y algunas técnicas más avanzadas tanto para la detección de rostros 2.2 como para la identificación de rostros 2.3.

2.2. Detección de rostros

2.2.1. Trabajos fundacionales

La detección de rostros es un caso particular de detección de objetos. Si bien hoy en día existen muchas aproximaciones usando redes neuronales, los métodos tradicionales no han dejado de usarse. Esto se debe en parte a su velocidad y bajo consumo de recursos. Además, a diferencia de los acercamientos basados en aprendizaje profundo, para este tipo de método generalmente se sabe por qué se comportan de la manera que lo hacen, por lo que es posible conocer su confianza. Zou *et al.* (Zou, Shi, Guo, y Ye, 2019) destacan tres métodos tradicionales: los detectores Viola-Jones (Viola y Jones, 2001), también conocidos como Haar Cascades, los histogramas de gradientes orientados (HOG) (Dalal y Triggs, 2005) (del inglés *histogram of oriented gradients*), y el modelo deformable basado en partes (DPM) (Felzenszwalb, McAllester, y Ramanan, 2008) (del inglés *deformable part-based model*). A continuación se brinda una breve descripción de cada uno de ellos.

El primero de estos métodos, Haar Cascades, es un método de aprendizaje automático que permite realizar la detección de objetos y rostros rápidamente. El modelo permite combinar múltiples clasificadores más débiles en forma de cascada y generar uno más fuerte. Esto hace posible descartar rápidamente el fondo de la imagen, y enfocarse en las áreas que potencialmente contienen el objeto de estudio. Los clasificadores se entrenan en base a características de Haar. Estas características son ventanas rectangulares de píxeles subdivididas en regiones rectangulares adyacentes, como se muestra en la figura 2.1. Se suman las intensidades de los píxeles en las regiones claras y se computa la diferencia con la suma de las intensidades en las regiones oscuras.

Los detectores HOG (Dalal y Triggs, 2005) se basan en evaluar los histogramas de orientaciones de gradiente de la imagen localmente normalizados en una cuadrícula densa.

Por último, DPM (Felzenszwalb y cols., 2008) surgió como una extensión a HOG y sigue la metodología de divide y conquistarás, descomponiendo el objeto en diferentes secciones para reconocerlo como conjunto de sus partes. De esta forma, un detector de este tipo consiste de un filtro global denominado raíz y varios filtros de partes, cuyos resultados (que se obtienen realizando el producto escalar entre un conjunto de pesos y un HOG) son tenidos en cuenta para la decisión de detección. Para entrenar los modelos, se utiliza una variante de SVM denominada SVM latente (LSVM), que permite, entre otras cosas, considerar partes dentro de las propias partes del objeto, ayudan a detectar objetos no sólo cuando están orientados de frente a la cámara y permiten representar el objeto en 3D. Este detector ganó tres ediciones consecutivas de la competencia VOC, consagrándose en su momento como lo mejor en términos de detección de objetos.

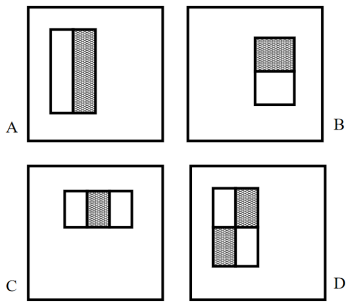


Figura 2.1: Ejemplos de características Haar relativas a una ventana de detección. Imagen extraída de Viola y Jones (Viola y Jones, 2001).

En los últimos años, las redes neuronales han representado un auge en las tareas que involucran aprendizaje automático, y las tareas de detección no se han quedado atrás. Es por ello que en la siguiente sección se presenta un modelo basado en redes neuronales, que conforma el estado del arte de la detección de

rostros.

2.2.2. Multi-task Cascaded Convolutional Networks (MTCNN)

El alineamiento de rostros es una tarea que permite detectar características geométricas en rostros (Gu y Kanade, 2009), para luego determinar por ejemplo la forma de los ojos, o de la boca. Debido a la relación inherente entre esta tarea y la detección de rostros, Zhang *et al.* (K. Zhang y cols., 2016) proponen realizar ambas en conjunto, aumentando sus desempeños. Un ejemplo de esto puede observarse en la figura 2.2, donde se muestra el resultado de aplicar MTCNN a una imagen de una persona, obteniendo el cuadro delimitador del rostro y cinco puntos fiduciarios.

Figura 2.2: Ejemplo del resultado de aplicar MTCNN a una imagen para detectar y alinear un rostro en ella. Imagen extraída de (de Paz Centeno, Iván, 2021).

Para realizar las tareas de detección y alineamiento en conjunto, Zhang *et al.* presentan una arquitectura compuesta por tres redes neuronales convolucionales (CNNs) dispuestas en cascada, cada una correspondiéndose con una etapa del procesamiento de la imagen. La información que retorna MTCNN podría ser utilizada para tareas como identificación y verificación de rostros, por lo que es muy útil en el ámbito de reconocimiento de rostros.

2.3. Reconocimiento de rostros

En esta sección se exponen los principales trabajos fundacionales del reconocimiento de rostros, así como algunos destacados de los últimos tiempos. Se considera la etapa del reconocimiento posterior a la detección, que puede involucrar la identificación o la verificación.

2.3.1. Trabajos fundacionales

Los pioneros en el área de reconocimiento de rostros son Bledsoe *et al.* (Bledsoe, 1966), quienes en 1964 comenzaron a utilizar computadoras para reconocer rostros humanos. Para esto marcaban puntos de referencia en los rostros, como los centros de los ojos o la boca, de forma manual.

Más adelante Goldstein *et al.* (Goldstein y Mackenberg, 1966) extendieron el trabajo de Bledsoe incluyendo veintiún puntos de referencia subjetivos. Si bien este sistema aportó una mejora en el desempeño, seguía implicando marcar los puntos de forma manual.

Sirovich y Kirby (Sirovich y Kirby, 1987) proponen en 1987 aplicar álgebra lineal al problema del reconocimiento de rostros, a través de lo que denominan *eigenfaces*. Esto dio pie para que en 1991 Turk y Pentland (Turk y Pentland, 1991) presentaran el sistema de clasificación Eigenfaces, siendo este considerado por muchos como la primera tecnología de reconocimiento de rostros funcional. Eigenfaces busca capturar la variación en una colección de imágenes de rostros, y usar esta información para luego codificar y comparar imágenes de rostros individuales de forma holística. Para esto utiliza la técnica de análisis de componentes principales (PCA), que permite describir un conjunto de datos en términos de nuevas variables no correlacionadas.

En 1996 Ojala *et al.* (Ojala, Pietikäinen, y Harwood, 1996) proponen utilizar Histogramas Local Binary Pattern (LBP) para el reconocimiento de rostros, permitiendo representar las imágenes como vectores de datos. El algoritmo es entrenado en base a un conjunto de imágenes de rostros. Para cada imagen, se obtiene un histograma representativo del rostro en ella mediante un conjunto de pasos que incluye realizar operaciones LBP sobre dichas imágenes. Al momento de llevar a cabo un reconocimiento, se calcula un histograma asociado a la imagen de entrada, y se lo compara con los calculados previamente. El más cercano será el que determinará la identificación.

Como alternativa a Eigenfaces, en 1997 Belhumeur *et al.* (Belhumeur, Hespanha, y Kriegman, 1997) propusieron el sistema Fisherfaces. Este tiene como base el álgebra lineal al igual que Eigenfaces, pero en vez de utilizar PCA hace uso del análisis discriminante lineal (LDA). Esta técnica permite representar los vectores de una misma clase cerca de un mismo punto del espacio, y representar vectores de diferentes clases lo más lejos posible entre sí. Esto no era posible con PCA, que solía tener problemas por ejemplo al encontrarse con imágenes de un mismo rostro pero con diferentes condiciones de iluminación.

Así como sucedió en el área de detección, se produjo un fuerte ascenso de las redes neuronales en el ámbito del reconocimiento facial. En las próximas subsecciones se describen algunas de ellas que se consideran destacables, y representan el estado del arte en la área.

2.3.2. DeepFace

DeepFace (Taigman, Yang, Ranzato, y Wolf, 2014) es un sistema de reconocimiento facial desarrollado por Facebook que se enfoca principalmente en

el alineamiento y representación de las imágenes. Estas tareas son realizadas en el pipeline de reconocimiento facial luego de detectar el rostro y antes de clasificarlo.

En la etapa de alineamiento, los autores utilizan un modelo 3D de un rostro que está basado en puntos fiduciarios para frontalizar el rostro detectado. Estos puntos son extraídos en varias iteraciones, a través de Support Vector Regression (SVR), y el descriptor visual está basado en Histogramas Local Binary Pattern (LBP).

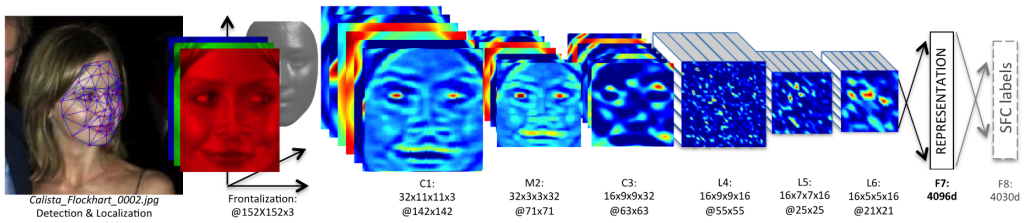


Figura 2.3: Esquema de la arquitectura de DeepFace. Imagen extraída de Taigman *et al.* (Taigman y cols., 2014).

La representación del rostro se obtiene mediante el uso de una red neuronal de nueve capas, cuya arquitectura puede observarse en la figura 2.3. Esta red incluye las etapas de frontalización, extracción de características y aplicación de filtros. Finalmente, se obtiene un vector de características que puede ser utilizado como representación del rostro y la distribución sobre las clases, asociada a la probabilidad de que el rostro pertenezca a cada una de dichas clases.

2.3.3. FaceNet

FaceNet (Schroff y cols., 2015) es un sistema que permite representar rostros en imágenes mediante representaciones vectoriales (del inglés *embeddings*), y luego obtener la similitud entre estos rostros a partir del cálculo de una distancia entre sus representaciones. Para esto se tiene que cumplir que dadas dos representaciones cualquiera de una misma persona, la distancia entre ellas sea menor a cierto umbral, y que a la vez esta supere dicho umbral para dos representaciones de personas distintas.

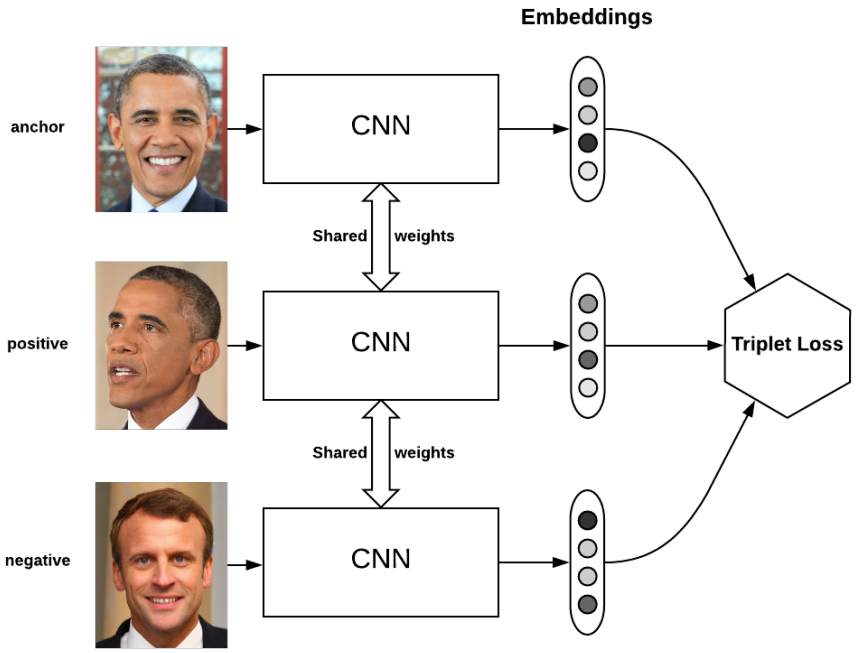


Figura 2.4: Ejemplo de aplicación de la función de pérdida de la tripleta. Imagen extraída de Moindrot (Moindrot, 2018).

El sistema es entrenado utilizando una CNN, y una función de pérdida, denominada por los autores pérdida de la tripleta (del inglés *triplet loss*), que es el eje principal de la propuesta de FaceNet. Esta función ajusta sus parámetros en función de tripletas con dos imágenes de la misma persona (una imagen ancla y un ejemplo positivo), y una de un sujeto diferente (un ejemplo negativo), como puede observarse en la figura 2.4. Una vez entrenado el sistema, se tiene una función de similitud que puede ser utilizada para verificar e identificar rostros.

2.3.4. ArcFace

Additive Angular Margin Loss (también conocido como ArcFace) propone un acercamiento con una nueva función de pérdida para mejorar el poder discriminativo del modelo de reconocimiento de rostros y estabilizar su entrenamiento. Si bien la función *softmax loss* es ampliamente utilizada en el aprendizaje automático, tiene el problema de no optimizar la representación vectorial para generar mayor similitud entre ejemplos de una misma clase y mayor diferencia entre ejemplos de distintas clases. Deng *et al.*(Deng, Guo, Xue, y Zafeiriou, 2019) intentan resolver este problema presentando ArcFace, y resaltan las siguientes ventajas del modelo. Es efectivo, ya que se ha logrado obtener resultados a nivel

del estado del arte en diez benchmarks de reconocimiento de rostros diferentes. Es sencillo de implementar, necesitándose una baja cantidad de líneas de código para hacerlo. Además, no necesita ser combinado con otras funciones de pérdida para lograr estabilidad y convergencia. Por último, es un método eficiente, requiriendo pocos recursos computacionales para el entrenamiento.

2.3.5. Circle Loss

En 2020, Sun *et al.* (Sun y cols., 2020) proponen una nueva función de pérdida titulada *Circle Loss* con el objetivo de maximizar la similitud intraclase (s_p) y minimizar la similitud interclase (s_n). Gran parte de las funciones de pérdida ya existentes (*p. ej.*, pérdida de la tripleta) intentan resolver esto reduciendo ($s_n - s_p$). Por otro lado, la función propuesta por los autores (nombrada *Circle Loss* por la forma circular de su límite de decisión) busca reducir ($\alpha_n s_n - \alpha_p s_p$), donde α_n y α_p son factores de ponderación. Esto permite ajustar de forma independiente s_n o s_p, desembocando en un proceso de optimización más flexible donde cada similitud aprende a su ritmo. A su vez, a diferencia de la utilización de funciones anteriormente propuestas, utilizar *Circle Loss* resulta en un único punto de convergencia objetivo.

Función de pérdida	Identificación ↑	
	R34	R100
Softmax	92,36	95,04
NormFace	92,62	95,27
AM-Softmax	97,54	98,31
ArcFace	97,68	98,36
Circle Loss	**97,81**	**98,50**

Cuadro 2.1: Resultados de exactitud rank-1 (%) para la tarea de identificación de rostros sobre el desafío MFC1, utilizando R34 y R100. Cuadro extraído de Sun *et al.* (Sun y cols., 2020). Se resalta en negrita el mejor resultado para cada red, y se indica con una flecha si es preferible un valor mayor o menor para la medida.

Como se puede apreciar en el cuadro 2.1, *Circle Loss* obtiene resultados de exactitud muy prometedores en la tarea de identificación de rostros sobre el desafío MegaFace Challenge 1, superando a funciones de pérdida del estado del arte como AM-Softmax y ArcFace. El desafío mencionado está creado en base al conjunto de imágenes MegaFace, el cual se utiliza para evaluar las tareas de verificación e identificación de rostros. En particular, el desafío evalúa los algoritmos incorporando de diez a un millón de distractores en el conjunto de imágenes sobre el cual se evalúa, siendo estos distractores personas desconocidas (Kemelmacher-Shlizerman, Seitz, Miller, y Brossard, 2016). Las funciones de pérdida comparadas en el cuadro 2.1 se evalúan utilizando las arquitecturas ResNet 34 (R34) y ResNet 100 (R100), siendo estas redes neuronales residuales

de 34 y 100 capas respectivamente. En el capítulo 3 se profundiza sobre este tipo de red neuronal.

2.4. Resumen y resultados

Con el fin de simplificar la información expuesta, se presenta al lector un cuadro (2.2) que resume los trabajos relacionados relevados en el área de detección y otro (2.3) que hace esto mismo para las tareas de reconocimiento de rostros. Para cada trabajo, se explicita cuál fue su principal aporte, así como sus autores y año de publicación.

El relevamiento de trabajos relacionados permite identificar diversos modelos que potencialmente se podrían incluir en la solución a utilizar, conociendo sus propuestas y resultados. A su vez, proporciona una visión de la evolución en las respectivas áreas, desde los trabajos fundacionales hasta los que representan actualmente el estado del arte. En ambas áreas, se observa que a partir de la década del 2010 los trabajos relevados comienzan a estar basados fundamentalmente en redes neuronales, como se puede ver en la sección 2.4.1.

Detección de rostros	
referencia	*resumen*
(Viola y Jones, 2001)	Propone el algoritmo Viola-Jones, que combina múltiples clasificadores entrenados en base a características de Haar.
(Dalal y Triggs, 2005)	Propone los detectores HOG, que evalúan los histogramas de orientaciones de gradiente de la imagen localmente normalizados en una cuadrícula densa.
(Felzenszwalb y cols., 2008)	Propone los detectores DPM, que descomponen el rostro en diferentes secciones para reconocerlo como conjunto de sus partes, extendiendo lo planteado por Dalal and Triggs (Dalal y Triggs, 2005).
(K. Zhang y cols., 2016)	Propone la arquitectura MTCNN, que realiza las tareas de alineamiento y detección de rostros en conjunto, utilizando CNNs dispuestas en cascada.

Cuadro 2.2: Resumen de los trabajos de detección de rostros relevados.

Reconocimiento de rostros	
referencia	*resumen*
(Bledsoe, 1966)	Marca puntos de referencia en los rostros en forma manual utilizando una computadora.
(Goldstein y Mackenberg, 1966)	Define veintiún puntos de referencia subjetivos, extendiendo lo planteado por Bledsoe et al. (Bledsoe, 1966).
(Sirovich y Kirby, 1987)	Propone aplicar álgebra lineal al problema del reconocimiento de rostros, a través de lo que denomina eigenfaces.
(Turk y Pentland, 1991)	Presenta el sistema de clasificación Eigenfaces, que utiliza PCA, basándose en lo planteado por Sirovich and Kirby (Sirovich y Kirby, 1987). Es considerado por muchos como la primera tecnología de reconocimiento de rostros funcional.
(Ojala y cols., 1996)	Utiliza Histogramas LBP, permitiendo representar los rostros como vectores de datos.
(Belhumeur y cols., 1997)	Presenta el sistema de clasificación Fisherfaces, que utiliza análisis discriminante lineal (LDA), basándose en Turk and Pentland (Turk y Pentland, 1991).
(Taigman y cols., 2014)	Propone el sistema de reconocimiento de rostros DeepFace, que alinea los rostros utilizando un modelo 3D basado en puntos fiduciarios, y representa vectorialmente los rostros utilizando una red neuronal.
(Schroff y cols., 2015)	Propone el sistema de reconocimiento de rostros FaceNet y la función de pérdida de la tripleta. Utiliza una CNN para representar los rostros vectorialmente.
(Deng y cols., 2019)	Propone la función de pérdida ArcFace loss.
(Sun y cols., 2020)	Propone la función de pérdida Circle Loss.

Cuadro 2.3: Resumen de los trabajos de reconocimiento de rostros relevados.

2.4.1. Resultados de los métodos presentados

En esta sección se presentan los resultados reportados por algunos de los trabajos presentados anteriormente, tanto en el área de detección como reconocimiento. Debido a su año de publicación, no se tiene resultados para la mayoría de ellos.

En el área de detección, los trabajos considerados reportan sus resultados sobre el conjunto de datos WIDER Face (Yang, Luo, Loy, y Tang, 2016), que es un conjunto referente en la temática. Este es también utilizado en la etapa de experimentación de este proyecto, por lo que es explicado con más detalle en el capítulo 5. WIDER Face está subdividido en tres subconjuntos: uno fácil, uno de dificultad media y uno difícil. Los resultados se compilan en el cuadro 2.4 sobre este conjunto.

Detección de rostros			
referencia	*medida AP* ↑		
	fácil	*media*	*difícil*
(Viola y Jones, 2001)	0,412	0,333	0,137
(K. Zhang y cols., 2016)	**0,851**	**0,820**	**0,607**

Cuadro 2.4: Resultados de medida AP de los trabajos de detección de rostros relevados sobre las particiones de dificultad fácil, media y difícil del conjunto de datos WIDER Face. Se resalta en negrita el mejor resultado para cada partición.

Con respecto al reconocimiento de rostros, se presentan resultados sobre la tarea de verificación. Esto es así porque la mayoría de los trabajos considerados presentan resultados para esta tarea, no siendo así con la tarea de identificación.

Reconocimiento de rostros	
referencia	*exactitud* ↑ ± *std*
(Turk y Pentland, 1991)	60,02 ± 0,79 [1]
(Taigman y cols., 2014)	97,35 ± 0,25
(Schroff y cols., 2015)	99,63 ± 0,09
(Deng y cols., 2019)	**99,83**
(Sun y cols., 2020)	99,73

Cuadro 2.5: Resultados de exactitud (%) en la tarea de verificación, de los trabajos de reconocimiento de rostros relevados sobre el conjunto de datos LFW, indicando la desviación estándar (*std*) para los trabajos que la reportan. Se resalta en negrita el mejor resultado.

Como puede observarse en el cuadro 2.5, el método que obtiene mejores resultados sobre el conjunto de datos Labeled Faces in the Wild (LFW) (Huang, Mattar, Berg, y Learned-Miller, 2008) (presentado con mayor profundidad en el capítulo 5) es ArcFace (Deng y cols., 2019). De todas formas, los resultados

[1] El resultado de este trabajo se obtiene siguiendo los paradigmas de LFW paradigma de imágenes restringidas (del inglés *image-restricted*) y paradigma sin datos externos (del inglés *no outside data results*), a diferencia de los posteriores que siguen los paradigmas de LFW paradigma no restringido (del inglés *unrestricted*) y paradigma con datos externos (del inglés *labeled outside data results*).

reportados por FaceNet (Schroff y cols., 2015) y Circle Loss (Sun y cols., 2020) difieren en menos del 0,2 % con respecto a ArcFace, lo que no representa una diferencia significativa. Dado que el conjunto de datos LFW fue presentado en 2008, los trabajos anteriores al 2008 son excluidos del cuadro 2.5, a excepción de Eigenfaces(Turk y Pentland, 1991), cuyo resultado se obtiene de un estudio posterior (Huang y cols., 2008). Se aclara que el resultado de este trabajo en particular se obtiene siguiendo un paradigma distinto a los demás, por lo que sus resultados no se pueden comparar de forma equitativa. De todas formas, la gran brecha que hay entre este resultado y los posteriores sugiere que un cambio de paradigma difícilmente lo acercaría a los resultados obtenidos por trabajos más recientes.

Tanto en la tarea de detección como en la de verificación, se aprecia una gran diferencia entre el resultado de los trabajos basados en redes neuronales y los fundacionales, que favorece a los primeros. En la tarea de detección se tiene un porcentaje de diferencia de al menos 106 %, mientras que en la de verificación se tiene uno de al menos 62 %.

Capítulo 3

Marco teórico

En este capítulo, se busca introducir al lector a los conceptos necesarios para tener una mejor comprensión del presente documento. Para lograr esto, primero se introduce el concepto de reconocimiento de rostros, que es la principal tarea que debe ser capaz de realizar la solución presentada en el proyecto.

A continuación se formalizan las tareas que lo componen, siendo estas detección, extracción de características, verificación e identificación. Además, la solución propuesta utiliza métodos basados en redes neuronales, las cuales a su vez tienen una alta relevancia en el estado del arte del reconocimiento de rostros. Es necesario entonces definir la noción de redes neuronales, así como algunos conceptos relativos a ellas.

Luego, se exponen los distintos tipos de aprendizaje a partir de los cuales estas redes son entrenadas. Se profundiza en el aprendizaje supervisado, dado que es el tipo de aprendizaje empleado para entrenar las redes utilizadas en este proyecto.

Se presenta también el concepto de red convolucional y la arquitectura ResNet, por ser arquitecturas utilizadas en la solución.

Por último, se introduce el meta-sistema operativo para robots ROS, utilizado en el proyecto para empaquetar la solución y ejecutar los experimentos, y se exponen los conceptos de nodo y tópico.

3.1. Reconocimiento de rostros

El reconocimiento de rostros es una tecnología biométrica capaz de identificar o verificar la identidad de una persona analizando las características presentes en su rostro.

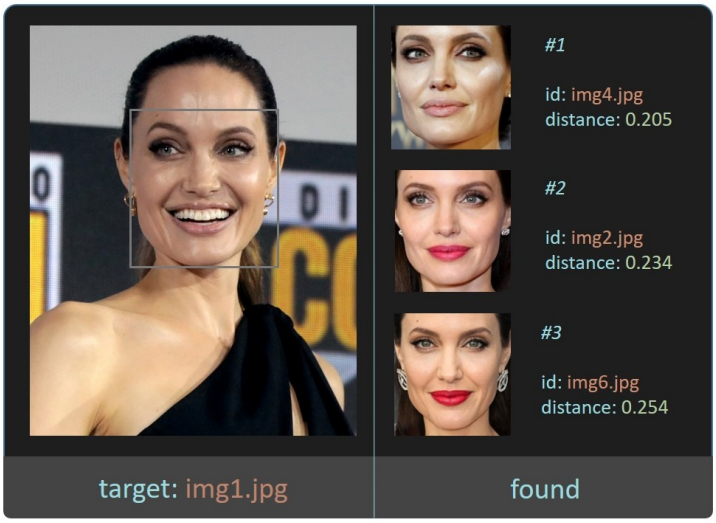


Figura 3.1: Ejemplo de detección e identificación de un rostro utilizando redes neuronales. Imagen extraída de (Serengil y Ozpinar, 2021).

Para realizar esta tarea, primero se debe detectar el rostro en la imagen, es decir, localizar el área de la imagen donde esté el rostro presente. Luego de haberse detectado, se construye una representación de dicho rostro analizando sus características. Las representaciones generadas se pueden utilizar más adelante para llevar a cabo las tareas de verificación o identificación.

La verificación de rostros implica determinar si dos rostros pertenecen a la misma persona, comparando sus representaciones asociadas. Por otro lado, la identificación de rostros consiste en comparar la representación de un rostro contra una base de datos ya existente, con el fin de determinar la identidad de la persona en base a la similitud entre las representaciones.

Como se vio en el capítulo anterior, los trabajos de reconocimiento de rostros que conforman el estado del arte están basados en redes neuronales. Además, como se verá más adelante, la solución propuesta en este trabajo también utiliza redes neuronales. Es por esto que se decide definir el concepto de redes neuronales en la siguiente sección, y presentar una breve descripción sobre algunos tipos de redes relevantes para este trabajo.

3.2. Redes neuronales

Las redes neuronales artificiales (en adelante ANN, por sus siglas en inglés), también llamadas simplemente redes neuronales, son algoritmos compuestos por un gran número de unidades de procesamiento interconectadas (neuronas) que trabajan juntas para resolver un problema específico (Gudikandula, 2019). Cada conexión tiene un peso w y un sesgo b asignados. La ANN es luego entrenada,

pudiendo seguir un método de aprendizaje supervisado, donde los datos de entrenamiento incluyen los resultados esperados, o de aprendizaje no supervisado, donde no se conoce en la etapa de entrenamiento cuáles son los resultados correctos (Donalek, 2011). Dado que en la solución propuesta en este proyecto, los modelos utilizados están preentrenados de forma supervisada, se pondrá foco en cómo se entrena la red siguiendo dicho paradigma. El entrenamiento se realiza a partir de información etiquetada (es decir, para la cual ya se sabe qué resultado se quiere que retorne la red), lo que provoca que estos pesos y sesgos se vayan ajustando, y por consecuencia, que la ANN obtenga un mayor desempeño en la resolución del problema en cada iteración. Una vez finalizado el entrenamiento, si este fue exitoso la red debería ser capaz de resolver el problema con nuevas entradas no conocidas.

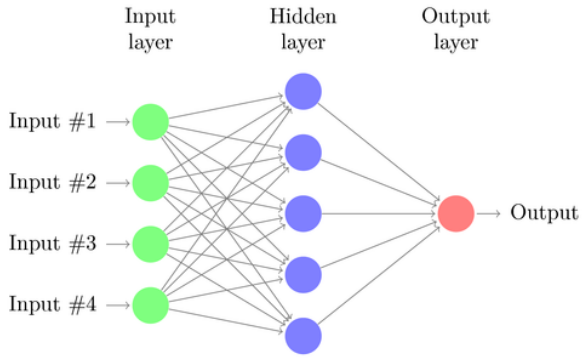


Figura 3.2: Ejemplo de ANN con una capa oculta. Imagen extraída de (Serengil y Ozpinar, 2021).

Las ANN se componen por tres tipos de capas: capa de entrada, capas ocultas, y capa de salida, como puede observarse en la figura 3.2. Cada una de estas capas se compone por un número de neuronas que deberá ser definido en la arquitectura de la red, así como el número de capas ocultas. La capa de entrada recibe información en crudo, para luego ser procesada por las capas ocultas. Estas reciben datos de la capa anterior (pudiendo esta ser la de entrada, u otra capa oculta), y aplican la siguiente función lineal:

$$\sum_i w_i x_i + b \tag{3.1}$$

donde w y b son los vectores de pesos y sesgos de cada conexión a través de la cual llegan las entradas a la neurona, y x el vector de entradas. Luego, al resultado obtenido se le aplica una función de activación, que tiene el propósito de incluir no-linealidad a la salida de la neurona. Finalmente, cuando los datos llegan a la capa de salida se vuelve a aplicar la ecuación 3.1 y una de activación, y se retorna el resultado obtenido.

Esta serie de pasos descritos se denomina *propagación hacia adelante*, y tiene como objetivo pasar los datos desde la capa de entrada a la de salida, para así realizar el procesamiento necesario y obtener los resultados. Sin embargo, si solo se llevara a cabo la *propagación hacia adelante*, no sería posible ajustar los pesos según el desempeño de la red para las entradas provistas. Esto es fundamental, porque es lo que le permite a la red aprender.

La actualización de pesos se realiza a través de un proceso de *propagación hacia atrás*. Luego de realizada la *propagación hacia adelante* se calcula el error que obtuvo la red, y este es propagado hacia atrás aplicando un método de optimización como *descenso por gradiente* para ajustar todos los pesos de la red, con el objetivo de reducir el error para las próximas iteraciones. Para calcular dicho error las redes utilizan *funciones de pérdida*, las cuales computan la distancia entre la salida del algoritmo y la esperada. El rol que cumplen estas funciones en el entrenamiento de las redes es crucial, lo que explica que varias de las propuestas destacables del estado del arte se basen en el planteo de una nueva función de pérdida, como por ejemplo el trabajo realizado por (Sun y cols., 2020) con su función *Circle Loss*. En definitiva, la propagación hacia atrás permite evaluar el desempeño actual del algoritmo, y utilizar esta información como retroalimentación para ajustar su funcionamiento.

Como se menciona en el capítulo 2, los métodos del estado del arte que implementan las tareas de detección y reconocimiento facial hacen uso de diferentes arquitecturas de redes neuronales, dentro de las cuales se encuentran las redes convolucionales y residuales. Se cree pertinente presentar una breve introducción de estos dos tipos de red.

3.2.1. Redes convolucionales

Este tipo de red es principalmente utilizada para clasificar imágenes, realizar análisis de grupos (del inglés *clustering*), y para el reconocimiento de objetos y rostros. La arquitectura de este tipo de redes suele consistir en múltiples capas convolucionales y de *submuestreo*, como *Max Pooling*. En las capas convolucionales se recorre la imagen comparándola con una serie de filtros para reconocer características. Esta comparación se realiza aplicando el producto escalar entre el filtro y la sección de la imagen que está siendo considerada, generando una matriz de salida como se muestra en la figura 3.3.

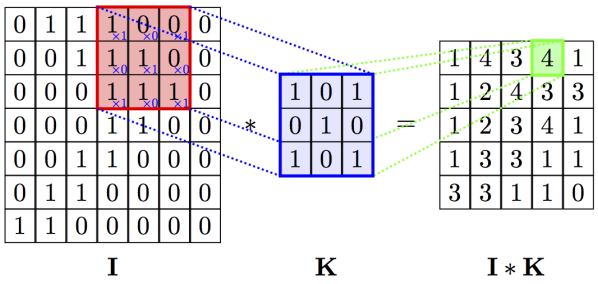


Figura 3.3: Aplicación de una convolución a una matriz I con un filtro K. Imagen extraída de (Veličković y cols., 2018).

Se debe resaltar que, para el caso de las imágenes a color, se va a tener una matriz para cada canal de color, por ejemplo, RGB. En este caso, también se tendrán tres filtros, uno por canal de color. Ahora, suponiendo que se trabaja con imágenes de 112x112 píxeles y 32 filtros, se generaría un mapa de características de tamaño (112x112x3) x 32, es decir, 1.204.224 neuronas. Realizar otra convolución con estos parámetros requeriría de un significativo poder computacional, que aumentaría aún más si se trabaja con resoluciones más grandes. Es por esto que se suele usar una capa de *submuestreo*, como *Max Pooling*, donde se selecciona el máximo de un conjunto de valores de una región del mapa de características, reduciendo la dimensionalidad de esta región y preservando las características más importantes obtenidas por cada filtro. Esta operación se puede apreciar en la figura 3.4 .

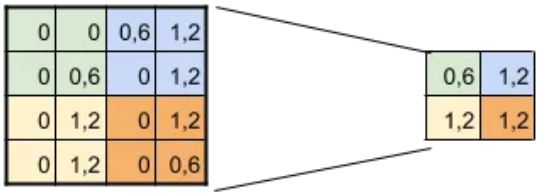


Figura 3.4: Aplicación de Max Pooling de 2x2 a una matriz de 4x4. Imagen extraída de (Barrios y Barrato, 2019).

3.2.2. ResNet

En general, en redes neuronales profundas aumentar la cantidad de capas ayuda a obtener mejores resultados, sin embargo, si esta aumenta demasiado puede llegar a disminuir el desempeño. Esto se debe a que en capas muy profundas, a la red se le dificulta ajustar los parámetros correctamente. He *et al.* (He, Zhang, Ren, y Sun, 2016) proponen una solución al *problema de degradación* a través de una nueva arquitectura de redes: las *redes residuales* (ResNet).

A continuación se explica el funcionamiento y utilidad de ResNet a través de un ejemplo.

Si se considera una CNN que utiliza una función de activación g, y sea $a[l]$ el resultado que proviene de la capa l de la red, la capa $l+1$ aplicaría una función lineal con el siguiente formato:

$$z[l+1] = W[l+1] * a[l] + b[l+1] \tag{3.2}$$

donde W y b representan los pesos y sesgos de cada capa respectivamente.

Luego, se aplicaría la función de activación de la siguiente manera:

$$a[l+1] = g(z[l+1]) \tag{3.3}$$

Análogamente, para el siguiente paso:

$$\begin{aligned} z[l+2] &= W[l+2] * a[l+1] + b[l+2], \\ a[l+2] &= g(z[l+2]) \end{aligned} \tag{3.4}$$

He *et al.* (He y cols., 2016) proponen, previo a aplicarle la función de activación a $z[l+2]$, sumarle $a[l]$ de la siguiente manera:

$$a[l+2] = g(z[l+2] + a[l]) \tag{3.5}$$

Esto se logra a partir de lo que se denomina una *conexión de salto*, la cual forma parte de un concepto introducido por (He y cols., 2016), denominado *bloque residual*. Este se compone por dos capas adyacentes de la red, y agrega la *conexión de salto* para hacer el pasaje de $a[l]$ dos capas en profundidad, como se ve en la figura 3.5.

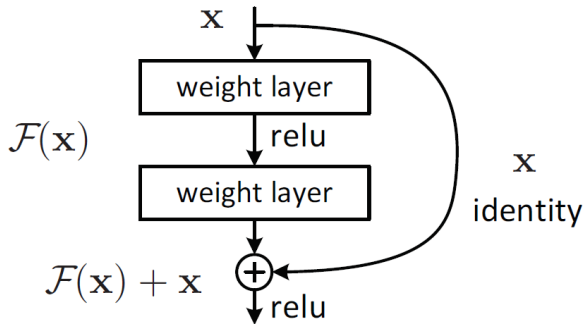


Figura 3.5: Bloque residual, donde x se corresponde con $a[l]$ y $F(x)$ con la función lineal aplicada por la capa correspondiente. Imagen extraída de (He y cols., 2016).

Sumar $a[l]$ a $z[l+2]$ ayuda a que la red pueda calcular fácilmente la función de identidad, dado que lo único que resta por hacer es llevar tanto el peso $W[l+1]$ como el sesgo $b[l+2]$ a 0, retornándose $a[l]$.

Como ahora es sencillo calcular la identidad, en el peor de los casos una capa muy profunda aplicará la identidad y el resultado se mantendrá igual, y ya no empeorará como sucedía en las *redes planas*, como las llaman He *et al.* (He y cols., 2016). En el mejor caso, esa capa profunda permitirá ajustar los parámetros de manera de obtener nueva información y, por lo tanto, un resultado aún mejor. Esto permite agregar cuantas capas se quiera a la red con la seguridad de que el error de entrenamiento no aumentará, sino que o se mantendrá constante, o disminuirá.

3.3. Robot Operating System (ROS)

ROS, por sus siglas en inglés *Robot Operating System*, es un conjunto de bibliotecas y herramientas que permiten la construcción de aplicaciones a ser utilizadas en robots (Stanford Artificial Intelligence Laboratory et al., s.f.). Las aplicaciones se componen por nodos organizados en una arquitectura de grafos, y la ejecución de estos nodos puede darse en una única computadora, o estar distribuida en múltiples equipos. Los nodos se comunican entre sí principalmente a través de lo que se denominan tópicos. Cada nodo se encarga de ejecutar una tarea particular de la aplicación, y es capaz de publicar o suscribirse a un tópico, permitiendo así el intercambio de mensajes entre nodos. Se puede ver en la figura 3.6 un ejemplo de comunicación entre nodos.

Figura 3.6: Ejemplo de comunicación entre un nodo publicador y múltiples nodos suscriptores a través de un tópico.

Se aclara que los nodos también se pueden comunicar mediante servicios, pero en este trabajo se utiliza únicamente la comunicación a través de tópicos.

Capítulo 4

Solución propuesta

En este capítulo se formaliza el problema a resolver, que implica esencialmente el desarrollo de un componente de reconocimiento facial a ser utilizado en un robot autónomo en una competencia internacional. Se presenta el objetivo de la competencia, y un análisis de sus pruebas más relevantes.

Luego, se introduce la arquitectura de la solución propuesta, que consiste principalmente de dos métodos basados en redes neuronales, MTCNN para detectar rostros y FaceNet-512 para la extracción de características. El modelo es también capaz de predecir la edad y el género de un rostro, para lo cual utiliza un sistema de predicción basado en redes convolucionales. Además, se presenta la solución en el contexto del meta-sistema operativo ROS, y se explica cómo interactúa con otros nodos de ROS.

A continuación, se extiende la descripción vista en estado del arte sobre los modelos utilizados, y se mencionan los principales parámetros configurables del sistema. Estos parámetros incluyen umbrales asociados a los métodos de detección y reconocimiento utilizados. El primero está asociado al tamaño del rostro en la imagen, el segundo a la confianza de las detecciones, y el último a la distancia entre representaciones de rostros. Además, se incluyen otros parámetros como uno para determinar qué tareas se quiere que realice el sistema, ofreciéndose reconocimiento, predicción de edad y predicción de género.

4.1. Problema a resolver

En el marco de la competencia internacional RoboCup@Home, el presente proyecto de grado busca desarrollar un componente de detección y reconocimiento de rostros para utilizar en el robot AlfredZ[1], el cual está siendo desarrollado por el grupo MINA de la Universidad de la República, y se espera que participe en la competencia en años próximos. Con el fin de lograr este objetivo, se describe la competencia y se analizan sus pruebas en la próxima sección.

[1] https://gitlab.fing.edu.uy/superrobot/alfred/-/wikis/AlfredZ-::-Robocup@Home

4.1.1. RoboCup@Home

La RoboCup@Home es una competencia internacional cuyo objetivo es fomentar el desarrollo de robots de servicio autónomos a partir de pruebas que evalúen sus habilidades, y que impulsen a los participantes a innovar en estas tecnologías. Estas pruebas se desarrollan en escenarios que se asemejan lo más posible a un ambiente hogareño, e incluyen disciplinas de navegación y mapeo, reconocimiento de objetos, personas y rostros, e interacción persona-computadora entre otras (*RoboCup@Home*, 2022). En la figura 4.1 se muestra un ejemplo de las interacciones que se dan entre robots y personas en esta competencia.

Figura 4.1: Robot interactuando con un participante en la edición del 2013 de la RoboCup@Home, en una prueba en la que se debe detectar a los participantes sedientos y darles una bebida. Imagen extraída de (*BvOF RoboCup2013 - RoboCup@Home*, 2013).

Con el propósito de determinar en qué escenarios y situaciones se expondrá al sistema de reconocimiento, se realiza un análisis de las pruebas de la RoboCup@Home (Matamoros, Rascon, y cols., 2019), identificando aquellas que involucran detección y reconocimiento. Se observa que este es el último manual disponible en la página oficial de la competencia, dado que el de 2020 no terminó de confeccionarse por la posposición de la competencia. En el cuadro 4.1 se presenta un resumen describiendo dichas pruebas.

prueba	*resumen*	*tareas*
Despedida	Alcanzar abrigos a una serie de invitados, y luego guiarlos a un taxi. Se debe reconocer a los invitados para entregarles el abrigo correcto. Como objetivo opcional, se propone guiar al taxi a la invitada de género femenino primero.	Detección Reconocimiento Predicción de género
Encuentra a mis amigos	Localizar y describir al menos a dos personas en base a su nombre.	Detección Reconocimiento Predicción de edad Predicción de género
Recepcionista	Llevar a dos invitados a la sala de estar, presentarlos y ofrecerles un lugar libre para sentarse. Se debe asociar a los rostros de los invitados con sus nombres y tragos favoritos. Como objetivo opcional, se propone ofrecer el sillón disponible a la persona de mayor edad.	Detección Reconocimiento Predicción de edad
Sirviendo las bebidas	Tomar órdenes y entregar bebidas a todas las personas que no tengan una. Se debe asociar los rostros de los invitados con sus órdenes.	Detección Reconocimiento
Restaurante	Tomar y servir al menos una orden en un restaurante. Se debe detectar a los comensales mientras hacen el pedido para mantener contacto visual.	Detección

Cuadro 4.1: Resumen de las pruebas de la RoboCup@Home donde se requieren tareas relevantes para el proyecto.

En base al análisis de las pruebas de la competencia, se definen los escenarios de prueba a utilizar en la sección experimental. En particular, se utilizará como referencia la prueba Recepcionista, Despedida y Sirviendo las bebidas (del inglés *Receptionist*, *Farewell* y *Serving the drinks*). Los escenarios de prueba serán especificados en el capítulo 5. Para un análisis más completo de dichas pruebas, se puede referir al apéndice A.

4.2. Solución propuesta

En esta sección se presenta la arquitectura de la solución y se profundiza en cada uno de sus módulos. A su vez, se explica por qué se utiliza ROS para empaquetar la solución, indicando también los nodos implementados y cómo se comunican entre ellos.

Por último, se presentan los parámetros que se pueden ajustar en el nodo desarrollado.

4.2.1. Arquitectura de la solución

La solución propuesta consiste en el desarrollo de un nodo de ROS, denominado *rostro_z*, que permite detectar y reconocer los rostros en imágenes en tiempo real, para lo cual utiliza modelos de detección y reconocimiento basados en redes neuronales. Además, la solución es capaz de predecir la edad y género de las personas en el campo de visión, utilizando modelos que también se basan en redes neuronales. Se observa que todos los modelos mencionados están preentrenados, es decir, no son entrenados en el contexto de este proyecto.

Se empaqueta la solución en un nodo de ROS para permitir la comunicación del componente con el resto de las funcionalidades que ofrece el robot AlfredZ, dado que el robot utilizará el meta-sistema ROS. Se espera que este robot tenga un tópico que publique imágenes captadas a tiempo real, y uno de los nodos que se suscriba a dicho tópico sea el presentado en este trabajo. De todas formas, se implementa un nodo auxiliar que cumple esta función para poder ejecutar los experimentos y utilizar la solución sin depender de otros nodos de AlfredZ. Este nodo se denomina *camara*, y se comunica con el nodo *rostro_z* a través del tópico *imagen*. La interacción entre estos dos nodos se pueden apreciar en el grafo presente en la figura 4.2.

Figura 4.2: Comunicación entre los nodos *camara* y *rostro_z* mediante el tópico *imagen*.

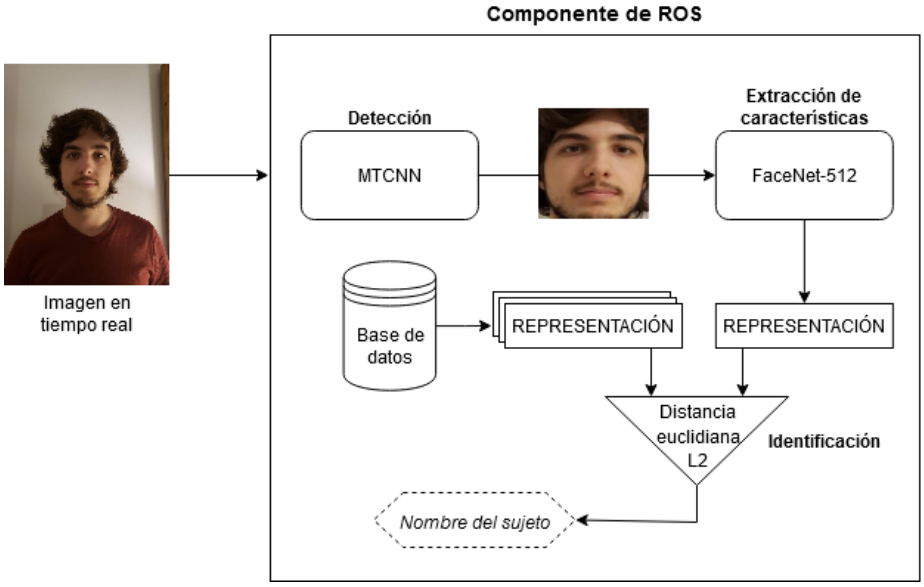


Figura 4.3: Diagrama a alto nivel de la arquitectura de la solución, y del flujo seguido en el nodo por una imagen capturada a tiempo real.

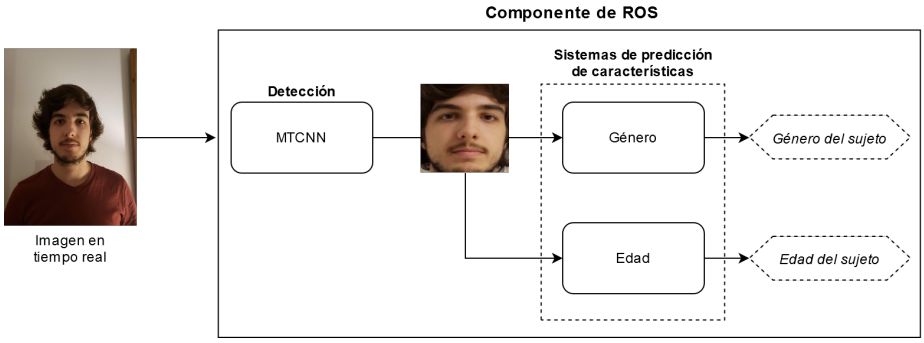


Figura 4.4: Diagrama a alto nivel de la arquitectura de la solución, y del flujo seguido en el nodo por una imagen capturada a tiempo real en el caso de la predicción de género y edad.

Como se puede apreciar en la figura 4.3, cuando el nodo de ROS recibe una imagen, intenta detectar un rostro usando MTCNN, y si la detección es exitosa, construye una representación de la imagen basándose en las características del rostro utilizando FaceNet-512. Luego, calcula la distancia entre esta representa-

ción y todas las almacenadas en la base de datos una por una, en búsqueda de aquella que minimice dicha distancia. En caso de que esta distancia sea menor al umbral de reconocimiento fijado, se retorna el nombre asociado a la representación. De otra forma, se retorna "Desconocido", y se considera el reconocimiento como fallido. También se considera que el reconocimiento falla si la detección no es exitosa, o el nombre retornado no se corresponde con el de la persona en la imagen, es decir, la identificación no es exitosa.

El nodo ofrece también predicción de edad y género, y el flujo de dichas predicciones puede observarse en la figura 4.4. Se observa que no es necesario realizar ambas predicciones a la vez, sino que se puede elegir cuál sistema de predicción utilizar. La solución permite predicción de características porque, al ver que la biblioteca Deepface utilizada para reconocimiento también ofrece estas predicciones, se decidió probarlas e incorporarlas a la solución. Sin embargo, este no es uno de los objetivos principales del proyecto, por lo que no se indagó sobre potenciales mejores métodos para llevar a cabo estas tareas.

Figura 4.5: (a-c) Ejemplos de reconocimiento de rostros, con el sujeto de prueba mirando en diferentes direcciones. (d) Ejemplo de predicción de edad y género para una mujer de 22 años.

La figura 4.5 demuestra el funcionamiento del nodo en tiempo real, realizando las tareas de predicción de características mencionadas y reconocimiento de rostros para una mujer de 22 años. Se observa un error de cinco años para la predicción de edad en este caso.

Si bien ya fueron descritas en el estado del arte (capítulo 2), se abordan a continuación las arquitecturas de detección y reconocimiento utilizadas con mayor profundidad. Se decide utilizar estas arquitecturas en la solución porque, según los resultados obtenidos tanto sobre conjuntos de datos como en escenarios reales, se determina que son las que más se adecúan a la RoboCup@Home de las implementaciones evaluadas. Cabe destacar que la justificación para haber experimentado sobre estas arquitecturas se describe en el capítulo 5.

4.2.2. Multi-task Cascaded Convolutional Networks (MT-CNN)

En 2016, Zhang *et al.* (K. Zhang y cols., 2016) proponen una arquitectura que realiza las tareas de alineamiento y detección de rostros en conjunto, debido a la relación inherente entre ellas. Dicha arquitectura está compuesta por tres redes neuronales convolucionales (CNNs) dispuestas en cascada, cada una correspondiéndose con una etapa del procesamiento de la imagen. Esto se puede observar en la figura 4.6. La entrada de esta cascada es un conjunto de imágenes compuesto por la imagen original en distintos tamaños. En la primera etapa (stage 1), una CNN poco profunda (P-Net) genera de forma rápida ventanas candidatas para rostros, utilizando *bounding box regression* para calibrarlas y combinando las que estén altamente superpuestas a través de *non-maximum suppression* (NMS). Luego, en la segunda etapa (stage 2) se descartan ventanas que no se correspondan con rostros a través de una CNN más compleja (R-Net), realizando también calibración a través de *bounding box regression*, y NMS para la fusión de ventanas superpuestas. Finalmente, una CNN muy similar a la anterior pero más compleja (O-Net) refina el resultado y retorna las posiciones en la imagen correspondientes a los rostros y sus características. En particular se retornan cinco posiciones por rostro: ojos izquierdo y derecho, nariz, y extremos izquierdo y derecho de la boca.

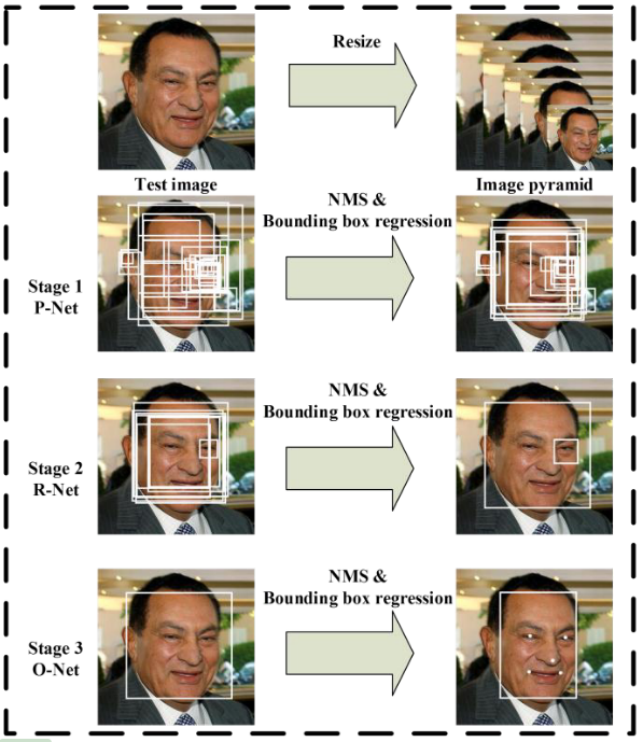


Figura 4.6: Etapas realizadas por MTCNN. Imagen extraída de Zhang *et al.* (K. Zhang y cols., 2016).

Esta arquitectura permite entonces realizar las tareas de detección y alineamiento de rostros en conjunto, a través de un diseño liviano que brinda desempeño en tiempo real. La información que retorna podría ser utilizada para tareas como identificación y verificación de rostros, por lo que es muy útil en el ámbito de reconocimiento de rostros.

4.2.3. FaceNet

Como se introdujo en la sección 2.3.3, FaceNet (Schroff y cols., 2015) es un sistema que permite representar rostros en imágenes como representaciones vectoriales en el espacio euclídeo. El objetivo es luego poder obtener la similitud entre estos rostros a partir del cálculo de una distancia entre sus representaciones. Esta distancia tiene que ser menor a cierto umbral si se trata de representaciones asociadas a una misma persona, y superarlo si se trata de distintas personas.

Figura 4.7: Funcionamiento de función de pérdida de la tripleta. Imagen extraída de Schroff *et al.* (Schroff y cols., 2015).

El sistema es entrenado utilizando una CNN, y una función de pérdida, denominada por los autores pérdida de la tripleta, que es el la base de la propuesta de FaceNet. Cada tripleta se define por una imagen ancla (del inglés *anchor*), una imagen de la misma persona como ejemplo positivo, y una imagen de una persona diferente como ejemplo negativo. En función de estas tres imágenes, FaceNet ajusta el ejemplo positivo para que su representación se acerque más a la imagen ancla que al ejemplo negativo, como se muestra en la figura 4.7, minimizando la distancia euclidiana L2 al cuadrado entre ellas. Lo que se quiere lograr entonces es que se cumpla la ecuación 4.1, donde x_i^a es la imagen ancla, x_i^p el ejemplo positivo, y x_i^n el negativo. Además, α es el margen forzado entre los positivos y negativos, T es el conjunto de todas las posibles tripletas en el conjunto de entrenamiento, con cardinalidad N, y f la función que genera una representación en base a una imagen.

$$||f(x_i^a) - f(x_i^p)||_2^2 + \alpha < ||f(x_i^a) - f(x_i^n)||_2^2 \,, \quad \forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in T \tag{4.1}$$

Según las imágenes que se encuentren disponibles en el conjunto de entrenamiento, la cantidad de posibles tripletas podría ser muy grande. Considerar todas al momento de entrenar provocaría que el entrenamiento sea un proceso muy lento. Para mejorar esto los autores proponen considerar solo tripletas difíciles o semi-difíciles, es decir, que no cumplan la ecuación 4.1, dado que las que cumplen la ecuación no contribuyen al entrenamiento y enlentecen el proceso. Una vez entrenado el sistema, se tiene una función de similitud que puede ser utilizada para verificar e identificar rostros.

En particular, la solución propuesta incluye una variante del método que utiliza representaciones de 512 bytes, utilizando el conjunto VGGFace2 (Cao, Shen, Xie, Parkhi, y Zisserman, 2018) para su entrenamiento, y la distancia euclidiana L2 para la comparación entre representaciones.

4.2.4. Parámetros de configuración

Si bien en el capítulo de experimentación (5) se fijan umbrales de tamaño del rostro, confianza y distancia, se remarca que en la solución estos son representados como parámetros en el código, por lo cual se puede cambiar su valor

de ser necesario. En el cuadro 4.2 se presentan estos umbrales, indicando sus nombres, para qué son utilizados, y sus valores por defecto.

umbral	*descripción*	*valor*
tamaño	Indica el mínimo tamaño necesario del rostro, en proporción al ancho y la altura de la imagen completa, para que una detección sea considerada como exitosa, en función del tamaño de la imagen.	1/16
confianza	Determina cuál es la mínima confianza necesaria para que una detección sea tomada como válida.	0,955
distancia	Determina cuál es la distancia mínima necesaria para que se considere que las representaciones corresponden al mismo sujeto.	1,040

Cuadro 4.2: Principales umbrales utilizados en la solución.

El umbral de tamaño del rostro se determina de forma experimental. Se prueba con distintos umbrales, considerando inversos de potencia de dos hasta 1/32. Luego, se posiciona un sujeto de prueba cerca de la cámara, y este se va alejando hasta que el sistema no logre detectarlo. Considerando las distancias a las que se especifica que se deben encontrar las personas en los experimentos de la sección 5.4, y que en la RoboCup@Home se contará con un público en el fondo de la imagen que puede actuar como ruido, se elige el valor de 1/16.

Con respecto a los umbrales de confianza y distancia, los valores por defecto indicados en el cuadro 4.2 fueron obtenidos en las pruebas (5.3.2.2) y (5.3.3) respectivamente. Se aclara que estos son los que se determinan para el método Esler, que es el que se recomienda utilizar por su desempeño obtenido en la etapa de experimentación. Para el caso de Deepface, el valor del umbral de distancia coincide con el de Esler, y el de confianza es 0,926.

A su vez, se tienen otros parámetros que puede llegar a ser interesante modificar. Por ejemplo, se puede determinar si se quiere que el modelo realice reconocimiento de rostros o predicción de características, de entre las cuales se puede elegir género y edad. También, se puede elegir si se quiere que el modelo construya una representación por persona o por cada imagen asociada a esa persona. Asimismo, se puede elegir el método a utilizar, que puede ser la implementación de Deepface o la de Esler.

Capítulo 5

Experimentación

Se tiene como objetivo del proyecto elegir qué algoritmo utilizar para implementar el nodo de ROS para reconocimiento de rostros. Es necesario entonces llevar a cabo un conjunto de experimentos sobre los posibles algoritmos para determinar cuál se adecúa mejor al objetivo del proyecto. Para ello, primero se realiza una etapa de pruebas experimentales utilizando conjuntos de imágenes públicos. Luego, se realizan pruebas en escenarios reales, haciendo uso de una cámara y corriendo el sistema en tiempo real. Para este último caso, se usa un conjunto de datos construido a medida con imágenes solicitadas a sujetos de prueba.

Las distintas tareas llevadas a cabo en cada etapa de experimentación se pueden apreciar en la figura 5.1. Las figuras verde y azul representan las dos grandes etapas de experimentación que se llevan a cabo, comentadas anteriormente. Luego, se descomponen estas etapas en distintas tareas (representadas en lila) y para cada tarea se especifican hitos (representados en amarillo).

Se observa que si bien se especifican pruebas de predicción de características en escenarios reales, estas no fueron ejecutadas, por lo que no se incluyen en el diagrama. La especificación de estas pruebas se puede encontrar en el anexo B.

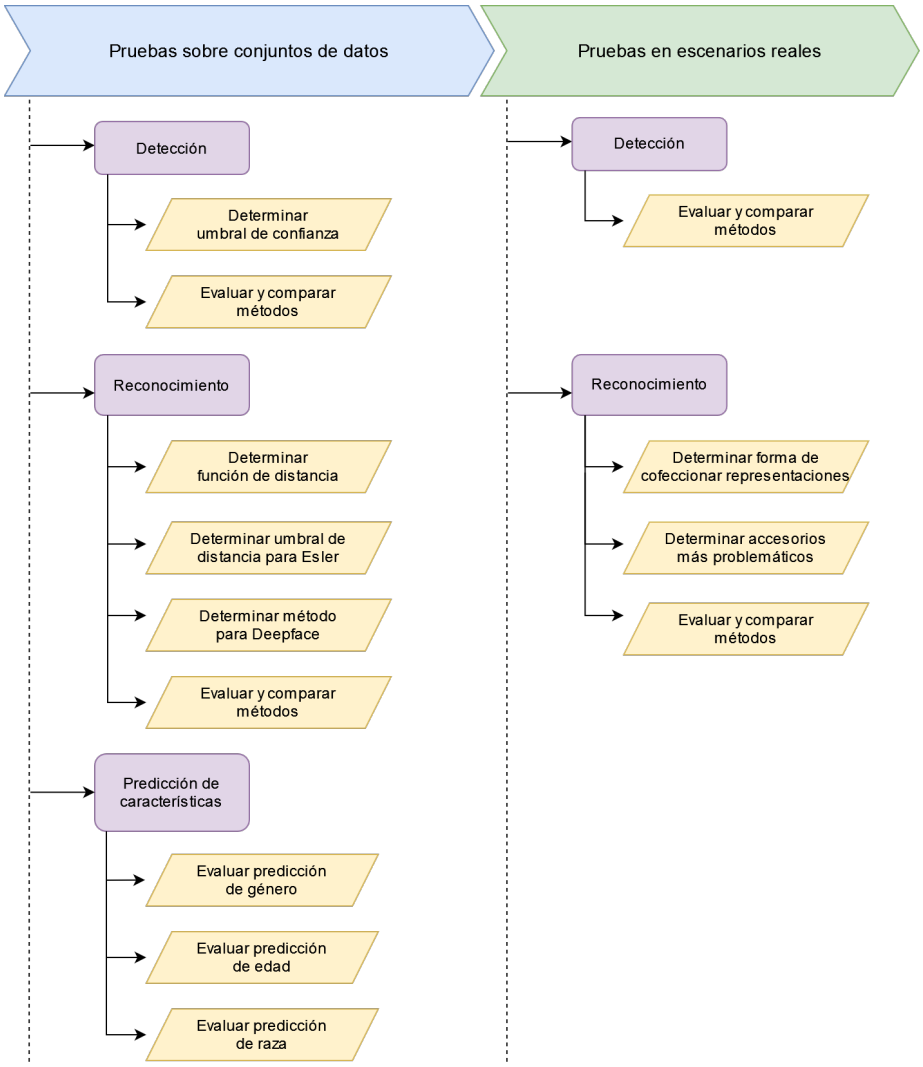


Figura 5.1: Esquema de las etapas que componen la fase de experimentación.

En este capítulo, primero se presentan los métodos a evaluar, así como las medidas de desempeño que se utilizan para hacerlo. El resto del capítulo se divide en dos secciones. En la primera, se especifican pruebas de detección, verificación y predicción de características sobre los conjuntos de datos, y se describen los conjuntos utilizados. En la segunda, se indican cuáles son las condiciones necesarias para ejecutar un conjunto de experimentos en escenarios reales, luego se establecen dos escenarios de prueba basados en las pruebas de la RoboCup@Home (Matamoros, Rascon, y cols., 2019), y se especifican los expe-

rimentos en base a ellos. La principal diferencia entre los dos escenarios es que en el primero se tiene una única persona en el campo de visión, y en el segundo se tienen múltiples personas. En el primer escenario, se realizan experimentos relativos a la detección de rostros, determinación de forma de construcción de representaciones, reconocimiento de rostros e incorporación de una nueva persona a la base de datos para su posterior reconocimiento. Por otro lado, en el segundo escenario se realiza una prueba para medir el desempeño del modelo reconociendo múltiples rostros.

5.1. Métodos a evaluar

Las pruebas se realizan sobre dos implementaciones basadas en redes neuronales. La primera es una implementación de FaceNet en PyTorch, denominada Esler (Esler, Tim, 2021) de ahora en más en referencia a su autor. Se remarca que FaceNet propone un modelo teórico, pero no presenta una implementación oficial. La segunda implementación es la provista por la biblioteca Deepface (Serengil y Ozpinar, 2021), que no se debe confundir con la arquitectura DeepFace presentada en el capítulo 2.

Se elige trabajar con estas implementaciones porque reportan resultados en GitHub (*GitHub*, 2022) a nivel del estado del arte, y se basan en métodos analizados en el capítulo 2. A su vez, estas implementaciones cuentan con más de tres mil estrellas y 600 bifurcaciones (del inglés *fork*) en GitHub, lo que las sitúa en una buena posición en comparación con otras implementaciones consideradas, dado que implica que tienen un mantenimiento y evaluación actualizada.

La biblioteca Deepface se destaca por ofrecer múltiples métodos para el reconocimiento de rostros. Para la identificación y verificación no solo ofrece una implementación de la arquitectura DeepFace (Taigman y cols., 2014), sino que también ofrece los métodos FaceNet (Schroff y cols., 2015) y ArcFace (Deng y cols., 2019) entre otros. También ofrece diversos métodos de detección, entre los cuales se encuentra MTCNN (K. Zhang y cols., 2016). Si bien la biblioteca también ofrece otros métodos, no se encuentran dentro de los estudiados en etapas anteriores del proyecto y, en general, reportan resultados inferiores en comparación con las otras propuestas.

Por otro lado, Esler incluye para la detección de rostros una implementación de MTCNN, y para la identificación y verificación una de FaceNet.

Con respecto a las tareas de predicción de edad, género y raza, la biblioteca Deepface ofrece una funcionalidad *analyze* que detecta estas características, utilizando redes neuronales con capas convolucionales, *Max Pooling* para el submuestreo, y la función de activación *softmax*. Dado que el método Esler no permite realizar predicción de características, para esta tarea se utiliza únicamente la biblioteca Deepface.

5.2. Medidas de desempeño

Las métricas que se utilizan para evaluar el desempeño en las tareas son la precisión, recuperación, exactitud, error absoluto medio (MAE), medida F1 y precisión promedio, del inglés *precision*, *recall*, *accuracy*, *mean absolute error*, *F1 score* y *average precision (AP)* respecivamente. Estas métricas se definen en el cuadro 5.1, y sus ecuaciones son las siguientes:

medida	*definición*
Precisión ↑	Proporción de las instancias que se etiquetan como positivas que realmente lo son (ecuación 5.1)
Recuperación ↑	Proporción de las instancias realmente positivas que son etiquetadas como positivas (ecuación 5.2)
Exactitud ↑	Proporción de las instancias clasificadas correctamente de todas las instancias (ecuación 5.3)
MAE ↓	Promedio de las diferencias entre los valores reales y los predichos (ecuación 5.4)
Medida F1 ↑	Medida que permite evaluar el desempeño combinando la precisión y recuperación (ecuación 5.5)
Medida AP ↑	Área bajo la curva PR que representa el balance entre precisión y recuperación (ecuación 5.6)

Cuadro 5.1: Definición de las medidas utilizadas para evaluar el desempeño en las tareas asociadas al reconocimiento de rostros. Todas las medidas toman valores porcentuales del 0 al 100, a excepción del MAE, cuyo valor mínimo es 0 y no tiene un valor máximo.

$$precisión = \frac{TP}{TP + FP} \quad (5.1) \qquad recuperación = \frac{TP}{TP + FN} \quad (5.2)$$

$$exactitud = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.3)$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \quad (5.4)$$

$$medida\ F1 = 2 * \frac{precisión * recuperación}{precisión + recuperación} \quad (5.5)$$

$$medida\ AP = \int_0^1 p(r)dr \quad (5.6)$$

En las ecuaciones anteriores, las instancias TP son las que el algoritmo clasifica como positivas y acierta, TN las que clasifica como negativas y acierta, FP las que son clasificadas como positivas pero no lo son, y FN las clasificadas como negativas siendo en realidad positivas. En particular, la métrica presentada en la ecuación 5.4 se utiliza para la prueba de predicción de edad, donde N representa la cantidad de individuos para los cuales se predice la edad, $\hat{y}_i$ representa la predicción arrojada por el algoritmo, y y_i la edad etiquetada del individuo. Con respecto a la ecuación 5.6, p hace referencia a la precisión y r a la recuperación.

Para evaluar la tarea de detección se utiliza la curva PR (Precisión-Recuperación) (Yohanandan, Shivy, 2020). Esta curva grafica la precisión en función de la recuperación. Para generar cada uno de los puntos de la curva, lo que se hace es ir variando un umbral para el valor de confianza de las regiones detectadas. Es decir, en cada punto de la gráfica se toman como clasificaciones positivas solo aquellas que tengan un valor de confianza mayor a determinado umbral. Cuanto más pequeño es este valor, mayor es la recuperación, ya que se aceptan regiones de casi cualquier valor de confianza, pero la precisión es más baja, ya que se aceptan predicciones con muy poca confianza, que puede que sean falsos positivos. La curva representa entonces el balance que hay entre precisión y recuperación. Esta curva permite elegir el valor de corte óptimo para el valor de confianza de acuerdo a la importancia que se le de a la recuperación y precisión. A su vez, en base a la curva se puede calcular la métrica AP, que se corresponde con el área bajo dicha curva. Cuanto más alto sea su valor, se puede obtener un mejor balance entre recuperación y precisión ajustando el umbral de confianza, resultando en un mayor resultado para ambas métricas.

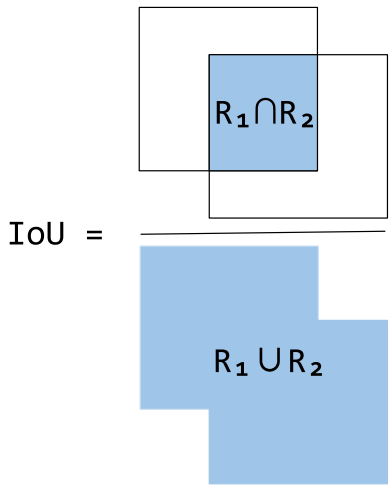


Figura 5.2: Visualización gráfica de la medida intersección sobre unión (IoU)

Para calcular los valores de precisión y recuperación presentes en las curvas

PR, se utiliza la métrica intersección sobre unión (del inglés *intersection over union* (IOU)). Esta medida se formula en la figura 5.2, donde R_1 representa la región de la imagen donde se detecta el rostro, y R_2 representa la región donde realmente está el rostro a detectar. Si dicha medida supera cierto umbral, la detección es considerada como exitosa.

5.3. Pruebas sobre conjuntos de datos

Esta sección está organizada de la siguiente forma. Primero, se describen los conjuntos de datos utilizados para la evaluación. Luego, se especifican pruebas de detección y verificación, donde se evalúa el desempeño de los modelos. Además, en el caso de la detección se determina el umbral de confianza asociado, y en el caso de la verificación se determina el umbral de distancia para Esler, el método de reconocimiento para Deepface, y la función de distancia para cada uno de ellos. Finalmente, se especifican pruebas de predicción de edad, género y raza. Para cada prueba se especifica su objetivo, desarrollo, y los resultados obtenidos por cada modelo junto a un análisis de estos.

5.3.1. Conjuntos de datos utilizados

Para evaluar la tarea de detección se utiliza el conjunto de datos WIDER Face (Yang y cols., 2016), dado que MTCNN, el modelo a evaluar, reporta resultados sobre este conjunto de datos. También, dado que FaceNet reporta resultados en LFW (Huang y cols., 2008), se utiliza dicho conjunto de imágenes para evaluar la identificación. En cuanto a la predicción de características, se busca utilizar un único conjunto de datos que permita evaluar raza, edad y género. Se decide entonces trabajar con UTKFace (Z. Zhang, Song, y Qi, 2017), dado que este se encuentra anotado con dicha información. Cabe destacar que LFW y UTKFace son de los conjuntos más utilizados en el área de reconocimiento y predicción de edad respectivamente, según lo reportado en (*The latest in Machine Learning — Papers With Code*, s.f.) y (*Kaggle*, s.f.). Los conjuntos mencionados incluyen rostros en diferentes escenarios y calidades (como tamaño e iluminación) lo que permite probar la robustez de los algoritmos frente a estas variaciones para cada una de las tareas. En esta sección se presenta una breve descripción de cada uno.

WIDER Face es un conjunto de datos que toma 32303 imágenes del conjunto WIDER (un conjunto de imágenes para el reconocimiento de eventos) y registra las regiones donde aparecen rostros, lo que permite evaluar el desempeño de modelos de detección de rostros. Las imágenes que se seleccionan incluyen desde imágenes en conciertos, eventos públicos y deportivos (que pueden incluir decenas de rostros), hasta imágenes en ambientes familiares y laborales (que suelen incluir una cantidad reducida de rostros). Esto puede observarse en la figura 5.3. El conjunto de datos se subdivide en tres conjuntos: uno fácil, uno de dificultad media y uno difícil. En la figura se muestran los distintos tipos de variaciones que incluye el conjunto, siendo esto escala, pose, oclusión, exposición,

iluminación y maquillaje (del inglés *scale*, *pose*, *occlusion*, *exposure*, *ilumination* y *makeup* respectivamente). Es en base a la notoriedad de estas variaciones en cada imagen que se define su grado de dificultad.

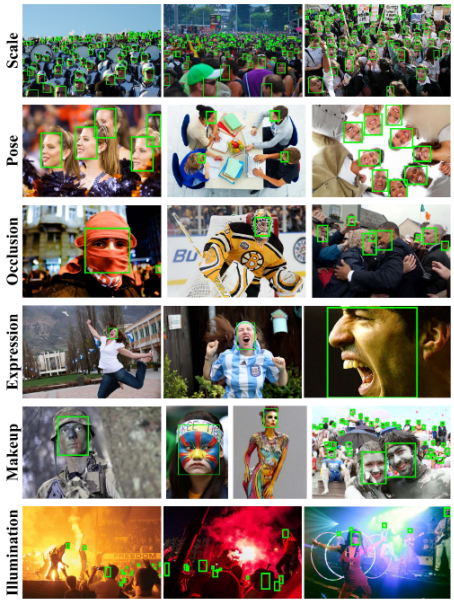


Figura 5.3: Conjunto de imágenes de muestra de WIDER Face (Yang y cols., 2016), que muestra los distintos tipos de variaciones que incluye el conjunto completo.

Figura 5.4: Conjunto de imágenes de muestra de LFW (Huang y cols., 2008). Los rectángulos amarillos representan los cuadros delimitadores del rostro para una misma persona.

LFW (Labeled Faces in the Wild) es una colección de imágenes de rostros recopilada en internet, conformada por más de 13000 imágenes. Cada imagen está anotada con el nombre de la persona que aparece en ella, y varias de las personas que se encuentran en la colección tienen dos o más imágenes de

su rostro. Es por esto que se puede utilizar para evaluar el desempeño de un modelo en tareas de identificación y verificación. Se muestran algunas de estas imágenes en la figura 5.4. En particular, se debe destacar que este conjunto se utiliza a través de la biblioteca *scikit-learn* (Pedregosa y cols., 2011), que agrupa las imágenes del conjunto en mil emparejamientos.

UTKFace es un conjunto de datos de más de 20000 imágenes anotadas indicando la raza, género y edad de la persona que aparece en ella, pudiéndose observar en la figura 5.5 algunas de las imágenes que lo componen. Esto permite evaluar el desempeño de modelos que predicen estas características. Se debe resaltar que, dado que se busca utilizar la biblioteca Deepface para la RoboCup@Home, se considera que los sujetos de prueba no serán menores a 15 años y no excederán los 65. Se restringe el conjunto de datos en base a este rango, debido a que el original contiene imágenes de personas de 0 a 120 años aproximadamente, lo que se considera un rango muy amplio para el uso práctico que se le quiere dar. El rango utilizado se determina empíricamente, en base a la experiencia de participantes de la competencia en ediciones anteriores, dado que no se encuentra información estadística sobre los participantes de la RoboCup@Home o restricciones de este tipo para ser voluntario en una prueba. En la figura 5.6 se grafican las proporciones del conjunto de datos obtenido.

Figura 5.5: Conjunto de imágenes de muestra de UTKFace (Z. Zhang y cols., 2017).

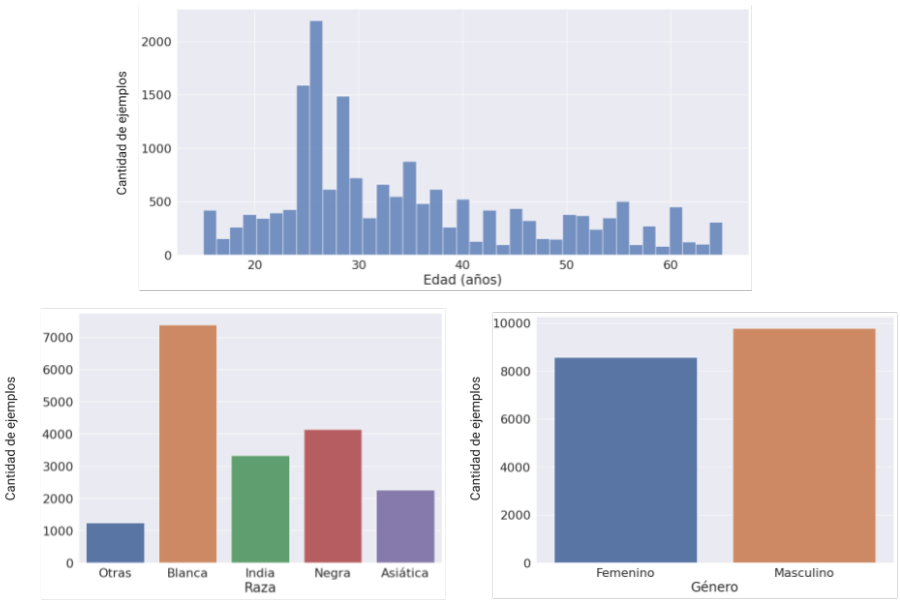


Figura 5.6: Proporciones por edad (en años), género y raza del conjunto de datos UTKFace (Z. Zhang y cols., 2017) restringido a las edades de 15 a 65.

Además, el conjunto de datos utilizado y la biblioteca Deepface trabajan con diferentes razas. Si bien ambas comparten las razas *blanca*, *negra*, *asiática* e *india* (del inglés *white*, *black*, *asian* e *indian*, respectivamente), la biblioteca también puede predecir razas como *de Medio Oriente* e *hispano latino* (del inglés *middle eastern* y *latino hispanic*) que no se encuentran explícitamente en el conjunto de datos. De todas formas, el conjunto tiene una clase *otros* (del inglés *others*) que contempla estas razas. Es por esto que se unifican las razas *de Medio Oriente* e *hispano latino* en *otros*.

Para probar los modelos, se generan tres subconjuntos seleccionando instancias al azar del conjunto de datos original buscando que se conserve la proporción de un atributo (género, edad o raza) en cada uno de los subconjuntos. Para el caso de la edad, se consideran intervalos de cinco años y se busca que haya una proporción similar en cada intervalo. Se observa que cada uno de los subconjuntos representa aproximadamente un 20 % del conjunto original. Esto se realiza para ver cómo cambian las medidas ante conjuntos balanceados según distintos atributos, y para poder identificar posibles sesgos en el modelo. Por último, dado que ya se calculan los resultados para tres subconjuntos diferentes (pero no disjuntos), se unifican los tres subconjuntos para tener un resultado adicional. Las proporciones por edad, género y raza de cada uno de estos subconjuntos pueden observarse en el apéndice B.

5.3.2. Pruebas de detección de rostros

En esta sección se presentan las distintas pruebas realizadas sobre las implementaciones de MTCNN propuestas por Deepface y Esler.

Los métodos de detección reportan tanto la predicción de áreas donde encuentran rostros como un valor de la confianza asociada a cada una de estas predicciones. Al momento de efectuar la detección de rostros, es habitual tener un umbral de confianza utilizado para filtrar las predicciones en base a los valores de confianza reportados por los modelos. Debido a esto, las pruebas presentadas a continuación se diferencian en pruebas de desempeño de los métodos, y pruebas para determinar este umbral de confianza para cada uno de los métodos.

5.3.2.1. Evaluación de desempeño

Objetivo:

- Evaluar el desempeño de las implementaciones detectando rostros en una imagen.

Desarrollo de la prueba:

Para el caso de la biblioteca Deepface, se utiliza la función *detect_faces* sobre cada imagen del conjunto de datos. Esta función retorna los rostros detectados en caso de detectar alguno. En el caso de Esler, al instanciar su modelo de MTCNN este retorna una función que es luego utilizada para obtener los rostros detectados, de forma análoga a la que ofrece Deepface.

En ambos casos se verifica que los rostros hayan sido detectados correctamente calculando la medida intersección sobre unión, utilizando un umbral de 0,5 para considerar la detección exitosa de un rostro. Esto se debe a que se evalúa sobre el conjunto de datos WIDER Face (Yang y cols., 2016), que define un umbral de 0,5 para la medida IoU. Se obtienen luego las curvas PR de las implementaciones. También se muestran las curvas PR de otros métodos a modo de referencia, en las figuras 5.7, 5.8 y 5.9. Se aclara que se utiliza el subconjunto de validación de WIDER Face para evaluar los métodos, dado que el subconjunto de evaluación no se encuentra disponible para utilizar libremente.

Resultados obtenidos:

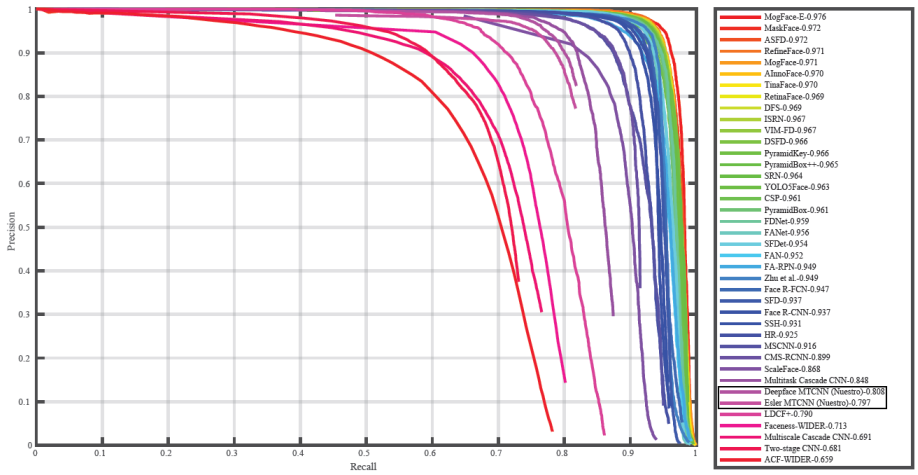


Figura 5.7: Resultados de la curva PR para Esler y Deepface sobre la partición fácil del subconjunto de validación de WIDER Face, comparándolos con otros métodos. Las curvas se obtienen con una secuencia de comandos provista por WIDER Face.

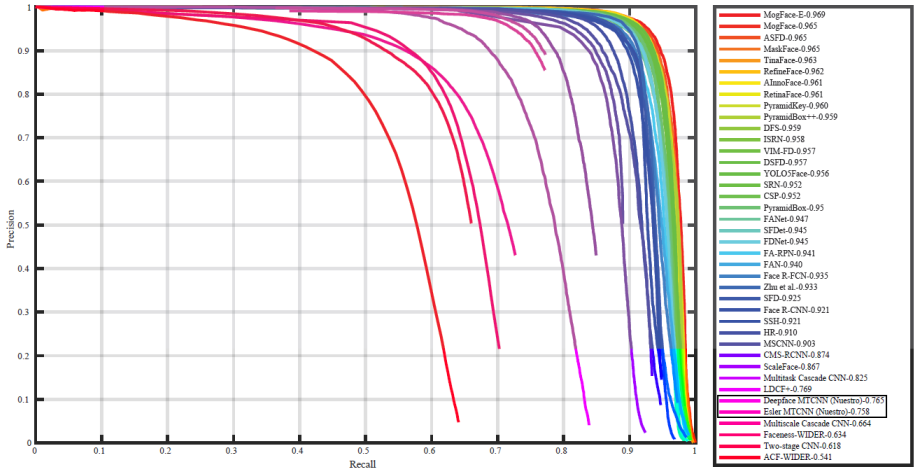


Figura 5.8: Resultados de la curva PR para Esler y Deepface sobre la partición media del subconjunto de validación de WIDER Face, comparándolos con otros métodos. Las curvas se obtienen con una secuencia de comandos provista por WIDER Face.

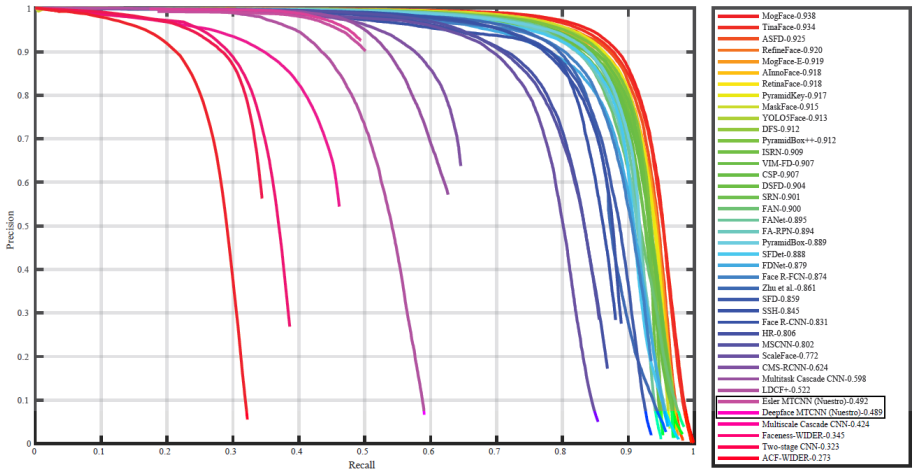
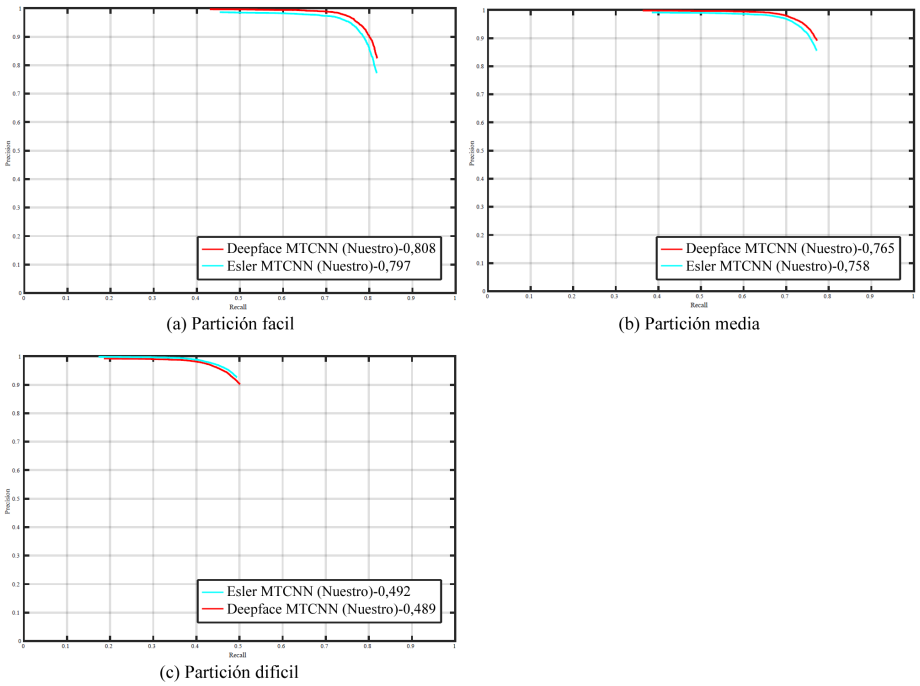


Figura 5.9: Resultados de la curva PR para Esler y Deepface sobre la partición difícil del subconjunto de validación de WIDER Face, comparándolos con otros métodos. Las curvas se obtienen con una secuencia de comandos provista por WIDER Face.



(a) Partición facil

(b) Partición media

(c) Partición dificil

Figura 5.10: Resultados de la curva PR para Esler y Deepface sobre las tres particiones del subconjunto de validación de WIDER Face. Las curvas se obtienen con una secuencia de comandos provista por WIDER Face.

Análisis de resultados:

Primeramente, con respecto a los resultados presentados en la figura 5.10, se observa que si bien el método de detección utilizado por Deepface tiene resultados de métrica AP algo superiores para la mayoría de los casos, ambos reportaron resultados muy similares.

Además, en las figuras 5.7, 5.8 y 5.9 se observa que comparados con otros resultados, los dos métodos evaluados tienen un valor de AP considerablemente menor a la mayoría de los otros métodos. Se cree que esto puede deberse a que los demás modelos entrenen y evalúen sobre particiones del mismo conjunto de datos (WIDER Face), mientras que los métodos que se están evaluando tomen un conjunto de datos diferente como base para el entrenamiento.

5.3.2.2. Determinación del umbral

Objetivo:

- Determinar el umbral de detección a utilizar.

Desarrollo de la prueba:

En base a los resultados obtenidos en la prueba anterior, se quiere determinar cuál es el punto de corte óptimo para determinar si una detección de un rostro se considera como exitosa.

Para esto se grafican las curvas PR utilizando la función *precision_recall_curve* de la biblioteca *scikit-learn*, la cual retorna diferentes umbrales de confianza, así como las precisiones, y recuperaciones resultado de utilizar dichos umbrales. Utilizando esta información, se prueba con distintos umbrales, realizando marcas en las curvas para poder observar la precisión y recuperación obtenidas para cada umbral, y se eligen dos marcas candidatas para cada método.

Resultados obtenidos:

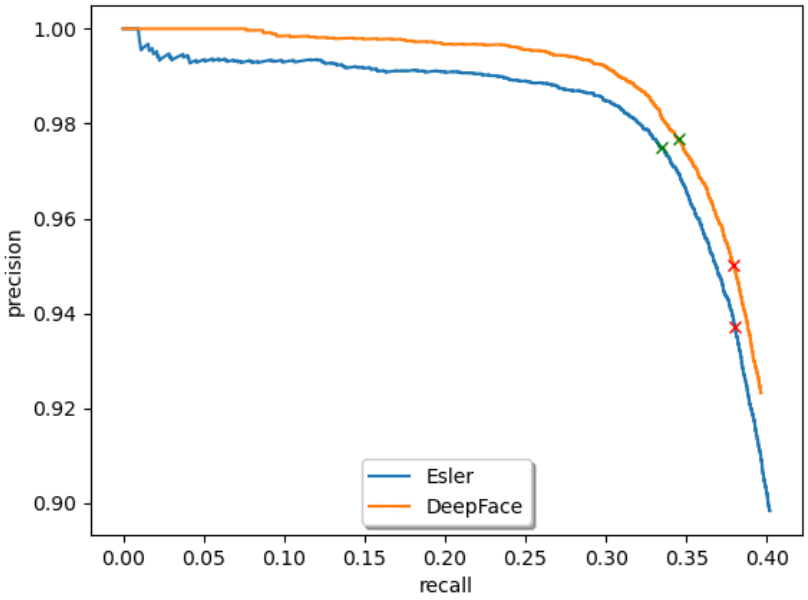


Figura 5.11: Resultados de la curva PR para Esler y Deepface sobre el subconjunto de validación de WIDER Face completo, marcando con cruces puntos candidatos para el umbral de detección.

Las marcas que se presentan en la figura 5.11 corresponden a los umbrales especificados en el cuadro 5.2.

	Esler	Deepface
Marcas rojas	0.955	0.926
Marcas verdes	0.826	0.808

Cuadro 5.2: Umbrales de corte para los distintos métodos según la marca.

Análisis de los resultados:

Las marcas realizadas en color verde tienen como objetivo encontrar el punto donde se comprometan menos ambas medidas, e implican la definición de un umbral de confianza para cada método. Dado que los valores de recuperación obtenidos utilizando dichos umbrales son considerablemente bajos (menores a 0,5), se consideró utilizar umbrales de menor valor, sacrificando la precisión en beneficio de la recuperación. Estos umbrales se corresponden a las marcas de color rojo en la figura 5.11.

Sin embargo, se debe tomar en cuenta que el conjunto de datos incluye imágenes con rostros muy pequeños (por ejemplo, decenas de personas en un desfile), que deben perjudicar significativamente la recuperación. Estos escenarios se alejan de los casos de uso de la RoboCup@Home y ambientes domésticos, donde generalmente las personas se encuentran más cerca de la cámara, y por lo tanto detectar un rostro en la imagen no es tan complejo como en dichos escenarios. De todas formas, esto es evaluado más adelante en las pruebas en escenarios reales, donde los escenarios se asemejan más a aquellos de la RoboCup@Home. Considerando esto último, y que en la competencia es probable que haya ruido en la imagen debido al público presente, se decide no sacrificar la precisión, y utilizar los umbrales correspondientes a las marcas verdes.

5.3.3. Prueba de verificación de rostros

Objetivo:

- Evaluar el desempeño de las implementaciones al verificar si pares de rostros pertenecen a la misma persona, y determinar los mejores valores para los parámetros de dichas implementaciones.

Desarrollo de la prueba:

El conjunto de datos LFW provee una partición de sus imágenes en pares etiquetados, lo que facilita la evaluación de la verificación de rostros. Para cada uno de estos pares se verifica si los rostros pertenecen a la misma persona, y se evalúa si el método acertó o no en base a las etiquetas provistas.

En el caso de la biblioteca Deepface, la verificación se hace utilizando la función *verify*, ofrecida por la propia biblioteca. Los modelos que se utilizan son los siguientes: DeepFace, ArcFace y FaceNet. De este último modelo, se utilizan las versiones que generan representaciones vectoriales de 128 y 512 dimensiones, que son las que ofrece la biblioteca. Además, la función *verify* puede ser invocada con distintas medidas de distancia: euclidiana, euclidiana L2 y distancia

coseno. Se realizan pruebas con cada una de estas distancias para determinar cuál reporta mejores resultados.

En el caso de Esler, la implementación ofrece un modelo InceptionResnetV1 preentrenado. Al instanciarlo, aplicándole *eval()* este retorna una función que dada una imagen de un rostro calcula su representación vectorial. Dado que al instanciarlo se puede configurar el conjunto de entrenamiento, que puede ser VGGFace2 (Cao y cols., 2018) o CASIA-Webface (Yi, Lei, Liao, y Li, 2014), se realizan pruebas sobre ambas versiones. Luego, para realizar la verificación se calcula la distancia entre las representaciones, utilizando la distancias euclidiana L2 y coseno.

Se observa que para las pruebas de Esler, también se quiere determinar cuál es el umbral de distancia óptimo. Es decir, cuando se comparan dos representaciones, cuál es la distancia máxima óptima para considerar que corresponden a una misma persona. Se tienen como referencia los umbrales de la biblioteca Deepface, y se prueba con distintos valores en un entorno cercano a estos. Los umbrales de esta biblioteca son 1,04 para la distancia euclidiana L2 y 0,30 para la distancia coseno.

Las distancias mencionadas previamente se definen en las ecuaciones 5.7, 5.8 y 5.10, donde P y Q son los vectores cuya distancia se pretende calcular, N su longitud, y p_i y q_i los elementos i de los vectores P y Q respectivamente. En particular, la distancia euclidiana L2 se corresponde a la distancia euclidiana, pero normalizando previamente cada vector con la norma L2 (formulada en la ecuación 5.9).

$$d_{euclidiana}(P,Q) = \sqrt{\sum_{i=1}^{N}(p_i - q_i)^2} \tag{5.7}$$

$$d_{cos}(P,Q) = \frac{P \cdot Q}{\|P\|_2 \, \|Q\|_2} = \frac{\sum_{i=1}^{N} P_i Q_i}{\sqrt{\sum_{i=1}^{N} P_i^2}\sqrt{\sum_{i=1}^{N} Q_i^2}} \tag{5.8}$$

$$norma_{L2}(X) = \sqrt{\sum_{k=1}^{N}(X_k)^2} \tag{5.9}$$

$$d_{L2}(P,Q) = d_{euclidiana}(norma_{L2}(P), norma_{L2}(Q)) \tag{5.10}$$

Resultados obtenidos:

Distancia	Umbral	Exactitud ↑	Rec. ↑	Prec. ↑	F1 ↑
Coseno	0,1	53,1	6,2	**100**	11,7
	0,2	68,1	36,2	**100**	53,2
	0,3	85,5	71,0	**100**	83,0
	0,4	94,0	88,0	**100**	93,6
	0,5	97,2	95,0	99,4	97,1
	0,6	**97,4**	96,6	98,2	**97,4**
	0,7	95,2	**96,8**	93,4	95,3
L2	0,8	91,1	82,2	**100**	90,2
	0,9	95,9	91,8	**100**	95,7
	1,04	**97,5**	95,8	99,2	**97,5**
	1,1	96,5	**96,8**	96,2	96,5
	1,2	92,9	**96,8**	89,8	93,2

Cuadro 5.3: Resultados (%) de exactitud, recuperación, precisión y medida F1 para el modelo entrenado con el conjunto VGGFace2 de Esler, utilizando las funciones de distancia coseno y euclidiana L2, y variando el umbral de distancia. Para cada distancia se resalta en negrita el mejor resultado de cada medida, y para cada medida se indica con una flecha si es preferible un valor mayor o menor.

Distancia	Umbral	Exactitud ↑	Rec. ↑	Prec. ↑	F1 ↑
Coseno	0,1	50,3	6,0	**100**	1,2
	0,2	55,9	11,8	**100**	21,1
	0,3	69,8	39,6	**100**	56,7
	0,4	84,4	68,8	**100**	81,5
	0,5	91,5	83,4	99,5	90,8
	0,6	**95,6**	92,4	98,7	**95,5**
	0,7	94,5	**95,4**	93,7	94,5
L2	0,8	78,7	57,4	**100**	72,9
	0,9	88,4	76,8	**100**	86,9
	1,04	93,2	87,2	99,1	92,8
	1,1	**95,5**	93,8	97,1	**95,4**
	1,2	92,0	**96,8**	88,3	92,4

Cuadro 5.4: Resultados (%) de exactitud, recuperación, precisión y medida F1 para el modelo entrenado con el conjunto CASIA-Webface de Esler, utilizando las funciones de distancia coseno y euclidiana L2, y variando el umbral de distancia. Para cada distancia se resalta en negrita el mejor resultado de cada medida, y para cada medida se indica con una flecha si es preferible un valor mayor o menor.

Modelo	Distancia	Exactitud ↑	Rec. ↑	Prec. ↑	F1 ↑
ArcFace	Coseno	**94,4**	**89,8**	98,9	**94,1**
	Euclidiana	79,8	62,6	95,4	75,6
	L2	93,2	87,2	**99,1**	92,8
DeepFace	Coseno	62,7	34,4	79,3	48,0
	Euclidiana	**66,6**	**64,0**	69,5	**65,7**
	L2	59,5	23,6	**83,7**	36,8
FaceNet-128	Coseno	**88,2**	**76,4**	**100**	**86,6**
	Euclidiana	80,1	60,4	99,7	75,2
	L2	78,2	56,4	**100**	72,1
FaceNet-512	Coseno	79,6	59,2	**100**	74,3
	Euclidiana	96,0	92,2	99,8	95,8
	L2	**97,6**	**96,0**	99,2	**97,6**

Cuadro 5.5: Resultados (%) de exactitud, recuperación, precisión y medida F1 para el método Deepface, utilizando las arquitecturas ArcFace, DeepFace, FaceNet-128 y FaceNet-512, y las funciones de distancia coseno, euclidiana y euclidiana L2. Para cada modelo se resalta en negrita el mejor resultado de cada medida, y para cada medida se indica con una flecha si es preferible un valor mayor o menor.

Análisis de los resultados:

En el caso de Esler, los mejores resultados se obtienen al utilizar el conjunto de datos de entrenamiento VGGFace2, y la distancia euclidiana L2 con un umbral de 1,04, como puede observarse en los cuadros 5.3 y 5.4. En este caso, el umbral que utiliza la biblioteca Deepface es el que arroja los mejores resultados. La segunda mejor combinación de parámetros se obtiene en este mismo conjunto de datos, utilizando la distancia coseno con un umbral de 0,6, e incluso supera el valor de recuperación de la primera configuración presentada. En este caso, el valor del umbral se aleja considerablemente del utilizado en la biblioteca Deepface (0,3). Finalmente, se decide seleccionar la primera, dado que es la que produce una mejor exactitud y medida F1, contemplando esta última tanto a la precisión como a la recuperación.

Se observa en el cuadro 5.5 que para Deepface, los mejores valores reportados se obtienen utilizando el modelo FaceNet-512, y la distancia euclidiana L2. Los resultados de este modelo son en general muy superiores a los demás. Es por esto que se decide utilizar esta combinación de modelo y distancia en las pruebas posteriores donde se utilice la biblioteca Deepface.

5.3.4. Pruebas de predicción de características

En esta subsección se especifican las pruebas asociadas a la predicción de características, ejecutadas con la función *analyze* de Deepface.

Se recuerda que se restringe el conjunto de datos de UTKFace para considerar únicamente personas en el rango etario de 15 a 65 años, y se generan

tres subconjuntos. El primero conserva la proporción de género, el segundo la proporción de raza y el tercero la proporción de rango etario. Se generan estos tres subconjuntos para ver cómo se comporta el modelo frente al balanceo por distintos atributos. Se reporta solamente el mejor y el peor resultado para no saturar con datos. Se puede acceder a los resultados completos en el apéndice B.

Se presentan además algunos resultados de referencia, pero no se podrían comparar de manera justa debido a que o evalúan sobre otro conjunto de datos, o no consideran las restricciones planteadas. Estos resultados incluyen los obtenidos por Sefik Ilkin Serengil, autor de la biblioteca Deepface, que evalúa la predicción de edad y género sobre IMDB-WIKI (Rothe, Timofte, y Gool, 2018), y la de raza sobre Fairface (Karkkainen y Joo, 2021).

5.3.4.1. Prueba de predicción de edad

Objetivo:

- Evaluar el desempeño del algoritmo al determinar la edad de las personas presentes en la imagen.

Desarrollo de la prueba:

Para cada imagen del conjunto de datos, se aplica la función *analyze* de Deepface, comparando la predicción retornada con la edad etiquetada de la persona. En base a esta comparación se calcula el MAE.

Resultados obtenidos y análisis de los resultados:

El mejor MAE que obtiene el modelo es de 8,0, utilizando el conjunto de datos balanceado por raza, mientras que el peor es de 12,2, que se da con el conjunto balanceado por edad. Se cree que esto último puede deberse a que, como el conjunto de entrenamiento (IMDB-WIKI) cuenta con más instancias de entre 20 y 40 años, pudo generar que el modelo aprenda a reconocer mejor a las personas de este rango etario. Dado que el conjunto recortado también tiene una concentración de personas en este rango, al forzar que deba existir la misma proporción para los distintos intervalos de edad se consideran menos personas de ese rango, y se cree que es por esto que los resultados son peores. En cambio, que el mejor resultado se de en el subconjunto balanceado por raza puede deberse a que este presenta mayor cantidad de gente de entre 20 y 40 años.

Si bien el conjunto sobre el cual se evalúa el modelo de Deepface se encuentra restringido, y por lo tanto no se lo podría comparar de forma justa con otros trabajos que evalúen sobre UTKFace, se presentan los resultados obtenidos por los siguientes trabajos a modo de referencia. Considerando los resultados reportados por Serengil, él obtiene un MAE de 7,88 con un modelo basado en *keras* (keras, 2022) (que es el que se encuentra en la biblioteca), lo cual no se aleja del resultado obtenido en el marco de este proyecto. Otro resultado interesante es el reportado por Berg *et al.* (Berg, Oskarsson, y O'Connor, 2021), quienes logran un MAE de 4,55, estando este valor al nivel del estado del arte. En este caso, el resultado presentado en este trabajo se aleja significativamente, siendo

el mejor resultado casi 1,76 veces peor. Además, (Geng, Smith-Miles, y Zhou, 2008) realizaron un estudio donde las personas debían predecir la edad observando imágenes de otras personas. Este estudio resultó en un MAE de 6,13, lo que es algo mejor que los resultados obtenidos.

De todas formas, los valores de MAE obtenidos indican que el modelo en general es capaz de distinguir correctamente entre sujetos de distintas generaciones, considerando una separación entre generaciones de alrededor de 25 años. Teniendo en cuenta que el objetivo de este proyecto es presentar una solución que se adapte a las pruebas de la RoboCup@Home, y que en la prueba asociada a la predicción de edad (*Recepcionista* 4.1) se pide solo determinar la persona de mayor edad, se considera que el modelo podría ser útil para cumplir el objetivo en caso de que la diferencia de edad entre dicha persona y el resto sea notoria.

5.3.4.2. Prueba de predicción género

Objetivo:

- Dado un rostro ya detectado, evaluar el desempeño del algoritmo al predecir el género de la persona asociada a él.

Desarrollo de la prueba:

Para cada imagen del conjunto de datos, se aplica la función *analyze* de la biblioteca Deepface, registrando el valor del género que esta predice. Se computa luego la exactitud del algoritmo, así como la precisión y recuperación para cada género.

Resultados obtenidos:

	Prec. ↑		Rec. ↑		Exac. ↑	
Género	Mejor	Peor	Mejor	Peor	Mejor	Peor
Masculino	72,3	66,2	99,3	98,9	77,6	74,5
Femenino	98,4	97,8	51,2	47,6		

Cuadro 5.6: Resultados (%) de precisión, recuperación y exactitud del sistema de reconocimiento de género provisto por la biblioteca Deepface, indicando el género asociado a la medida si corresponde, obtenidos en base a las tres particiones de UTKFace previamente mencionadas.

Análisis de los resultados:

En relación al género, se observa que esta vez se obtienen resultados muy dispares. Si bien la precisión para el sexo femenino y la recuperación para el masculino son considerablemente altos, estando siempre por encima de 97,8 %, la precisión para hombres se sitúa por debajo del 72,3 %, mientras que la recuperación para mujeres no supera el 51,2 %. Además, el modelo obtiene una exactitud máxima de 77,6 %.

Se teoriza que estos resultados pueden haber sido influidos por el hecho de que el conjunto de datos de entrenamiento tiene muchos más hombres que

mujeres, provocando que el modelo tienda a etiquetar a las personas con el género masculino. Esto se reafirma viendo el valor de la recuperación para el sexo masculino (de las instancias masculinas, cuántas efectivamente etiquetó el modelo como masculinas) y la precisión para el sexo femenino (de las que se etiquetan como femenino, cuántas realmente lo eran). Este último resultado tiene la ventaja de que si se etiqueta como femenino, es sumamente probable que dicha persona sea de sexo femenino, y esto es de utilidad porque el objetivo opcional de la competencia implica el reconocimiento de una mujer.

Como referencia, los resultados de precisión y recuperación reportados por Serengil son más parejos, rondando entre 95,1 % y 98,5 %, y obteniendo una exactitud del 97,44 %. A su vez, Sheoran *et al.* (Sheoran, Joshi, y Bhayani, 2020) obtienen una exactitud de 94,9 % sobre UTKFace. Si bien estos resultados son significativamente superiores a los obtenidos por el modelo de Deepface, se recuerdan nuevamente las restricciones particulares del conjunto de evaluación, y el carácter opcional del objetivo de la RoboCup@Home que se plantea cumplir con dicho modelo (*Despedida* 4.1).

5.3.4.3. Pruebas de predicción de raza

Objetivo:

- Dado un rostro ya detectado, evaluar el desempeño del algoritmo al predecir la raza de la persona asociada a él.

Desarrollo de la prueba:

Para cada imagen del conjunto de datos, se aplica la función *analyze* de la biblioteca Deepface, registrando el valor de la raza que esta predice. Se computa luego la exactitud de la implementación.

Resultados obtenidos y análisis de los resultados:

La mejor exactitud obtenida es de 70,7 %, y se logra en el conjunto balanceado por edad. Esta partición contiene más cantidad de personas de raza blanca, que también son la clase mayoritaria en el conjunto de datos completo. Puede que eso explique por qué se obtuvieron mejores resultados en dicha partición. De hecho, la segunda partición con mejores resultados es la balanceada por género, y también contiene una mayor cantidad de personas blancas.

Por otro lado, el peor resultado obtenido es de 66,2 %, y se da en la partición balanceada por raza. Es probable que suceda algo similar a lo que sucede con la edad: al forzar que haya una cantidad similar de personas de cada raza, se disminuye la cantidad de personas a las que le es más fácil clasificar. En este caso, se tiene una mayor proporción de personas de raza blanca en el conjunto de entrenamiento.

Teniendo en cuenta las condiciones particulares del conjunto de datos utilizado para evaluar, se toma como referencia una vez más a Serengil, quien obtiene una exactitud de 68,0 %. Este valor se encuentra aproximadamente en el promedio de los dos resultados extremos obtenidos en este trabajo. Naveen y Srivastava (Venkat y Srivastava, 2018) evalúan utilizando UTKFace, y obtienen un resultado del 80,8 %, lo que es considerablemente mayor a los obtenidos en

este trabajo. De todas formas, la predicción de raza no es contemplada dentro de las pruebas de las últimas ediciones de la RoboCup@Home (Matamoros, Rascon, y cols., 2019), pero los resultados pueden ser interesantes para futuras ediciones.

5.4. Pruebas en escenarios reales

Luego de evaluar los distintos métodos de detección y reconocimiento utilizando imágenes de bases de datos, se quiere realizar pruebas a tiempo real, en escenarios similares a los que se especifican en el manual de pruebas de la RoboCup@Home. Estas pruebas son ejecutadas en una computadora Sony VAIO con un procesador i7-3532QM CPU @ 2.20GHz, una memoria RAM de 8 GiB, y una placa de video AMD Radeon HD 7600M Series. Se utiliza una cámara web Xiaomi modelo CMSXJ22A, la cual tiene un sensor de 1/2,7 2MP, un lente de 3,6mm con un ángulo de 90 grados, una resolución de 1920x1080, y una tasa de fotogramas de 30FPS. En términos de software, se trabaja con Ubuntu 20.04 y ROS Noetic. Para diferenciar estas pruebas de las definidas en el manual de la RoboCup@Home, de ahora en adelante estas pruebas serán referidas como experimentos.

Cada escenario se corresponde con al menos una prueba diferente de la RoboCup@Home, y especifica un conjunto de experimentos. En ciertos casos se incorporan variaciones como la utilización de accesorios, distintos peinados y variaciones en la iluminación, que permiten probar la robustez de los algoritmos frente a ellas.

Para evaluar el desempeño de los métodos se utilizan la precisión, recuperación y exactitud, definidas en el cuadro 5.1, además de registrar el tiempo de ejecución del experimento. En general, se toman diez sujetos de prueba para cada experimento, y se realizan diez ejecuciones por cada uno de ellos. Las medidas reportadas corresponden al promedio de estas ejecuciones, o de lo contrario, se explicita a cuántas ejecuciones corresponden. A su vez, se explicita en cada experimento qué métricas de las mencionadas son utilizadas para evaluar el desempeño.

Los modelos sobre los cuales se realizan los experimentos son, para la detección los modelos de MTCNN que ofrecen Deepface y Esler, y para la identificación los modelos de FaceNet-512 propuestos por ambas implementaciones. En cuanto a la distancia, se utiliza la euclidiana L2. Se utilizan estos modelos y distancias porque son los que reportan mejores resultados en las pruebas sobre los conjuntos de datos, los cuales se encuentran en la sección 5.3.

Para las tareas de predicción de edad y género, se utiliza la función *analyze* de la biblioteca Deepface. Se observa que la implementación de Esler no permite la identificación de estas características.

Esta sección está organizada de la siguiente forma. Primero, se especifican los requisitos necesarios para la ejecución de los experimentos. Luego, los escenarios donde estos son ejecutados, especificando para cada uno de ellos sus experimentos asociados, con los resultados obtenidos y un análisis de estos. Fi-

nalmente, se presentan conclusiones considerando los experimentos realizados, junto con una tabla que resume los resultados obtenidos.

5.4.1. Requisitos

Para la ejecución de los experimentos, se debe tener una carpeta inicial de imágenes para distintas personas, que deberá encontrarse en el directorio *.ros* y titularse *img*. Se deben tener cinco imágenes por persona , cumpliendo los perfiles presentados en la figura 5.12. Esto es así porque en el experimento 1.1 (5.4.2.2) se evalúa generar las representaciones de los rostros con distintas cantidades de imágenes por persona en la base de datos, comenzando por una, hasta cinco imágenes.

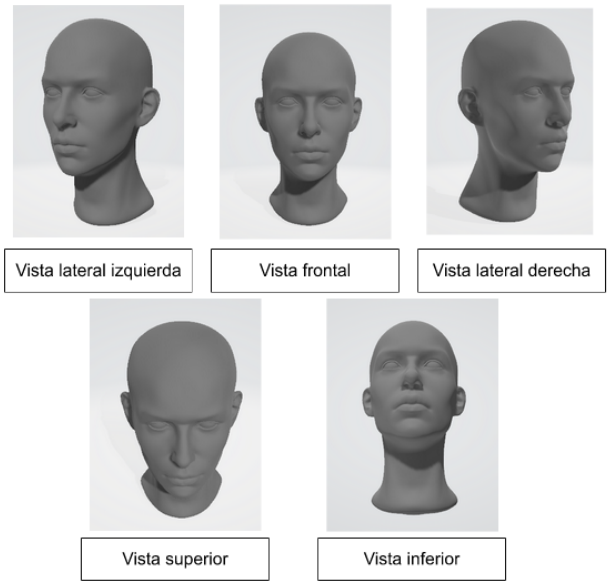


Figura 5.12: Imágenes representativas de los distintos perfiles que se toman en cuenta para un sujeto en la confección de la base de datos y ejecución de las pruebas. Modelo tridimensional extraído de (cgspektor, 2022).

Dado que en los experimentos ejecutados en este proyecto se trabaja con un conjunto que incluye veinte sujetos de prueba, si se quisieran replicar las pruebas se debería contar con esa cantidad de personas en la base de datos. El nombre de cada imagen debe seguir el siguiente formato:

`<nombre de la persona>-X`

donde `X` representa el número de la imagen de esa persona. Por ejemplo, si se tienen tres imágenes de Juana Pérez, los archivos tienen los siguientes nombres:

```
Juana_Perez-1
Juana_Perez-2
Juana_Perez-3
```

Se decide trabajar con veinte sujetos en la base de datos considerando que el robot doméstico podría llegar a interactuar con un número de personas cercano a este en un escenario hogareño, tomando en cuenta los integrantes del hogar y visitantes frecuentes, como familia y amigos. Las imágenes que componen el conjunto utilizado en este proyecto se presentan en la figura 5.13.

Figura 5.13: Imágenes en base a las cuales se construye el conjunto de representaciones utilizado en los distintos experimentos.

Estas imágenes deben ser tomadas con una fuente luz superior, y son utilizadas para generar la base de datos necesaria para luego identificar los rostros en los experimentos. Además, para ciertos experimentos se utilizan imágenes con fuente de luz lateral. Por ejemplo, en el experimento 1.0 del escenario I (5.4.2.1) se utilizan para probar la robustez de los métodos frente a variaciones de iluminación. Para replicar estos experimentos, se debería contar con un conjunto de dieciocho sujetos, considerando los perfiles presentados en la imagen 5.12 pero tomados con una fuente de luz lateral, dado que fue lo que se utilizó en el marco de este proyecto.

Para cada uno de los experimentos que se plantean a continuación, se requiere confeccionar un archivo de texto plano (`.txt`) que contenga los valores esperados. Estos valores son comparados con las salidas reales, para poder así

conocer el éxito o fracaso del experimento, y calcular las métricas asociadas a su ejecución.

5.4.2. Escenario I

Frente a la cámara se encuentra una única persona de interés, situándose aproximadamente a un metro de la cámara.

Este escenario aplica a la prueba *Recepcionista* (definida en el cuadro 4.1), cuando llega un nuevo invitado.

5.4.2.1. Experimento 1.0

Objetivo:

- Determinar el desempeño de los modelos de detección a la hora de construir la base de datos.

Flujo:

1. Para cada imagen de la base de datos, se corre el algoritmo de detección, guardando el recorte generado a la hora de encontrar una cara, o en su defecto, la imagen en su totalidad.
2. Observar y analizar manualmente cada imagen para determinar si los rostros que se detectan se corresponden con la realidad.

Salidas:

- Imágenes con los rostros detectados, en caso de detectar alguno, o la imagen original en caso contrario.
- Valores de precisión y recuperación al detectar los rostros.

Resultados obtenidos:

	Esler		Deepface	
Posición de luz	Prec. ↑	Rec. ↑	Prec. ↑	Rec. ↑
Superior	**100**	**100**	**100**	**100**
Lateral	97,4	84,4	**97,7**	**92,2**

Cuadro 5.7: Resultados (%) de precisión y recuperación para el experimento 1.0. Para cada posición de luz se resalta en negrita los resultados del modelo que tienen un mayor valor, y para cada medida se indica con una flecha si es preferible un valor mayor o menor.

Análisis de los resultados:

Se observa que al construir la base de datos utilizando imágenes que tienen una fuente de luz superior, ambos modelos son capaces de detectar un rostro

en todas las imágenes. En el caso de las imágenes con luz lateral, si bien la diferencia entre las precisiones es pequeña (representando un 0,3 %) se observa que la recuperación de los modelos tiene una diferencia de 7,8 %. Esta diferencia no es despreciable, implicando que el método de detección de Deepface presenta una mayor tolerancia a dificultades en la iluminación, pudiendo así detectar más rostros que Esler bajo estas circunstancias.

Se debe aclarar que para ciertas imágenes de luz lateral, se nota una diferencia de iluminación muy significativa entre el hemisferio izquierdo y el derecho del rostro, lo cual podría explicar por qué no se detectan los rostros para algunas imágenes. En la figura 5.14, se muestra una imagen del conjunto de datos donde se percibe una diferencia significativa de iluminación entre los hemisferios del rostro.

Figura 5.14: Rostro perteneciente al conjunto de imágenes de luz lateral confeccionado, que refleja el desbalance de iluminación entre los hemisferios del rostro.

De todas formas, según participantes previos de la competencia, el lugar donde se ejecutarán las pruebas de la RoboCup@Home estará muy bien iluminado, por lo que el rostro estará iluminado de manera uniforme. El conjunto de imágenes que considera iluminación superior es entonces el que más se adecúa a esta realidad, por lo cual se decide no seguir trabajando con el que presenta luz lateral como base para generar las representaciones.

Figura 5.15: Rostro perteneciente al conjunto de imágenes de luz lateral confeccionado, donde se detecta además un pequeño rostro inesperado de fondo.

Se destaca también que ambos métodos detectan en una imagen en particular un rostro muy pequeño que se encuentra en un cuadro situado al fondo de la imagen, como puede observarse en la figura 5.15. Si bien esta detección es descartada para el cálculo de las métricas de precisión y recuperación, se valora la capacidad de los métodos de poder detectar un rostro de ese tamaño.

5.4.2.2. Experimento 1.1

Objetivos:

- Determinar qué variante de almacenamiento de las representaciones vectoriales es la que da mejores resultados, siendo las variantes a considerar:
  - Una representación vectorial por persona, promediando las de cada imagen de la persona
  - Una representación vectorial por imagen
- Determinar la cantidad óptima de imágenes por persona, considerando de una a cinco, siendo este valor la cantidad de imágenes que se tiene por persona en el conjunto. Para esto se requiere tener diferentes perfiles del rostro, por lo que se utilizan los indicados en la figura 5.12. Estos se van agregando en ese orden, al ir aumentando la cantidad de imágenes por persona.

Flujo:

El flujo presentado a continuación se ejecuta para cada una de las variantes presentadas en los objetivos de este experimento.

1. Detectar los rostros que se encuentren en la imagen, descartando aquellos cuyo cuadro delimitador sea menor a 1/16 de la imagen.

2. Seleccionar el rostro más cercano, es decir, aquel cuyo cuadro delimitador sea de mayor tamaño.

3. Reconocer el rostro detectado, comparando su representación vectorial con las almacenadas en las bases de datos. En caso de que haya múltiples coincidencias, se considera aquella que tenga menor distancia entre las representaciones.

Salidas:

- Mensaje indicando el éxito o fracaso del experimento para cada una de las variantes, siendo este mensaje "Éxito" o "Fracaso" respectivamente.

- Tiempo de ejecución del experimento para cada variante.

- El nombre de la persona reconocida para cada una de las variantes. En caso de no lograr reconocer el rostro, se retorna "Desconocido".

Formato del archivo con los valores esperados:

```
<nombre de la persona a reconocer>
```

Resultados obtenidos:

	Esler			
	Representación por persona		Representación por imagen	
# Imág.	Exac. ↑	Tiempo Ejec. ↓	Exac. ↑	Tiempo Ejec. ↓
5	**100 ± 0**	**0,256 ± 0,017**	**100 ± 0**	0,269 ± 0,023
4	**100 ± 0**	**0,268 ± 0,020**	**100 ± 0**	0,270 ± 0,016
3	**100 ± 0**	**0,267 ± 0,014**	99,0 ± 3,0	0,268 ± 0,009
2	**100 ± 0**	**0,261 ± 0,015**	**100 ± 0**	0,263 ± 0,012
1	**100 ± 0**	**0,257 ± 0,006**	**100 ± 0**	0,261 ± 0,022
Prom.	**100 ± 0**	**0,262 ± 0,14**	99,8 ± 1,4	0,266 ± 0,016

Cuadro 5.8: Resultados de exactitud (%), tiempo de ejecución (s) y sus promedios para el experimento 1.1 y el método Esler, indicando la desviación estándar con un (±) donde corresponda. Se resalta en negrita los resultados de la forma de representación y cantidad de imágenes que obtiene un mayor valor de exactitud y menor valor de tiempo de ejecución. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

	Deepface			
	Representación por persona		Representación por imagen	
# Imág.	Exac. ↑	Tiempo Ejec. ↓	Exac. ↑	Tiempo Ejec. ↓
5	99,0 ± 0,32	**0,943 ± 0,036**	**100 ± 0**	0,986 ± 0,047
4	99,0 ± 0,32	**0,949 ± 0,049**	**100 ± 0**	0,973 ± 0,056
3	**100 ± 0**	**0,976 ± 0,041**	**100 ± 0**	0,987 ± 0,054
2	**100 ± 0**	**0,936 ± 0,041**	**100 ± 0**	0,942 ± 0,046
1	**96,0 ± 0**	**0,257 ± 0,041**	**96,0 ± 12,6**	0,970 ± 0,034
Prom.	98,8 ± 4,8	**0,933 ± 0,030**	**99,2 ± 5,7**	0,972 ± 0,049

Cuadro 5.9: Resultados de exactitud (%), tiempo de ejecución (s) y sus promedios para el experimento 1.1 y el método Deepface, indicando la desviación estándar con un (±) donde corresponda. Se resalta en negrita los resultados de la forma de representación y cantidad de imágenes que obtiene un mayor valor de exactitud y menor valor de tiempo de ejecución. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

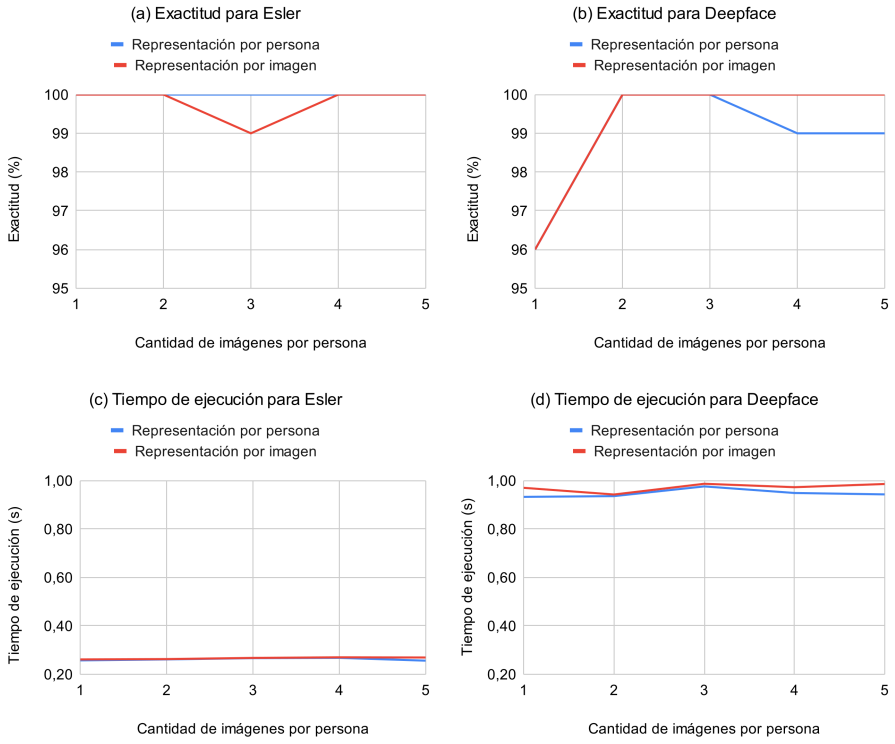


Figura 5.16: (a-b) Exactitud para Esler y Deepface en el experimento 1.1, variando la cantidad de imágenes por persona utilizada al confeccionar la base de datos. (c-d) Tiempo de ejecución para Esler y Deepface en el experimento 1.1, variando la cantidad de imágenes por persona utilizada al confeccionar la base de datos.

Análisis de los resultados:

Como se puede observar en los cuadros 5.8 y 5.9, y en la figura 5.16, el tiempo de ejecución de las variantes de representación por imagen y por persona son muy parecidos para ambos modelos. Además, la versión que considera una representación por imagen suele demorar unas décimas de segundo más, lo cual tiene sentido porque se están comparando más representaciones para cada ejecución. Se observa que el tiempo de ejecución de Deepface es considerablemente mayor al de Esler, siendo este último cerca de 3,6 veces más rápido.

Los resultados de exactitud de ambos métodos son muy similares, tanto considerando representaciones generadas por imágenes como por persona. Se debe destacar que si bien Esler obtiene mejores resultados, ambos modelos alcanzan resultados muy cercanos al 100 %. En el caso de Deepface, se puede apreciar en la figura 5.16(b) que la exactitud presenta una mejora al utilizar a partir de dos

imágenes por persona.

Dado lo anterior, se considera que se carece de la información suficiente para concluir cuál es la mejor manera de almacenar las representaciones, y cuántas imágenes por persona utilizar. Es por esto que se decide realizar un nuevo experimento, considerando imágenes de los sujetos de prueba con variaciones de iluminación, accesorios y posición del pelo, considerando también que estas variaciones se dan con frecuencia en el contexto hogareño.

5.4.2.2.1 Experimento 1.1 con variaciones

Se ejecuta el experimento nuevamente para ambos modelos considerando imágenes estáticas que involucren las siguientes variaciones:

- Luz lateral
- Accesorios:
  - Lentes
  - Lentes de sol
  - Auriculares
  - Gorro y sombrero
  - Tapabocas
  - Bufanda
- Peinados
  - Cerquillo
  - Pelo recogido

Esta vez, dado que son imágenes estáticas, se ejecuta una sola vez el experimento en lugar de diez veces por cada combinación de método de reconocimiento, forma de generación de representación y cantidad de imágenes. Además, se aclara que no se trabaja con igual cantidad de imágenes por variación.

Resultados obtenidos:

	Esler			
	Representación por persona		Representación por imagen	
#Imág.	Exac. ↑	Tiempo Ejec. ↓	Exac. ↑	Tiempo Ejec. ↓
5	**87,4**	**0,749**	86,7	0,814
4	**86,7**	**0,729**	**86,7**	0,738
3	**87,4**	0,738	85,9	**0,732**
2	**86,7**	**0,728**	85,9	0,740
1	**84,4**	**0,733**	**84,4**	0,741
Prom.	**86,5 ± 1,2**	**0,735 ± 0,009**	85,9 ± 0,9	0,753 ± 0,034

Cuadro 5.10: Resultados de exactitud (%), tiempo de ejecución (s) y sus promedios para el experimento 1.1 con variaciones y el método Esler, indicando la desviación estándar con un (±) donde corresponda. Se resalta en negrita los resultados de la forma de representación y cantidad de imágenes que obtiene un mayor valor de exactitud y menor valor de tiempo de ejecución. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

	Deepface			
	Representación por persona		Representación por imagen	
#Imág.	Exac. ↑	Tiempo Ejec. ↓	Exac. ↑	Tiempo Ejec. ↓
5	**83,7**	**1,654**	83,0	1,706
4	**81,5**	1,702	**81,5**	**1,678**
3	**82,2**	**1,630**	81,5	1,675
2	79,3	**1,620**	**80,0**	1,626
1	**80,0**	**1,613**	**80,0**	1,659
Prom.	**81,3 ± 1,8**	**1,644 ± 0,036**	81,2 ± 1,3	1,669 ± 0,029

Cuadro 5.11: Resultados de exactitud (%), tiempo de ejecución (s) y sus promedios para el experimento 1.1 con variaciones y el método Deepface, indicando la desviación estándar con un (±) donde corresponda. Se resalta en negrita los resultados de la forma de representación y cantidad de imágenes que obtiene un mayor valor de exactitud y menor valor de tiempo de ejecución. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

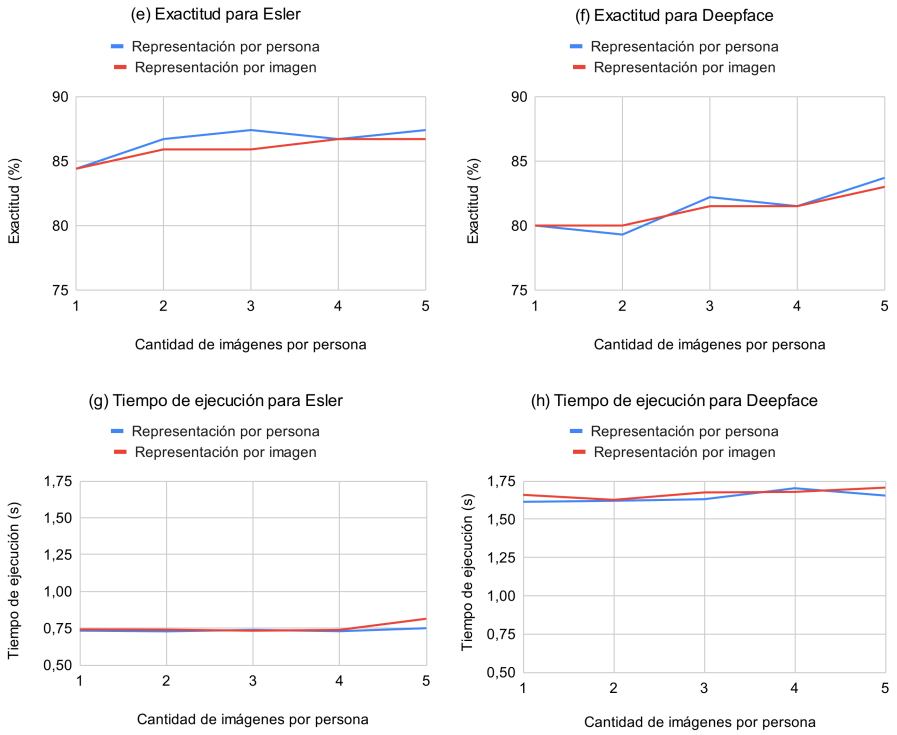


Figura 5.17: (e-f) Exactitud para Esler y Deepface en el experimento 1.1 con variaciones, variando la cantidad de imágenes por persona utilizada al confeccionar la base de datos. (g-h) Tiempo de ejecución para Esler y Deepface en el experimento 1.1 con variaciones, variando la cantidad de imágenes por persona utilizada al confeccionar la base de datos.

Análisis de los resultados:

Si bien no hay diferencias muy notorias, se observa en los cuadros 5.10 y 5.11, que la variante de representación por persona tiene un desempeño levemente superior, tanto en exactitud como en tiempo. Además, se observa que al aumentar la cantidad de imágenes por persona se obtiene una mejor exactitud, y el tiempo solo aumenta mínimamente. Esto último se puede apreciar mejor en la figura 5.17.

Si bien comparar los métodos de reconocimiento no forma parte de los objetivos del experimento, se observa que Esler a priori tiene un mejor desempeño. De todas formas, la elección de cuál método utilizar está sujeta a próximos experimentos enfocados particularmente en su comparación.

Tomando en cuenta todo esto, se decide trabajar con una representación por persona, considerando cinco imágenes en su armado.

Por último, si bien no se presentan los resultados por variación en este capítu-

lo, cabe destacar que las variaciones que más dificultades causan a los modelos son el tapabocas y el gorro. También, se observa que las imágenes que incluyen una sola variación en general no comprometen el desempeño del modelo, y se necesita una combinación de estos para hacerlo. Los resultados de exactitud por variación se pueden observar en el apéndice C.

5.4.2.3. Experimento 1.2

Objetivo:

- Reconocer a la persona que se encuentra en la imagen.

Flujo:

1. Ejecutar los pasos 1-3 especificados en el experimento 1.1 (5.4.2.2).

Salidas:

- Mensaje indicando el éxito o fracaso del experimento, siendo este mensaje "Éxito" y "Fracaso" respectivamente.
- Tiempo de ejecución del experimento.
- El nombre de la persona reconocida. En caso de no lograr reconocer el rostro, se retorna "Desconocido".

Formato del archivo con los valores esperados:

```
<nombre de la persona a reconocer>
```

Caso de uso en la prueba Recepcionista:
El robot debe ser capaz de reconocer a una persona que ya tiene registrada en su base de datos, para presentársela a nuevos invitados.
Resultados obtenidos:

Esler		Deepface	
Exac. ↑	Tiempo Ejec. ↓	Exac. ↑	Tiempo Ejec. ↓
100 ± 0	**0,272 ± 0,035**	92,0 ± 25,3	1,215 ± 0,078

Cuadro 5.12: Resultados de exactitud (%) y tiempo de ejecución (s) para el experimento 1.2. Se resalta en negrita los resultados del modelo que tienen un mayor valor de exactitud y menor valor de tiempo de ejecución. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

Análisis de los resultados:
Como se desprende del cuadro 5.12, la diferencia de exactitud entre los modelos es de un 8,0 %, que no es despreciable. De todas formas, ambos modelos tuvieron una exactitud superior al 90,0 %, lo cual no se aleja del nivel del estado

del arte. Por esto, se decide no descartar el modelo Deepface en base a estos resultados.

En cuanto al tiempo de ejecución, Deepface tiene un orden de magnitud mayor que Esler, siendo aproximadamente 4,5 veces más lento. Esta diferencia en desempeño podría llegar a ser definitoria, pero para el caso de uso que se le va a dar, que demore poco más de un segundo no afectaría en demasía la experiencia de uso en un ambiente doméstico.

Por último, se resalta que la varianza reportada para la exactitud de Deepface se debe a un valor atípico, donde se confundió a un sujeto de prueba con otro en el 80 % de las ejecuciones asociadas a ese sujeto. Si se removiera este valor, se obtendría una varianza nula. Se cree que esta confusión pudo haberse debido a que el sujeto a ser reconocido está afeitado en las imágenes tomadas para construir la base de datos, pero al momento de realizar el experimento, este tiene barba.

5.4.2.3.1 Experimento 1.2 con variaciones

Considerando que los modelos logran reconocer la mayoría de los rostros, se plantea repetir el experimento pero incluyendo distintos accesorios en la toma para ver cómo se comportan los modelos en condiciones no óptimas, presentando dificultades con las cuales el algoritmo se tendría que enfrentar si se lo utilizara en un ambiente doméstico.

Los accesorios utilizados son tapabocas, lentes, lentes de sol y gorro. Como se vio en el experimento detallado en (5.4.2.2.1), estos accesorios son los que generan más dificultades para los modelos. Dado que en general utilizando solo un accesorio los modelos son capaces de reconocer los rostros, los experimentos se ejecutan combinando los accesorios, como se muestra en el cuadro 5.14.

Esler		Deepface	
Exac. ↑	Tiempo Ejec. ↓	Exac. ↑	Tiempo Ejec. ↓
52,4 ± 45,7	**0,253 ± 0,038**	42,7 ± 44,3	1,161 ± 0,056

Cuadro 5.13: Resultados de exactitud (%) y tiempo de ejecución (s) para el experimento 1.2 con variaciones. Se resalta en negrita los resultados del modelo que tienen un mayor valor de exactitud y menor valor de tiempo de ejecución. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

	Esler	Deepface
Accesorios	Exac. ↑	Exac. ↑
Gorro y Lentes de sol	63,0 ± 48,1	**71,0 ± 40,9**
Gorro y Lentes	**77,0 ± 40,0**	74,0 ±39,8
Tapabocas y Gorro	18,0 ± 34,6	**25,0 ± 39,2**
Tapabocas y Lentes de sol	**35,0 ± 47,4**	19,0 ± 35,4
Tapabocas y Lentes	**63,0 ± 41,4**	23,0 ± 38,9

Cuadro 5.14: Resultados de exactitud (%) y tiempo de ejecución (s) para el experimento 1.2 con variaciones, por combinación de accesorio. Se resalta en negrita los resultados del modelo que tienen un mayor valor de exactitud. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

Análisis de los resultados:

Como se puede apreciar en el cuadro 5.13, si bien Esler obtiene mejores resultados, ambos modelos disminuyen considerablemente su exactitud y aumentan su varianza. Esto indica que el modelo se torna más impredecible con la inclusión de múltiples accesorios. La diferencia entre los modelos en este caso es de 9,7 %, que es muy cercana a la obtenida en el experimento anterior. En cuanto al tiempo de ejecución, se mantiene la tendencia de que Esler es más rápido, siendo Deepface aproximadamente 4,6 veces más lento.

Figura 5.18: Rostro perteneciente al conjunto de imágenes con accesorios utilizado para evaluar los métodos, que representa cuán cubierto puede quedar el rostro utilizando lentes de sol y tapabocas a la vez.

En cuanto a la resiliencia de los modelos frente a accesorios, la combinación de tapabocas y lentes de sol, y la de tapabocas y gorro son las que más dificultan a los modelos. Esto se puede apreciar en el cuadro 5.14. Particularmente, los

experimentos que incluyen tapabocas son en los que se producen peores resultados. Esto puede deberse a que el tapabocas es el accesorio que cubre la mayor parte del rostro (en particular, la boca y la nariz), como se puede ver en la figura 5.18.

5.4.2.4. Experimento 1.3

Objetivo:

- Incorporar una persona a la base de datos, para luego detectarla y reconocerla.

Flujo:

1. Detectar los rostros que se encuentren en la imagen, descartando aquellos cuyo cuadro delimitador sea menor a 1/16 de la imagen.
2. Seleccionar el más cercano, es decir, aquel cuyo cuadro delimitador sea de mayor tamaño.
3. Intentar reconocer el rostro detectado, comparando su representación vectorial con las almacenadas en la base de datos. Al no encontrarse la persona en la base de datos, esta no debería ser reconocida, por lo que se la considera como un rostro nuevo a almacenar en la base de datos. En caso de reconocer a la persona, se considera que el experimento falla, por lo que se finaliza dicho experimento.
4. Tomar cinco fotografías de la nueva persona a reconocer (en base a los cinco perfiles especificados en la figura 5.12), calcular sus representaciones vectoriales, solicitar su nombre, y agregar la relación entre las representaciones y el nombre a la base de datos.
5. Ejecutar los pasos 1-3 especificados en el experimento 1.1 (5.4.2.2).

Salidas:

- Mensaje indicando el éxito o fracaso del experimento, siendo este mensaje "Éxito" y "Fracaso" respectivamente.
- Tiempo de ejecución del reconocimiento.
- Tiempo de ejecución de la actualización de la base de datos, agregando las representaciones vectoriales del nuevo sujeto.
- El nombre de la persona reconocida en el paso 4. En caso de no lograr reconocer el rostro, se retorna "Desconocido".

Formato del archivo con los valores esperados:

```
<nombre de la persona a detectar y reconocer>
```

Caso de uso en la prueba Recepcionista: Al llegar un nuevo invitado, el robot debe ser capaz de almacenar la relación entre su nombre y su rostro para luego poder presentarlo a otros invitados.

Resultados obtenidos:

Esler		Deepface	
Tiempo Ejec. BD ↓	Tiempo Ejec. ↓	Tiempo Ejec. BD ↓	Tiempo Ejec. ↓
1,019 ± 0,016	**0,248 ± 0,020**	**0,505 ± 0,069**	0,921 ± 0,027

Cuadro 5.15: Resultados de tiempo de ejecución (s) para el reconocimiento y la actualización de la base de datos, para el experimento 1.3. Se resalta en negrita los resultados del modelo que tienen un menor tiempo de ejecución. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

Esler	Deepface
Exac. ↑	Exac. ↑
63,3 ± 39,4	**80,0 ± 37,1**

Cuadro 5.16: Resultados de exactitud (%) para el experimento 1.3. Se resalta en negrita los resultados del modelo que tienen un mayor valor de exactitud. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

Análisis de los resultados:

Se observa en el cuadro 5.16 que Deepface obtiene una mejor exactitud que Esler, teniendo una diferencia de 16,7 %. En cuanto al tiempo de ejecución, se aprecia en el cuadro 5.15 que se mantiene la tendencia de que Esler es más rápido que Deepface. Sobre el tiempo de actualización de la base de datos, esta vez es Deepface el modelo que obtiene mejores resultados, actualizando la base de datos en la mitad del tiempo. Este hecho llama la atención a los autores de este trabajo, y revisando la solución implementada se observa que en el caso de Esler se está incluyendo en el tiempo de actualización la instanciación de los modelos, lo que se especula que puede estar provocando la diferencia reportada. Una vez más, realizar esta actualización en poco más de un segundo no cambiaría en demasía la experiencia de uso.

Cabe resaltar que existen dos maneras de fallar en este experimento. La primera es que se reconozca un sujeto no presente en la base de datos, y la segunda es no reconocer al nuevo sujeto una vez agregado a esta. Se observa que en general, se producen fallos del primer tipo. En particular, algunos de estos fallos se producen para todas o casi todas las ejecuciones asociadas a un sujeto (al menos el 80 %). Esto genera que la varianza adquiera un valor considerable. Uno de los casos que genera una exactitud nula se da entre hermanos que comparten varios rasgos faciales, como puede apreciarse en la figura 5.19.

Figura 5.19: Rostros de hermanos con rasgos faciales similares, lo que provoca que en el experimento 1.3 los métodos los confundan.

5.4.3. Escenario II

Se encuentran múltiples personas frente a la cámara, algunas paradas y otras sentadas. Estas personas están ubicadas a aproximadamente dos metros de la cámara, una junto a la otra. Se consideran personas paradas y sentadas porque en ciertas pruebas de la competencia, esta es la disposición de los participantes en el escenario. Además, esto permite evaluar la capacidad de reconocimiento de los métodos desde distintos ángulos del rostro.

Este escenario aplica a las pruebas *Despedida* y *Sirviendo las bebidas*, especificadas en el cuadro 4.1.

5.4.3.1. Experimento 2.1

Objetivo:

- Detectar y reconocer a todas las personas en la imagen.

Flujo:

1. Detectar los rostros que se encuentren en la imagen, descartando aquellos cuyo cuadro delimitador sea menor a 1/16 de la imagen.

2. Para cada rostro detectado, reconocerlo comparando su representación vectorial con las almacenadas en la base de datos. En caso de que haya múltiples coincidencias, se considera aquella que tenga menor distancia entre las representaciones.

Formato del archivo con los valores esperados:

```
<nombre de la persona que se está más a la izquierda>
...
<nombre de la persona que se está más a la derecha>
```

Salidas:

- Mensaje indicando el éxito o fracaso del experimento, siendo este mensaje "Éxito" y "Fracaso" respectivamente.
- Tiempo de ejecución del experimento.
- Los nombres de las personas reconocidas, en el orden en el que se encuentran en la imagen, de izquierda a derecha. En caso de no reconocer alguno de los rostros, se retorna "Desconocido" para la persona en cuestión.

Caso de uso en la prueba Despedida: El robot debe poder reconocer a las personas en la imagen, para entregarles el abrigo correcto al despedirlas.

Caso de uso en Sirviendo las bebidas: El robot debe poder reconocer a las personas en la imagen, para entregarles el trago correcto.

Variaciones:

- Cantidad de personas en la imagen:
  - Cinco: dos paradas y tres sentadas.
  - Tres: una parada y dos sentadas.

Resultados obtenidos:

	Esler		Deepface	
Cant. suj. prueba	Exac. ↑	Tiempo Ejec. ↓	Exac. ↑	Tiempo Ejec. ↓
3	**100 ± 0**	**0,535 ± 0,035**	**100 ± 0**	1,156 ± 0,809
5	**100 ± 0**	**0,815 ± 0,036**	50 ± 51,3	1,860 ±0,785

Cuadro 5.17: Resultados de exactitud (%) y tiempo de ejecución (s) para el experimento 2.1, por cantidad de sujetos de prueba. Se resalta en negrita los resultados del modelo que tienen un mayor valor de exactitud y menor tiempo de ejecución. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

Análisis de los resultados:

Tomando en cuenta tres sujetos de prueba, los modelos no tienen ningún fallo en sus ejecuciones. Teniendo en cuenta cinco sujetos, se aprecia en el cuadro 5.17 que el desempeño de Deepface decae hasta alcanzar un 50,0 % de exactitud. Cabe aclarar que este valor hace referencia a haber tenido ejecuciones exitosas del experimento. Es decir, se reconoce a todas las personas en el orden correspondiente, considerando la prueba como fallida en caso de no reconocer a alguna de ellas. Sin embargo, si se considera el reconocimiento de cada persona en el campo de visión de manera individual, el modelo habría alcanzado una exactitud de 90,0 %.

Resulta relevante aclarar que, para las ejecuciones con cinco sujetos, en el caso de Deepface se presentan resultados para el doble de ejecuciones que en el de Esler. Esto se debe a que, luego de realizar las diez ejecuciones correspondientes de Deepface, al resultar extraños los resultados obtenidos, se considera que pueden ser mejorables, debiéndose estos quizás a una ejecución atípica, por lo que se realiza una segunda tanda de diez ejecuciones del experimento. Sin embargo, los resultados obtenidos en esta segunda tanda empeoran con respecto a la primera. Dado que ambas tandas aportan información relevante, se decide presentar el promedio de ambas.

En los experimentos realizados, hay dos sujetos que en ciertas ejecuciones de Deepface no son reconocidos, pero sí lo son por el otro modelo. Cabe destacar que el sujeto que causa mayores dificultades presenta barba en el momento de ejecutar el experimento, y no la tiene al momento de tomar las imágenes para confeccionar la base de datos. Es posible que sea esta característica la que dificulta al modelo Deepface en el reconocimiento. Dado que la barba se encuentra en la misma zona del rostro donde se coloca el tapabocas, y que se reporta en el experimento 1.2 con variaciones (5.4.2.3.1) que este accesorio es uno de los que genera más dificultades, esto puede ser un indicio de que dicha zona sea una zona sensible para los modelos. Esto resulta relevante, dado que es altamente probable que cambie el aspecto de la barba de las personas en un ambiente doméstico, y aunque su uso es menos frecuente, también es posible que ingresen al hogar con un tapabocas colocado luego de estar en la calle.

En cuanto al tiempo de ejecución, una vez más se mantiene la tendencia de que Esler es más rápido.

5.4.4. Compilación de resultados

	Esler		Deepface	
Prueba	Exac. ↑	Tiempo Ejec. ↓	Exac. ↑	Tiempo Ejec. ↓
1.1	**100 ± 0**	**0,262 ± 0,14**	98,8 ±4,8	0,933 ± 0,030
1.1 var.	**86,5 ± 1,2**	**0,735 ± 0,009**	81,3 ± 1,8	1,644 ± 0,036
1.2	**100 ± 0**	**0,272 ± 0,035**	92,0 ± 25,3	1,215 ± 0,078
1.2 var.	**52,4 ± 45,7**	**0,253 ± 0,038**	42,7 ± 44,3	1,161 ± 0,056
1.3	63,3 ± 39,4	**0,248 ± 0,020**	**80,0 ± 37,1**	0,921 ± 0,027
2.1 (3 suj.)	**100 ± 0**	**0,535 ± 0,035**	**100 ± 0**	1,516 ± 0,809
2.1 (5 suj.)	**100 ± 0**	**0,815 ± 0,036**	50 ± 51,3	1,860 ± 0,785

Cuadro 5.18: Compilación de los resultados de exactitud (%) y tiempo de ejecución (s) por experimento, incluyendo los experimentos con variaciones y las ejecuciones con tres y cinco sujetos para el experimento 2.1. Los resultados reportados son promedios, y no todos se obtienen considerando la misma cantidad de ejecuciones. Se resalta en negrita los resultados del modelo que tiene un mayor valor de exactitud y menor tiempo de ejecución. Además, se indica para cada medida con una flecha si es preferible un valor mayor o menor.

Como puede apreciarse en el cuadro 5.18, Esler obtiene un mejor desempeño que Deepface en todas las pruebas, a excepción de la 1.3 donde Deepface lo supera, y la 2.1, con tres sujetos de prueba, donde ambos obtienen una exactitud del 100 %. En otras palabras, Esler reporta mejores resultados al reconocer una persona que ya se encuentra en la base de datos, incluso con accesorios, y reconociendo múltiples personas en un mismo campo de visión. Por otro lado, Deepface obtiene mejores resultados cuando se introduce un nuevo sujeto que debe detectar, confeccionar su representación y actualizar la base de datos con dicha representación, para luego reconocerlo.

En cuanto al tiempo de ejecución, Esler es al menos 2,282 veces más rápido que Deepface. En la prueba donde se da una mayor diferencia, la 1.2 con accesorios, Esler es 4,467 veces más rápido. En promedio, Esler es 3,395 veces más rápido. Se debe tener en cuenta que este promedio se calcula en base al tiempo de ejecución promedio de cada prueba, y que no para todas las pruebas se realizó la misma cantidad de ejecuciones.

Dados estos resultados, existen tres variables de gran importancia a la hora de decidir qué modelo utilizar, que determinan qué tan probable es que el modelo sufra una confusión al incorporar un sujeto nuevo a la base de datos (como las que se dan en la prueba 1.3, planteada en la sección 5.4.2.4). La primera variable a considerar es cuántas personas se encuentran representadas en la base de datos. Cuanta más gente haya, hay más oportunidades para que el modelo confunda individuos. Del análisis de la pruebas se desprende que la

cantidad de gente participando en una prueba está en el orden de las unidades. La segunda variable es qué tan frecuentemente se va a actualizar la base de datos de manera automática con nuevos sujetos, que justamente es la instancia donde se puede producir el tipo de confusión mencionada. Si bien esto sucede en algunas pruebas, como Recepcionista, la poca cantidad de gente representada en la base de datos parece indicar que esto no sea un problema. La tercera variable a considerar es qué tan parecida es la gente que está tomando parte de la prueba. Claramente, cuanto más se parezcan dos personas, más difícil será distinguirlas. Esta última variable está sujeta al azar, dado que los participantes de cada prueba pueden ser elegidos de manera aleatoria. En base a esto, y a los resultados presentados en esta subsección, se concluye que Esler es la implementación que mejor se ajusta a este proyecto.

5.5. Principales decisiones tomadas

Se presenta en el cuadro 5.19 las principales decisiones tomadas a lo largo de la etapa de experimentación. Esto se hace debido a que estas decisiones afectan a la construcción del nodo y a todas las pruebas que se realizan luego de haber tomado cada decisión. Por ejemplo, una vez determinado el umbral de confianza, este es utilizado para todas las pruebas posteriores. Para cada una de estas decisiones, se indica la prueba o experimento del cual se desprende.

referencia	*decisión*
5.3.2.2	Con respecto a los umbrales de confianza en la etapa de detección, se decide no sacrificar la precisión en pos de una mejor recuperación. Se utilizan entonces los umbrales de confianza que comprometen menos a ambas métricas. Estos son 0,955 para Esler y 0,926 para Deepface.
5.3.3	Para Deepface, se utiliza el modelo de reconocimiento FaceNet-512, y distancia euclidiana L2.
5.3.3	Para Esler, se utilizan la función de distancia euclidiana L2, con un umbral de 1,04 y el conjunto de entrenamiento VGGFace2.
5.4.2.2	Ante la posibilidad de confeccionar una representación por persona, o una por imagen, se decide ir por la opción de una por persona.
5.4.2.2	Se decide considerar cinco imágenes por persona a la hora de construir una representación.
-	Utilizar la implementación de Facenet-512 implementada por Tim Esler, en lugar de la versión de Deepface de esta arquitectura, usando como modelo de detección MTCNN.

Cuadro 5.19: Principales decisiones tomadas en base a los resultados obtenidos en las pruebas realizadas en la etapa de experimentación. Para cada decisión se referencia la prueba asociada, a excepción de la última que surge como conclusión general de todas las pruebas ejecutadas.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

El desarrollo de este proyecto permite conocer algunos métodos de reconocimiento de rostros que conforman el estado del arte, así como los resultados que estos obtienen. Se destaca de estos FaceNet (Schroff y cols., 2015), que si bien fue presentado en el año 2015 sigue encontrándose dentro de los métodos con mejor desempeño, alcanzando una exactitud de 99,63 % sobre LFW. Este método es el utilizado en la solución propuesta en este proyecto, tanto en la implementación basada en Deepface como la que utiliza Esler. Cabe aclarar que la exactitud mencionada es la que reportan los autores del artículo que presenta el método, mientras que las obtenidas sobre LFW en este trabajo son 97,5 % para Esler y 97,6 % para Deepface. Otro método destacable es ArcFace (Deng y cols., 2019), que obtiene sobre este mismo conjunto de datos una exactitud de 99,83 %, superando a FaceNet. Sin embargo, dado que la implementación que provee la biblioteca Deepface de este método obtiene resultados significativamente inferiores (no superando el 94,4 %), y dada la carencia de implementaciones de este método al momento de elegir una implementación a utilizar, se opta por las implementaciones de FaceNet mencionadas. Considerando los resultados que reportan los autores para ambos métodos, así como los otros estudiados, se puede afirmar que el nivel del estado del arte es muy alto en el área de reconocimiento de rostros, obteniendo la mayoría de los métodos analizados una exactitud superior al 99,0 % sobre LFW.

En cuanto a la predicción de edad y género, los modelos presentados no obtienen resultados al nivel del estado del arte, considerando como estado del arte a (Berg y cols., 2021) en el área de predicción de edad, y a (Sheoran y cols., 2020) en el área de predicción de género. De todas formas, se determina que estos podrían ser útiles para la competencia. Esto es porque, para el modelo de predicción de género, el objetivo opcional es reconocer a la invitada de géne-

ro femenino, y para dicho género se alcanzan valores de precisión de al menos 97,8 %. Además, para la predicción de edad el objetivo es detectar al invitado de mayor edad, y dado que el modelo alcanza valores de MAE de entre 8,0 y 12,2 en distintos subconjuntos, se considera que si la diferencia de edad es relativamente notoria, el modelo podría detectarlo de forma correcta. Finalmente, se recuerda que en caso de fallo los objetivos opcionales no restan puntos, por lo que utilizarlos solamente puede beneficiar al equipo en la competencia.

A su vez, las pruebas de robustez permiten identificar que el accesorio que causa más problemas a los métodos al momento de identificar a una persona es el tapabocas, siendo las combinaciones de este con el gorro y los lentes de sol las más problemáticas. Esto se vincula también a las dificultades que presentan los modelos frente a los cambios de barba, lo que parece indicar una sensibilidad por parte de los modelos en la zona del rostro asociada. De todas formas, en general un solo accesorio no compromete el desempeño del modelo, y se necesita una combinación de estos para hacerlo.

Por otra parte, realizar un análisis de las pruebas de la RoboCup@Home permite tener una visión más clara de los escenarios en los que estará expuesto el robot a la hora de utilizar el sistema de reconocimiento. Este análisis posibilita trazar pruebas específicas a la situación, y presentar conclusiones más certeras. Estas pruebas indican que Esler obtiene un tiempo de ejecución de al menos el doble que Deepface para todos los experimentos en escenarios reales y una mayor exactitud en la mayoría de estos. Se da una excepción en la prueba 1.3, donde Esler confunde a los nuevos sujetos con sujetos ya existentes en la base de datos, y Deepface obtiene mejores resultados con una diferencia significativa. Sin embargo, como se vio en la sección 5.4.4, esto no parecería representar un problema, sobre todo porque en la competencia se tendrá una cantidad pequeña de personas en la base de datos.

En base a los resultados y análisis presentados en este documento, se determina que el modelo que más se adecúa a la competencia es la implementación de FaceNet-512 provista por Tim Esler (Esler, Tim, 2021), utilizando MTCNN como método de detección. De todas formas, se recuerda que el nodo de ROS desarrollado podría utilizarse con cualquiera de los dos modelos de reconocimiento.

6.2. Trabajo futuro

A partir de lo investigado y realizado en este proyecto, se desprende una serie de líneas que resulta interesante desarrollar. Estas son presentadas a continuación.

Si bien se especificaron e implementaron parcialmente experimentos en escenarios reales de predicción de características, la necesidad de incorporar otros experimentos de detección y reconocimiento llevó a que la ejecución de los primeros fuera aplazada (por estar estos asociados a objetivos opcionales) y finalmente no llevada a cabo. Debido a esto, como trabajo futuro se podría realizar la ejecución de estos experimentos y así poder evaluar esta tarea en un escenario

real. Estas especificaciones se incluyen en el apéndice B, y abarcan la predicción de edad y género.

Además, se recuerda que se evaluó solamente un modelo para cada tarea de predicción de características. Es por esto que resulta interesante evaluar otros modelos y compararlos con el evaluado en este proyecto como trabajo futuro.

En la sección 5.4.2.2 se plantea una prueba para determinar cuál es la manera más conveniente de construir representaciones, y se opta por construir una representación por persona. De todas formas, dado que esto es un parámetro de la función implementada en este trabajo, se podría establecer un mecanismo como trabajo futuro para que, en caso de un reconocimiento fallido, se utilice la otra manera de generar representaciones. Esto puede ser especialmente útil en situaciones donde no se esté totalmente de frente al sujeto a detectar, o este no esté mirando directamente al robot. Esto es así porque, si bien en la confección de una representación por persona se consideran distintos perfiles de esta, elaborar una representación por cada imagen de dicha persona hará que se realicen más comparaciones (y más específicas de cada perfil), pudiendo así tener más oportunidades de reconocimiento. Si bien en los escenarios de la RoboCup@Home el robot estará ubicado de frente al individuo a detectar, y este estará mirando al robot, esta mejora puede ser muy útil en el contexto hogareño.

Por último, otro posible trabajo futuro que se desprende de esta tesis es realizar nuevas ejecuciones del experimento asociado a la incorporación de rostros a la base de datos (5.4.2.4) para Esler, probando con umbrales de distancia más pequeños. Esto surge ante los bajos resultados que este método obtiene con respecto a Deepface en dicho experimento, confundiendo rostros nuevos con otros ya presentes en la base de datos.

Referencias

Abney, S. (2002). Bootstrapping. En *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 360–367).

Amos, B., Ludwiczuk, B., Satyanarayanan, M., y cols. (2016). Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, *6*(2).

Anwar, A. (2021, Jun). *A beginner's guide to regression analysis in machine learning.* `https://towardsdatascience.com/a-beginners-guide-to-regression-analysis-in-machine-learning-8a828b491bbf`. Medium. (visitado en 2021-08-08)

Asiri, S. (2018, Jun). *Machine learning classifiers. what is classification?* Towards Data Science. Descargado de `https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623` (visitado en 2022-10-29)

Avila Camacho, J. (2020, Mar). *Support vector regression (svr).* `https://www.jacobsoft.com.mx/es_mx/support-vector-regression/`. JacobSoft Consultoría y Capacitación. (visitado en 2021-08-08)

Barrios, J., y Barrato, J. I. (2019, Jun). *Redes neuronales convolucionales.* `https://www.juanbarrios.com/redes-neurales-convolucionales/`. Fusionando las Tecnologías Digitales y la Salud Juan Barrios.

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

Benhur J, S. (2020). *A friendly introduction to siamese networks.* `https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab17522942`. (visitado en 2021-07-29)

Berg, A., Oskarsson, M., y O'Connor, M. (2021). Deep ordinal regression with label diversity. En *2020 25th international conference on pattern recognition (icpr)* (pp. 2740–2747).

Bledsoe, W. W. (1966). The model method in facial recognition. *Panoramic Research Inc., Palo Alto, CA, Rep. PR1*, *15*(47), 2.

Bvof robocup2013 - robocup@home. (2013). `https://flic.kr/p/eWKYYW`. (visitado en 2022-10-22)

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., y Zisserman, A. (2018). Vggface2: A dataset for recognising faces across pose and age. En *2018 13th ieee international conference on automatic face & gesture recognition (fg 2018)* (pp. 67–74).

cgspektor. (2022). *Female head 2t.* `https://skfb.ly/oxySZ`. ("Female Head 2"(https://skfb.ly/oxySZ) por cgspektor está bajo la licencia Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/). Visitado en 2022-11-01.)

Chollet, F. (2021). *Deep learning with python.* Simon and Schuster.

Chollet, F. (2022). *Keras: the python deep learning api.* `https://keras.io/`.

Dalal, N., y Triggs, B. (2005). Histograms of oriented gradients for human detection. En *2005 ieee computer society conference on computer vision and pattern recognition (cvpr'05)* (Vol. 1, pp. 886–893).

de Paz Centeno, Iván. (2021). *Mtcnn.* `https://github.com/ipazc/mtcnn`. (visitado en 2022-10-29)

Deng, J., Guo, J., Xue, N., y Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. En *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 4690–4699).

Donalek, C. (2011). Supervised and unsupervised learning. En *Astronomy colloquia. usa* (Vol. 27).

Eirea, M., y Peña, F. (2022). *Estado del arte.* `https://gitlab.fing.edu.uy/manuel.eirea/proyecto-de-grado/-/blob/master/estado%20del%20arte/Estado_del_arte.pdf`. (visitado en 2022-10-26)

Esler, Tim. (2021). *Face recognition using pytorch.* `https://github.com/timesler/facenet-pytorch`. (visitado en 2022-11-25)

Everitt, B. S., Landau, S., Leese, M., y Stahl, D. (2011). Cluster analysis: Wiley series in probability and statistics. *Southern Gate, Chichester, West SussexUnited Kingdom: John Wiley & Sons*.

Fabre, R., Albar, B., Dussieux, C., Joffroy, L., Li, Z., Pinet, C., ... Loty, S. (2019). *Catie robotics@ home 2019 team description paper.* `http://robotics-catie-fr.catie-na.fr/wp-content/uploads/2019/10/Catie_Robotics_TDP_2019.pdf`. (visitado en 2021-08-08)

Felzenszwalb, P., McAllester, D., y Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. En *2008 ieee conference on computer vision and pattern recognition* (pp. 1–8).

fiducial point. (n.d.) mcgraw-hill dictionary of scientific & technical terms, 6e. (2003). (s.f.). `https://encyclopedia2.thefreedictionary.com/fiducial+point`. (visitado en 2021-08-08)

Geng, X., Smith-Miles, K., y Zhou, Z.-H. (2008). Facial age estimation by nonlinear aging pattern subspace. En *Proceedings of the 16th acm international conference on multimedia* (pp. 721–724).

Github. (2022). `https://github.com`.

Goldstein, A. G., y Mackenberg, E. J. (1966). Recognition of human faces from isolated facial features: A developmental study. *Psychonomic Science*, *6*(4), 149–150.

Gu, L., y Kanade, T. (2009). Face alignment. En *Encyclopedia of biometrics* (pp. 291–294). Boston, MA: Springer US. Descargado de `https://doi.org/10.1007/978-0-387-73003-5_186` doi: 10.1007/978-0-387-73003-5_186

Gudikandula, P. (2019). *A beginner intro to neural networks.* `https://purnasaigudikandula.medium.com/a-beginner-intro-to-neural-networks-543267bda3c8`. (visitado en 2021-08-05)

He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Huang, G. B., Mattar, M., Berg, T., y Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained envi-

ronments. En *Workshop on faces in'real-life'images: detection, alignment, and recognition.*

jibo. (2021). https://jibo.com. (visitado en 2021-07-16)

Kaggle. (s.f.). https://www.kaggle.com. (Visitado en 2022-12-01.)

Karkkainen, K., y Joo, J. (2021). Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. En *Proceedings of the ieee/cvf winter conference on applications of computer vision* (pp. 1548–1558).

Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., y Brossard, E. (2016). The megaface benchmark: 1 million faces for recognition at scale. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4873–4882).

Kortli, Y., Jridi, M., Al Falou, A., y Atri, M. (2020). Face recognition systems: A survey. *Sensors*, *20*(2). Descargado de https://www.mdpi.com/1424-8220/20/2/342 doi: 10.3390/s20020342

The latest in machine learning — papers with code. (s.f.). https://paperswithcode.com/. (Visitado en 2022-12-11.)

Lfw : Results. (2019). http://vis-www.cs.umass.edu/lfw/results.html. (visitado en 2022-09-20)

Liu, J., Zhang, Z., Tang, B., y Chen, X. (2017). Wrighteagle@ home 2017 team description paper. *RoboCup@ Home*.

Matamoros, M., Rascon, C., Wachsmuth, S., Moriarty, A. W., Kummert, J., Hart, J., ... St-Pierre, M. (2019). *Robocup@home 2019: Rules and regulations (draft).* http://www.robocupathome.org/rules/2019_rulebook.pdf. (visitado en 2021-05-26)

Matamoros, M., Viktor, S., y Paulus, D. (2019). Trends, challenges and adopted strategies in robocup@ home. En *2019 ieee international conference on autonomous robot systems and competitions (icarsc)* (pp. 1–6).

Memmesheimer, R., Müller, D., Kramer, I., Yann, N., Wettengel, T. E., Buchhold, L., ... cols (2019). *Robocup@ home open platform league 2019-homer@ unikoblenz (germany).* https://www.researchgate.net/profile/Raphael-Memmesheimer-2/publication/337732704_RoboCupHome_Open_Platform_League_2019_-_homerUniKoblenz_Germany/links/5de790494585159aa45fb0a6/RoboCupHome-Open-Platform-League-2019-homerUniKoblenz-Germany.pdf. (visitado en 2021-07-02)

Mendoza, J. (s.f.). *¿qué es un modelo svm?* https://estadisticamente.com/que-es-un-modelo-svm/. (visitado en 2021-08-08)

Mettler, T., Sprenger, M., y Winter, R. (2017). Service robots in hospitals: new perspectives on niche evolution and technology affordances. *European Journal of Information Systems*, *26*(5), 451–468.

Milliez, G. (2018). Buddy: A companion robot for the whole family. En *Companion of the 2018 acm/ieee international conference on human-robot interaction* (pp. 40–40).

Mitchell, T., Buchanan, B., DeJong, G., Dietterich, T., Rosenbloom, P., y Wai-

bel, A. (1990). Machine learning. *Annual review of computer science*, *4*(1), 417–433.

Moindrot, O. (2018). *Triplet loss and online triplet mining in tensorflow.* `https://omoindrot.github.io/triplet-loss`. (visitado en 2022-10-13)

Ojala, T., Pietikäinen, M., y Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, *29*(1), 51–59.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Prakash, J. (2021). *Non maximum suppression: Theory and implementation in pytorch.* `https://learnopencv.com/non-maximum-suppression-theory-and-implementation-in-pytorch/`. (visitado en 2022-10-29)

Pranav, K. (2019, Dec). *Ranked accuracy.* `https://medium.com/analytics-vidhya/ranked-accuracy-11bdaef795e3`. Analytics Vidhya. (visitado en 2022-10-29)

Real academia española: Diccionario de la lengua española, 23.ª ed., [versión 23.4 en línea]. (s.f.). (visitado en 2021-08-08)

Robocup@home. (2022). `https://athome.robocup.org/`. (visitado en 2022-10-20)

Rosebrock, A. (2020). *Object detection: Bounding box regression with keras, tensorflow, and deep learning.* `https://www.pyimagesearch.com/2020/10/05/object-detection-bounding-box-regression-with-keras-tensorflow-and-deep-learning/`. (visitado en 2022-10-29)

Rothe, R., Timofte, R., y Gool, L. V. (2018). Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, *126*(2-4), 144–157.

Salton do Prado, K. (2017). *Face recognition: Understanding lbph algorithm.* `https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b`. (visitado en 2022-10-29)

Savage, J., Martell, R., Estrada, H., Cruz, J., Negrete, M., Marquez, J., ... Cruz, J. (2019). *Pumas@home 2019 team description paper.* `https://biorobotics.fi-p.unam.mx/wp-content/uploads/Publications/Papers/2019/TDPRobocup2019.pdf`. (visitado en 2021-10-11)

Schroff, F., Kalenichenko, D., y Philbin, J. (2015). Facenet. *A unified embedding for face recognition and clustering*.

Serengil, S. I., y Ozpinar, A. (2021). Hyperextended lightface: A facial attribute analysis framework. En *2021 international conference on engineering and emerging technologies (iceet)* (p. 1-4). Descargado de `https://doi.org/10.1109/ICEET53442.2021.9659697` doi: 10.1109/ICEET53442.2021.9659697

Sheoran, V., Joshi, S., y Bhayani, T. R. (2020). Age and gender prediction using deep cnns and transfer learning. En *International conference on computer vision and image processing* (pp. 293–304).

Singh, A. (2020). *Feature engineering for images: A valuable introduction to the hog feature descriptor.* https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/. Analytics Vidhya. (visitado en 2021-10-29)

Sirovich, L., y Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Josa a*, *4*(3), 519–524.

Stanford Artificial Intelligence Laboratory et al. (s.f.). *Robotic operating system.* Descargado de https://www.ros.org

Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., y Wei, Y. (2020). Circle loss: A unified perspective of pair similarity optimization. En *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 6398–6407).

Taigman, Y., Yang, M., Ranzato, M., y Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1701–1708).

Terpstra, T., Owens, T., Hashemian, A., y Voitel, T. (2021). Depth mapping accuracy evaluation of the microsoft kinect v2 motion capture sensor. *J Assoc Crime Scene Reconstr*, 25.

Turk, M., y Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, *3*(1), 71–86.

van der Burgh, M., Lunenburg, J., van Beek, L., Geijsberts, J., Janssen, L., Aleksandrov, S., ... van de Molengraft, M. (2019). *Tech united eindhoven @home 2019 team description paper.* https://www.techunited.nl/uploads/Zorgrobots/Kwalificatie/Tech_United_Eindhoven_TDP_2019.pdf. (visitado en 2021-08-08)

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., y Bengio, Y. (2018). Graph Attention Networks. *International Conference on Learning Representations*. Descargado de https://openreview.net/forum?id=rJXMpikCZ

Venkat, N., y Srivastava, S. (2018). *Ethnicity detection using deep convolutional neural networks.* https://www.researchgate.net/publication/329355417_Ethnicity_Detection_Using_Deep_Convolutional_Neural_Networks. (visitado en 2021-12-01)

Viola, P., y Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. En *Proceedings of the 2001 ieee computer society conference on computer vision and pattern recognition. cvpr 2001* (Vol. 1, pp. I–I).

Yang, S., Luo, P., Loy, C.-C., y Tang, X. (2016). Wider face: A face detection benchmark. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 5525–5533).

Yi, D., Lei, Z., Liao, S., y Li, S. Z. (2014). Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*.

Yohanandan, Shivy. (2020). *map (mean average precision) might confuse you!* https://towardsdatascience.com/map-mean-average

`-precision-might-confuse-you-5956f1bfa9e2`. (visitado en 2022-09-30)

Zhang, K., Zhang, Z., Li, Z., y Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, *23*(10), 1499–1503.

Zhang, Z., Song, Y., y Qi, H. (2017). Age progression/regression by conditional adversarial autoencoder. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 5810–5818).

Zou, Z., Shi, Z., Guo, Y., y Ye, J. (2019). Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*.

Glosario

Análisis de grupos El análisis de grupos examina conjuntos de datos para evaluar si pueden ser resumidos de manera tal que se encuentre un significado asociado a dichos grupos, pudiendo encontrar un número relativamente pequeño de grupos cuyos elementos estén relacionados y que se diferencien de aquellos que se encuentran en otros grupos (Everitt, Landau, Leese, y Stahl, 2011).

Aprendizaje automático Área que busca construir programas que mejoren automáticamente con experiencia (Mitchell y cols., 1990).

Boostrapping Problema de aprendizaje automático en el que se busca inducir un clasificador a partir de una pequeña cantidad de ejemplos anotados y una gran cantidad de ejemplos sin anotar (Abney, 2002).

Bounding box regression Método de aprendizaje profundo para la detección de objetos que permite predecir las coordenadas del cuadro delimitador como valores continuos (Rosebrock, 2020).

Clasificación Proceso de predecir la clase a la que pertenece un conjunto de datos (Asiri, 2018).

Descriptor visual Representación simplificada de una imagen que contiene solamente la información más importante de esta (Singh, 2020).

Exactitud rank-1 Medida que se define como el porcentaje de veces que la clasificación predicha coincide con la etiquetada. Se diferencia de la exactitud rank-N, donde lo que se mide es el porcentaje de veces que la clasificación etiquetada se encuentra en el top N de clasificaciones predichas más probables (Pranav, 2019).

Función de pérdida Las funciones de pérdida computan la distancia entre la salida del algoritmo y la esperada. Esto permite evaluar el desempeño actual del algoritmo, y utilizar esta información como retroalimentación para ajustar su funcionamiento (Chollet, 2021) .

Histograma Representación gráfica de una distribución de frecuencias por medio de rectángulos, cuyas anchuras representan intervalos de la clasificación y cuyas alturas representan las correspondientes frecuencias (rae, s.f.).

Intersección sobre Unión Métrica que permite cuantificar el porcentaje de superposición entre dos cuadros delimitadores. Se calcula como la intersección de los cuadros sobre su unión (Prakash, 2021).

Local Binary Pattern Descriptor visual que dada una imagen, un radio, y un número de vecinos a considerar etiqueta cada uno de los píxeles de la imagen con un número binario. Para calcular este número se le asigna a cada vecino un uno si su valor es mayor o igual al del píxel central, y un cero si es menor (considerando un canal de color). Luego estos números son concatenados en un orden arbitrario, pero que debe respetarse para cada píxel, resultando en un único número binario. Esto se debe repetir para cada píxel de la imagen (Salton do Prado, 2017).

Non-maximum suppression Algoritmo que permite seleccionar una entidad de un conjunto de entidades superpuestas. Recibe como entrada un conjunto de predicciones P, un puntaje de confianza de cada predicción, y un umbral IoU. Como primer paso selecciona la predicción s con mayor puntaje de confianza, quitándola de P. Luego compara las predicciones en P con s y descarta las que tengan un IoU mayor al umbral, y si aún quedan predicciones en P vuelve al primer paso. Esto se repite hasta que P quede vacío (Prakash, 2021).

Paradigma con datos externos Paradigma de entrenamiento con LFW que permite utilizar datos externos etiquetados (*LFW : Results*, 2019).

Paradigma de imágenes restringidas Paradigma de entrenamiento con LFW que permite utilizar únicamente los pares etiquetados provistos (*LFW : Results*, 2019).

Paradigma no restringido Paradigma de entrenamiento con LFW que permite utilizar la información de identidad de las personas para generar pares etiquetados adicionales (*LFW : Results*, 2019).

Paradigma sin datos externos Paradigma de entrenamiento con LFW que no permite utilizar datos externos (*LFW : Results*, 2019).

Puntos fiduciarios Punto de marca visible en el campo de visión de un instrumento óptico, que sirve como referencia o para tomar una medida (fiduc, s.f.).

Redes siamesas Redes neuronales que consisten en dos o más subredes idénticas, con la misma configuración de parámetros y pesos. Son capaces de

determinar si dos imágenes pertenecen a una misma identidad sin necesidad de diversos ejemplos por identidad, y tampoco volver a entrenar la red cada vez que se le presenta una identidad nueva. Para esto aprenden una función de similitud que luego pueden aplicar a cualquier par de imágenes (Benhur J, 2020).

Regresión Consiste en ajustar una función perteneciente a una familia de funciones a los datos de entrenamiento bajo una función de error (Anwar, 2021).

Support Vector Machine Conjunto de algoritmos de aprendizaje supervisado para resolver tareas de clasificación que, dado un conjunto de datos en un espacio, separa dichos datos en dos espacios lo más amplios posibles mediante un hiperplano. Este hiperplano se denomina vector de soporte, y queda definido por los dos puntos más cercanos de cada clase (Mendoza, s.f.).

Support Vector Regression Es una variante del modelo de análisis de SVM, pero en lugar de ser utilizado para clasificar, se usa como esquema de regresión para predecir valores. En este caso, se define un margen de tolerancia alrededor del vector con el objetivo de minimizar el error, tolerando entonces parte de ese error (Avila Camacho, 2020).

Apéndice A

Pruebas de la RoboCup@Home

En este apéndice se presentan las pruebas de la competencia que resultan relevantes para este proyecto, de forma simplificada, haciendo énfasis en los aspectos más relevantes para el proyecto.

A.1. Despedida

Objetivo:

- Alcanzarle a dos invitados sus abrigos y llevarlos de a uno a un taxi.

Observaciones:

- Cada persona tiene asignado un nombre y un abrigo de diferente color.
- Hay cinco personas en la habitación, dos paradas y tres sentadas.

Desarrollo:

1. Dos personas indican simultáneamente al robot que quieren retirarse agitando su mano y gritando.
2. El robot reconoce a las personas que quieren retirarse, va a buscar sus abrigos y se los entrega.
3. El robot identifica al taxista, que viste ropa llamativa y porta un paraguas.
4. El robot escolta a cada invitado al taxi uno a uno y luego retorna a la arena.

Tareas relevantes:

- Detectar al taxista.

- Puntos adicionales: Reconocer a los invitados para entregarles el abrigo correcto.
- Puntos adicionales: Identificar a la invitada mujer.

A.2. Encuentra a mis amigos

Objetivo:
Localizar y describir al menos dos personas sólo conociendo su nombre.
Observaciones:

- Hay cuatro personas en la habitación y cada una tiene un nombre asignado.

Desarrollo:

1. El operador indica el nombre de la persona que quiere localizar.
2. El robot localiza e identifica a la persona.
3. El robot retorna al operador y provee una descripción física unívoca de la persona y la localización de esta utilizando puntos de referencia como muebles.

Tareas relevantes:

- Detectar al operador.
- Reconocer a las personas por su nombre.
- Puntos adicionales: Reconocer y describir un tercer invitado.

A.3. Recepcionista

Objetivo:
Llevar a dos invitados a la sala de estar, presentarlos y ofrecerles un lugar libre para sentarse.
Observaciones:

- Cada invitado tiene un nombre y bebida asignados.
- Hay una persona denominada John que ya se encuentra en la sala y es conocida por el robot.
- Las personas se mueven dentro de la habitación a medida que llegan los invitados.

Desarrollo:

1. El robot pregunta su nombre y bebida preferida a un invitado que acaba de llegar.

2. El robot guía al invitado a la sala y se lo presenta a John. Al presentar al invitado, el robot debe apuntarlo, decir su nombre y su bebida preferida.

3. El robot ofrece un lugar libre para que el invitado se siente.

4. Se deben repetir estos pasos con al menos un invitado, y siempre se debe presentar el nuevo invitado a John y a cada uno de los invitados que llegaron primero.

Tareas relevantes:

- Asociar rostros de los invitados con sus nombres y tragos favoritos.
- Apuntar a la persona que está siendo introducida.
- Puntos adicionales: Ofrecer el sillón a la persona de mayor edad.

A.4. Sirviendo las bebidas

Objetivo:
Tomar órdenes y entregar bebidas a todas las personas que no tengan una.
Observaciones:

- Hay cinco personas en la habitación, dos paradas y tres sentadas.
- Cada una tiene un nombre asignado y, o bien porta una bebida o tiene una orden de bebida asignada.

Desarrollo:

1. El robot identifica a las personas que no tienen una bebida y toma su orden.

2. El robot se dirige al bar para recoger las bebidas y luego se dirige nuevamente hacia los invitados.

3. El robot entrega la bebida correcta al invitado y dice su nombre mientras se la entrega. En caso de tener una bandeja, se asegura de que el invitado esté tomando la bebida correcta.

4. Si la bebida que el invitado había ordenado no estaba disponible, el robot toma la orden nuevamente y realiza los pasos anteriores.

Tareas relevantes:

- Asociar los rostros de los invitados a sus órdenes y nombres.

A.5. Restaurante

Objetivo:
Tomar y servir al menos una orden en un restaurante.
Observaciones:

- La prueba se realiza en un restaurante real, donde pueden haber clientes y mozos reales. Estos pueden llegar a actuar como ruido para el robot al momento de hacer la detección del rostro y gestos de los clientes a los que tiene que atender.

Desarrollo:

1. El robot se dirige a una mesa.
2. El cliente realiza una orden. Mientras esto sucede el robot debe mantener contacto visual, ya que sino el equipo es penalizado.
3. El robot se dirige a la cocina, donde recibe los objetos de la orden.
4. El robot se dirige nuevamente a la mesa para colocar la orden sobre ella.

Tareas relevantes:

- Detectar el rostro de los comensales y mantener contacto visual mientras hacen el pedido.

A.6. Generalidades

En general, cada vez que un robot interactúe con una persona, se considera adecuado que establezca contacto visual. De todas formas, en las pruebas que se encuentran descritas en el manual esto no se explicita, a excepción de la prueba Restaurante, que indica que si no se establece contacto visual al tomar la orden se penaliza al equipo. Las pruebas donde se considera que el único aspecto relevante para el proyecto es mantener contacto visual con las personas, pero esto no es un requisito de la prueba, son excluidas de este apéndice.

Apéndice B

Predicción de características

En este apéndice se presentan las proporciones de los distintos subconjuntos de UTKFace (Z. Zhang y cols., 2017) mencionados en la sección de experimentación (5), así como las proporciones del conjunto original. Luego, se presentan los resultados completos para las tareas de predicción de edad, raza y género. Por último, se especifican los experimentos en escenarios reales de predicción de características, cuya ejecución queda planteada como trabajo futuro.

B.1. Proporciones de los subconjuntos de UTKFace

En la figura B.1 se presentan las proporciones de edad, género y raza de los subconjuntos balanceados confeccionados en este proyecto, para evaluar la predicción de edad y género en las pruebas especificadas en 5.3.4.1 y 5.3.4.2. Además, en la figura B.2 se presentan las proporciones de edad, género y raza para el conjunto original de UTKFace, sin restringir por edad y sin balancear.

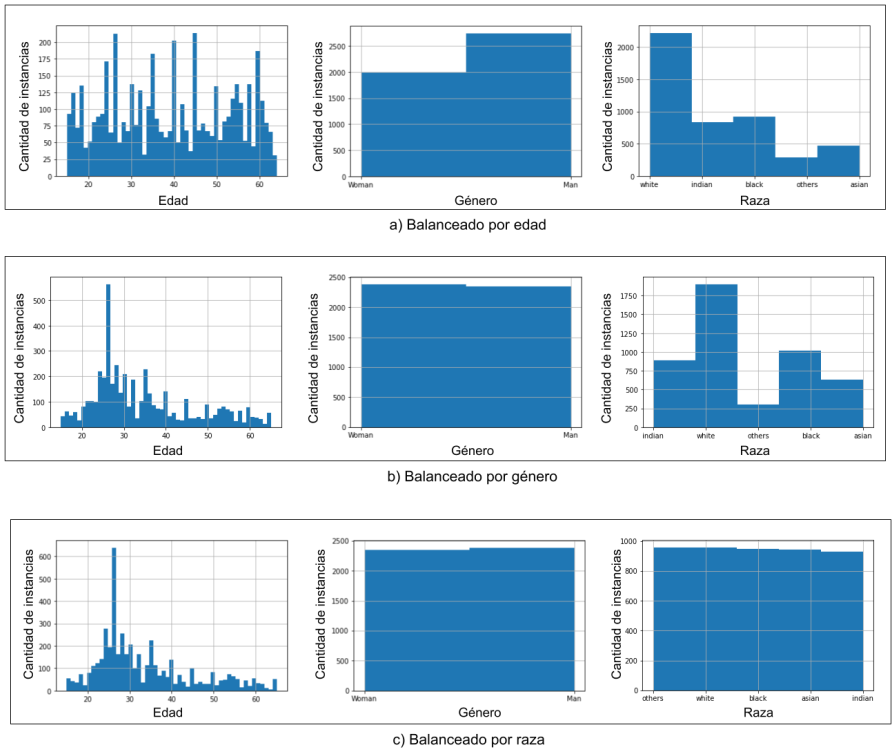


Figura B.1: Proporciones de edad, género y raza para los subconjuntos de UTK-Face (Z. Zhang y cols., 2017) balanceados por edad (a), género (b) y raza (c).

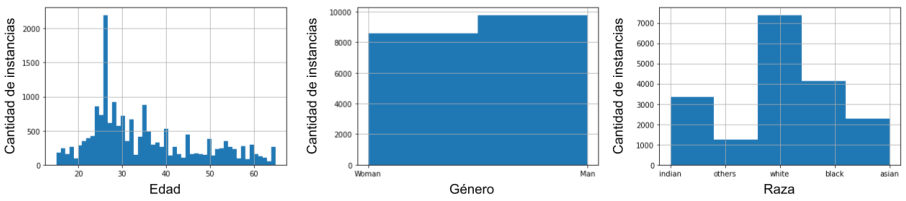


Figura B.2: Proporciones de edad, género y raza para UTKFace (Z. Zhang y cols., 2017).

B.2. Resultados de predicción de características

Se presentan en distintos cuadros los resultados de MAE para la predicción de edad (B.1), los resultados (%) de exactitud (B.2), precisión, recuperación y medida F1 para el género femenino (B.3) y masculino (B.4) en la tarea de predicción de género, y por último, los resultados (%) de exactitud para la predicción de raza (B.5) .

B.2.1. Predicción de edad

Conjunto	MAE ↓
Balanceado por edad	12,2
Balanceado por género	8,7
Balanceado por raza	**8,0**
Unión	9,5

Cuadro B.1: Resultados de MAE por subconjunto para la tarea de predicción de edad. Se resalta en negrita el resultado para el subconjunto en el que se obtiene un menor valor.

B.2.2. Predicción de género

Conjunto	Exac. ↑
Balanceado por edad	**77,6**
Balanceado por género	75,2
Balanceado por raza	74,4
Unión	75,8

Cuadro B.2: Resultados de exactitud (%) por subconjunto para la tarea de predicción de genero. Se resalta en negrita el resultado del subconjunto en el que se obtiene un mayor valor.

Conjunto	Prec. ↑	Rec. ↑	F1. ↑
Balanceado por edad	**98,1**	47,7	64,2
Balanceado por género	97,9	**51,2**	**67,3**
Balanceado por raza	98,4	50,2	66,5
Unión	98,2	49,7	66,0

Cuadro B.3: Resultados (%) de precisión, recuperación y medida F1 por subconjunto para el genero femenino en la tarea de predicción de género. Se resalta en negrita los resultados de los subconjuntos en los que se obtiene un mayor valor para cada medida.

Conjunto	Prec. ↑	Rec. ↑	F1. ↑
Balanceado por edad	**72,3**	**99,3**	**83,7**
Balanceado por género	67,3	98,9	80,1
Balanceado por raza	66,3	99,2	79,4
Unión	68,8	99,2	81,3

Cuadro B.4: Resultados (%) de precisión, recuperación y medida F1 por subconjunto para el genero masculino en la tarea de predicción de género. Se resalta en negrita los resultados de los subconjuntos en los que se obtiene un mayor valor para cada medida.

B.2.3. Predicción de raza

Conjunto	Exac. ↑
Balanceado por edad	**70,7**
Balanceado por género	69,2
Balanceado por raza	66,2
Unión	69,2

Cuadro B.5: Resultados (%) de exactitud por subconjunto para la tarea de predicción de raza. Se resalta en negrita el resultado del subconjunto en el que se obtiene un mayor valor.

B.3. Experimentos de predicción de características

Se especificaron e implementaron parcialmente dos experimentos que no se pudieron ejecutar. Estos se presentan a continuación. Se mantiene la numeración presentada en la sección 5.4, correspondiéndose el experimento 1.4 con el escenario I (5.4.2) y el 2.1 con el escenario II (5.4.3).

B.3.1. Experimento 1.4

Objetivos:

- Ordenar correctamente a tres personas que se presentan individualmente al robot según sus edades.

Flujo:

1. Detectar los rostros que se encuentren en la imagen, descartando aquellos cuyo cuadro delimitador sea menor a 1/16 de la imagen.
2. Seleccionar el más cercano, es decir, aquel cuyo cuadro delimitador sea de mayor tamaño.
3. Predecir la edad de la persona detectada.
4. En caso de que la persona en cuestión no sea la tercera en llegar, esperar por entrada del usuario:
   - En caso de que ingrese s (sí), repetir los pasos 1-4.
   - En caso de que ingrese n (no), detener la ejecución del experimento.
5. Ordenar las personas según sus edades.

Salidas:

- Mensaje indicando el éxito o fracaso del experimento, siendo este mensaje "Éxito" y "Fracaso" respectivamente.
- Tiempo de ejecución del experimento.
- Las edades predichas en el paso 3.
- Las posiciones de las personas según el ordenamiento realizado.

Formato del archivo de salidas esperadas:

```
<posición de la primera persona según el ordenamiento por edad>
<posición de la segunda persona según el ordenamiento por edad>
<posición de la tercera persona según el ordenamiento por edad>
```

Caso de uso en Recepcionista: El robot debe ser capaz de reconocer a la persona de mayor edad para ofrecerle el mejor asiento.

B.3.2. Experimento 2.2

Objetivos:

- Reconocer a las invitadas de sexo femenino.

Flujo:

1. Detectar los rostros que se encuentren en la imagen, descartando aquellos cuyo cuadro delimitador sea menor a 1/16 de la imagen.
2. Predecir el género de las personas detectadas.

Salidas:

- Mensaje indicando el éxito o fracaso del experimento, siendo este mensaje "Éxito" y "Fracaso" respectivamente.
- Tiempo de ejecución del experimento.
- Género de las personas detectadas, en el orden en el que se encuentran en la imagen, de izquierda a derecha.

Formato del archivo de salidas esperadas:

```
<género de la persona que está más a la izquierda en la imagen>
...
<género de la persona que está más a la derecha en la imagen>
```

Caso de uso en Despedida: El robot debe poder reconocer a la invitada mujer que quiere retirarse, para así escoltarla primero al taxi.

Apéndice C

Experimento 1.1 con variaciones

En este apéndice se presentan los resultados de exactitud completos del experimento 1.1 con variaciones. En el cuadro C.1, se consideran las imágenes que tienen exactamente una variación. En el C.2 y C.3, se consideran las imágenes que tienen una combinación de al menos dos variaciones, donde el primero muestra cada combinación, mientras que el segundo resume las ejecuciones por variación.

Variación	Exac. ↑ Esler	Exac. ↑ Deepface	Total
L. de sol	100	100	3
Lentes	100	100	4
Tapaboca	100	100	4
Auriculares	100	100	2
Cerquillo	100	100	1
Gorro	66,7	66,7	3
Pelo hacia atrás	100	100	2
Luz lateral	87,8	86,7	90

Cuadro C.1: Resultados de exactitud (%) para Esler y Deepface por variación, y el total de imágenes para cada una. Las imágenes utilizadas tienen exactamente una variación.

Var. 1	Var. 2	Var. 3	Exac. ↑ Esler	Exac. ↑ Deepface	Total
L. de sol	Tapaboca	-	0	0	2
Lentes	Cerquillo	Luz lateral	100	100	1
Lentes	Sombrero	-	100	100	1
L. de sol	Gorro	Bufanda	100	100	1
Lentes	Tapaboca	-	100	0	2
L. de sol	Gorro	-	100	100	2
L. de sol	Gorro	Tapaboca	0	0	1
Lentes	Gorro	Bufanda	100	100	2
Lentes	Auriculares	-	100	100	2
L. de sol	Pelo hacia atrás	-	100	100	1
Bufanda	Cerquillo	-	100	100	1
Bufanda	Cerquillo	Luz lateral	100	50	2
Lentes	Gorro	-	100	50	2
Lentes	Gorro	Tapaboca	0	0	1
Lentes	Pelo hacia atrás	-	100	100	1
Lentes	Luz lateral	-	100	100	3
Cerquillo	Luz lateral	-	100	100	1

Cuadro C.2: Resultados de exactitud (%) para Esler y Deepface por combinación de variaciones, y total de imágenes por combinación. Las imágenes utilizadas tienen al menos dos variaciones.

Variación	Exac. ↑ Esler	Exac. ↑ Deepface	Total
L. de sol	57,1	57,1	7
Lentes	93,3	73,3	15
Tapaboca	33,3	0	6
Bufanda	100	83,3	6
Pelo hacia atrás	100	100	2
Cerquillo	100	80,0	5
Sombrero	100	100	1
Gorro	77,8	66,7	9
Auriculares	100	100	2
Luz lateral	100	100	7

Cuadro C.3: Resultados de exactitud (%) para Esler y Deepface por variación, y total de imágenes por cada una. Las imágenes consideradas tienen al menos dos variaciones.