UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

SISTEMA DE VERIFICACIÓN DE IDENTIDAD ENFOCADO EN IMÁGENES DE ROSTROS AFECTADAS POR OCLUSIONES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

JOAQUÍN MAXIMILIANO ALIAGA GONZÁLEZ

PROFESOR GUÍA:
JOSE MANUEL SAAVEDRA RONDO

MIEMBROS DE LA COMISIÓN:
SERGIO FERNANDO ASTUDILLO TORRES
FRANCISCO JAVIER RIVERA SERRANO

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE: Ingeniero Civil Eléctrico
POR: **JOAQUÍN MAXIMILIANO ALIAGA GONZÁLEZ**
FECHA: 30 de marzo de 2021
PROF. GUÍA: JOSE MANUEL SAAVEDRA RONDO

SISTEMA DE VERIFICACIÓN DE IDENTIDAD ENFOCADO EN IMÁGENES DE ROSTROS AFECTADAS POR OCLUSIONES

En el presente trabajo se busca proponer un sistema basado en inteligencia artificial para resolver el problema que presenta la empresa GeoVictoria S.A, correspondiente a verificar la identidad de personas utilizando rostros que están ocluidos por mascarillas.

Para ello, primero es necesario obtener una base de datos de entrenamiento, utilizando datos públicos, y generar una base de datos de evaluación utilizando los datos privados, proporcionados por la empresa, la cual está separada en 4 casos para comprender el comportamiento tanto del modelo base como de los modelos propuestos en cada uno.

Luego se realiza una propuesta de varios modelos basados en las últimas publicaciones científicas relacionadas, los cuales son entrenados y se obtienen sus resultados, tanto durante el entrenamiento como en la etapa de evaluación.

Al analizar los resultados obtenidos se obtiene que los modelos, cuya arquitectura está basada en redes tripletas, mejoran considerablemente las métricas de evaluación al verificar rostros que están ocluidos por mascarilla. Sin embargo, estos presentan métricas muy por debajo del modelo base en los casos en que no existen oclusiones.

Por otra parte, el modelo actual de la empresa muestra tener los mejores resultados en todos los casos evaluados, excepto cuando existen oclusiones, lo cual afirma la hipótesis inicial de que el modelo falla ante este escenario.

En función de los resultados se propone un sistema compuesto por el modelo de la empresa y por la arquitectura basada en tripletas, los cuales debiesen funcionar de manera separada dependiendo del caso.

De manera general, se cumple el objetivo de proponer una arquitectura basada en inteligencia artificial para mejorar los resultados en la verificación de identidad cuando el rostro está ocluido por el uso de mascarilla.

A mi yo pasado, por el esfuerzo y el aguante,
a mi yo futuro, porque los desafíos nunca paren.

Agradecimientos

Parto agradeciéndome, por haber soñado en grande, por el esfuerzo, la dedicación, por las decisiones correctas y los errores. Por levantarme siempre, a pesar de todo. Por tener la valentía de seguir.

Le agradezco a mi familia, en especial a mi madre y mi hermana, las mujeres de mi vida, mis ángeles guardianes. Por haber estado siempre ahí para mi, por su apoyo y amor incondicional. Por sus consejos, sus retos, sus palabras de aliento, por haberme demostrado siempre lo orgullosas que estaban de cada uno de mis logros.

Te agradezco Lissett, por tu compañía, por caminar juntos, por sacarme del camino cómodo y mostrarme otros destinos, otros sueños. Gracias por haber estado ahí, ayudándome, escuchándome, soportándome. Gracias por creer en mí cuando ni yo mismo lo hacía. Gracias por ser esa niña que despierta mi niño interior y me invita a jugar, a conocer, a descubrir. El hogar que hemos formado es el lugar donde quiero estar, el escondite que me da cobijo cuando el mundo se me hace muy grande.

A los “me lleva”, por no tomarse todo tan en serio, por las risas, los carretes, las historias, la confianza. Por tantos entrenamientos aguantados, por las peleas y el apoyo, por sudarla codo a codo hasta el último segundo.

A mi padre, por la relación que hemos ido cultivando, mejorando, arreglando. Por haberme enseñado a ser curioso, a querer aprender como funciona todo, por enseñarme que hay que desarmar para armar.

Por último al Taekwondo y todas las personas que lo componen, por haberme enseñado a resistir, a seguir pateando aunque el cuerpo ya no pueda más. El tatami siempre fue un espacio en que lo único importante era el aquí y el ahora, todo lo demás quedaba fuera.

Mente fría y corazón en llamas.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Antecedentes

A diferencia del trabajo tradicional, en donde un empleado realiza sus tareas dentro de un espacio establecido en la oficina de la empresa, hoy en día existen compañías que proporcionan trabajos fuera de sus dependencias físicas, como es el caso de reponedores que pertenecen a una misma empresa pero desarrollan sus labores en distintos supermercados.

Para este tipo de empresas, se vuelve un desafío el hecho de controlar la asistencia de su personal, puesto que sus trabajadores no están concentrados en un mismo espacio o inclusive deben cambiar de lugar durante su jornada laboral.

El control de asistencia se hace necesario en distintos rubros, sobre todo en los tipos de trabajos antes mencionados, pues impacta directamente sobre distintas áreas, como la calidad del trabajo ofrecido por la compañía, la distribución de tareas, los pagos de sueldos, el cobro de la empresa por los servicios prestados, entre otros.

Un mal control de la asistencia del personal puede conllevar a errores en los pagos de sueldos, lo cual deriva en pérdidas para la empresa cuando se pagan más días u horas de las realmente trabajadas o problemas legales cuando sucede lo contrario. Por otra parte, se podrían generar problemas en la ejecución del trabajo al considerar, por ejemplo, como asistente a una persona que no está.

Actualmente, en el mercado existen soluciones para control de asistencia que requieren la instalación de una máquina para su función y utilizan distintas tecnologías como timbraje de tarjetas, uso de huella dactilar, de biometría ocular, tarjetas con tecnología RFID, etc.

Sin embargo, dichas soluciones no se ajustan al contexto de estas empresas, puesto que se tendría que comprar una máquina de control para cada uno de los lugares en que los empleados trabajan, lo cual representa una inversión considerable. Además, esta vía supone mayor complejidad puesto que sería necesario un sistema central que obtenga la información de las distintas máquinas.

1.2. Relevancia de la investigación

En las últimas décadas, se ha visto un incremento considerable de dispositivos móviles con procesadores integrados de alta capacidad de cómputo y con conexión a internet, lo cual los transforma en una excelente herramienta de procesamiento.

Sumado a lo anterior, el acceso a este tipo de tecnologías, por parte de la población mundial, ha aumentado vertiginosamente, llegando al punto en que cada persona posee por lo menos 1 de estos dispositivos.

Por tanto, la creación de servicios que utilicen tanto la capacidad de dichos aparatos como la información disponible en internet, se convierte en una excelente oportunidad de negocios y de desarrollar nuevas tecnologías que hagan más cómodo el diario vivir.

Por otro lado, es necesario que los estudios científicos salgan de los ambientes controlados de los laboratorios académicos y se prueben en ambientes industriales o “de la vida real”, puesto que es en ese punto en que dichos estudios generan realmente un avance en la sociedad.

En síntesis, la importancia de este trabajo radica en la oportunidad de desarrollar un sistema que interactúa directamente con las personas y en el desafío de enfrentar problemas no considerados en un contexto de laboratorio.

1.3. Motivación

En base a los problemas antes descritos, surge la necesidad de resolver estos problemas utilizando las últimas investigaciones realizadas en el campo de visión computacional y tratando de llevarlas a un ambiente de uso real, en que no necesariamente se cumplen las condiciones impuestas en un laboratorio.

Por tanto, la motivación de este trabajo se forja en el interés de llevar las investigaciones a aplicaciones industriales que interactúan directamente con las personas, para así impactar en el desarrollo, tanto industrial como social.

De manera más específica, el hecho de desarrollar sistemas que utilizan imágenes para generar “máquinas inteligentes” genera un alto interés de parte del autor, puesto que se relaciona directamente con el campo de estudio.

1.4. Formulación del problema

La empresa GeoVictoria S.A, que presta servicios de gestión de asistencia a otras empresas, ha desarrollado un sistema que utiliza herramientas de inteligencia artificial para la verificación de la identidad del trabajador, utilizando distintas fuentes de datos como biometría

facial o de voz, el cual ha demostrado tener bastante efectividad en ambientes controlados [1]. Dichas herramientas corresponden a modelos de redes neuronales que son entrenadas con una amplia cantidad de datos.

Este sistema presenta mejoras significativas respecto a las soluciones tradicionales dado que no se requiere la instalación de máquinas en los lugares de trabajo, sino que basta con que los trabajadores porten un teléfono celular con cámara y acceso a internet. Tampoco es necesario que dichos dispositivos tengan una gran capacidad de cómputo, puesto que el procesamiento es realizado en servidores ubicados *en la nube*.

Sin embargo, en el contexto específico de biometría facial, existen ciertas dificultades que enfrenta este servicio y este trabajo de titulo se centra en una de ellas.

Esta dificultad consiste en tener rostros que están ocluidos por el uso de mascarillas, fenómeno que no había sido caso de estudio en Chile[2], pero que ha tenido un aumento considerable debido a la pandemia del COVID-19, que ha generado la necesidad de utilizar mascarilla en los espacios de trabajo. Esto supone un problema, puesto que el uso de mascarilla conlleva que se tapen elementos de información valiosa como lo son la boca, las mejillas y la nariz, lo cual genera que los modelos, entrenados con imágenes de rostros totalmente descubiertos, disminuyan considerablemente su tasa de verificaciones correctas y, por tanto, dejan de ser atractivas para las empresas que contratan estos servicios.

1.5. Objetivos

El objetivo general es desarrollar un modelo de inteligencia artificial para hacer verificación de identidad del personal dependiente de una empresa, donde los rostros vienen afectados por problemas de oclusión por mascarillas.

Como objetivos específicos se tienen:

- Generación de una base de datos de entrenamiento y de evaluación, a partir de datos públicos y de datos proporcionados por la empresa.
- Propuesta de una arquitectura basada en redes neuronales con enfoque en mejorar los resultados de verificación en casos con mascarilla.

1.6. Alcances

Como parte de los alcances de este trabajo se considera la creación de una base de datos de evaluación utilizando los registros de uso del servicio de la empresa, la cual se utilizará para medir el rendimiento del sistema actual y del sistema propuesto.

[1] Ambiente controlado: Situación en que se pueden obtener datos de la persona siempre en el mismo contexto, por ejemplo, una foto directa del rostro tomada por la cámara de un celular

[2] Dado que no existe la costumbre cultural de utilizar mascarilla en la vida cotidiana, al contrario de otros países como China.

Además, se considera la creación de una base de datos de entrenamiento que contenga imágenes de personas con y sin mascarilla, la cual se confeccionará a partir de datos públicos disponibles en *la web*.

En la misma línea, se considera la propuesta, implementación y evaluación de una arquitectura basada en redes neuronales que haya sido probada anteriormente en otras investigaciones científicas. Dicha arquitectura tendrá como objetivo mejorar los resultados de verificación en el caso de personas que utilizan mascarilla, respecto al sistema actual.

Finalmente, se considera parte de este trabajo un análisis de las fuentes de error y de los resultados obtenidos, añadiendo una propuesta de mejora y de avances futuros.

No se consideran parte de este trabajo la optimización de los hiperparámetros del modelo propuesto, ni la puesta en producción del mismo.

1.7. Estructura del documento

El presente documento se estructura de la siguiente manera:

1. Comienza con un Resumen, que engloba los aspectos más importantes de la memoria
2. Sigue con la presente Introducción, en que se da cuenta de los antecedentes, la relevancia de la investigación, la motivación, se formula el problema, los objetivos y una breve descripción metodológica.
3. Luego se presenta el Marco Teórico, en que se detallan los conceptos necesarios y suficientes para abordar el desarrollo de la memoria.
4. Posteriormente se dan a conocer las investigaciones más recientes pertinentes al caso, en el capítulo de Estado del Arte
5. A continuación se explica con detalle la Metodología utilizada y cada una de las etapas llevadas a cabo
6. Posterior a ello se da cuenta de los Resultados obtenidos y se hace un análisis de estos en forma de Discusión
7. Por último se dan las Conclusiones respecto al trabajo realizado, en torno al cumplimiento o no de los objetivos propuestos
8. Finalmente se detalla la Bibliografía utilizada

Capítulo 2

Marco Teórico y Estado del Arte

2.1. Marco Teórico

2.1.1. Algoritmos de aprendizaje

Según Goodfellow et al. [1], un algoritmo de aprendizaje de máquinas es “un algoritmo que es capaz de aprender de los datos”. Pero ¿qué significa que un algoritmo aprenda? Mitchell [2] lo define como “Un programa de computador se dice que aprende de una experiencia E respecto a algún tipo de tarea T y una medida de desempeño P, si su desempeño en T, medido por P, mejora con la experiencia E”.

2.1.2. Tipos de aprendizaje de máquinas

2.1.2.1. Aprendizaje Supervisado

La experiencia E en los algoritmos de aprendizaje supervisado corresponde a bases de datos compuestas de características y, además, de etiquetas u objetivos[1]. Por tanto el algoritmo intenta aprender la forma de relacionar las características de una entrada con su etiqueta correspondiente. El término tiene su origen en la perspectiva de un “profesor o instructor” que permitiéndole ver la etiqueta correcta al sistema, lo guía por el procedimiento correcto.

2.1.2.2. Aprendizaje No Supervisado

La experiencia E en los algoritmos de aprendizaje no supervisado corresponde a bases de datos que contienen muchas características y en que el objetivo del algoritmo es aprender propiedades útiles de la estructura de los ejemplos disponibles, para intentar implícita o explícitamente aprender la distribución de probabilidad $p()$ o algún atributo interesante de dicha distribución[1]. En contraste con el caso supervisado, este término se origina en el hecho de que el algoritmo debe aprender a representar los datos por si mismo, puesto que cada entrada de los datos no tiene una etiqueta asignada.

2.1.3. Oclusiones en imágenes

Las oclusiones corresponden a objetos, reales o artificiales, que cubren información importante de la imagen. En el contexto de imágenes de rostros de personas, se pueden definir

6 tipos diferentes de oclusión[3]:

- Accesorios faciales: corresponden a objetos que las personas utilizan en su rostro como anteojos, lentes de sol, bufandas, gorros, etc.
- Oclusiones externas: pueden ser otras partes del cuerpo como las manos u otros objetos externos.
- Campo de visión limitado: casos en que se observa solo una fracción del rostro.
- Auto-oclusiones: casos en que la persona no está de frente a la cámara sino que la tiene rotada.
- Iluminación extrema: casos en que una parte del rostro está muy iluminada respecto al resto.
- Oclusión artificial: objetos creados de manera digital que ocluyen la imagen, como rectángulos negros/blancos o ruido aleatorio.

2.1.4. Métricas de evaluación para sistemas de verificación de identidad

Para un sistema que funciona con biometría, es necesario enrolar a los usuarios que están autorizados por dicho sistema. Se entiende por enrolar (en un contexto biométrico), el proceso en que se toman los datos que identifican a una persona, como su nombre, fotografías de su cara, fotografías de su iris, huella dactilar, entre otras.

Dichos datos son obtenidos una sola vez y se guardan en una base de datos. Luego al momento en que un usuario desea ser autorizado, debe entregar información biométrica, por ejemplo una fotografía de su rostro, con lo cual el sistema buscará en su base de datos la imagen que fue utilizada durante el enrolamiento y entregará un porcentaje de confianza de cuán parecidas son ambas imágenes.

Para validar que la persona sea efectivamente quién dice ser, se define un valor mínimo que debe ser a lo menos igualado por el porcentaje de confianza para que la persona sea considerada válida. Este umbral mínimo es conocido como la **sensibilidad** o **confianza mínima** del sistema. Por tanto, un nivel mayor de sensibilidad implica que la exigencia para autorizar a un usuario es mayor.

2.1.4.1. Tasas de error en sistemas biométricos

Este sistema de ingreso puede incurrir en fallas, por tanto, es necesario tener métricas para evaluar su desempeño y buscar mejorarlas en caso de que sea necesario. En esta línea, Conrad et al. [4] definen 3 métricas para evaluar un sistema biométrico, las cuales son:

- **Tasa de falsos rechazos o falsos negativos**: conocida comúnmente por FNR (por sus siglas en inglés), corresponde al caso en que una persona que está efectivamente autorizada, es rechazada por el sistema verificador. Esto también se conoce como *error de tipo I*.

- **Tasa de falsas aceptaciones o falsos positivos:** conocida comúnmente por FPR (por sus siglas en inglés), corresponde al caso en que una persona que no se encuentra en los archivos del sistema, es autorizada por el verificador. Esto también se conoce como *error de tipo II*.

- **Tasa de error de crossover** (CER por sus siglas en inglés): también conocida como *equal error rate* (EER), describe el desempeño general del sistema biométrico. A medida que la sensibilidad del sistema aumenta, la tasa de falsas aceptaciones disminuye pues el sistema es más riguroso. Sin embargo, también puede aumentar la tasa de falsos rechazos, por la misma causa. En caso contrario, la FNR disminuye, pero existe la posibilidad de que la FPR aumente. Por tanto se define el EER como el punto en que ambas tasas de error se igualan, como se observa en la Figura 2.1.

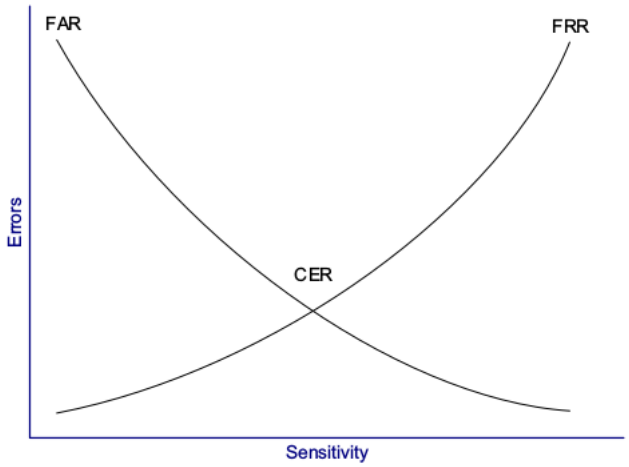


Figura 2.1: Tasa de error de crossover (o EER)[4]. Las curvas FAR y FRR corresponden a FPR y FNR, respectivamente.

2.1.4.2. Curva de ROC

Se define la **tasa de verdaderos positivos** (TPR por sus siglas en inglés) como la cantidad porcentual de veces en que el sistema acepta a una persona que está efectivamente autorizada.

Usando dicho valor junto a la tasa de falsos positivos, se puede construir otra métrica que sirve principalmente para comparar modelos y determinar de manera general si uno es mejor que el otro. Esta métrica se llama *Receiver Operating Characteristic* (ROC) y se obtiene tomando un arreglo de distintos umbrales (o niveles de confianza) y calculando las tasas TPR y FPR para cada umbral.

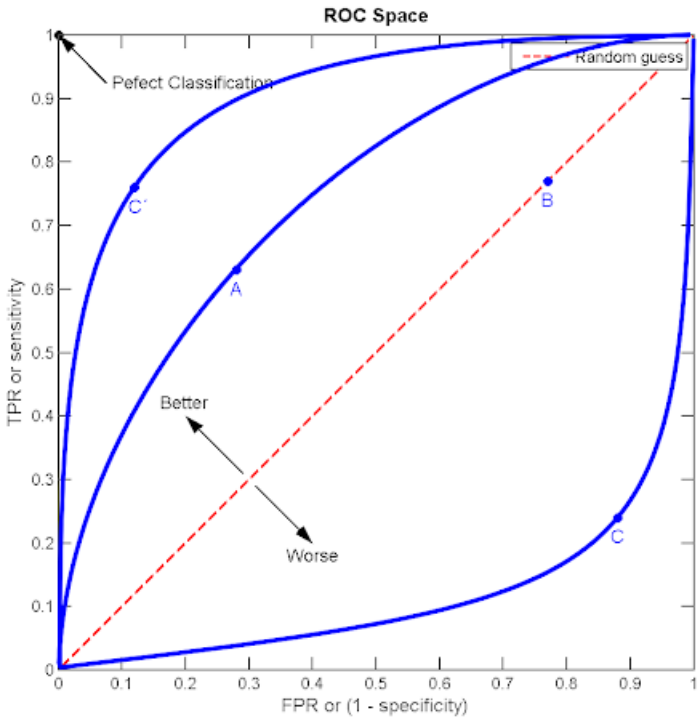


Figura 2.2: Curva de ROC.

En la Figura 2.2 se observa un diagrama clásico de una curva de ROC. De la inspección visual de esta se pueden obtener 3 resultados.

- Un modelo cuya curva está sobre otra es un mejor modelo, puesto que para un mismo nivel de error de falsos positivos, se tiene una mayor tasa de verdaderos positivos.
- Un modelo cuya curva está en la diagonal es idéntico a modelar su salida como un evento aleatorio con igual probabilidad de ocurrencia para cada clase, por ende el modelo en sí no aporta mayor información al problema.
- Un modelo cuya curva está debajo de la diagonal es el peor escenario posible, puesto que ni siquiera mejora los resultados comparados con una salida aleatoria con igual probabilidad de ocurrencia para cada clase.
- El modelo teóricamente perfecto será aquel cuya curva sea una vertical, es decir, que logra la máxima tasa de verdaderos positivos para cualquier valor de falsos positivos.

Puesto que la inspección visual no siempre entrega resultados claros, lo más común es computar el **área bajo la curva** (AUC) y elegir el modelo según cual tenga mayor área. Otro criterio es elegir una tasa de falsos positivos aceptable y elegir el modelo que tenga mayor tasa de verdaderos positivos.

2.1.5. Redes neuronales convolucionales

En el campo de visión por computadora, lo que se busca, en esencia, es obtener la máxima cantidad de información de una imagen o de una serie de imágenes consecutivas (como un

video). Hasta la década pasada, lo común era enfocar los recursos y las investigaciones en la **ingeniería de características**, cuya misión era desarrollar técnicas para extraer características (información) de la imagen.

Así es como se desarrollaron los filtros para detectar bordes como el filtro de Gabor [5], de Canny [6] o el uso de histogramas como vectores para representar imágenes[7], entre otras. El problema principal de estas técnicas es que se utilizaban muchos recursos para desarrollarlas y eran poco generalizables, es decir, funcionaban sólo para ciertos tipos de imágenes, con cierta luminosidad, ángulo, etc.

En este sentido, las redes neuronales convolucionales marcaron un hito sin precedentes, puesto que utilizando parámetros entrenables fueron capaces de generar representaciones de las imágenes con el único requerimiento de tener una cantidad suficiente de imágenes durante el entrenamiento. Estos modelos demuestran tener una alta capacidad de representación semántica y son útiles para diversas tareas como clasificación [8], detección [9] o segmentación [10], entre otras.

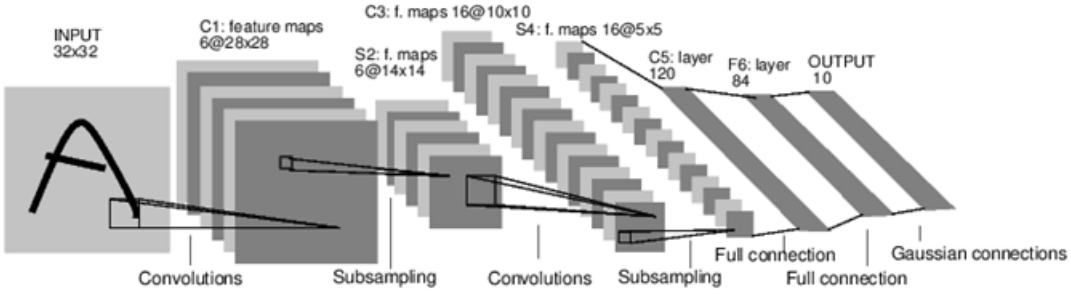


Figura 2.3: Arquitectura básica de una red neuronal convolucional.

En la Figura 2.3 se observa una estructura típica de una red convolucional, la cual está compuesta principalmente por una seguidilla de convoluciones con distinto tamaño de entrada y salida.

Existen arquitecturas de redes convolucionales ampliamente conocidas. Dentro de las contemporáneas está **AlexNet** (2012) [11], una red compuesta por 8 capas y 60 millones de parámetros entrenables, la cual se hizo famosa luego de ganar la competencia ImageNet y que revolucionó el estado del arte al utilizar nuevas funciones de activación conocidas como *Rectified Linear Units*. A partir de esta red se empezaron a desarrollar arquitecturas mucho más "profundas" (mayor cantidad de capas). Dentro de estas está **VGG-16**[12], que logró superar los resultados de AlexNet utilizando 16 capas.

En la Figura 2.4 se observa una representación simplificada de AlexNet y VGGNet.

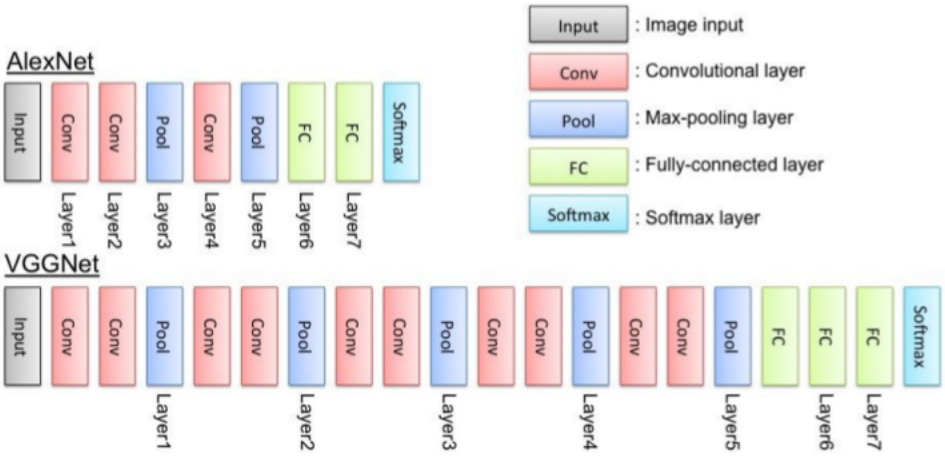


Figura 2.4: Representación simplificada de AlexNet y VGGNet.

Una arquitectura que cambió radicalmente la tendencia que existía hasta la época, de crear redes cada vez con más capas, fue **Resnet**[8], cuya principal novedad fue agregar *conexiones residuales*, consistentes en agregar los resultados de capas anteriores en capas siguientes.

Dado lo anterior, se obtiene la capacidad de añadir más información de entrada para las capas posteriores sin necesidad de aumentar el tamaño de la red. Por esta razón, estas conexiones se conocen como *skip connections*. Cabe destacar que estas conexiones evitan el problema de *vanishing gradient* y además, con el uso de *batch normalization*, se evita el sobreajuste.

En la Figura 2.5 se observa la arquitectura de la Resnet50. En la notación, **B** corresponde a una capa de *batch normalization*, **R** a la función de activación **ReLU**, *max pool 3x3* corresponde a tomar el máximo por regiones de 3x3 y *global average pool* a reemplazar una matriz completa (un canal) por su valor máximo.

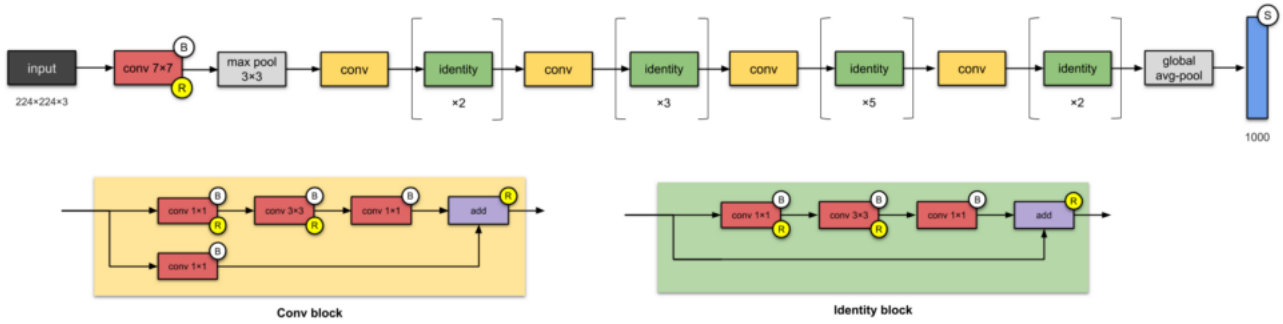


Figura 2.5: Representación de la arquitectura de Resnet50.

2.1.6. Redes Siamesas

Se conoce como redes siamesas a aquellas arquitecturas compuestas por dos subredes idénticas que comparten la gran mayoría o la totalidad de sus pesos y son entrenadas de manera simultánea. A las subredes que componen la arquitectura se les conoce como **backbone**.

Son especialmente útiles para problemas en los que se necesita conocer el nivel de similitud entre 2 imágenes, por ejemplo, en el caso de verificación de identidad utilizando rostros, se utiliza una imágen base (o ***anchor***) y una imagen de prueba, ambas imágenes son las entradas a la red siamesa y esta entrega dos vectores de características (o ***embeddings***).

Finalmente, para conocer la similitud entre ambos rostros basta con utilizar alguna función de distancia (como la distancia Euclideana) o de similitud (como la similitud coseno) entre ambos *embeddings*.

En la Figura 2.6 se observa una representación de red siamesa, en que todos los pesos de los *backbones* son compartidos, es decir, existe sólo una subred que es utilizada dos veces.

El entrenamiento de estas redes se hace con pares de imágenes, los cuales pueden ser de dos tipos:

- Par positivo: tanto el *anchor* como la imagen de prueba pertenecen a la misma clase.
- Par negativo: el *anchor* es de una clase distinta a la imagen de prueba.

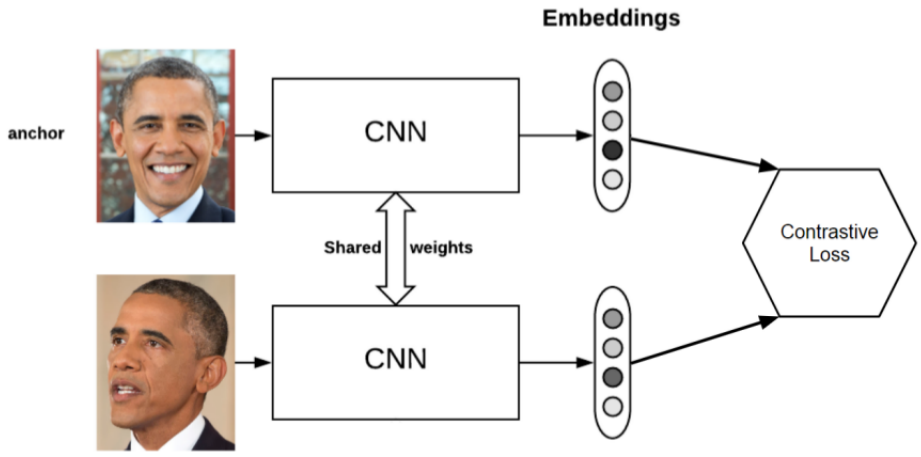


Figura 2.6: Representación de una Red Siamesa para el caso de verificación de identidad utilizando rostros.

Finalmente, la función a optimizar en el caso de las redes siamesas se conoce como ***contrastive loss***, cuya expresión matemática se detalla en la ecuacion 2.1

$$L_i = y_i D_{w,i}^2 + (1 - y_i)\{\max(0, \lambda - D_{w,i})\}^2 \tag{2.1}$$

Donde $D_{w,i}$ corresponde a la distancia L2 entre los *embeddings* y λ al margen entre dichos

vectores, como se expresa en la ecuación 2.2

$$D_{w,i} = L_2(f_w(x_i^1), f_w(x_i^2)) \tag{2.2}$$

En términos generales, el *contrastive loss* está diseñado para minimizar la distancia entre los *embeddings* de pares positivos y maximizarla en el caso de los negativos.

2.1.7. Redes Tripletas

Este tipo de redes sigue la misma filosofía que las redes siamesas, pero en este caso se utilizan 3 subredes. Al igual que en el caso anterior, estas subredes son idénticas y comparten la mayoría o la totalidad de sus pesos.

La diferencia en este caso es que la red es entrenada con tríos de imágenes donde siempre se tendrá el *anchor*, el par positivo y el par negativo. En la Figura 2.7 se observa una representación de una Red Tripleta.

Está arquitectura fue diseñada pensando en los casos en que puede existir alta similitud visual entre el par positivo y el par negativo y cuyas diferencias son sólo detalles. Por ejemplo, en el caso de la verificación por rostros, de manera abstracta 3 fotos de rostros pertenecen a la clase persona, pero se diferencian en las características faciales y, en muchos casos, dichas características pueden diferenciarse sólo en algunos detalles que incluso para el ojo humano puede ser complejo de distinguir.

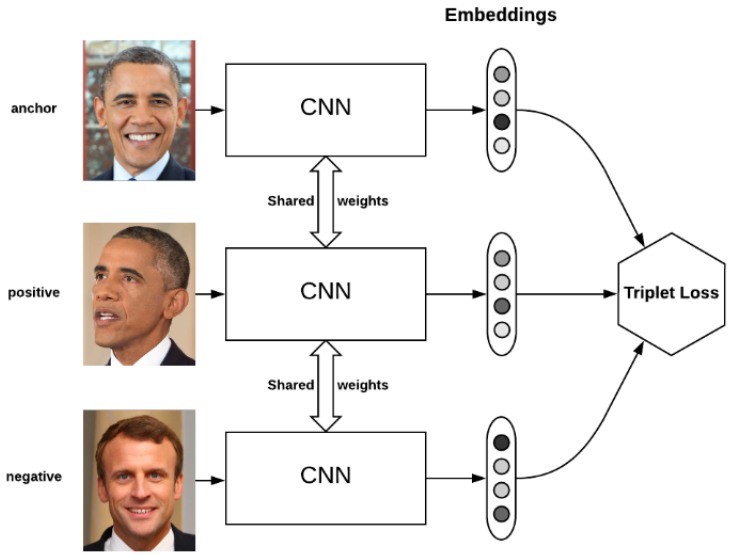


Figura 2.7: Representación de una Red Tripleta para el caso de verificación de identidad utilizando rostros.

La función objetivo se conoce como ***Triplet Loss*** y su tarea es minimizar la distancia entre el par positivo y maximizar la distancia entre el par negativo, manteniendo un margen λ entre ellos. Matemáticamente esto se escribe como la ecuacion 2.3

$$D_{w,i}(a,+) + \lambda < D_{w,i}(a,-) \tag{2.3}$$

Por tanto la función de *Loss* es de la forma:

$$L_i = \max(D_{w,i}(a,+) - D_{w,i}(a,-) + \lambda, 0) \tag{2.4}$$

Donde λ es un hiperparámetro del sistema.

2.2. Estado del arte

2.2.1. Detección de rostros en imágenes

En los casos de uso real en que se quiere verificar la identidad de una persona por medio de una imagen, primero es necesario detectar el rostro de dicha persona. En este sentido, la detección de rostros es un problema bastante estudiado y, en general, bien resuelto para una gran variedad de escenarios.

Un detector de objetos, propuesto en el año 2001, conocido como ***Haar Cascade*** [13] fue uno de los trabajos en aprendizaje de máquinas que causó un antes y un después en la detección de rostros dado los altos resultados obtenidos para el poder computacional existente en aquella época.

Este detector propuso 3 innovaciones: la primera corresponde a un nuevo tipo de representación de la imagen llamado *Imagen integral*, el cual permite una computación más eficiente. El segundo es un sistema de aprendizaje basado en el algoritmo *AdaBoost*, el cual selecciona un número reducido de características criticas y produce clasificadores extremadamente eficientes. Por último, la tercera innovación es un método para combinar de manera creciente más clasificadores en forma de cascada para enfocar la detección sobre las zonas importantes y descartar rápidamente el fondo de la imagen.

Con la irrupción de los modelos basados en redes convolucionales y el aumento del poder computacional, se han desarrollado nuevos modelos que superan significativamente los resultados obtenidos por modelos antiguos. Entre ellos están ***RetinaFace*** [14] y ***Multi-task Cascaded Convolutional Networks*** (MTCNN) [15], los cuales son los que tienen los mejores resultados hasta la fecha.

Respecto a este último, está compuesto por 3 etapas conocidas como redes P, R y O. La primera etapa (P), se encarga de generar regiones propuestas que pueden ser de interés. La segunda etapa (R) se encarga de refinar las regiones de la etapa anterior y la tercera etapa (O) entrega el resultado final, compuesto por el *bounding box* que enmarca el rostro y los puntos de interés del mismo como ojos, boca, labios, etc. En la Figura 2.8 se observa un diagrama explicativo del proceso ocurrido en cada etapa.

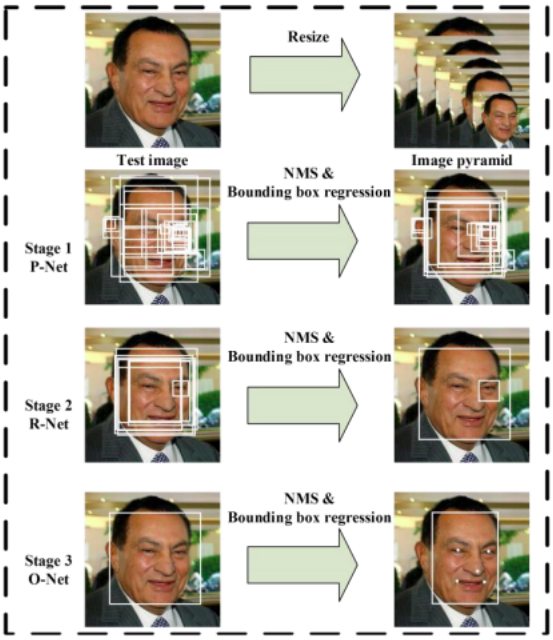


Figura 2.8: Diagrama explicativo del proceso ocurrido en la red MTCNN compuesto por las subredes o etapas P, R y O.

2.2.2. Verificación de identidad en presencia de oclusiones

Según Zeng et al. [3], se pueden definir dos enfoques generales en los estudios de verificación de identidad en presencia de oclusiones. Estos son los modelos enfocados en el **reconocimiento de una fracción del rostro** y los enfocados en la **detección de la oclusión**.

2.2.2.1. Reconocimiento de una fracción del rostro

Este enfoque busca construir un vector de características robusto y que logre capturar la información solamente de las zonas no ocluidas.

Una investigación que generó un avance significativo en esta área es la hecha por Liao et al. [16] en que diseñan un modelo que no necesita alinear el rostro (como lo hacían la mayoría de las investigaciones anteriores a esta). Para ello, utilizan descriptores múltiples de puntos clave (*MKD* por sus siglas en inglés) que generan un vector de características cuya dimensión depende de la información disponible en la imagen. Si bien esta investigación es considerada un hito en el área, se requieren varias imágenes de rostros en distintas posiciones para cubrir todas las variantes posibles, lo cual lo hace muy sensible a los datos de entrenamiento.

Utilizando técnicas recientes, He et al. [17] proponen un modelo de Red convolucional multiescala con entrenamiento supervisado (*MDSCNN* por sus siglas en inglés). Este trabajo comprende la utilización de varias redes entrenadas con distintas zonas de la cara para extraer sus características y formar un vector, combinando los resultados de cada red. Si bien demostró tener buenos resultados, se necesitaron 55 redes para lograrlo, por tanto, no es una solución viable a este problema que tiene la restricción de tener un costo computacional no muy elevado y un tiempo de respuesta bajo.

2.2.2.2. Detección de oclusiones

Este enfoque busca detectar primero la zona ocluida y con esta información hacer el reconocimiento de identidad.

En esta línea, Oh et al. [18] proponen dividir la imagen en N regiones no sobrepuestas y aplicar PCA, a cada una para obtener una representación, que fue previamente entrenada con imágenes sin oclusiones. Luego, a cada zona le aplican un clasificador del tipo *nearest-neighbour* que entrega la presencia o no presencia de oclusiones en dicha zona. Finalmente, utilizan **LNMF** (*Local Non-negative Matrix Factorization*) para la etapa de reconocimiento. El problema de esta solución es que es muy sensible a los datos de entrenamiento y que no es robusta puesto que tiende a detectar solo aquellas clases que tienen mayor probabilidad de ocurrencia.

Un trabajo similar es realizado por Zhaohua Chen et al. [19], en que utilizan un clasificador binario del tipo *Support Vector Machine* sobre una imagen dividida en zonas no superpuestas y luego excluyen las zonas ocluidas para la etapa de reconocimiento. Si bien muestran una mejora puesto que no utilizan las zonas ocluidas en la comparación, su desempeño está lejos de ser útil para un uso industrial.

Dado que las zonas tapadas en una imagen le añaden ruido a su representación como vector y, por tanto, corrompen el reconocimiento, se han propuesto modelos que utilizan *deep learning* para generar mejores representaciones. En este sentido Yizhang Xia et al. [20], proponen una red neuronal convolucional que detecta oclusiones en 4 áreas de interés: ojo izquierdo, ojo derecho, nariz y boca. Si bien este enfoque produce una mejoría significativa respecto a las investigaciones de la época, se ve muy limitado porque las zonas de detección están fijas, lo cual lo hace sensible a las rotaciones o a oclusiones en zonas no consideradas. Además, la etapa de reconocimiento depende directamente de los resultados de la etapa de detección, por tanto, los resultados serán equívocos si el primer paso falla.

Siguiendo este mismo enfoque, Son *et al.* [21] proponen una arquitectura llamada ***Pairwise Differential Siamese Network***(PDSN), la cual tiene como objetivo lograr reconocer personas a pesar de que parte de su rostro esté ocluido y sin importar el tipo de oclusión.

Para lograr esto, Song et al. [21] diseñaron una arquitectura de redes siamesas cuya salida es un espacio de características de 3 dimensiones, el cual es utilizado como entrada para un módulo generador de máscaras, cuyo propósito es construir una máscara que, al aplicarla al espacio de características, logre disminuir la importancia de las zonas ocluidas en la imagen y así reconocer a la persona, utilizando sólo las zonas visibles del rostro. En la Figura 2.9 se observa la arquitectura de esta red.

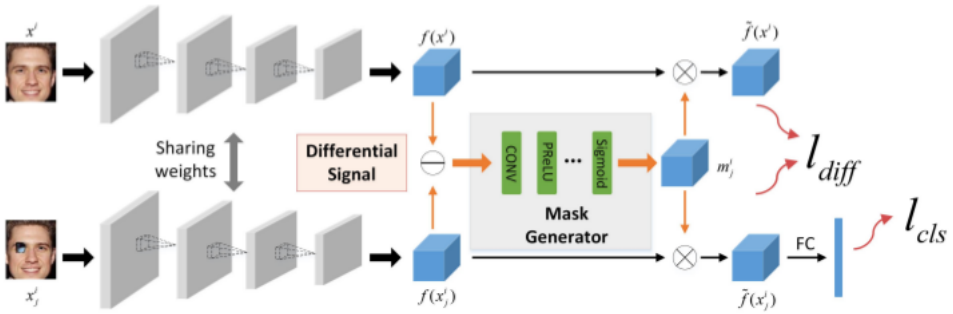


Figura 2.9: Arquitectura de la red PDSN [21]

El módulo generador de máscaras está compuesto por una capa de *Batch Normalization*, seguida de una capa Convolucional con función de activación PReLU, seguida de una última capa de *Batch Normalization* con una función de activación sigmoide, para que las salidas estén en el rango [0,1]. Cabe mencionar que el tamaño de salida es idéntico al de entrada, para que la máscara sea aplicable directamente al espacio de características obtenido del *backbone*.

Los autores proponen una función de pérdida compuesta por un *loss* diferencial (l_{diff}) y uno de clasificación. El objetivo de la parte diferencial es que el resultado de aplicar la máscara aprendida tanto a la imagen limpia como a la ocluida sean lo más parecidos posible, mientras que el de clasificación es para verificar que las características obtenidas del mismo proceso sean suficientes para identificar a una persona.

Matemáticamente, se expresa el *loss* diferencial como se muestra en la ecuación 2.5, donde $M_\theta(\cdot)(x)$ es la multiplicación matricial entre la máscara aprendida y el espacio de características, x^i es el *anchor* y x^i_j es la imagen ocluida.

$$L_{diff} = ||M_\theta(\cdot)f(x^i) - M_\theta(\cdot)f(x^i_j)||_1 \tag{2.5}$$

Por tanto, utilizando la entropía cruzada como función para medir la clasificación, la función de *loss* es de la forma (2.6):

$$L = -\sum_i \log\big(p_{y_i}(F(M_\theta(\cdot)f(x^i_j)))\big) + \lambda||M_\theta(\cdot)f(x^i) - M_\theta(\cdot)f(x^i_j)||_1 \tag{2.6}$$

Donde λ es un hiperparámetro fijado en 10 por los autores, para hacer que los distintos componentes de la función objetivo tengan la misma escala.

Cabe destacar que para el entrenamiento, cada imagen fue dividida en 25 zonas y cada zona de ocluyó de manera artificial. Con esto, se entrenaron 25 generadores de máscaras distintos, uno por cada zona, con lo que obtuvieron un diccionario de máscaras, es decir, la máscara a utilizar sobre el espacio de características dependía de la o las zonas ocluidas en la imágen de prueba.

Por tanto, durante la etapa de inferencia, la imagen se procesaba primero con una red externa que detectaba la zona ocluida y en base a dicho resultado se escogía cuál máscara utilizar.

Si bien este trabajo muestra mejoras importantes sobre otros del estado del arte y es ampliamente generalizable, se considera computacionalmente muy complejo de lograr, puesto que se debe entrenar de manera separada una red para cada zona ocluida, lo cual implica demasiado tiempo si no se tiene un poder de computación suficiente. Además, agregan la complejidad de tener que entrenar una red externa que sea capaz de detectar y segmentar zonas ocluidas.

Capítulo 3

Metodología

3.1. Enfoque metodológico

Se utilizará un enfoque metodológico **cuantitativo**, puesto que se busca medir y comparar los resultados de verificación de identidad del sistema propuesto con el sistema utilizado actualmente en la empresa. Se escoge este enfoque puesto que el objetivo es lograr un sistema que sea mejor que el existente y que funcione dentro de las limitaciones de recursos computacionales.

Cabe mencionar que, para todo el desarrollo, se utilizará el lenguaje de programación **python 3** con la librería de redes neuronales artificiales ***pytorch***. El entrenamiento de las redes se realizará en una máquina virtual creada utilizando los servicios de ***Google Cloud Platform***.

3.2. Formalización del problema

El problema a resolver se formaliza de la siguiente manera.

Existe un sistema de verificación de identidad que presenta errores en sus resultados cuando la imagen de entrada es de una persona que utiliza mascarilla. Estos errores comprenden tanto falsos positivos como falsos negativos.

Se busca proponer un nuevo sistema basado en redes neuronales convolucionales, enfocado en la verificación de identidad para imágenes de rostros con mascarilla y que disminuya los errores detallados anteriormente.

Para entrenar dicho sistema se utilizarán bases de datos públicas y, para evaluarlo, se utilizarán los datos ofrecidos por la empresa que corresponden al registro de uso de su aplicación de verificación actual.

Las bases de datos utilizadas, la arquitectura actual, la propuesta y la metodología de entrenamiento y evaluación, se detallan a continuación.

3.3. Bases de datos

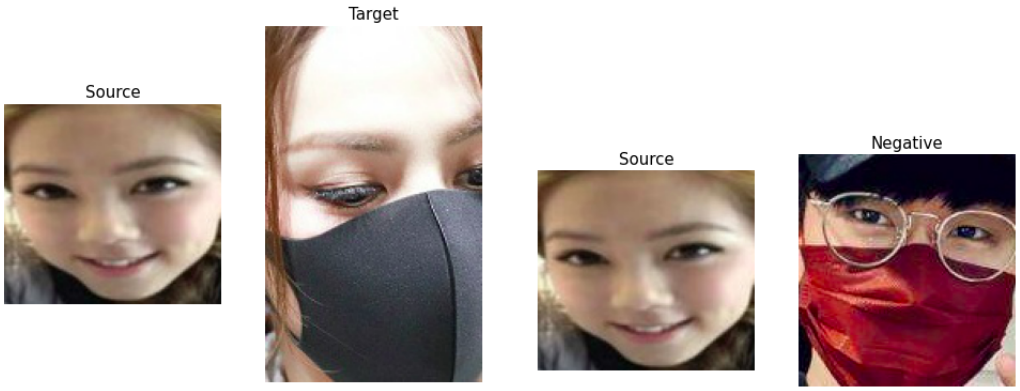
3.3.1. Bases de datos de entrenamiento

Para el entrenamiento se utilizaron dos bases de datos distintas, detalladas a continuación:

1) **Real Masked Face Database**[22] (RMFD): Esta base de datos está compuesta por 525 identidades (personas distintas) y, en total, se tienen 5.000 fotos con mascarilla y 90.000 sin mascarilla. **Cabe mencionar que estas imágenes corresponden en su totalidad a personas con características asiáticas.**

Para generar una base de datos con pares positivos, se obtuvieron todas las combinaciones posibles entre una imagen sin mascarilla y todas las imágenes con mascarilla para una misma persona. Luego de eliminar aquellas imágenes que por algún motivo estaban corrompidas, se obtuvieron 393.912 pares de imágenes pertenecientes a 403 personas distintas.

Por último, para generar pares negativos, a cada registro de la base de datos creada se le agregó al azar una imagen con mascarilla de una identidad distinta. En la figura 3.1 se observan ejemplos de pares positivos y negativos.



(a) Par positivo (b) Par negativo

Figura 3.1: Ejemplos de par positivo (a) y par negativo (b) en RMFD.

2) **CASIA-webface**[23]: Esta base de datos está compuesta por 1.005 identidades y, en total, se tienen 82.563 fotografías. Estas fotografías corresponden principalmente a personajes famosos cuyas características raciales son bastante variadas, lo cual podría conllevar a un modelo que tenga menos sesgo racial y sea más generalizable.

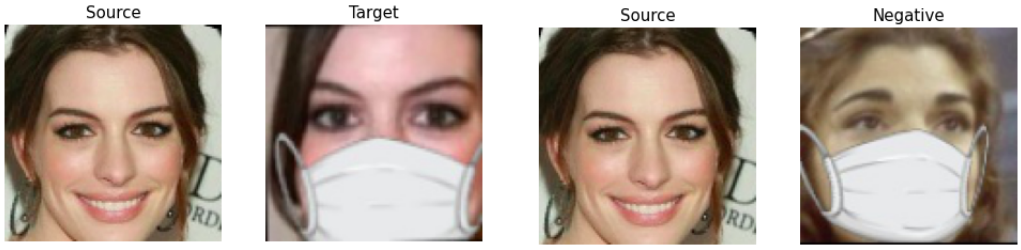
Dado que no se tienen imágenes de estas personas con mascarillas, se utilizó un modelo de redes neuronales convolucionales que detecta puntos relevantes del rostro y, utilizando dichos puntos, se adhirió una mascarilla ficticia a la imagen, logrando una base de datos con las mismas personas utilizando mascarillas artificiales, con lo cual se obtuvieron los pares positivos.

Al igual que con la base de datos anterior, se hizo una limpieza de las imágenes corrompidas y un cruce entre una imagen sin mascarilla y todas las imágenes con mascarilla para una

misma persona. Dado que en este caso se tiene una cantidad considerablemente mayor de registros, la base de datos obtenida del cruce supera el millón de pares.

En base a lo anterior, se decidió obtener una muestra aleatoria de 400.000 registros para que los resultados sean comparables entre las bases de datos. Cabe destacar que la muestra aleatoria cumple con la condición de que cada identidad se repite a lo menos 4.000 veces.

Como resultado de dicho procesamiento, se obtuvieron 400.000 registros correspondientes a 395 identidades distintas y, análogo al caso anterior, se obtuvieron pares negativos agregando al azar una persona de identidad distinta para cada registro. En la figura 3.2 se observan ejemplos de pares positivos y negativos.



(a) Par positivo (b) Par negativo

Figura 3.2: Ejemplos de par positivo (a) y par negativo (b) en CASIA webface.

3.3.2. Base de datos de evaluación

La empresa Geovictoria, además de su propio modelo, utiliza en paralelo un sistema de verificación de identidad provisto por los servicios de Amazon Web Services (AWS). Utilizando un archivo que contiene 592.00 registros de uso de este servicio, se obtuvo un histograma de los niveles de confianza (probabilidad de que dos imágenes pertenezcan a la misma persona) generados por AWS, el cual se observa en la figura 3.3.

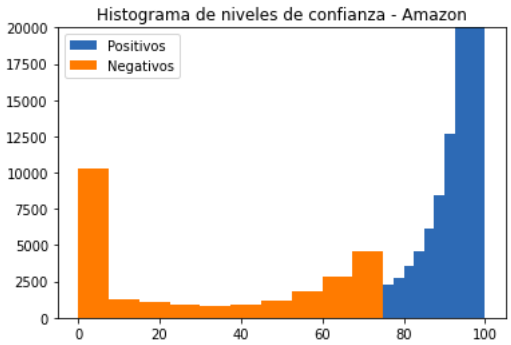


Figura 3.3: Histograma de niveles de confianza obtenidos de los registros del servicio de verificación de identidad de Amazon.

Es preciso notar que existen 359.076 registros (60.65 %) con nivel de confianza mayor a 99 %, por tanto el histograma fue limitado en su frecuencia a 20.000 para poder observar con mayor detalle el resto de rangos de confianza.

Del histograma se obtienen los siguientes datos:

- El valor de confianza mínimo para considerar un par como positivo es de 75 %, por tanto no existe cruce entre los registros positivos (azul) y negativos (naranjo).
- Del total de registros, AWS considera 25.825 (22.92 %) como pares negativos

Considerando los resultados obtenidos del análisis del histograma, se realizó lo siguiente:

- Se tomaron 100 registros aleatorios con nivel de confianza mayor a 99 % (pares positivos) y se revisaron manualmente, no encontrándose ningún falso positivo. Por tanto, se tomó una muestra de 10.000 registros de este grupo y se formó un grupo llamado **positivos fáciles**.
- Se tomaron registros aleatorios con nivel de confianza menor a 50 % (pares negativos) y se revisaron manualmente. En este caso se formó un grupo con los verdaderos negativos, llamado **negativos fáciles**. Sin embargo, se encontraron falsos negativos, correspondientes al 51 % de los ejemplos revisados. Estos fueron guardados en un grupo llamado **positivos difíciles**.
- Se tomaron registros con confianza entre 75 % y 90 % y se marcaron como **negativos difíciles**, todos aquellos que correspondían a falsos positivos.
- Para cada imagen revisada, se anotó si la persona usaba o no mascarilla.

Como resultado de la anotación anterior se lograron 4 grupos de evaluación:

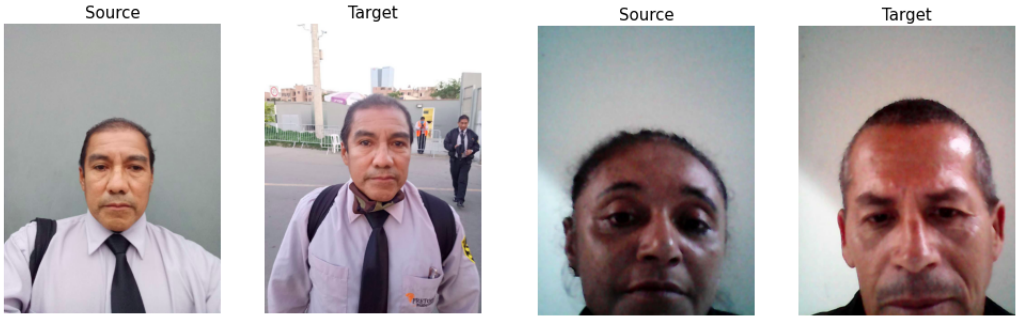
- **Ejemplos Fáciles:** Compuesto por 10.000 pares positivos y 56 negativos de los grupos positivos y negativos fáciles. En la Figura 3.4 se observan ejemplos de estos casos.

(a) Par positivo (b) Par negativo

Figura 3.4: Ejemplos de par positivo (a) y par negativo (b), en casos fáciles.

- **Ejemplos Difíciles:** Compuesto por 308 pares positivos y 307 negativos de los grupos positivos y negativos difíciles. En la Figura 3.5 se observan ejemplos de estos casos. En el par positivo (a) la dificultad está presente en la presencia de otras personas en la

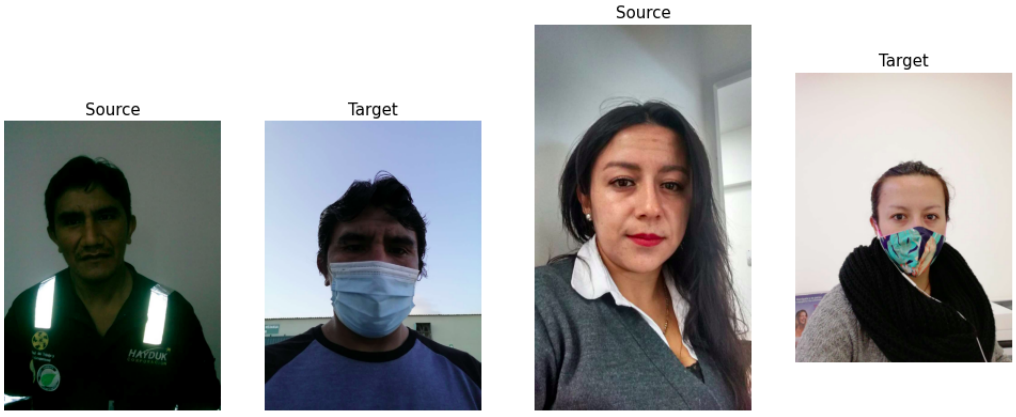
imagen, mientras que en el par negativo (b), a pesar de ser personas distintas, existen rasgos faciales y condiciones de luminosidad similares que podrían generar confusiones en el sistema.



(a) Par positivo (b) Par negativo

Figura 3.5: Ejemplos de par positivo (a) y par negativo (b), en casos difíciles.

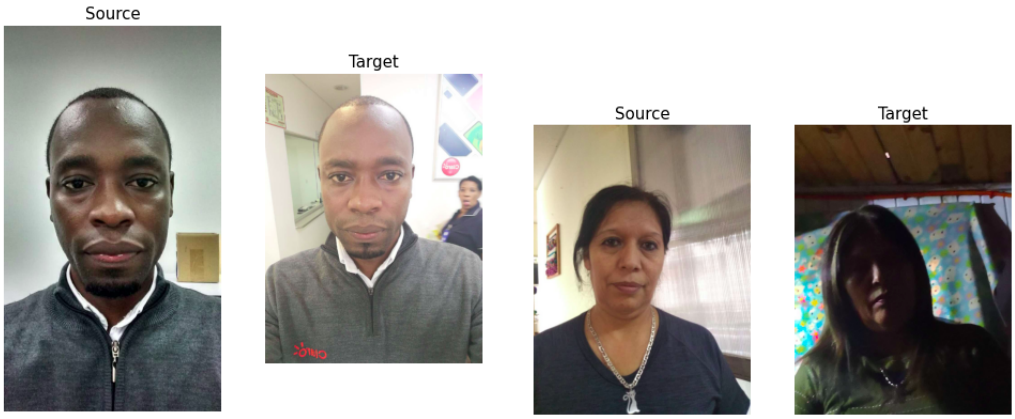
- **Ejemplos con mascarilla:** Compuesto por 743 pares positivos y 47 negativos de las imágenes anotadas con personas con mascarillas. En la Figura 3.6 se observan ejemplos de estos casos.



(a) Par positivo (b) Par negativo

Figura 3.6: Ejemplos de par positivo (a) y par negativo (b), en casos con mascarilla.

- **Ejemplos sin mascarilla:** Compuesto por 226 pares positivos y 111 negativos, de los ejemplos difíciles que no tienen mascarilla. En la Figura 3.7 se observan ejemplos de estos casos.



(a) Par positivo (b) Par negativo

Figura 3.7: Ejemplos de par positivo (a) y par negativo (b), en casos sin mascarilla.

A continuación se muestra en la tabla 3.1, un resumen de las bases de datos creadas para la evaluación.

Grupo	Pares positivos	Pares negativos
Fáciles	10.000 (99.40 %)	56 (0.60 %)
Difíciles	308 (50.08 %)	307 (49.20 %)
Con mascarilla	743 (94.05 %)	47 (5.95 %)
Sin mascarilla	226 (67.06 %)	111 (32.94 %)

Tabla 3.1: Resumen de las bases de datos de evaluación.

3.4. Arquitectura base

La arquitectura utilizada actualmente por el sistema de verificación está compuesto por dos redes neuronales.

La primera corresponde a una red MTCNN [15], descrita en la sección 2.2.1, cuyo objetivo es detectar y recortar el rostro de la imagen original.

El tamaño de salida de la imagen es fijo y corresponde a 112x112 píxeles con 3 canales. Además, si en la imagen existe más de un rostro, se retorna el que tenga mayor porcentaje de confianza.

La segunda corresponde a una red *Inception Resnet V1* cuyo *backbone* fue detallado en la sección 2.1.4 y se le agrega un módulo *Inception* [24]. El objetivo de esta red es obtener los *embeddings* de las imágenes de entrada para luego compararlos utilizando la función de similitud coseno. El tamaño del embedding obtenido para cada imagen es de 512x1

3.5. Arquitecturas propuestas

Se proponen 2 arquitecturas, una siamesa y una tripleta, ambas basadas en la red PDSN, propuesta en la publicación “Occlusion Robust Face Recognition Based on Mask Learning With Pairwise Differential Siamese Network” [21], detallada en la sección 2.2.2.2.

Ambas arquitecturas consideran una etapa previa que es la detección y recorte del rostro desde la imagen original, lo cual se realiza utilizando una red MTCNN [15], con las mismas configuraciones que la arquitectura base.

Además, para ambas arquitecturas el *backbone* de la red corresponde a una Resnet50 [8], la cual fue previamente entrenada utilizando una función objetivo llamada ***Additive Angular Margin Loss*** que se describe en [25]. Esta red entrega una salida compuesta por 512 canales de tamaño 7x7.

Los pesos son compartidos completamente y no cambian durante el entrenamiento del generador de máscaras, por tanto en términos prácticos sólo existe una red que es utilizada varias veces.

3.5.1. Red Siamesa

Esta red está basada en PDSN [21] pero con ciertos cambios que se detallan a continuación.

El módulo generador de máscaras se entrenó con pares de imágenes en que la imagen de prueba siempre era de la misma persona con mascarilla, por tanto, sólo se obtuvo una máscara especializada en la detección y supresión de la zona ocluida por la mascarilla y no un diccionario de máscaras cómo en el modelo original.

Esto genera que no exista la necesidad de tener un detector de oclusiones previo al sistema y que se reduzca la cantidad de datos necesarios para el entrenamiento.

Se utiliza el mismo *loss* propuesto en [21] que fue detallado en la ecuación 2.6. En la Figura 3.8 se observa un diagrama de la arquitectura.

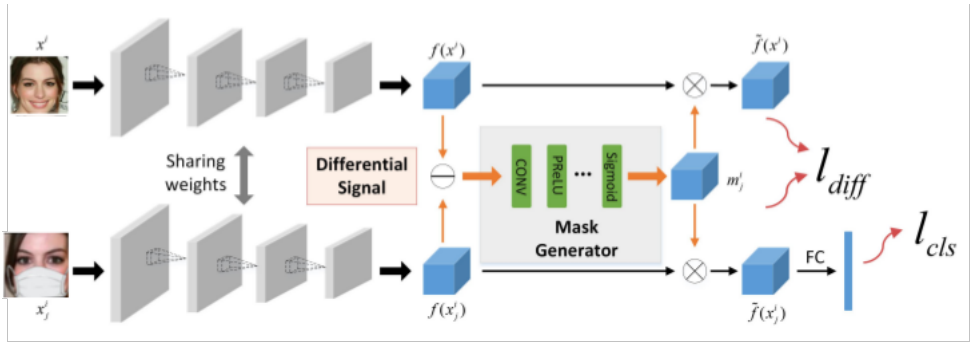


Figura 3.8: Diagrama de la arquitectura de redes siamesas, basada en PDSN.

3.5.2. Red Tripleta

Esta red corresponde a una versión tripleta de PDSN [21]. La diferencia en este caso es que el entrenamiento se realiza con tríos de imágenes, consistentes en la imagen base, el par positivo y el par negativo, por lo que la red debiese aprender no sólo a igualar las características faciales, sino también a diferenciarlas para evitar los falsos positivos.

Dada la arquitectura en forma de tripletas, se utilizó como función objetivo un *triplet loss* como el enunciado en la ecuación 2.4, con un margen $\lambda = 1.5$

A dicha función se le sumó un *loss* de clasificación para mantener la misma línea de la función original 2.6, utilizando un $\delta = 2$, quedando la función objetivo de la forma 3.1.

$$L_i = -\sum_i \log\left(p_{y_i}(F(M_\theta(\cdot)f(x_j^i)))\right) + \delta \cdot \max(D_{w,i}(a,+) - D_{w,i}(a,-) + \lambda, 0) \tag{3.1}$$

En la Figura 3.9 se observa un diagrama de la arquitectura, la diferencia con la red siamesa es que aquí se entrega un par positivo y un par negativo durante el entrenamiento y l_{diff} corresponde a 2.4.

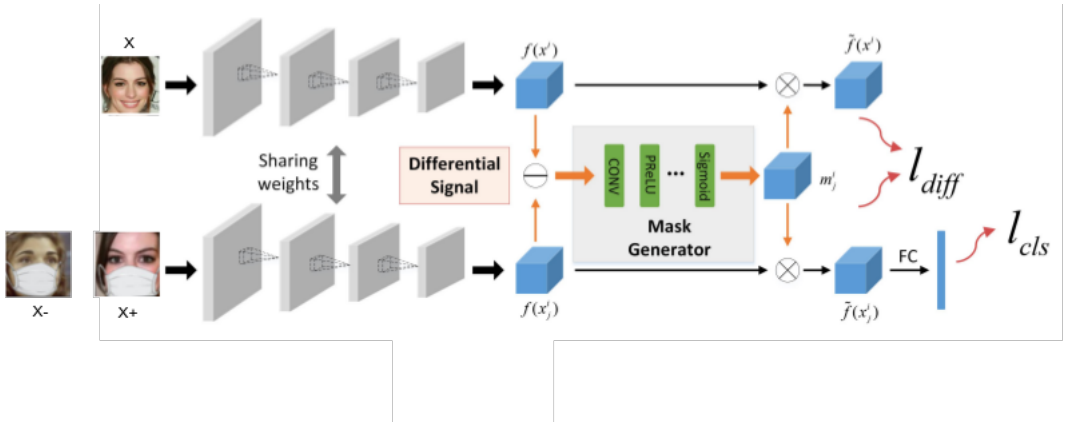


Figura 3.9: Diagrama de la arquitectura de redes tripletas, la diferencia con la red siamesa es que aquí se entrega un par positivo y un par negativo durante el entrenamiento y l_{diff} corresponde a 2.4.

3.6. Entrenamiento de los modelos

Puesto que se tienen dos arquitecturas y dos bases de entrenamiento distintas, al entrenar cada arquitectura con cada base de datos se obtienen 4 modelos. Además, dados los resultados preliminares, se entrenó un quinto modelo, según se detalla a continuación.

- **RMFD-PDSN**: Red siamesa PDSN entrenada con los datos de RMFD.
- **CASIA-PDSN**: Red siamesa PDSN entrenada con los datos de CASIA-webface.
- **RMFD-TRIPLET**: Red tripleta entrenada con los datos de RMFD.

- **CASIA-TRIPLET**: Red tripleta entrenada con los datos de CASIA-webface.
- **RC-TRIPLET**: Red tripleta entrenada primero con los datos de RMFD y luego con CASIA-webface.

Cada base de datos se dividió en dos grupos, uno de entrenamiento con el 80 % de los datos y otro de validación con el 20 % restante, procurando conservar la distribución de clases en cada uno. Además, todos los modelos fueron entrenados por 20 épocas y para la evaluación se escogió la época con mayor *accuracy* en la clasificación de identidad, utilizando el par positivo ocluido, medido sobre el grupo de validación.

3.7. Evaluación de los modelos

La evaluación de los modelos se hizo utilizando las bases de datos creadas para este fin y descritas en la sección 3.3.2.

Primero se realizó la evaluación sobre la arquitectura base, la cual fue nombrada como ***Bioapi*** por la empresa. Luego se evaluó el *backbone* de la red PDSN, el cual fué nombrado ***Arcface*** puesto que es el nombre con el que se le conoce públicamente.

Finalmente, se realizó la evaluación de los 5 modelos previamente entrenados, descritos en la sección anterior.

Cabe mencionar que la evaluación se realizó pasando de manera secuencial cada uno de los pares de imágenes de las bases de datos y obteniendo los *embeddings* de la imagen base y de su par, que podía ser positivo o negativo. Luego se computó la similitud coseno entre ambos vectores y se guardó dicho valor.

Una vez obtenida la similitud para cada registro y conociendo previamente si eran par positivo o negativo, se utilizó un arreglo de umbrales desde 0 hasta 1, obteniendo con esto la cantidad de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

3.7.1. Métricas y criterios de evaluación

Con los datos obtenidos se hizo un gráfico de las curvas de ROC para cada modelo y su correspondiente área bajo la curva (*AUC*). Utilizando dicha área se escogió el mejor modelo para cada caso de evaluación.

Para el mejor modelo en cada caso se calculó el punto de *equal error rate* (EER) descrito en la sección 2.1.3, el cual entrega el umbral para el cual se igualan los errores de falsas aceptaciones y de falsos rechazos. Utilizando dicho umbral se calculó el *accuracy* del modelo.

Para comparar los resultados se siguió el siguiente criterio: si el mejor modelo, según su AUC, era la arquitectura base, se obtuvo el EER, umbral y *accuracy* para el segundo mejor modelo. En cambio, si la mejor arquitectura no era la base, se obtuvo el EER, umbral y *accuracy* de ésta.

3.8. Impacto del módulo generador de máscaras

Para cuantificar el impacto del módulo generador de máscaras, se realizaron experimentos excluyendolo, a modo de estudio de ablación.

Este estudio constó de entrenar el *backbone* de la arquitectura propuesta, utilizando como salida la última capa convolucional del mismo y las mismas funciones de *loss*. Con esto se obtuvieron 4 modelos, los cuales fueron nombrados según se detalla a continuación:

- **RMFD-RESNET**: Red siamesa compuesta por Resnet, entrenada con los datos de RMFD.
- **CASIA-RESNET**: Red siamesa compuesta por Resnet, entrenada con los datos de CASIA.
- **RMFD-TRIPLET-RESNET**: Red tripleta compuesta por Resnet, entrenada con los datos de RMFD.
- **CASIA-TRIPLET-RESNET**: Red tripleta compuesta por Resnet, entrenada con los datos de CASIA.

Los modelos obtenidos fueron evaluados solamente en las bases de datos con mascarillas y sin mascarillas, para luego poder comparar los resultados y cuantificar el impacto del módulo generador de máscaras.

Capítulo 4

Resultados y discusión

4.1. Resultados

4.1.1. Resultados de los modelos sin el módulo generador de máscaras

Al evaluar los modelos entrenados sin el módulo generador de máscaras, se obtuvieron los siguientes resultados:

Utilizando las bases de datos con y sin mascarillas se calculó el área bajo la curva (AUC) y el *accuracy* en el punto de *equal error rate* (EER). Dichos resultados se muestran en la Tabla 4.1

	Sin mascarilla			**Con mascarilla**		
Modelo	AUC	EER	Accuracy	AUC	EER	Accuracy
CASIA-RESNET	0.730	31.6 %	68.2 %	0.530	46.5 %	42.4 %
RMFD-RESNET	0.708	31.6 %	68.2 %	0.529	47.5 %	40.7 %
CASIA-TRIPLET-RESNET	0.727	32.3 %	67.6 %	0.530	46.6 %	42.3 %
RMFD-TRIPLET-RESNET	0.716	32.3 %	67.6 %	0.537	47.0 %	41.6 %

Tabla 4.1: Resultados de los modelos sin el módulo generador de máscaras.

4.1.2. Resultados de entrenamiento y validación

Durante el entrenamiento se midió la variación de la función de *loss* y el *accuracy* de clasificación utilizando la imagen ocluida.

En la Figura 4.1 se observa la evolución de la función objetivo y en la Figura 4.2 el *accuracy* a través de las épocas.

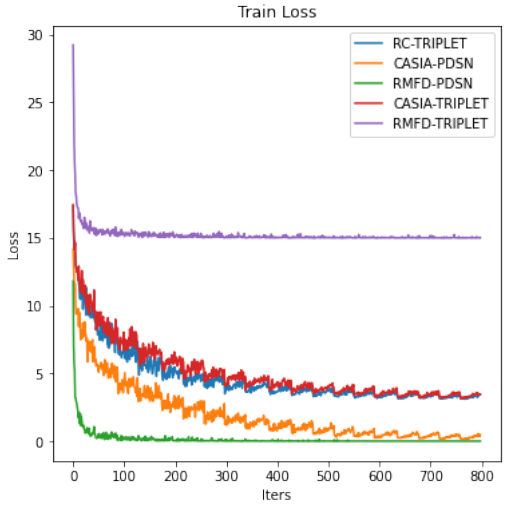


Figura 4.1: *Loss* durante el entrenamiento de los distintos modelos.

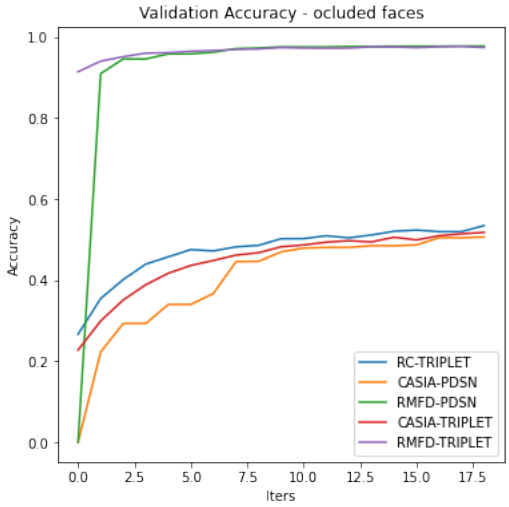


Figura 4.2: *Accuracy* en las etapas de validación durante el entrenamiento de los distintos modelos, medido utilizando el *embedding* resultante de la imagen ocluida.

4.1.3. Resultados de evaluación

Utilizando los modelos entrenados y las bases de datos creadas para evaluación, se obtuvieron los resultados mostrados a continuación. En cada tabla se marcó en negrita el modelo con mejor área bajo la curva.

4.1.3.1. Casos fáciles

Para el caso de los ejemplos fáciles, en la Figura 4.3 se observan las curvas de ROC y en la Tabla 4.2 el AUC para cada modelo.

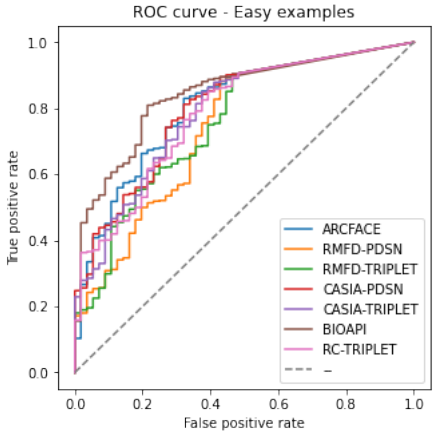


Figura 4.3: Curva de ROC para los distintos modelos en el caso de ejemplos fáciles.

Modelo	AUC
BIOAPI	0.842
ARCFACE	0.806
RMFD-PDSN	0.743
RMFD-TRIPLET	0.756
CASIA-PDSN	0.798
CASIA-TRIPLET	0.789
RC-TRIPLET	0.785

Tabla 4.2: Área bajo la curva para cada modelo en el caso de ejemplos fáciles.

Dado que el modelo con mayor AUC es **BIOAPI**, en la Figura 4.4 se muestran las tasas de falsas aceptaciones y falsos rechazos para determinar el punto de EER y en la Tabla 4.3, el valor de dicho punto, el umbral y el *accuracy* usando dicho umbral. También se muestran dichos valores para ARCFACE, el modelo con segundo mayor AUC obtenido.

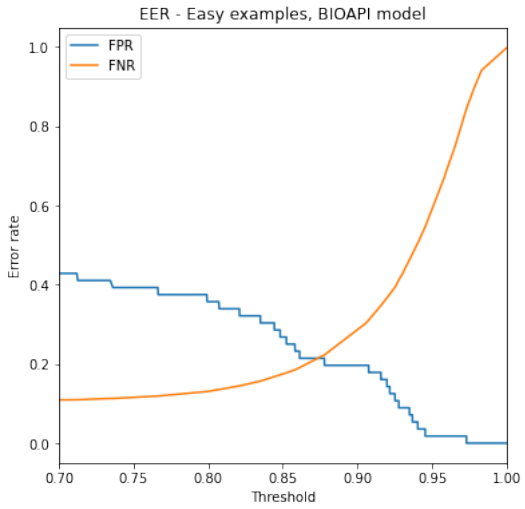


Figura 4.4: Tasa de falsas aceptaciones y tasa de falsos rechazos en función del umbral, para el caso de ejemplos fáciles.

Modelo	BIOAPI	ARCFACE
EER	21.4 %	26.5 %
100 %-EER	78.6 %	73.5 %
Accuracy	77.7 %	73.6 %
Umbral	87.7 %	99.7 %

Tabla 4.3: Error, 100 %-error, umbral y *accuracy* para BIOAPI y el segundo mejor modelo, para el caso de ejemplos fáciles.

4.1.3.2. Casos difíciles

Para el caso de los ejemplos difíciles, en la Figura 4.5 se observan las curvas de ROC y en la Tabla 4.4 el AUC para cada modelo.

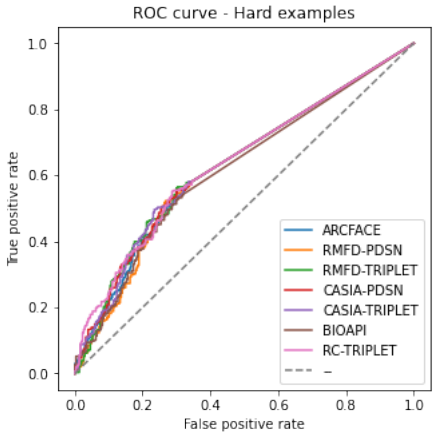


Figura 4.5: Curva de ROC para los distintos modelos en el caso de ejemplos difíciles.

Modelo	AUC
BIOAPI	0.6188
ARCFACE	0.6316
RMFD-PDSN	0.6254
RMFD-TRIPLET	0.6347
CASIA-PDSN	0.6351
CASIA-TRIPLET	0.6353
RC-TRIPLET	0.6389

Tabla 4.4: Área bajo la curva para cada modelo en el caso de ejemplos difíciles.

Dado que el modelo con mayor AUC es **RC-TRIPLET**, en la Figura 4.6 se muestran las tasas de falsas aceptaciones y falsos rechazos para determinar el punto de EER y en la Tabla 4.5, el valor de dicho punto, el umbral y el *accuracy* usando dicho umbral. También se muestran dichos valores para BIOAPI.

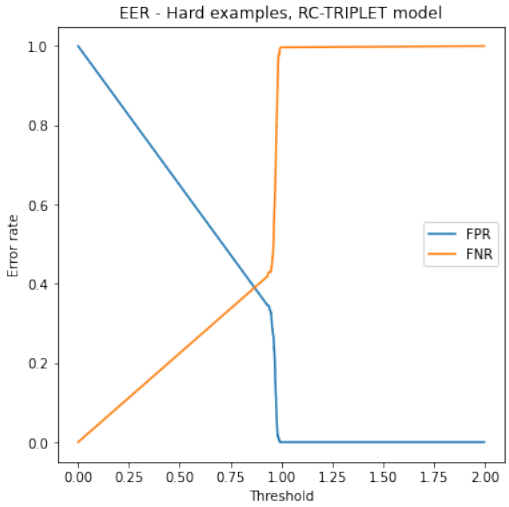


Figura 4.6: Tasa de falsas aceptaciones y tasa de falsos rechazos en función del umbral, para el caso de ejemplos difíciles.

Modelo	RC-TRIPLET	BIOAPI
EER	39.0 %	40.2 %
100 %-EER	61.0 %	59.8 %
Accuracy	61.8 %	61.1 %
Umbral	86.83 %	33.63 %

Tabla 4.5: Error, 100 %-error, umbral y *accuracy* para el mejor modelo y BIOAPI, en el caso de ejemplos difíciles.

4.1.3.3. Casos con mascarilla

Para el caso de los ejemplos con mascarilla, en la Figura 4.7 se observan las curvas de ROC y en la Tabla 4.6 el AUC para cada modelo.

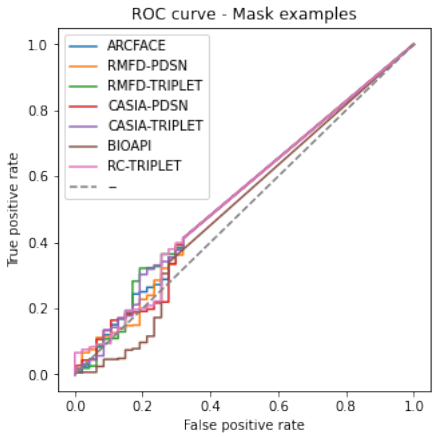


Figura 4.7: Curva de ROC para los distintos modelos en el caso de ejemplos con mascarillas.

Modelo	AUC
BIOAPI	0.506
ARCFACE	0.541
RMFD-PDSN	0.538
RMFD-TRIPLET	0.545
CASIA-PDSN	0.536
CASIA-TRIPLET	0.544
RC-TRIPLET	0.541

Tabla 4.6: Área bajo la curva para cada modelo en el caso de ejemplos con mascarillas.

Dado que el modelo con mayor AUC es **RMFD-TRIPLET**, en la Figura 4.8 se muestran las tasas de falsas aceptaciones y falsos rechazos para determinar el punto de EER y en la Tabla 4.7, el valor de dicho punto, el umbral y el *accuracy* usando dicho umbral. También se muestran dichos valores para BIOAPI.

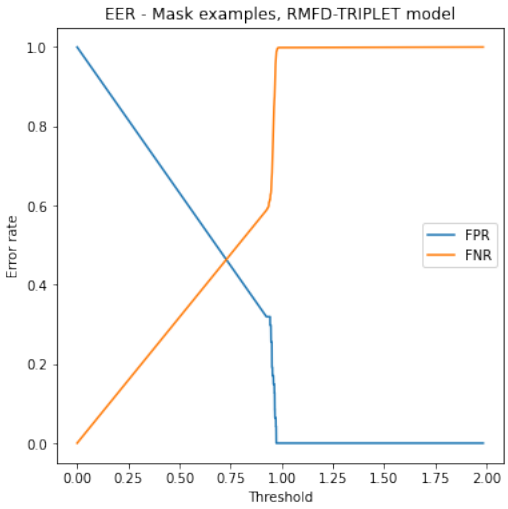


Figura 4.8: Tasa de falsas aceptaciones y tasa de falsos rechazos en función del umbral, para el caso de ejemplos con mascarilla.

Modelo	RMFD-TRIPLET	BIOAPI
EER	46.35 %	47.7 %
100 %-EER	53.65 %	52.3 %
Accuracy	42.7 %	36.2 %
Umbral	72.9 %	27.6 %

Tabla 4.7: Error, 100 %-error, umbral y *accuracy* para el mejor modelo y BIOAPI, en el caso de ejemplos con mascarilla.

4.1.3.4. Casos sin mascarilla

Para el caso de los ejemplos sin mascarilla, en la Figura 4.9 se observan las curvas de ROC y en la Tabla 4.8 el AUC para cada modelo.

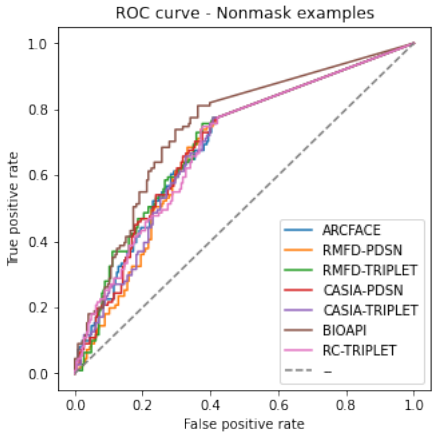


Figura 4.9: Curva de ROC para los distintos modelos en el caso de ejemplos sin mascarillas.

Modelo	AUC
BIOAPI	0.746
ARCFACE	0.696
RMFD-PDSN	0.682
RMFD-TRIPLET	0.702
CASIA-PDSN	0.696
CASIA-TRIPLET	0.688
RC-TRIPLET	0.694

Tabla 4.8: Área bajo la curva para cada modelo en el caso de ejemplos sin mascarillas.

Dado que el modelo con mayor AUC es **BIOAPI**, en la Figura 4.10 se muestran las tasas de falsas aceptaciones y falsos rechazos para determinar el punto de EER y en la Tabla 4.9, el valor de dicho punto, el umbral y el *accuracy* usando dicho umbral. También se muestran dichos valores para RMFD-TRIPLET, el modelo con segundo mayor AUC obtenido.

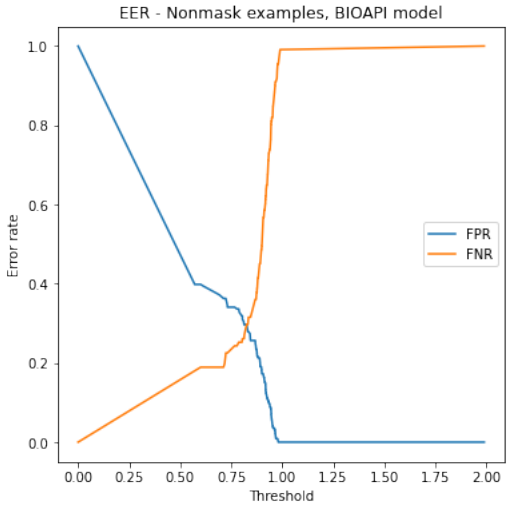


Figura 4.10: Tasa de falsas aceptaciones y tasa de falsos rechazos en función del umbral, para el caso de ejemplos sin mascarilla.

Modelo	BIOAPI	RMFD-TRIPLET
EER	29.6 %	35.0 %
100 %-EER	70.4 %	65.0 %
Accuracy	70.3 %	65.2 %
Umbral	82.6 %	94.8 %

Tabla 4.9: Error, 100 %-error, umbral y *accuracy* para BIOAPI y el segundo mejor modelo, en el caso de ejemplos sin mascarilla.

4.2. Discusión

4.2.1. Entrenamiento y validación

Se observa en la Figura 4.1 que los modelos RMFD-PDSN y CASIA-PDSN llegan a niveles de *loss* similares y muy cercanos a cero, pero en el primer caso la convergencia es mucho más acelerada. Por otra parte, los modelos CASIA-TRIPLET y RC-TRIPLET convergen a un *loss* más alto que los casos anteriores, cercano a 5 y con una velocidad similar. Por último, el modelo RMFD-TRIPLET es quien logra peor *loss*, estancándose rápidamente en un valor cercano a 25.

Sin embargo, al observar el *accuracy* en la Figura 4.2, los mejores resultados son obtenidos por RMFD-PDSN y por RMFD-TRIPLET, llegando ambos a niveles cercanos al 98 %. Si bien el primer caso es esperable puesto que obtiene un bajo *loss*, el segundo está fuera de todo pronóstico.

Una posible explicación de este fenómeno es que, dado que el *loss* de RMFD-TRIPLET no decrece con el pasar de las iteraciones, pero *accuracy* es alto, la función objetivo esté completamente dominada por el término que busca minimizar la distancia entre pares positivos y maximizarla entre negativos. Esto quiere decir que la arquitectura de la red, específicamente el módulo generador de máscaras, no es capaz de generar espacios de características numéricamente distintos para personas distintas y sólo está restándole importancia a la zona ocluida.

Además, se observa que el *accuracy* en los modelos que fueron validados con la base de datos CASIA son cercanos al 50 %, independiente de la arquitectura utilizada. Mientras que la validación con RMFD entrega resultados muy cercanos al 98 %. Esto se puede interpretar como que las personas que componen esta última base de datos son más fáciles de identificar, lo que se puede deber a que las fotos ocluidas corresponden a situaciones reales y no ficticias como es el caso de CASIA.

Lo anterior se respalda en los resultados obtenidos para RC-TRIPLET, el que se puede entender como un modelo que se pre-entrenó en RMFD y luego se hizo *transfer learning* con los datos de CASIA y, a pesar de este procedimiento, la capacidad de clasificar de este modelo sobre dicha base de datos no comprende una mejora significativa respecto al resto.

Por último, cabe mencionar que en este caso el *accuracy* no es una medida completamente representativa del comportamiento de los modelos puesto que las clases en las bases de datos no están necesariamente equilibradas.

4.2.2. Casos fáciles

Al observar las curvas de ROC en la Figura 4.3, se observa claramente que BIOAPI es muy superior al resto de modelos, lo cual se confirma al ver los datos de la Tabla 4.2. Según estos mismos datos el modelo que le procede es ARCFACE.

Un primer análisis de estos resultados es que, dado que los ejemplos fáciles están compuestos principalmente por rostros de personas que aparecen de manera frontal, con un ta-

maño apropiado de la cara respecto a la imagen y sin oclusiones, la máscara generada sólo induce una pérdida de información, puesto que el módulo que la genera fue entrenado para esto. Este es un comportamiento esperado, dado que la arquitectura no fue diseñada para detectar, de alguna manera, la presencia o no presencia de oclusiones y generar una salida en función de eso.

Como se muestra en la Tabla 4.3, la diferencia de la métrica 100 %-EER entre ambos modelos es de un 5.1 % y se mantiene una diferencia similar en el resto de las métricas. Esto se aleja de lo esperado, puesto que el modelo ARCFACE es ampliamente utilizado y en general tiene muy buenos resultados en las distintas bases de datos públicas para verificación de identidad, mientras que la arquitectura de BIOAPI (*Inception Resnet V1*) no suele ser tan utilizada.

Algo que llama profundamente la atención es que el umbral óptimo de ARCFACE es del 99.7 %, es decir, la sensibilidad del modelo es extremadamente alta. Este comportamiento demuestra que los ejemplos son efectivamente fáciles, puesto que con dicha sensibilidad tan alta, se alcanza un *accuracy* del 73.6 %.

4.2.3. Casos difíciles

Al observar las curvas de ROC de la Figura 4.5, todos los modelos muestran un comportamiento similar, lo cual es ratificado por los datos de la Tabla 4.4, en la que se ve que las diferencias están en el segundo o tercer decimal. Según dichos datos, el mejor modelo es RC-TRIPLET.

Se observa también que CASIA-PDSN y CASIA-TRIPLET son los modelos que siguen al mejor. Un primer resultado obtenido de estos datos es que los modelos entrenados con CASIA, para este caso, logran mejores resultados, indiferente de la arquitectura. Esto se puede deber al hecho de que las características faciales de los sujetos en CASIA, son más heterogéneas y más similares al caso de uso que en RMFD, por tanto la red aprende a procesar de mejor manera dichas características.

Al comparar RC-TRIPLET con BIOAPI usando los datos de la Tabla 4.5, se obtiene que la diferencia porcentual de la métrica 100 %-EER es poco significativa, siendo del 1.2 %, lo que se ve reflejado también en el *accuracy* que obtienen. Esto se produce porque el *dataset* de casos difíciles se compone tanto de imágenes con mascarilla como sin mascarilla y, como se vio en los resultados de casos fáciles (que no presentan mascarillas), las arquitecturas propuestas no funcionan bien en dichos casos.

Sin embargo, como se observa en la siguiente sección, si están mejorando la verificación en los casos con mascarillas, por lo que el resultado es una ponderación entre ambos casos.

4.2.4. Casos con mascarilla

Al observar las curvas de ROC de la Figura 4.7, todos los modelos muestran un comportamiento similar, salvo en ciertas regiones. Esto es ratificado por los datos de la Tabla 4.6, en la que se ve que el mejor modelo es RMFD-TRIPLET, aunque distanciado solo en 0.001 de CASIA-TRIPLET, lo cual entrega un primer resultado concreto. Los modelos entrenados

con una red tripleta funcionan mejor que el resto de las arquitecturas evaluadas cuando las imágenes contienen mascarillas.

Lo anterior se debe a dos situaciones. La primera es que la máscara obtenida del generador está realizando lo que debería, que es suprimir la importancia de la zona ocluida al momento de obtener el *embedding* de la imagen de prueba, lo cual será ratificado más adelante.

La segunda, es que la red tripleta permite aprender las diferencias que existen entre dos rostros de los cuales sólo se tiene información de la mitad del rostro hacia arriba.

Sin embargo, se esperaba que el modelo RC-TRIPLET presentara mejores resultados que el resto, puesto que es una red tripleta que fue entrenada con más datos, pero los resultados de la Tabla 4.6 muestran lo contrario a las suposiciones teóricas.

Otro punto importante del análisis de las curvas obtenidas en este caso, es que todos los modelos muestran ser idénticos o peores a un modelo aleatorio, según se explicó en la sección 2.1.4.2 y que sólo al superar el 20 % de falsos positivos algunos modelos comienzan a mostrar un mejor comportamiento. Se visualiza incluso que BIOAPI muestra ser peor que un modelo aleatorio hasta, a lo menos, un 30 % de falsos positivos, lo que refuerza la hipótesis inicial de que el sistema actual funciona mal para los casos con mascarillas.

Al comparar RMFD-TRIPLET con BIOAPI usando los datos de la Tabla 4.7, se tiene que la diferencia porcentual de los errores es de apenas un 1.35 %. Sin embargo, los resultados de *accuracy* se distancian en un 6.5 % a favor de RMFD-TRIPLET, por lo que se puede considerar que sí existe una mejora sustancial al utilizar dicho modelo al momento de verificar identidades con rostros ocluidos por mascarillas.

4.2.5. Casos sin mascarilla

Al observar las curvas de ROC de la Figura 4.9, se tiene que BIOAPI es notoriamente mejor que el resto de los modelos, lo que se condice con los resultados obtenidos para los casos fáciles. El segundo mejor modelo, según los datos de AUC de la Tabla 4.8 es RMFD-TRIPLET, teniendo una diferencia de 0.44 respecto a BIOAPI. El resto de los modelos tienen resultados bastante similares entre sí, diferenciándose en uno o dos decimales.

Al comparar BIOAPI contra RMFD-TRIPLET, según los datos de la Tabla 4.9, se tiene un 15.4 % de diferencia en 100 %-EER, lo cual corresponde a una diferencia importante y reafirma que BIOAPI es superior verificando cuando el rostro no tiene mascarilla.

4.2.6. Impacto del módulo generador de máscaras

Para el caso sin mascarillas, se considera el modelo RMFD-TRIPLET, el cual logra los mejores resultados y se comparan dichos valores con los obtenidos al eliminar el módulo generador de máscaras, resultando lo expuesto en la Tabla 4.10

Estos son resultados esperables, puesto que se podría suponer que, al tratarse de rostros sin mascarillas, el módulo generador de máscaras está suprimiendo una zona con información

RMFD-TRIPLET	Con módulo	Sin módulo
EER	35.0 %	32.3 %
Accuracy	65.2 %	67.6 %

Tabla 4.10: Comparación de resultados del mejor modelo obtenido para el caso sin mascarillas, utilizando y sin utilizar el módulo generador de máscaras.

valiosa para comparar a las personas, por lo cual el modelo sin el módulo obtiene resultados porcentualmente mejores.

Para el caso con mascarillas, también se considera el modelo RMFD-TRIPLET, el cual logra los mejores resultados y se comparan dichos valores con los obtenidos al eliminar el módulo generador de máscaras, resultando lo expuesto en la Tabla 4.11

RMFD-TRIPLET	Con módulo	Sin módulo
EER	46.35 %	47.0 %
Accuracy	42.7 %	41.6 %

Tabla 4.11: Comparación de resultados del mejor modelo obtenido para el caso sin mascarillas, utilizando y sin utilizar el módulo generador de máscaras.

Los resultados antes expuestos revelan que el uso del módulo genera un aumento del $1.1\,\%$, lo cual indica levemente que el módulo generador de máscaras tiene un impacto sobre la representación de la imagen y por tanto sobre la verificación.

Sin embargo, el impacto no es tan significativo como se esperaba teóricamente, según los resultados de los autores que lo propusieron [21]. Esto puede deberse a distintos factores, entre ellos, que la forma de entrenar el módulo fue simplificada respecto al original y que no se obtuvo una máscara promedio fija, como sugieren los autores, sino que simplemente se utilizaron los pesos aprendidos para obtener la máscara en función de las imágenes de entrada.

4.2.7. Resultados generales

De manera general se obtiene que BIOAPI funciona significativamente mejor que el resto de los modelos en los casos en que no existen mascarillas, lo cual es avalado por las métricas obtenidas tanto para casos fáciles como para casos sin mascarilla. Además, al observar su desempeño en los casos con mascarilla se reafirma la hipótesis inicial, de que el sistema falla ante este tipo de oclusiones.

Para el caso específico de mascarillas, se tiene que tanto RMFD-TRIPLET como CASIA-TRIPLET obtienen resultados significativamente mejores que el resto de los modelos, pero que al combinar ambas bases de entrenamiento los resultados empeoran. También es interesante notar que ARCFACE obtiene resultados comparativos a los dos modelos antes señalados, lo cual podría dar un indicio de que un modelo compuesto solamente por un *backbone*, con arquitectura de redes tripletas y entrenado con *triplet loss* 2.4 podría ser incluso mejor que las arquitecturas propuestas en este trabajo.

En base a los resultados obtenidos, se tiene que el problema de verificar identidades de personas que utilizan mascarillas no queda resuelto, porque, si bien se mejoraron las métricas respecto al modelo original, aún están lejos de ser lo suficientemente buenas para ser aceptadas como solución.

Lo anterior se debe a que este es un problema complejo de resolver, puesto que se debe tomar una decisión utilizando la mitad de la información, es decir, lograr decidir si dos imágenes de rostros corresponden o no a la misma persona sólo viendo la mitad de la cara. Esto es difícil incluso para el humano, por tanto se requieren otras estrategias y mayor capacidad de representación para lograr resultados aceptables.

Un ejemplo de la dificultad del problema, se observa en la Figura 4.11, en que los aspectos generales no permiten decidir rápidamente si ambas imágenes corresponden o no a la misma persona y es necesario enfocarse en detalles específicos, como la zona inferior del ojo o la forma de las cejas.

Inclusive, existen elementos distractores entre una imagen y otra, como la forma del cabello, que podría hacer suponer que son personas distintas.

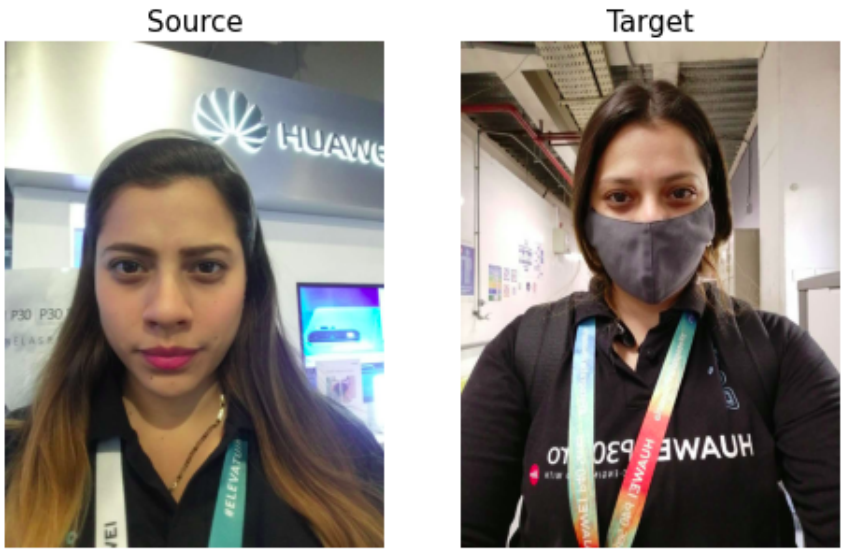


Figura 4.11: Ejemplo de la dificultad que supone verificar la identidad de una persona utilizando sólo la mitad del rostro no ocluida. Esta imagen corresponde a un par positivo.

4.2.8. Fuentes de error en las imágenes de evaluación

Aparte de los errores cometidos por cada modelo en cada una de los casos de evaluación, los datos en sí poseen fuentes de error que producen que los desempeños óptimos no puedan ser alcanzados.

Por una parte, una fuente de error inherente a los datos de evaluación es que fueron anotados por humanos, por lo tanto pueden existir casos que estén marcados como negativos cuando en realidad eran positivos o viceversa. Este tipo de error es muy complejo de superar, puesto que aunque se intentaran corregir las anotaciones con otros humanos, estos últimos tendrían su propio sesgo e inducirían inevitablemente un factor de error.

Otra fuente de error proveniente de los datos es la calidad de los mismos y esto sí se podría mejorar por medio de algún pre-procesamiento previo a utilizar un modelo de verificación para lograr su comportamiento óptimo. Dichas fuentes pueden ser individualizadas como:

- Imágenes con los rostros rotados respecto a la horizontal, es decir, que la persona se ve de manera horizontal, en cualquiera de los dos sentidos.
  Esto podría mejorarse con un detector de ojos, el cual podría entregar la posición respecto al centro de la imagen en que se encuentran y con eso rotar el rostro para que quede en la posición correcta. En la Figura 4.12 se observa un ejemplo de este caso.

- Imágenes con los rostros volteados respecto a la vertical. Esto sucede principalmente porque se utiliza la cámara trasera de un dispositivo móvil para enrolar a la persona y la cámara frontal al momento de hacer verificación, por lo que las características faciales aparecen cambiadas (por ejemplo, un lunar que está al lado derecho del rostro aparece al lado izquierdo).
  Esto podría mejorarse utilizando la imagen de prueba original y la imagen de prueba

volteada respecto a la vertical, con lo cual se obtendrían dos *embeddings* que podrían ser concatenados y utilizar ese resultado para la verificación.

- Heterogeneidad en el tamaño del rostro respecto a la imagen en las imágenes de enrolamiento, lo cual es producido porque no existe una distancia estándar entre la persona y la cámara al momento de enrolar.
  Dado que la empresa tiene una aplicación para realizar este servicio, se podría solucionar indicando un tamaño mínimo que debe ocupar el rostro en la imagen al momento de enrolar y de verificar. En la Figura 4.13 se observa un ejemplo de este caso.

- Cambios físicos en las personas debido al paso del tiempo. Esto podría solucionarse si el sistema de verificación tuviera un tiempo máximo que, al ser cumplido, requiera que la persona sea enrolada nuevamente para así guardar el nuevo estado de las características de su rostro.

- Presencia de otras personas en la imagen de verificación. Este es un problema más complejo, puesto que un enfoque podría ser dejar el rostro detectado que ocupa mayor espacio en la imagen, suponiendo que la persona que desea verificarse cumple con dicha condición. Sin embargo, existen casos, como el mostrado en la Figura 4.14 en que aparece un rostro de mayor tamaño correspondiente a una publicidad puesta en el lugar donde se tomó la fotografía.

- Por último, aunque un menor medida, existen otros tipos de oclusiones que aparecen en las imágenes, como lentes, gorros, bufandas, pelo, entre otros. Para resolver dicho problema sería necesario implementar un modelo de verificación que sea generalizable respecto a las oclusiones, lo cual podría corresponder a un trabajo futuro. Un ejemplo de esta situación se observa en la Figura 4.15.

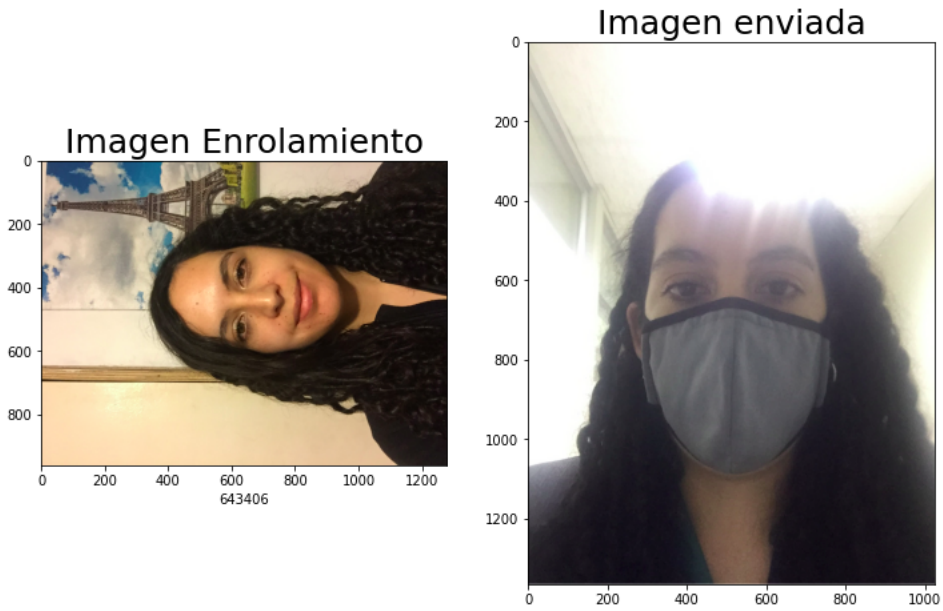


Figura 4.12: Ejemplo de rostro rotado respecto a la horizontal.

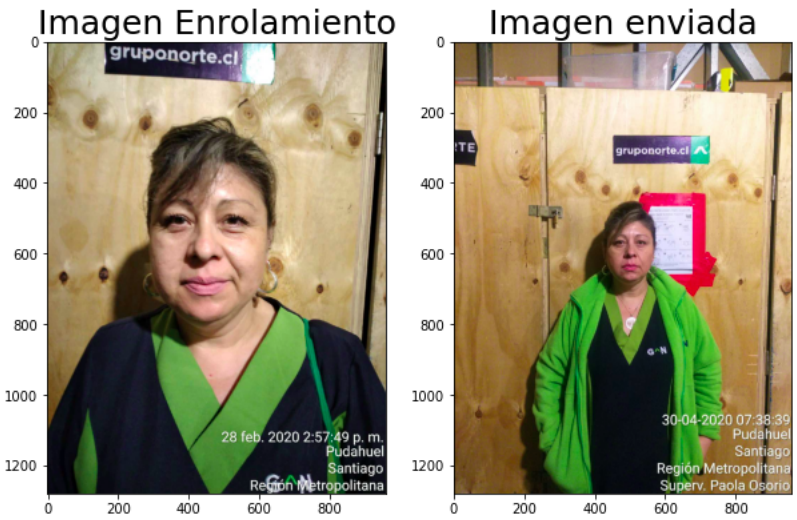


Figura 4.13: Ejemplo de rostro lejano respecto a la cámara.

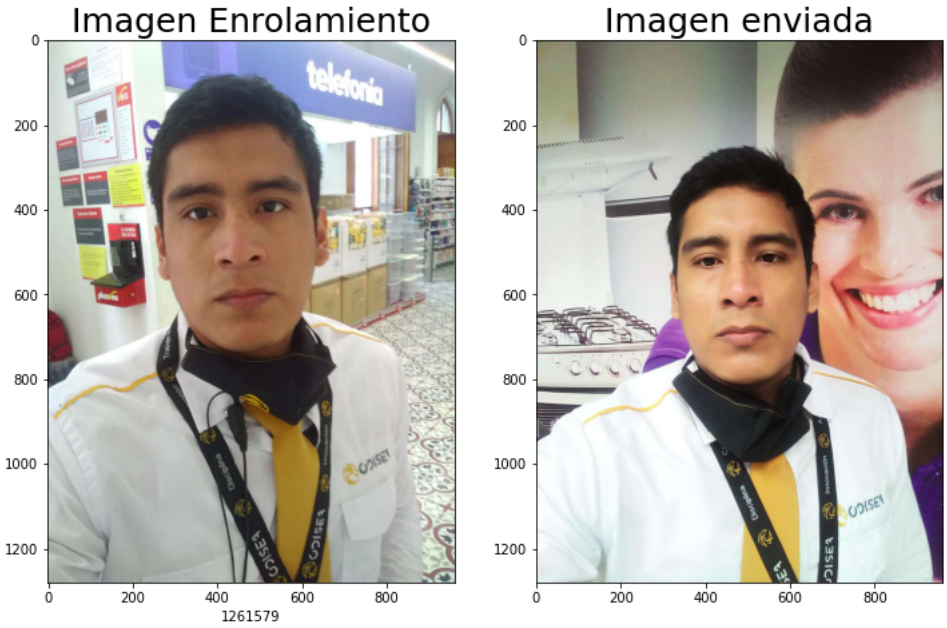


Figura 4.14: Ejemplo de dos rostros que aparecen en la imágen, pero sólo uno corresponde a una persona real.

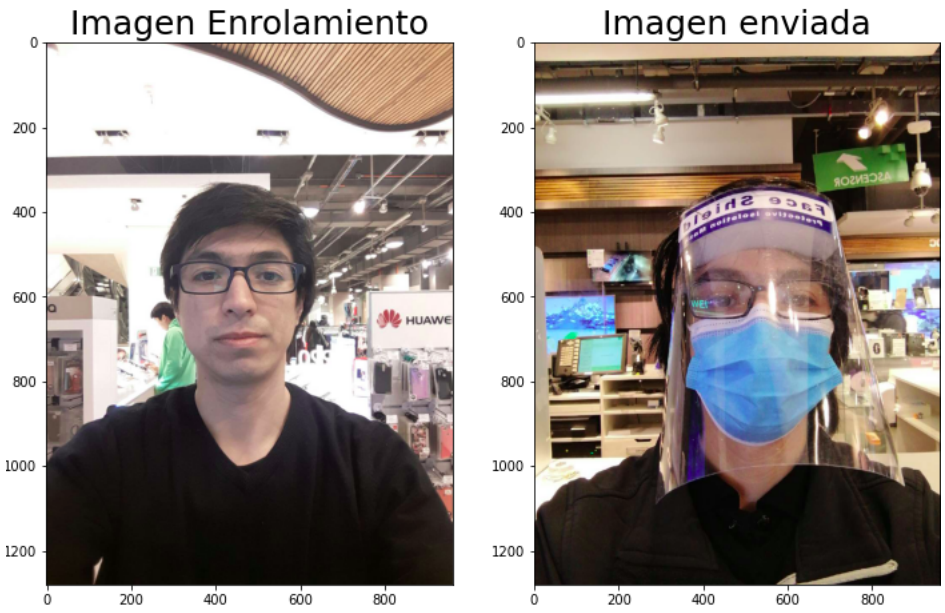


Figura 4.15: Ejemplo de diferentes oclusiones presentes en la imagen.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

De manera específica se puede concluir lo siguiente:

- De todos los modelos evaluados, aquellos con arquitectura de redes tripletas presentan mejores resultados al momento de verificar personas que utilizan mascarilla.
- El módulo generador de máscara genera un impacto sobre la representación de la imagen y mejora levemente los resultados cuando existe oclusión, así como los empeora cuando no existe. Esta leve incidencia sobre los resultados se puede deber a las simplificaciones y diferencias que se utilizaron en la arquitectura propuesta en este trabajo, respecto a lo propuesto por los autores de dicho módulo.
- El modelo actual de la empresa muestra ser superior al resto en todos los casos, excepto cuando hay oclusiones presentes. Esto conlleva a la idea de que los modelos propuestos pueden ser innecesariamente complejos y que dicha complejidad está empeorando la capacidad de la red en obtener características.
- Dado lo anterior, un enfoque de redes tripletas estándar, compuestas sólo por un *backbone* y sin módulos extra agregados, podrían suponer una mejor solución al problema.
- Existen fuentes de error externas al modelo, de las cuales la gran mayoría podrían ser resueltas mediante pre-procesamiento, lo cual impactaría positivamente en el comportamiento del modelo verificador.

De manera general, se concluye lo siguiente:

- Se logró el objetivo general que es desarrollar un modelo de inteligencia artificial para hacer verificación de identidad, cuando los rostros están afectados por oclusiones provocadas por mascarilla
- Se logró el objetivo específico de generar una base de datos de entrenamiento y de evaluación a partir de datos públicos y privados, para ser utilizados por las arquitecturas propuestas.

- Se logró el objetivo específico que es proponer una arquitectura basada en redes neuronales convolucionales que mejorara las tasas de verificación de identidad en los casos en que el rostro está ocluido por mascarillas.
- El sistema idóneo, según los resultados obtenidos, sería uno compuesto por un sistema de pre-procesamiento de la imagen, seguido de un detector de mascarillas. Si se obtiene que la imagen presenta mascarilla, se debe utilizar RMFD-TRIPLET para la verificación. En caso contrario, se debe utilizar BIOAPI.
- La investigación realizada es importante puesto que demuestra que existen modelos basados en inteligencia artificial que pueden resolver este tipo de problemas que se presentan en contextos de uso real y fuera de un laboratorio. Además, se dan las guías para seguir perfeccionando o buscando un modelo distinto a los propuestos que pueda tener mejores resultados.

5.2. Trabajo futuro

En base a los resultados y a los análisis de los mismos, además de la discusión detallada en esta investigación, se propone como trabajo futuro lo siguiente:

Diseñar una arquitectura de redes tripletas utilizando alguno de los *backbones* expuestos en esta investigación u otro presente en las publicaciones científicas y entrenarla con una función de *triplet loss* 2.4 clásica, sin añadir módulos extras.

Desarrollar en un sistema de pre-procesamiento de la imagen que solucione los problemas expuestos en la sección 4.2.6, para llevar al óptimo el comportamiento del modelo de verificación.

Integrar un modelo previo que sea capaz de detectar la presencia de mascarilla en la imagen, para así elegir si utilizar el modelo base o el modelo entrenado para solucionar dicho problema.

Bibliografía

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[2] T. Mitchell, *Machine Learning*, ser. McGraw-Hill International Editions. McGraw-Hill, 1997. [Online]. Available: https://books.google.cl/books?id=EoYBngEACAAJ

[3] D. Zeng, R. Veldhuis, and L. Spreeuwers, "A survey of face recognition techniques under occlusion," 2020.

[4] E. Conrad, S. Misenar, and J. Feldman, "Chapter 5 - domain 5: Identity and access management (controlling access and managing identity)," in *Eleventh Hour CISSP® (Third Edition)*, third edition ed., E. Conrad, S. Misenar, and J. Feldman, Eds. Syngress, 2017, pp. 117 – 134. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B978012811248900005X

[5] R. Mehrotra, K. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern Recognition*, vol. 25, no. 12, pp. 1479 – 1494, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/003132039290121X

[6] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[7] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image

recognition," 2015.

[13] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.

[14] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," 2019.

[15] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, p. 1499–1503, Oct 2016. [Online]. Available: http://dx.doi.org/10.1109/LSP.2016.2603342

[16] S. Liao, A. K. Jain, and S. Z. Li, "Partial face recognition: Alignment-free approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1193–1205, 2013.

[17] L. He, H. Li, Q. Zhang, Z. Sun, and Z. He, "Multiscale representation for partial face recognition under near infrared illumination," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016, pp. 1–7.

[18] H. J. Oh, K. Lee, and S. Lee, "Occlusion invariant face recognition using selective local non-negative matrix factorization basis images," *Image Vision Comput.*, vol. 26, pp. 1515–1523, 11 2008.

[19] Zhaohua Chen, Tingrong Xu, and Zhiyuan Han, "Occluded face recognition based on the improved svm and block weighted lbp," in *2011 International Conference on Image Analysis and Signal Processing*, 2011, pp. 118–122.

[20] Yizhang Xia, Bailing Zhang, and F. Coenen, "Face occlusion detection based on multi-task convolution neural network," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015, pp. 375–379.

[21] L. Song, D. Gong, Z. Li, C. Liu, and W. Liu, "Occlusion robust face recognition based on mask learning with pairwise differential siamese network," 10 2019, pp. 773–782.

[22] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei, H. Chen, Y. Miao, Z. Huang, and J. Liang, "Masked face recognition dataset and application," 2020.

[23] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," 2014.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[25] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," 2019.