Universidad Continental

FACULTAD DE INGENIERÍA

Escuela Académico Profesional de Ingeniería de Sistemas e Informática

Tesis

Reconocimiento facial para el control de acceso a las instalaciones de FUDEC Perú, 2022

Andrei Saavedra Rivera

Para optar el Título Profesional de
Ingeniera de Sistemas e Informática

Huancayo, 2023

Universidad Continental

INFORME DE CONFORMIDAD DE ORIGINALIDAD DE TESIS

A	:	Dr. Felipe Gutarra Meza Decano de la Facultad de Ingeniería
DE	:	Dr. Job Daniel Gamarra Moreno Asesor de tesis
ASUNTO	:	Remito resultado de evaluación de originalidad de tesis
FECHA	:	20 de julio de 2023.

Con sumo agrado me dirijo a vuestro despacho para saludarlo y en vista de haber sido designado asesor de la tesis titulada: "Reconocimiento facial para el control de acceso a las instalaciones de FUDEC Perú, 2022", perteneciente al estudiante Andrei Saavedra Rivera, de la E.A.P. de Ingeniería de Sistemas e Informática; se procedió con la carga del documento a la plataforma "Turnitin" y se realizó la verificación completa de las coincidencias resaltadas por el software dando por resultado 20 % de similitud (informe adjunto) sin encontrarse hallazgos relacionados a plagio. Se utilizaron los siguientes filtros:

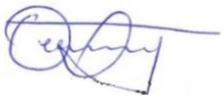
- Filtro de exclusión de bibliografía — SI [X] NO [ ]
- Filtro de exclusión de grupos de palabras menores (N° de palabras excluidas:) — SI [ ] NO [X]
- Exclusión de fuente por trabajo anterior del mismo estudiante — SI [ ] NO [X]

En consecuencia, se determina que la tesis constituye un documento original al presentar similitud de otros autores (citas) por debajo del porcentaje establecido por la Universidad.

Recae toda responsabilidad del contenido de la tesis sobre el autor y asesor, en concordancia a los principios de legalidad, presunción de veracidad y simplicidad, expresados en el Reglamento del Registro Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales – RENATI y en la Directiva 003-2016-R/UC.

Esperando la atención a la presente, me despido sin otro particular y sea propicia la ocasión para renovar las muestras de mi especial consideración.

Atentamente,



Asesor de tesis

Cc.
Facultad
Oficina de Grados y Títulos
Interesado(a)

Universidad Continental

DECLARACIÓN JURADA DE AUTENTICIDAD

Yo, Andrei Saavedra Rivera, identificado(a) con Documento Nacional de Identidad No. 73455073, de la E.A.P. de Ingeniería de Sistemas e Informática de la Universidad Continental, declaro bajo juramento lo siguiente:

1. La tesis titulada: "Reconocimiento facial para el control de acceso a las instalaciones de FUDEC Perú, 2022", es de mi autoría, la misma que presento para optar el Título Profesional de Ingeniero de Sistemas e Informática.
2. La tesis no ha sido plagiada ni total ni parcialmente, para la cual se han respetado las normas internacionales de citas y referencias para las fuentes consultadas, por lo que no atenta contra derechos de terceros.
3. La tesis es original e inédita, y no ha sido realizado, desarrollado o publicado, parcial ni totalmente, por terceras personas naturales o jurídicas. No incurre en autoplagio; es decir, no fue publicado ni presentado de manera previa para conseguir algún grado académico o título profesional.
4. Los datos presentados en los resultados son reales, pues no son falsos, duplicados, ni copiados, por consiguiente, constituyen un aporte significativo para la realidad estudiada.

De identificarse fraude, falsificación de datos, plagio, información sin cita de autores, uso ilegal de información ajena, asumo las consecuencias y sanciones que de mi acción se deriven, sometiéndome a las acciones legales pertinentes.

20 de Abril de 2023.

Andrei Saavedra Rivera

DNI. No. 73455073

Cc.
Facultad
Oficina de Grados y Títulos
Interesado(a)

Tesis Saavedra Rivera

ORIGINALITY REPORT

20%	19%	6%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.wikizero.com Internet Source	1%
2	Submitted to Universidad de Alicante Student Paper	1%
3	www.electronicid.eu Internet Source	1%
4	syfeed.com Internet Source	1%
5	Submitted to Infile Student Paper	1%
6	latam.kaspersky.com Internet Source	<1%
7	docs.microsoft.com Internet Source	<1%
8	repositorio.ucv.edu.pe Internet Source	<1%
9	es.m.wikipedia.org Internet Source	<1%

163	Silva Flores Ricardo David. "Implementación de una red neuronal convolucional en un FPGA para la clasificación de piezas de manufactura", TESIUNAM, 2022 Publication	<1 %
164	hdl.handle.net Internet Source	<1 %
165	repositorio.uss.edu.pe Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

AGRADECIMIENTOS

Agradezco sincera y profundamente a mi madre Rocío Marisol Rivera Aliaga, por ser parte de este proceso en mi evolución profesional, su dedicación, apoyo, comprensión en este proceso fue fundamental para terminar mi carrera profesional.

Agradezco también a mi abuelo Rufino Rivera Santillán quien siempre me brindo el apoyo para ser un profesional con principios y una persona de bien. Finalmente agradezco a toda mi familia, que siempre me apoyaron para terminar la carrera con éxito.

DEDICATORIA

El presente trabajo va dedicado a mi querida madre Rocío Marisol Rivera Aliaga, que sé que se siente orgullosa y feliz por ver que me desarrollo profesionalmente. Dedico también este logro a mi abuelo Rufino Rivera, por ser mi mentor y mi protector para culminar con éxito esta carrera.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

La problemática de la presente investigación se enfocó en el crecimiento exponencial de los índices de delincuencia que afectan al patrimonio público y/o privado en nuestro país. Asimismo, se resaltaron las limitaciones funcionales con las que cuentan los sistemas de seguridad, tanto tecnológicos como biométricos, que no garantizan una seguridad completa que permita resguardar el patrimonio. De esta manera, se desarrolló un sistema que está compuesto por un software de reconocimiento facial y un circuito de control de acceso que en conjunto permiten un funcionamiento seguro, continuo y automático. El software de reconocimiento facial se implementó en base al algoritmo de redes neuronales siamesas que demostró su superioridad frente a otros algoritmos en las pruebas de campo realizadas en el presente trabajo, logrando alcanzar un 94% de precisión en su accionar. Además, cuenta con una detección facial que mejora la calidad de imágenes capturadas por medio de un porcentaje que se le asigna al rostro detectado en base a su nitidez. A su vez este, permite la automatización del proceso de control de acceso. Del mismo modo, cuenta con un proceso de verificación de vida por medio del conteo de parpadeos que impide que se realice una suplantación de identidad. El cual, obtuvo una precisión del 98% durante las pruebas de campo. Así, se brinda acceso mediante un circuito de Arduino que abre un cerrojo de selenoide una vez que se haya realizado un reconocimiento facial correcto. El proyecto tiene como objetivo mitigar y prevenir cualquier acto delictivo que ponga en riesgo el patrimonio de la empresa FUDEC Perú, a través del uso de la inteligencia artificial.

Palabras clave: Red neuronal artificial, Red neuronal siamés, OpenCV, Arduino, Scrum, Tensorflow, Aplicativo Desktop.

ABSTRACT

The problem of this research focused on the exponential growth of crime rates affecting public and/or private assets in our country. It also highlighted the functional limitations of the security systems, both technological and biometric, which do not guarantee complete security to protect the patrimony. Thus, a system was developed that is composed of facial recognition software and an access control circuit that together allow a secure, continuous, and automatic operation. The facial recognition software was implemented based on the Siamese neural network algorithm, which demonstrated its superiority over other algorithms in the field tests carried out in this work, achieving 94% accuracy in its operation. In addition, it has a face detection that improves the quality of captured images by means of a percentage assigned to the detected face based on its sharpness. This, in turn, allows the automation of the access control process. Likewise, it has a life verification process by counting blinks that prevents identity theft. This obtained an accuracy of 98% during field tests. Thus, access is provided through an Arduino circuit that opens a selenoid lock once a correct facial recognition has been performed. The project aims to mitigate and prevent any criminal act that puts at risk the assets of the company FUDEC Peru, using artificial intelligence.

Keywords: Artificial Neural Network, Siamese Neural Network, OpenCV, Arduino, Scrum, Tensorflow, Desktop Application.

INTRODUCCIÓN

En la última década, Perú ha incrementado su tasa de delincuencia en gran magnitud por lo que la necesidad de protección y seguridad es altamente necesaria hoy en día. Una de las medidas que tomaron bastante fuerza fueron las medidas biométricas que a través del análisis de las características de una persona se lograba su autenticación y reducía en gran medida los actos delictivos. De la misma manera, la importancia que tienen las herramientas tecnológicas para la vida cotidiana ha dado un giro sorprendente, un caso de ellos es el uso exponencial de cámaras de video vigilancia instaladas en organizaciones y viviendas con el fin de resguardar la integridad del patrimonio. Sin embargo, se han visto afectadas por sus limitaciones funcionales y por las diversas modalidades de actos delictivos que exigen una mayor seguridad.

Un problema que se presenta en la empresa FUDEC Perú en los últimos años, es el consumo excesivo de horas del personal de seguridad dentro del proceso de control de acceso a las instalaciones, este proceso impide el buen funcionamiento del personal que enfoca la mayor parte de sus horas laborales sobre este proceso de seguridad.

El presente trabajo de investigación tiene como objetivo el desarrollo de un sistema de reconocimiento facial, haciendo uso de las redes neuronales siamesas, con el objetivo de mantener un control de acceso automático, seguro y preciso en las instalaciones de FUDEC Perú. De esta manera, se busca resguardar el patrimonio frente a actos delictivos que supongan un riesgo para la organización.

Por lo cual este estudio, se organiza en cinco capítulos:

Capítulo I: Planteamiento del Estudio, donde se explica la problemática del proceso de control de acceso del personal, así mismo se identifica tanto el problema general como específico del caso de estudio. Se presentan los objetivos relacionados al problema, la justificación y finaliza con la operacionalización variables.

Capítulo II: Marco teórico, donde se detalla el estado del arte relacionado a las Redes Neuronales Artificiales, Redes neuronales siameses y su participación en el reconocimiento facial para el control de acceso.

Capítulo III: Metodología, donde se indica la metodología, su estructura y el modelo a seguir durante el desarrollo de la tesis.

Capítulo IV: Análisis y diseño de la solución, se identifican las necesidades, se diseñan las interfaces en base a los requerimientos, se define la arquitectura de la solución física y lógica.

Capítulo V: Construcción, donde se realiza el preprocesamiento de datos, modelado, desarrollo y entrenamiento de la red neuronal siamés, incluyendo el cálculo de su porcentaje de acierto y pérdida o error. Cabe resaltar que también se hace un comparativo de la tecnología aplicada con otros algoritmos de reconocimiento facial para determinar la superioridad de la implementación del proyecto.

CAPÍTULO I. PLANTEAMIENTO DEL ESTUDIO

1.1. Planteamiento del problema

Durante las últimas dos décadas en el Perú se ha registrado un crecimiento exponencial de la delincuencia, no solo en su capital sino en sus diferentes departamentos. Durante el último año, pese a estar en una pandemia, la delincuencia se registró como uno de los problemas más importantes que afectan a nuestro país (1). Véase en la Figura 1.

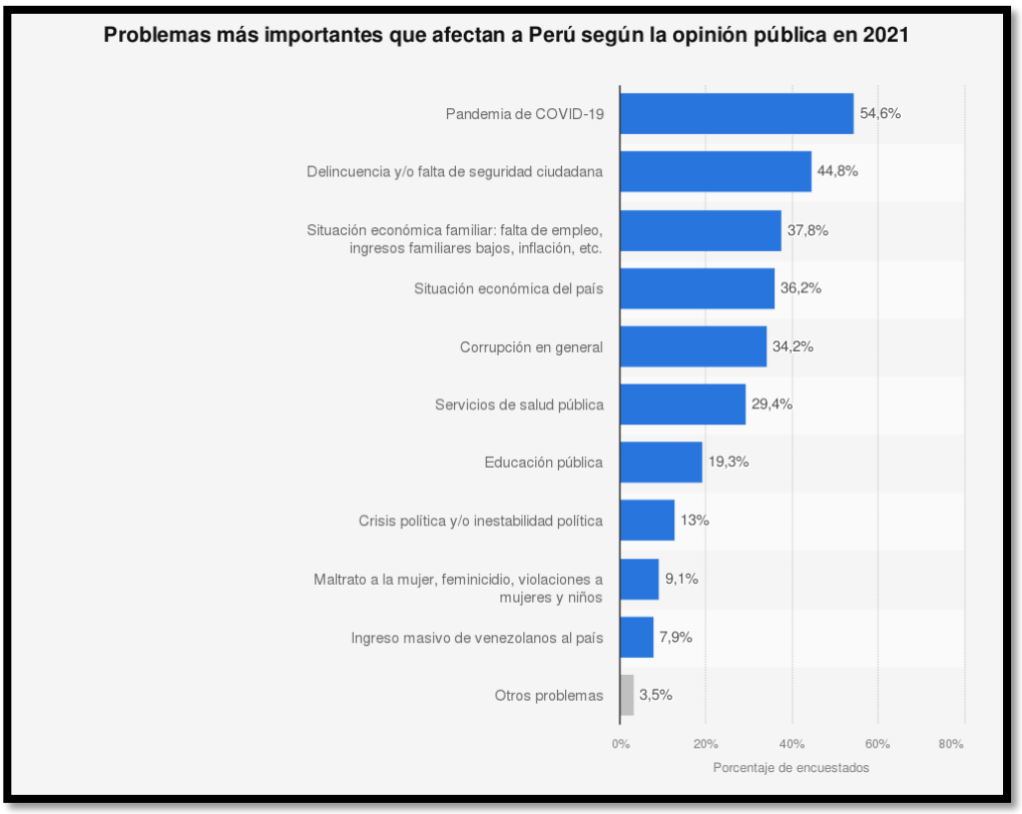


Figura 1. Porcentaje de actos delictivos por departamentos

Fuente: (1)

De la misma manera, si se toman en cuenta los datos estadísticos de denuncias por comisión de delitos en los últimos años se puede evidenciar una decadencia en el sector de seguridad social. Véase en la Figura 2.

Perú: Denuncias por comisión de delitos, según departamento
Octubre - Diciembre, 2019 - 2021

Departamento	2019 Oct - Dic	2020 Oct - Dic	2021 Oct - Dic	Variación 2021 / 2019 Absoluta	Variación 2021 / 2019 %	Variación 2021 / 2020 Absoluta	Variación 2021 / 2020 %
Total	**120 504**	**89 723**	**114 394**	**-6 110**	**-5,1**	**24 671**	**27,5**
Amazonas	1 652	1 361	1 345	-307	-18,6	-16	-1,2
Áncash	4 356	3 390	3 754	-602	-13,8	364	10,7
Apurímac	1 173	1 112	1 295	122	10,4	183	16,5
Arequipa	6 648	5 701	7 253	605	9,1	1 552	27,2
Ayacucho	1 619	1 474	1 865	246	15,2	391	26,5
Cajamarca	2 997	3 064	3 049	52	1,7	-15	-0,5
Prov. Const. del Callao	4 927	3 426	4 039	-888	-18,0	613	17,9
Cusco	4 205	3 882	4 095	-110	-2,6	213	5,5
Huancavelica	362	456	640	278	76,8	184	40,4
Huánuco	1 957	2 093	2 545	588	30,0	452	21,6
Ica	4 875	3 666	4 514	-361	-7,4	848	23,1
Junín	4 625	3 768	5 372	747	16,2	1 604	42,6
La libertad	6 827	4 961	6 274	-553	-8,1	1 313	26,5
Lambayeque	7 293	4 714	6 881	-412	-5,6	2 167	46,0
Lima metropolitana 1/	42 864	27 725	38 535	-4 329	-10,1	10 810	39,0
Departamento de lima 2/	3 833	3 255	3 498	-335	-8,7	243	7,5
Loreto	1 780	1 358	1 862	82	4,6	504	37,1
Madre de dios	750	633	730	-20	-2,7	97	15,3
Moquegua	663	566	834	171	25,8	268	47,3
Pasco	587	642	678	91	15,5	36	5,6
Piura	7 025	4 680	6 122	-903	-12,9	1 442	30,8
Puno	2 002	1 580	2 068	66	3,3	488	30,9
San martín	2 053	2 087	2 360	307	15,0	273	13,1
Tacna	1 295	1 161	1 452	157	12,1	291	25,1
Tumbes	1 430	1 180	1 382	-48	-3,4	202	17,1
Ucayali	2 706	1 788	1 952	-754	-27,9	164	9,2

1/ Denominación establecida mediante Ley N° 31140, comprende los 43 distritos de la provincia de Lima.
2/ Denominación establecida mediante Ley N° 31140, constituido por las provincias de Barranca, Cajatambo, Canta, Cañete, Huaral, Huarochiri, Haura, Oyón y Yauyos.
Fuente: Ministerio del Interior - Sistema de Denuncias Policiales-SIDPOL.
Elaboración: Instituto Nacional de Estadística e Informática.

Figura 2. Denuncias por comisión de delitos, según departamento.

Fuente: (1)

Y si a estos datos, se los clasifican a través de los principales indicadores de seguridad ciudadana, se demuestra que los delitos contra el patrimonio representan más del 70% del total de delitos en el país. Véase en la Figura 3.

Perú: Principales Indicadores de Seguridad Ciudadana
2016 - 2021 y Enero - Marzo 2022

Indicador	2016	2017	2018	2019	2020	2021	Ene-Mar 2022
I DENUNCIAS POR COMISIÓN DE DELITOS							
1.1 Denuncias por comisión de delitos	**355 876**	**399 869**	**466 088**	**446 508**	**320 819**	**403 071**	**... a/**
Contra el patrimonio	242 653	265 219	315 542	296 760	189 656	247 672	...
Contra la vida, el cuerpo y la salud	44 342	50 597	49 577	44 983	33 927	39 302	...
Contra la seguridad pública	38 150	49 385	53 595	46 305	37 673	51 935	...
Contra la libertad	20 428	22 660	29 079	35 259	32 073	36 336	...
Otros 1/	10 303	12 008	18 295	23 201	27 490	27 826	...
1.2 Denuncias por vehículos robados	**17 544**	**18 106**	**19 084**	**20 159**	**13 984**	**19 991**	**5 442**
Vehículos recuperados	**12 991**	**12 676**	**14 865**	**13 690**	**10 309**	**12 108**	**3 204**
1.3 Denuncias de accidentes de tránsito	**116 659**	**107 913**	**90 056**	**95 800**	**57 396**	**74 624**	**20 104 b/**
1.4 Denuncias de trata de personas	**539**	**725**	**734**	**509**	**372**	**535**	**75**
1.5 Personas detenidas	**111 233**	**135 036**	**150 575**	**162 505**	**178 512**	**173 616**	**48 054**
II DENUNCIAS POR COMISIÓN DE FALTAS							
Denuncias por faltas	**264 793**	**274 345**	**84 132**	**84 345**	**49 398**	**54 672**	**13 765 c/**
III BANDAS DESARTICULADAS							
Bandas desarticuladas	**4 525**	**4 148**	**5 132**	**4 839**	**3 365**	**4 158**	**1 158**
IV VIOLENCIA FAMILIAR Y SEXUAL							
4.1 Denuncias por violencia familiar	**164 488**	**187 270**	**222 376**	**276 322**	**238 704**	**241 911**	**63 762**
Física	73 413	76 011	111 428	116 458	97 088	97 952	26 152
Psicológica	54 927	69 969	97 308	133 653	124 157	125 759	33 249
Otro 2/	36 148	41 290	13 640	26 211	17 459	18 200	4 361
4.2 Denuncias por violencia sexual	**5 683**	**7 113**	**7 789**	**8 255**	**7 987**	**9 840**	**2 473**
Hombre	**395**	**492**	**527**	**495**	**486**	**620**	**134**
Menor de 18 años	322	386	414	374	372	443	101
De 18 y más	73	106	113	121	114	177	33
Mujer	**5 288**	**6 621**	**7 262**	**7 760**	**7 501**	**9 220**	**2 339**
Menor de 18 años	3 768	4 486	4 641	4 902	4 824	5 820	1 456
De 18 y más	1 520	2 135	2 621	2 858	2 677	3 400	883

Figura 3. Principales indicadores de Seguridad Ciudadana, 2016-21

Fuente: (1)

En consecuencia, hoy en día el servicio de protección y seguridad para el patrimonio, tanto público como privado, es altamente demandado.

La continua evolución y éxito de la tecnología a nivel de sistemas de seguridad y de vigilancia ha llevado a que la gran mayoría de organizaciones públicas y privadas tengan la necesidad de adquirir equipos que le faciliten el resguardo de sus patrimonios (2). La calidad del servicio que brinda un sistema de cámaras de vigilancia supone un menor riesgo para el patrimonio. Sin embargo, no es suficiente tener el registro de algún acontecimiento, sino también poder controlarlo y reaccionar al evento de manera inmediata.

Como Perú y muchos otros países poseen una delincuencia casi incontrolable. De manera que, en los últimos años se han desarrollado y aplicado diversas estrategias y tecnologías que buscan prevenir y mitigar el impacto de robos y asaltos. El uso de nuevas tecnologías ha ido en aumento de manera sustancial y esto ha significado un cambio en los hábitos financieros, comerciales y sociales. Por esto, el desarrollo de sistemas de controles de acceso, de intrusión, de videovigilancia, de geolocalización, de reconocimientos biométricos, de comunicación y de gestión de central de alarmas han sido la primera respuesta de las organizaciones (3). La aplicación de sistemas de videovigilancia y de alarmas tienen un impacto sobre la delincuencia dirigiendo sus esfuerzos en tratar de prevenirlos y de obtener evidencias del acto delictivo. Sin embargo, no tienen una funcionalidad completa que logre prevenir y mitigar robos o accesos no autorizados a organizaciones y sus áreas restringidas.

La inteligencia artificial ha ido evolucionando de una manera exponencial, como consecuencia de la revolución tecnológica, así propone soluciones y aplicaciones más eficientes sobre diferentes áreas, en especial sobre la seguridad (4). De esta manera, se resalta una rama de la inteligencia artificial que es la visión artificial. Esta, permite a través de un modelo de aprendizaje automático adquirir, procesar y analizar imágenes del mundo real con el fin de producir información relevante que promueva un accionar inmediato (5). Por tanto, el reconocimiento e identificación facial es posible con la tecnología actual. La cuál, dotaría a los sistemas de seguridad con una mayor eficiencia y precisión con el fin de prevenir y mitigar riesgos de seguridad.

Un problema que se presentando en la empresa FUDEC Perú, es el tiempo que toma el proceso de control de acceso a las instalaciones de la organización. Para este proceso se encarga a uno de los empleados que verifique las identificaciones y permisos de cada persona que intenta ingresar.

Al realizar un mapeo de las instalaciones de la organización, se registró que es posible la implementación de un sistema de seguridad que permita controlar el acceso de manera automática y autónoma haciendo uso de sistema de cámaras de vigilancia.

El presente trabajo de investigación tiene como objetivo el desarrollo de un sistema de reconocimiento facial, haciendo uso de las redes neuronales siamesas, con el objetivo de mantener un control de acceso automático, seguro y preciso en las instalaciones de FUDEC Perú. De esta manera, se busca resguardar el patrimonio frente a actos delictivos que supongan un riesgo para la organización.

1.2. Formulación del problema

1.2.1. Problema general

Basándose en lo previamente expuesto, se plantea el siguiente problema:

¿Cómo influye el reconocimiento facial en la mejora del control de acceso a las instalaciones de FUDEC Perú?

1.2.2. Problemas específicos

- ¿Cómo influye la red neuronal siamés en la mejora del control de acceso a las instalaciones de FUDEC Perú?
- ¿Cómo influye la verificación de vida en la mejora del control de acceso a las instalaciones de FUDEC Perú?
- ¿Cómo influye la detección facial en la mejora del control de acceso a las instalaciones de FUDEC Perú?

1.3. Objetivos

1.3.1. Objetivo general

Determinar la influencia del reconocimiento facial en la mejora del control de acceso a las instalaciones de FUDEC Perú

1.3.2. Objetivos específicos

- Determinar la influencia de la red neuronal siamés en la mejora del control de acceso a las instalaciones de FUDEC Perú

- Determinar la influencia de la verificación de vida en la mejora del control de acceso a las instalaciones de FUDEC Perú
- Determinar la influencia de la detección facial en la mejora del control de acceso a las instalaciones de FUDEC Perú

1.4. Justificación e importancia

1.4.1. Justificación social

El objetivo de desarrollar un sistema de reconocimiento facial que mantenga un control continuo del acceso a las instalaciones de FUDEC Perú es controlar el ingreso del personal autorizado. Así, prevenir todo ingreso no autorizado mitigando el impacto de actos delictivos que pongan en riesgo el patrimonio de la organización y su personal.

1.4.2. Justificación teórica

La aplicación de las redes neuronales siamesas, conjuntamente con el uso de la función de pérdida de triplete, se emplea sobre la tecnología de reconocimiento facial que aportará conocimiento sobre su utilidad en el procesamiento de imágenes. Además, los resultados del proyecto servirán en futuros trabajos que tengan como base el análisis de imágenes por medio de las redes neuronales siamesas en el desarrollo de la medida biométrica del tipo reconocimiento facial.

1.4.3. Importancia

La importancia de la presente investigación radica en promover el uso de la tecnología para optimizar procesos como el de control de acceso que permitan reducir tiempos y costos dentro de una organización. Además, se busca prevenir cualquier acto delictivo que ponga en riesgo el patrimonio de la organización.

1.5. Hipótesis y descripción de variables

1.5.1. Hipótesis General

El sistema de reconocimiento facial influye de manera positiva sobre la mejora del control de acceso a las instalaciones de FUDEC Perú.

1.5.2. Hipótesis Específica

La red neuronal siamés influye de manera positiva sobre la mejora del control de acceso a las instalaciones de FUDEC Perú.

La verificación de vida influye de manera positiva sobre la mejora del control de acceso a las instalaciones de FUDEC Perú.

La detección facial influye de manera positiva sobre la mejora del control de acceso a las instalaciones de FUDEC Perú.

1.6. Operacionalización de variables

Tabla 1. Operacionalización de variables

VARIABLE	DEFINICIÓN	DIMENSIÓN	INDICADOR	INSTRUMENTOS
Control de acceso	Es el método que garantiza y fuerza a los usuarios a autenticarse a través de alguna medida que permite mantener un control en el ingreso y salida de los usuarios. (6)	Puntos estratégicos Instalación Configuración	• Tiempo de espera en el acceso	• Resultados del acceso
Reconocimiento facial	"El reconocimiento facial es una manera de identificar o confirmar la identidad de una persona mediante su rostro. Los	Verificación de vida Red neuronal siamés	• Precisión de la verificación • Precisión de la	• Matriz de confusión • Resultados de la verificación • Resultados

	sistemas de reconocimiento facial se pueden utilizar para identificar a las personas en fotos, videos o en tiempo real.” (6)	Detección facial	identificación • Precisión de la detección	de la detección

CAPÍTULO II. MARCO TEÓRICO

2.1. Antecedentes del problema

2.1.1. Antecedentes nacionales

A continuación, se presentan los antecedentes de la investigación, se revisaron y analizaron diversas fuentes relacionadas a la investigación como artículos científicos y proyectos de investigación, los cuales nos ayudaron a generar valor a la presente tesis.

En el trabajo de investigación (6) "Análisis comparativo de los algoritmos fisherfaces y lbph para el reconocimiento facial en diferentes condiciones de iluminación y pose, Tacna - 2015" se tuvo como finalidad comparar la eficacia o aciertos del algoritmo Fisherfaces y LBPH, que actualmente son los más populares en la aplicación de la tarea de reconocimiento facial. Así, para darle uso a esta medida biométrica a través de las cámaras de seguridad como respuesta al incremento de la delincuencia en Perú.

Para lo cual, emplearon una población controlada de 20 personas que permitieron realizar las diferentes pruebas a través de un software desarrollado en el entorno .net y mediante el uso de la librería OpenCV. Las pruebas se realizaron bajo las mismas condiciones con el fin de tener resultados que no contengan ningún tipo de sesgo y que sean lo más representativos posibles.

Los resultados obtenidos mostraron la superioridad del algoritmo Fisherfaces sobre el LBPH con un 8% mayor de precisión. Así, recomiendan la utilización del algoritmo Fisherfaces para la elaboración y desarrollo de software para cámaras de videovigilancia siempre y cuando se coteje con el algoritmo LBPH.

De esta manera, el presente antecedente contribuye con información y conocimiento necesario acerca de los dos algoritmos más populares en tareas de reconocimiento facial que permitirán una fácil implementación y posterior comparación con los resultados obtenidos en el presente estudio.

En el artículo científico (7) "Procesamiento de video usando Apache

Hadoop con OpenCV y JavaCV para reconocimiento facial" se presenta una solución tecnológica e informática de bajo coste que permite analizar el flujo de video de las cámaras de seguridad para la identificación de personas haciendo uso de las librerías OpenCV y JavaCV. La cual, contribuye con en aplacar el impacto de la creciente inseguridad ciudadana en Ecuador, resaltando su aplicación sobre el Sistema de Servicio Integrado de Seguridad ECU-911 que contiene a los delincuentes más buscados en el país de los cuatros mundos.

Para la construcción del sistema, se desarrolló en base a 4 aspectos. En primer lugar, se hace referencia a la tecnología HDFS que permitió tener un servicio de almacenamiento distribuido para datos de video. Como segundo punto, se resalta el uso del Fuse DFS característico de Hadoop que permitió las acciones de lectura, escritura y operaciones directas sobre el sistema de archivos distribuido localmente. En el tercer punto, se implementan las librerías JavaCV y OpenCV que permitieron el procesamiento del flujo de video. Finalmente, se utilizó la programación MapReduce de Hadoop que procesó los datos del video de una manera simultánea así logrando que el rendimiento tenga una dependencia del número de archivos que se analicen. Véase la Figura 4.

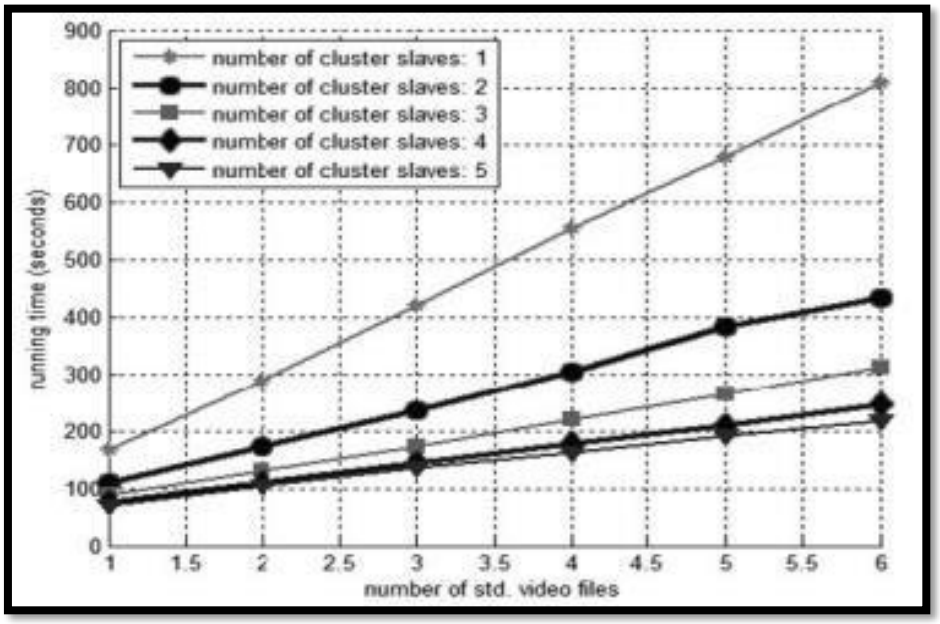


Figura 4. Tiempo de respuesta de opencv

Fuente: (7)

A través del desarrollo del producto se concluye que la propuesta a una pequeña escala el rendimiento es muy satisfactorio, resaltando que la rapidez depende del poder computacional y el número de archivos a analizar a través del MapReduce. Por otro lado, se resalta que el sistema de reconocimiento facial también puede trabajar utilizando imágenes, que reemplazarían a los fotogramas y el rendimiento sería mayor.

Por tal razón, la investigación contribuye con conocimiento en el manejo del flujo de video y su rendimiento con el uso de la librería OpenCV que permite conectarse tanto cámaras de seguridad como a webcams a través de diferentes protocolos que garantizan un funcionamiento continuo.

En el trabajo investigación (8) "Sistema de vigilancia biométrico facial para el control delincuencial en la división policial Chimbote" se menciona la importancia de la existencia de las cámaras de vigilancia en puntos estratégicos que permitan monitorear las calles y con la aplicación de algoritmos de reconocimiento facial lograr la identificación de requisitoriados con el fin de identificar amenazas que atenten contra los ciudadanos. Para la investigación se usaron las 64 cámaras de video vigilancia del distrito de Chimbote. Se resalta la evaluación de las dos formas de reconocimiento facial tanto el geométrico (basado en rasgos) y el fotométrico (basado en lo visual). Para la investigación se aplicó el algoritmo de correspondencia entre agrupaciones de grafos elásticos (EBGM). Este algoritmo crea una estructura dinámica que proyecta el rostro sobre una planilla elástica que se situa en los puntos estratégicos de un rostro. Así, se evalúa el comportamiento de todos los píxeles de la imagen. Véase la Figura 5.

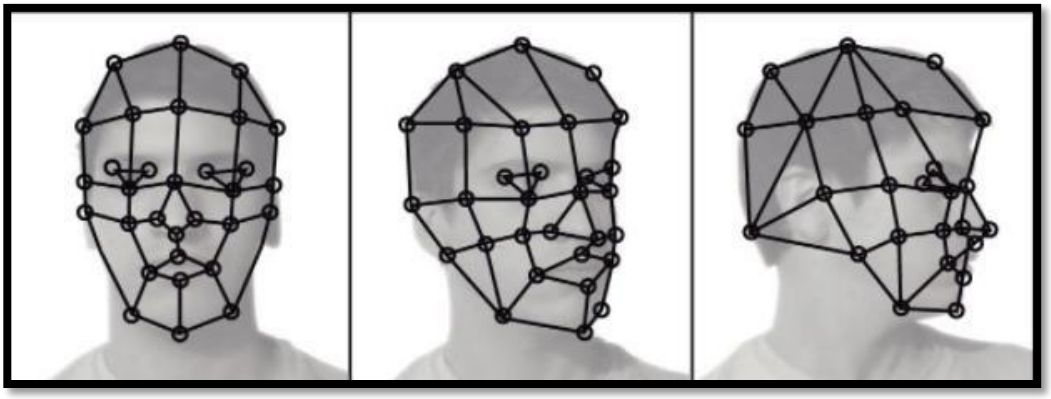


Figura 5. Mapeo de las características faciales

Fuente: (8)

En conclusión, el presente estudio brinda información sobre los protocolos de conexión rtp y rtsp que permiten una conexión continua con las cámaras de seguridad. Además, aporta conocimiento que permite diferenciar entre los tipos de reconocimiento facial tanto geométrico como fotométrico.

En el trabajo de investigación (9) "Diseño de un sistema de seguridad física mediante Reconocimiento Facial a través del flujo de video, siguiendo las mejores prácticas de las normas ISO 80601". La tesis se basa en el diseño de un sistema de seguridad física con la aplicación de reconocimiento facial para el control de accesos de los usuarios que ingresan y transitan por las áreas restringidas de una empresa del sector minero. Para la propuesta del diseño del sistema de seguridad se resaltaron las ventajas que implicaba la aplicación de reconocimiento facial en las instalaciones de la empresa. Se destacó la no manipulación de un tercero que proponía una reducción de tiempo de control y reducción en costos. Además, es un sistema higiénico que en tiempos de pandemia proponía un control sin ningún tipo de contacto, como método no intrusivo. Finalmente, se catalogó como un sistema que tiene un accionar en tiempo real lo cual mitiga el impacto de las amenazas. Dentro de la investigación se menciona que para una mayor precisión en cuanto a la identificación se debe tener en cuenta la posición del sujeto y fondo, calidad de imagen, colores e iluminación, peinado y expresión, anteojos y mascarillas. Se

recomienda registrar a los usuarios en todos estos diferentes aspectos para el entrenamiento del modelo con el fin de tener un reconocimiento preciso. Resaltan el uso de las cámaras de seguridad con un mínimo de calidad de HD 720P para lograr una mayor eficacia.

Finalmente, se resalta su aporte sobre el desarrollo de un sistema de reconocimiento facial siguiendo las mejoras prácticas de la norma ISO 80601. Las cuales permitieron, en un contexto de pandemia, evitar la cercanía entre el personal que se encontraba en las instalaciones de la empresa minera y garantizar un accionar seguro por parte del sistema de reconocimiento facial.

En el artículo científico (10) "Aplicación de deep learning para el reconocimiento facial con la presencia de oclusiones en el contexto de la pandemia covid 2021" resaltan la aplicación del Deep Learning para lograr un reconocimiento facial preciso, incluso en personas que usen accesorios en la cara como lentes o mascarillas faciales en estos contextos de pandemia. La investigación propone una metodología de cuatros fases: 1) Creación del DataSet 2) Entrenamiento de la red neuronal 3) Ejecución 4) Análisis del resultado. Se aplicó el algoritmo VGG16, que está previamente entrenado, en base a un dataset de 2400 imágenes que permitieron el entrenamiento del modelo. Como resultado se obtuvo un 71% de precisión en la identificación facial. En la Figura 6 se puede evidenciar las clases que hacen referencia al personal que se trató de identificar en base a diferentes algoritmos para obtener la precisión final del modelo. Véase la Figura 6.

CLASE	TIPOS				PORCENTAJE
	SS	SC	CS	CC	
Roy Uscamayta	100%	90%	80%	60%	82.5%
Jemima Elias	100%	80%	100%	70%	87.5%
Angiela Rojas	100%	80%	70%	40%	72.5%
Milagros Alfaro	40%	40%	30%	20%	32.5%
Luis Salazar	90%	80%	80%	70%	80%
PORCENTAJE TOTAL					71%

Figura 6. Comparación de la precisión entre algoritmos de reconocimiento facial

Fuente: (10)

El resultado del entrenamiento se encuentra en un porcentaje total que estaba dentro del rango objetivo que garantiza la efectividad de una aplicación de reconocimiento facial.

Se concluye que el uso de la librería VGG16 permitió la transferencia de aprendizaje hacia el entrenamiento. Así, con un entrenamiento de 500 ciclos de entrenamiento en un periodo de 8 horas garantizando un entrenamiento eficiente para la aplicación del algoritmo. Asimismo, se resalta su contribución de conocimiento en el algoritmo que facilitará su implementación y servirá de guía para poder comparar los resultados con algoritmos basados en inteligencia artificial.

En el trabajo de investigación (11) "Implementación de un sistema de gestión de seguridad electrónica con Machine Learning dirigido a Prosegur Perú para gestión de seguridad en viviendas de Lima Metropolitana" se resalta la aplicación de la tecnología de Machine Learning para aplicarlas sobre los productos de Prosegur con el objetivo de alcanzar una mayor seguridad en viviendas de Lima Metropolitana. De esta manera, se resaltan los algoritmos de redes neuronales como materia prima para el desarrollo de un modelo de reconocimiento facial que permita a los clientes mantener un monitoreo en tiempo real de personas que ingresan a su vivienda. Se busca implementar un sistema que no solo permita el reconocimiento facial, sino que alarme a los dueños de la vivienda si una persona no registrada está en la casa. Así, se busca lograr una reducción de los crimines vinculados al patrimonio y mitigar el impacto de estos.

Finalmente, se resalta el aporte de conocimiento sobre la aplicación del reconocimiento facial en tiempo real de manera continua, haciendo uso del flujo de video de las cámaras de vigilancia. Así, proponiendo una implementación de un accionar mucho más rápido que permitirá prevenir cualquier tipo de ingreso no autorizado, que afecta significativamente la precisión del sistema.

2.1.2. Antecedentes internacionales

A continuación, se presentan los antecedentes de la investigación, se revisaron y analizaron diversas fuentes relacionadas a la investigación como tesis y artículos científicos, los cuales nos ayudaron a generar valor a la presente tesis.

En el artículo científico (12) “Reconocimiento facial con base en imágenes” se presenta un proceso de desarrollo de sistemas de reconocimiento facial en base a 5 etapas que garantizarán un funcionamiento óptimo. La investigación parte de la problemática de la creciente ineficacia de los sistemas de biométricos y su precisión en su accionar, culpando al no procesamiento y debido control de los factores externos como la iluminación, el fondo de la imagen, ángulo y opacidad.

El proceso establecido por la investigación consiste cinco etapas continuas: 1) Detección facial o de rostro 2) Acondicionamiento 3) Normalización 4) Extracción de características 5) Reconocimiento. La primera etapa permite la localización del rostro y la posterior segmentación de la imagen. Así, a través del acondicionamiento se logrará localizar los componentes y la escala a la que se encuentra el rostro. Siguiendo el proceso, la normalización procesarán las imágenes de modo que atenúen los efectos de los cambios de iluminación, opacidad y distancias. Consecuentemente, el cuarto proceso logrará aportar la información necesaria para diferenciar los rostros. Finalmente, el patrón extraído se comparará con la base de datos para realizar el reconocimiento.

El estudio pudo determinar que el proceso planteado tuvo un impacto positivo sobre la medida biométrica de reconocimiento facial logrando una precisión del 90%. Por tal razón, se resalta el aporte de conocimiento en sistemas de reconocimiento facial que trabajan con imágenes o capturas de video. Las cuales, permiten obtener una precisión mayor en cuanto al reconocimiento. Sin embargo, es una opción que dilata el tiempo de identificación.

En el artículo científico (13) “A Siamese Long Short-Term Memory Architecture for Human Re-identification” se presenta una arquitectura innovadora de memoria a corto plazo (LSTM) siamesa que puede procesar regiones de imágenes secuencialmente que mejoran la capacidad de re-identifiación que consiste en identificar a una persona en dos puntos diferentes a través del uso de las cámaras de

vigilancia con el objetivo de tener un sistema de videovigilancia inteligente que promueva una mayor seguridad ciudadana.

El proyecto se basa en la creación de una red neuronal siamés que tome un par de imágenes como entradas para poder aplicar una diferencia sobre los pixeles procesados y determinar si pertenecen a la misma persona. Para lo cual, se utiliza el algoritmo de optimización descenso de gradiente estocástico que una vez que calculen los pesos (W) para cada de las imágenes se propagarán hacia atrás con el fin de minimizar el error lo máximo posible. Así, para el entrenamiento tratan las imágenes en lotes de 100 pares y es realizado a través de 20 épocas con una parada temprana en base al performance del algoritmo cross-validation. Y se probaron los datasets Market-1501, CUHK03 y VIPeR.

Se concluyó que la red LSTM puede propagar selectivamente información contextual y así mejorar la capacidad de identificación de características locales en dos imágenes. Así se garantiza que la red a través de la propagación sea capaz de memorizar información relevante que permitirá la reidentificación.

Finalmente, se resalta el aporte en cuanto al uso de las redes neuronales siamesas y la arquitectura a implementar para promover una identificación facial rápida con una cantidad mínima de imágenes en el dataset.

En el trabajo de investigación (14) “Diseño e Implementación de un Sistema Embebido de Reconocimiento Facial para el Control de Acceso usando Deep Learning” se presenta una solución de bajo costo con un procesador NCS2 de Intel que proporciona la capacidad de procesamiento suficiente para desarrollar una solución eficaz. Así, resaltando que en los últimos años se ha incrementado el uso de identificadores biométricos por el nivel de seguridad que ofrecen pero que debido a la exigencia de hardware no se pueden desarrollar sistemas sofisticados por la barrera económica.

Para la implementación de la solución se presentó un modelo de arquitectura que permite visualizar el flujo del sistema embebido de reconocimiento facial. Así, comenzando el flujo desde el entrenamiento

para poder continua con el proceso de optimización e inferencia para poder enviar los datos procesados a la aplicación de usuario. Véase la Figura 7.

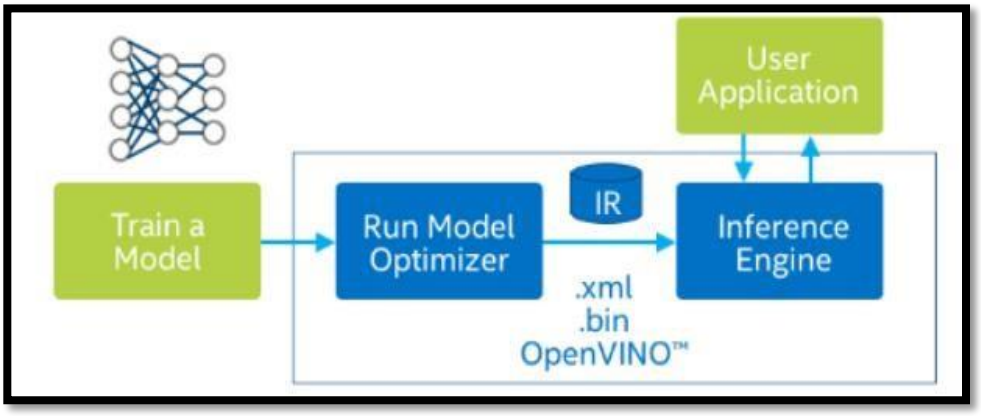


Figura 7. Arquitectura del sistema de reconocimiento facial

Fuente: (14)

Se aplica un modelo de red neuronal con librerías de Python en específico la librería OpenCV que permitió generar un sistema de control de accesos con reconocimiento e identificación facial de tipo embebido. Para la implementación de la solución se usó un microcontrolador Raspberry PI 3 con un procesador NCS2 que conectados en un circuito controlador de apertura de cerradura controlan el acceso frente a las señales que envía el algoritmo de reconocimiento facial.

De tal manera, es necesario resaltar el conocimiento sobre los procesadores de placa reducida que permiten un procesamiento rápido y además permiten integrar una cámara de gama media que facilita el proceso de instalación y mantenimiento del sistema de reconocimiento facial.

En el artículo científico (15) “Facial Recognition using Convolutional Neural Networks and Implementation on Smart Glasses” se desarrolla un reconocimiento facial en lentes inteligentes que garanticen la precisión del reconocimiento y su fácil portabilidad. El cual, soluciona las limitantes de esta medida biométrica como la luz, la posición, el ángulo, la distancia y el fondo que impiden un buen reconocimiento.

Para el desarrollo de los lentes inteligentes se utiliza una Raspberry Pi de bolsillo que contendrá la red neuronal y la base de datos para la identificación. Para esto, se implementará una pequeña cámara al lado derecho de los lentes inteligentes que al capturar una imagen la enviarán a la CPU para evaluar si coincide. Así, presentan su diagrama de flujo de hardware. Véase la Figura 8.

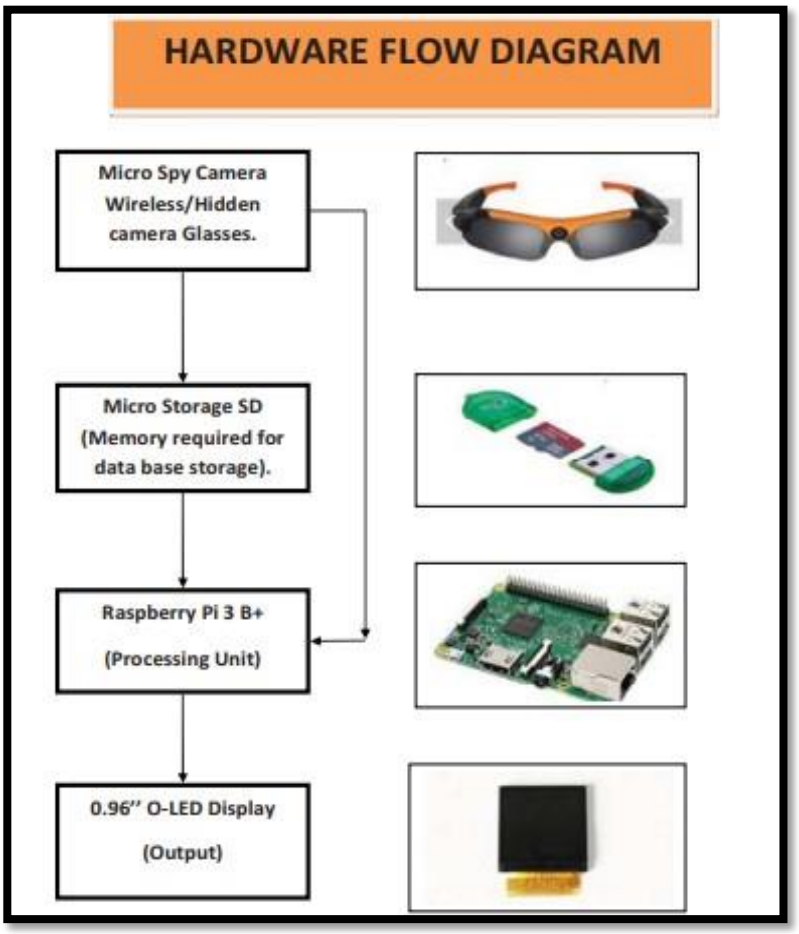


Figura 8. Arquitectura de hardware de un software de reconocimiento facial

Fuente: (15)

Se resalta su aporte en el uso redes neuronales convolucionales para el reconocimiento facial y su aplicación sobre la autenticación para mejorar la seguridad de una organización. Asimismo, su aplicación sobre gafas inteligentes que son superiores a las cámaras en cuanto a la portabilidad y su

frontalidad. Como resultado se obtuvo una precisión del 98.5% usando un conjunto de 2500 imágenes que sirvieron para el desarrollo del modelo.

En el artículo científico (16) “Siamese Neural Networks for One-shot Image Recognition” menciona la importancia de las redes neuronales siamesas y su contribución sobre la identificación facial en tiempo real. Las redes neuronales siamesas pertenecen a los algoritmos one-shot que si bien es un concepto todavía inmaduro se comprueba que para la clasificación de imágenes en video tiene un gran potencial debido a que la mayoría de los algoritmos evalúan tratan de evaluar el video por fotograma mientras que la SNN toma una captura del video y evalúa la imagen en base a una base de datos lo cual promueve un mayor tiempo de respuesta y eficiencia. Así, al evaluar una imagen no la compara con cada una de las imágenes dentro de la base de datos, sino que las compara simultáneamente.

Véase la Figura 9.

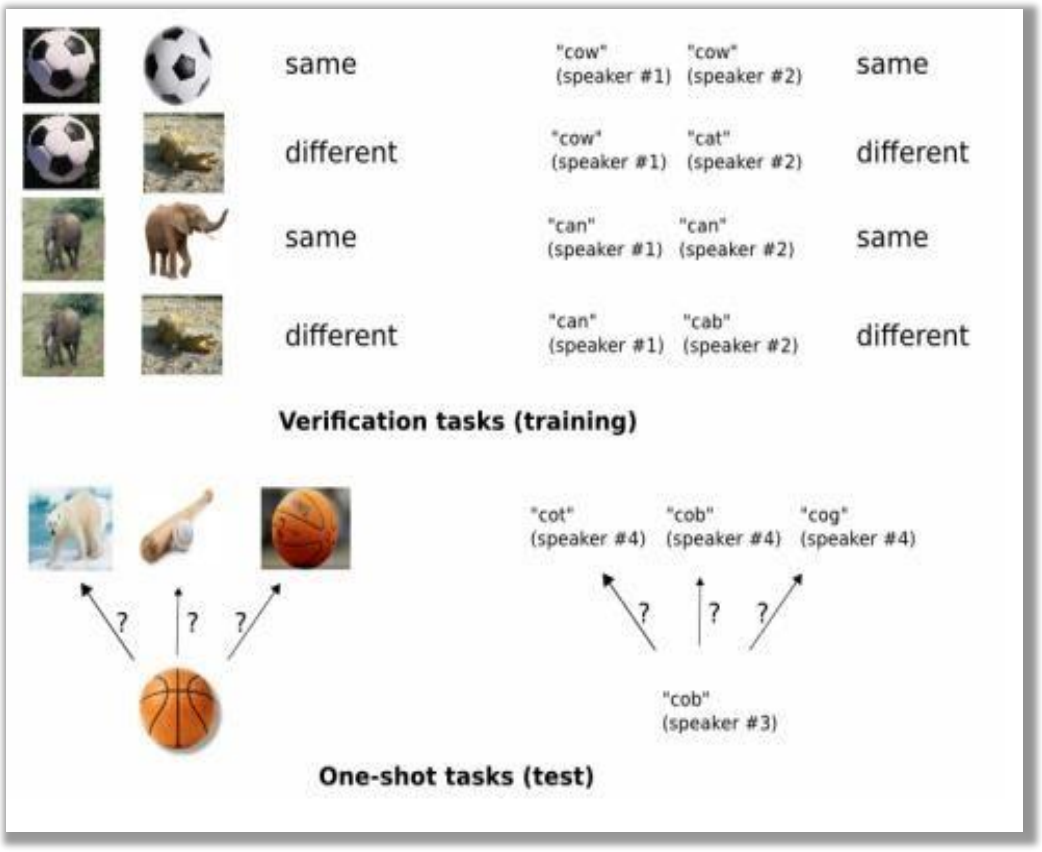
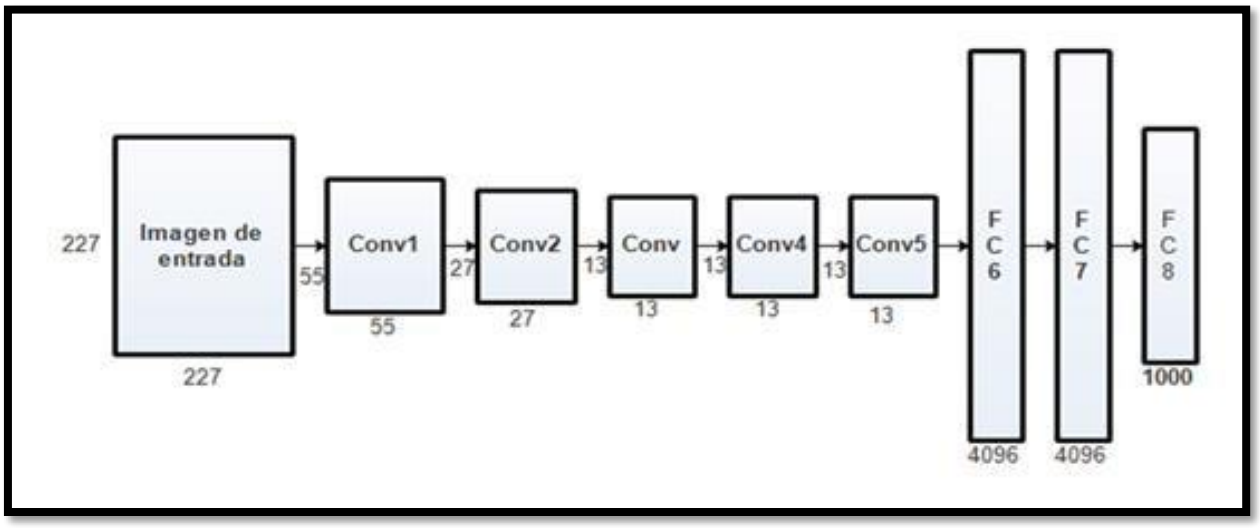


Figura 9. Aprendizaje de una sola imagen

Fuente: (16)

Además, brindan una estructura de red neuronal convolucional para poder aplicarla sobre la clasificación de imágenes. Véase la Figura 10.



.

Figura 10. Arquitectura de una red neuronal siames

Fuente: (16)

De esta manera, se evalúan las características de la cara para poder compararlas con las reconocidas. Así, se logra clasificar las imágenes y tener un resultado con una alta precisión.

En conclusión, el presente estudio contribuye con una información precisa de la forma en como desarrollar una red neuronal siamés a través de sus diferentes capas en cuanto a profundidad y ancho de la misma. Además, realiza un aporte en cuanto al uso de la función de pérdida de triplete que permitirá un entrenamiento personalizado y con mejores resultados para tareas de procesamiento de imágenes.

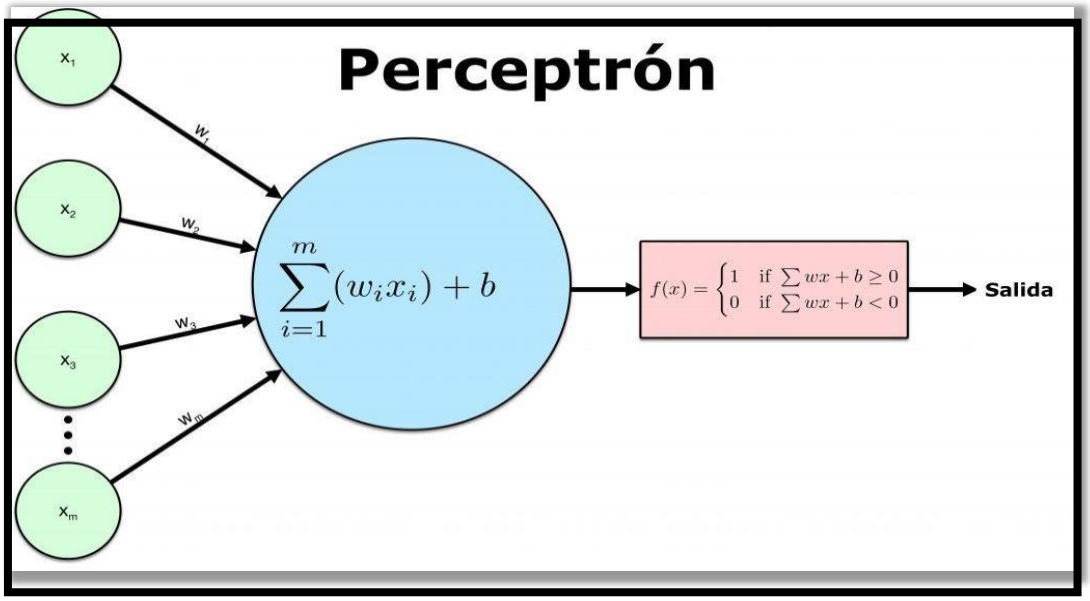
2.2. Bases teóricas

2.2.1. Redes neuronales artificiales

Según (17) se tratan de "Un modelo computacional altamente eficiente que ha ido evolucionando a partir de diversas aportaciones prácticas y científicas, cuyo objetivo es dar solución a los problemas de la misma manera que un cerebro humano. Consiste en un conjunto de neuronas artificiales que están interconectadas. La información tiene un flujo desde la entrada, pasando por una fase de procesamiento y finalmente produce unos valores de salida".

En la actualidad las redes neuronales se utilizan para una gran variedad de tareas, como la clasificación, visión por computador, predicción y el procesamiento de lenguaje natural, que son complicados de resolverlos con una programación básica.

Cada neurona está interconectada con otra a través de conexiones que representan el procesamiento que se hace desde una capa de neuronas a otra. En estas conexiones, el resultado de la neurona anterior es procesado a través de un valor de peso. Del mismo modo, al finalizar el proceso se aplica un peso que modifica el valor del resultado final. Este proceso es debido a la función de activación. El cual, determinará si la neurona es activada o no. Véase en la Figura 11.



.

Figura 11. Arquitectura de un perceptrón

Fuente: (17)

Una red neuronal artificial está compuesta por:

- Entrada o Input, hace referencia a los datos de entrada de la red o a los resultados de la neurona anterior.

- Sumador, representa la operación de suma de todas las entradas o inputs de la red, equilibrándolas con un umbral o peso correspondiente.
- La función de activación supone la activación o no de la neurona en base al resultado obtenido del procesamiento anterior.
- Salida o Output, hace referencia al resultado o señal que se obtiene al final de la red neuronal o que será destinado hacia otra neurona.

2.2.1.1. Tipos de redes neuronales artificiales

Según (18), se presentan los siguientes tipos de redes neuronales artificiales:

- Red neuronal monocapa: "La red neuronal monocapa se corresponde con la red neuronal más simple, está compuesta por una capa de neuronas que proyectan las entradas a una capa de neuronas de salida donde se realizan los diferentes cálculos. Su aprendizaje requiere conocer los valores esperados, es decir, que exige un aprendizaje supervisado". Véase la Figura 12.

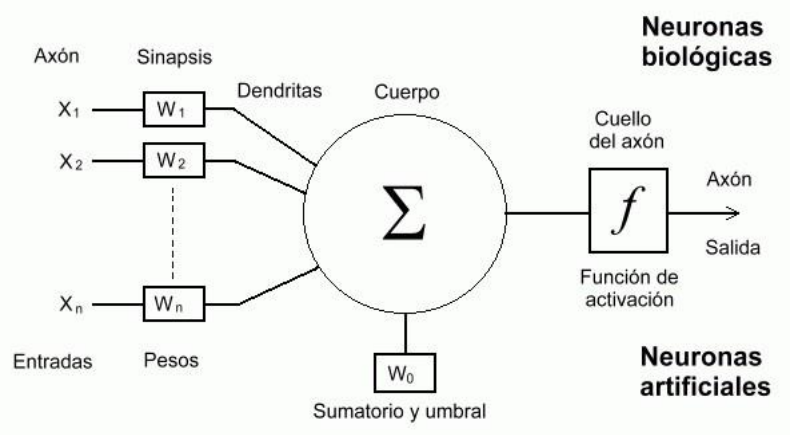


Figura 12. Red neuronal monocapa

Fuente: (18)

- Red neuronal multicapa: "El perceptrón multicapa es una red neuronal artificial (RNA) formada por múltiples capas, de tal manera que tiene capacidad para resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón (también llamado perceptrón simple)". Véase la Figura 13.

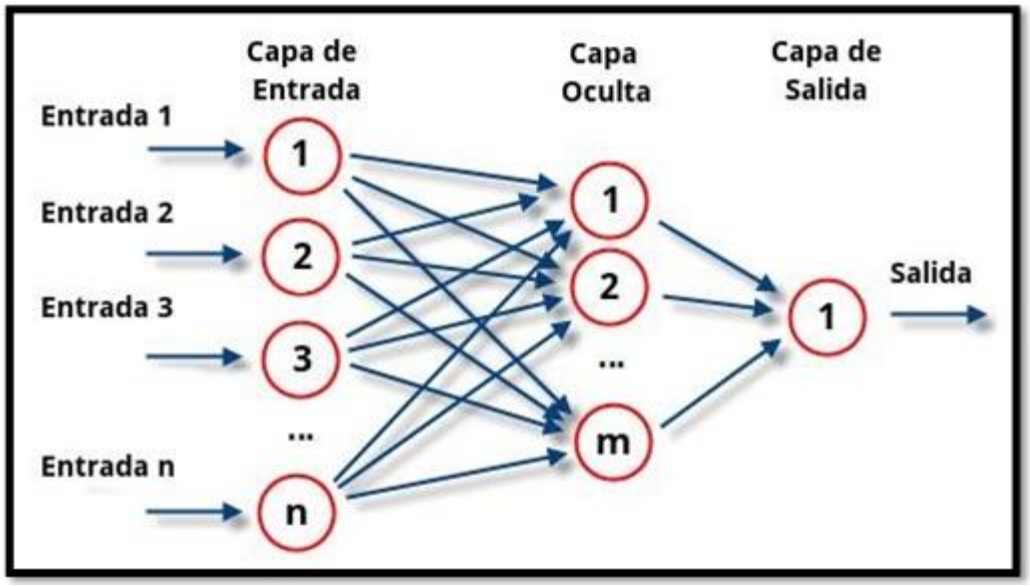


Figura 13. Red neuronal multicapa

Fuente: (18)

- Red neuronal convolucional (CNN): “Las CNN son versiones regularizadas de perceptrones multicapa. Los perceptrones multicapa generalmente significan redes completamente conectadas, es decir, cada neurona en una capa está conectada a todas las neuronas en la siguiente capa. La "conectividad total" de estas redes las hace propensas al sobreajuste de datos. Las formas típicas de regularización, o de prevención del sobreajuste, incluyen: penalizar los parámetros durante el entrenamiento (como la caída del peso) o recortar la conectividad (conexiones omitidas, abandonos, etc.)”. Las CNN se adaptan un enfoque diferente hacia la regularización: aprovechan la estructura jerárquica en los datos y ensamble patrones de complejidad creciente utilizando patrones simples en sus filtros. Por lo tanto, en una escala interconectada y compleja, las CNN están en el extremo inferior. En la Figura 17 se puede visualizar la arquitectura tradicional de una red convolucional del tipo perceptrón multicapa. Así, evidenciando las diferentes capas que se

van procesando a través de convoluciones y que demuestran su procesamiento interno como varias redes neuronales trabajando de forma continua. Véase la Figura 14.

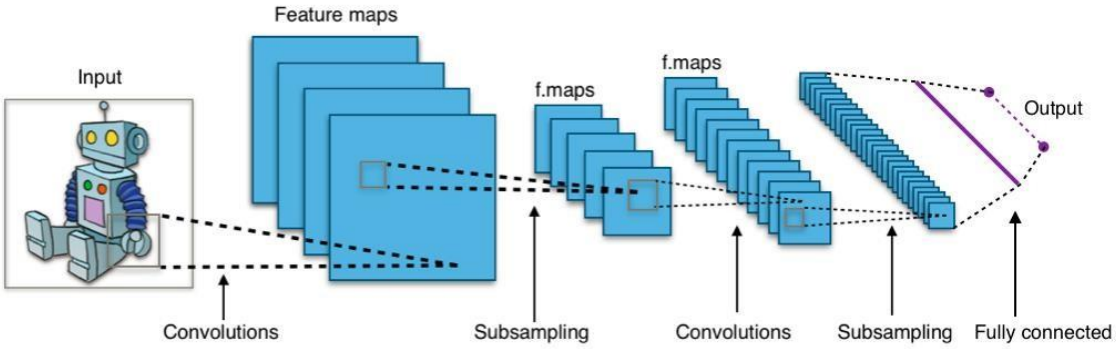


Figura 14. Red neuronal convolucional

Fuente: (18)

- Red neuronal recurrente: "Es una clase de redes neuronales artificiales donde las conexiones entre nodos forman un gráfico dirigido o no dirigido a lo largo de una secuencia temporal. Esto le permite exhibir un comportamiento dinámico temporal. Derivados de redes neuronales feedforward, los RNN pueden usar su estado interno para procesar secuencias de entradas de longitud variable" (19). Esto las convierte en tareas aplicables tales como el reconocimiento de escritura a mano conectado o el reconocimiento de voz. Las redes neuronales recurrentes están continuamente validadas a través del test de Turing y pueden ejecutar programas arbitrarios para procesar secuencias de inputs. Véase la Figura 15.

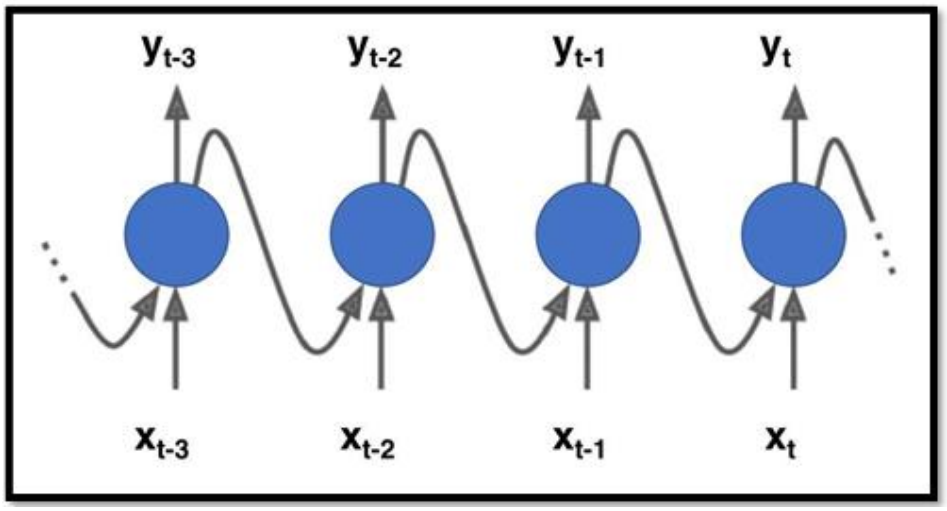


Figura 15. Red neuronal recurrente

Fuente: (18)

2.2.1.2. Aprendizaje de una red neuronal artificial

Las redes neuronales pueden solucionar los problemas que se han presentado durante décadas al desarrollar aplicaciones que exigen unos requerimientos de mayor capacidad a través de la programación tradicional. Como hemos visto, su sencilla estructura oculta su complejidad a nivel de programación y matemático. Las redes neuronales funcionan propagando inputs, procesos y sesgos hacia adelante. Sin embargo, es en el proceso inverso de propagación hacia atrás donde la red aprende a través del error a determinar los cambios exactos que se deben aplicar a los promedios y sesgos para producir un resultado exacto (19).

Desde el punto de vista del aprendizaje por máquina, el aprendizaje consiste en minimizar la diferencia entre el resultado real y el que se obtuvo como resultado. Este proceso es arduamente tedioso y consume muchos recursos e infraestructura del procesador, como pone de manifiesto el tiempo que tarda en ejecutarse una revolución o época. Favorablemente, este aprendizaje solo se realiza al momento de desarrollar la red neuronal y no cada vez que se necesita su aplicación.

Lo que ha atraído el mayor interés en las redes neuronales es su capacidad de predicción, clasificación o agrupación que se da de manera automática. Dada una determinada tarea a resolver, y una clase de

funciones F, el aprendizaje consiste en utilizar un conjunto de datos pasados para encontrar la función que resuelve la tarea de forma óptima. Esto consiste en la definición de una función de coste (20), Véase la Figura 16:

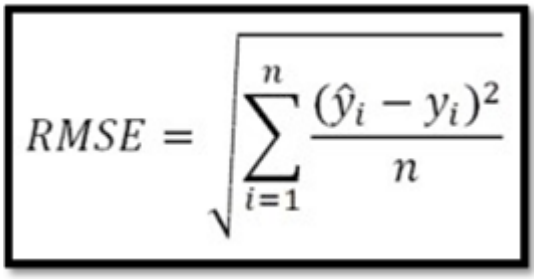


Figura 16. Función de Coste

Fuente: (20)

La función de coste C es un concepto importante en el mundo del aprendizaje automático, ya que representa lo lejos que una solución particular se encuentra del resultado óptimo al problema a resolver. Los algoritmos de aprendizaje buscan a través de su procesamiento encontrar una función o proceso que represente el menor costo posible. Así, con la gran magnitud de problemas y con requerimientos cada vez más exigentes se plantean diferentes formas de aprendizaje que cumplan con las expectativas. De esta manera, se plantearon las siguientes formas de aprendizaje (20):

- El aprendizaje supervisado, “Es una rama de Machine Learning , un método de análisis de datos que utiliza algoritmos que aprenden iterativamente de los datos para permitir que los ordenadores encuentren información escondida sin tener que programar de manera explícita dónde buscar. El aprendizaje supervisado es uno de los tres métodos de la forma en que las máquinas "aprenden": supervisado, no supervisado y optimización” (20).
- El aprendizaje no supervisado, “Es una de las formas en que Machine Learning (ML) "aprende" los datos. El aprendizaje no supervisado tiene datos sin etiquetar que el algoritmo tiene que intentar entender por sí mismo. El aprendizaje supervisado es en el que se etiquetan los conjuntos de datos para que haya una clave de respuestas con la que la máquina pueda medir su precisión. Si Machine

Learning fuera un niño que aprendiera a andar en bicicleta, el aprendizaje supervisado es el padre que corre detrás de la bicicleta y la sostiene en posición vertical. El aprendizaje no supervisado consiste en entregar la bicicleta, darle palmaditas en la cabeza al niño y decirle buena suerte "(20).

- El aprendizaje por refuerzo, "Es un área del aprendizaje automático inspirada en la psicología conductista; donde la máquina aprende por sí sola el comportamiento a seguir en base a recompensas y penalizaciones. Las técnicas de aprendizaje automático supervisado y no supervisado están revolucionando nuestra industria" (20).

2.2.1.3. Red neuronal siamés

La siguiente red neuronal siamés, véase la Figura 17 "Es una red neuronal artificial que usa los mismos pesos mientras trabaja en tándem en dos vectores de entrada diferentes para calcular vectores de salida comparables. A menudo, uno de los vectores de salida se calcula previamente, formando así una línea base contra la cual se compara el otro vector de salida.Esto es similar a la comparación de huellas dactilares , pero se puede describir más técnicamente como una función de distancia para el hashing sensible a la localidad" (21).

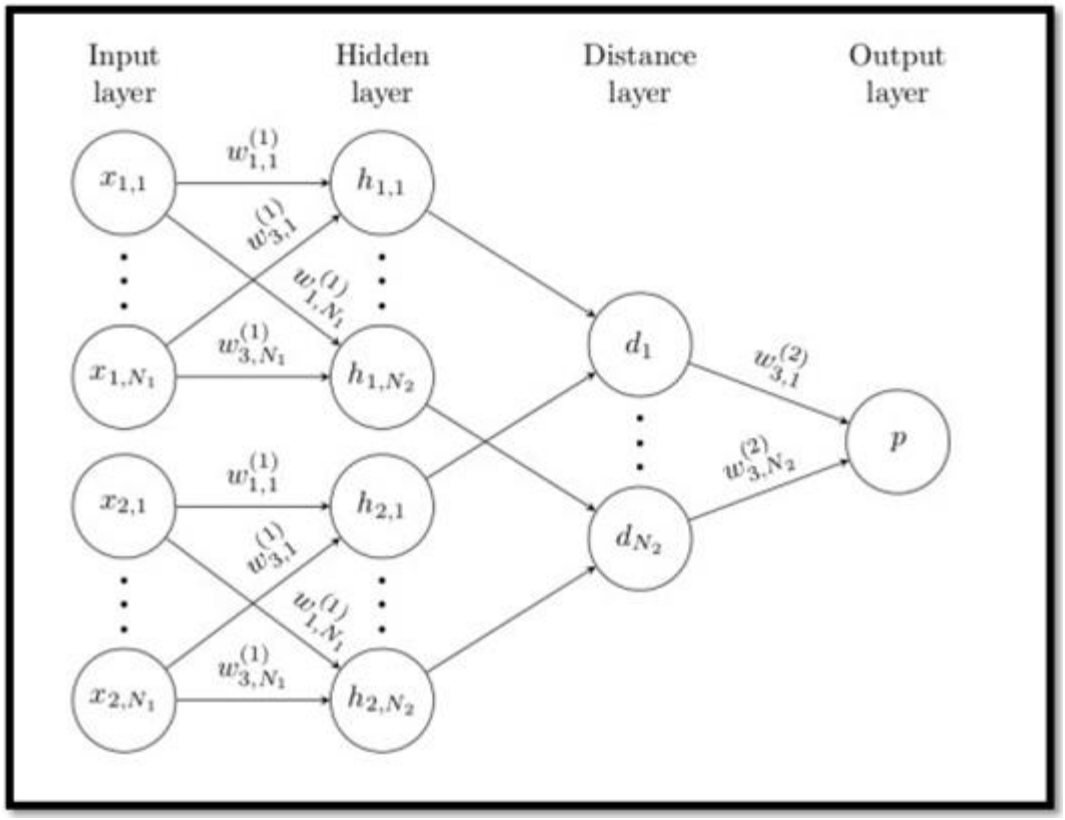


Figura 17. Estructura de una red neuronal siamés

Fuente: (21)

Es posible modificar la arquitectura de una red siamés para su aplicación en otros campos que requieran de una verificación de dos entradas. Se recomienda modificar las capas escondidas que centran en el procesamiento de las entradas, más no la función de similaridad. Esto se usa frecuentemente para soluciones que quieran comparar entradas parecidas en diferentes conjuntos de datos.

"El One-shot learning surge de la necesidad de aprender las características requeridas para la tarea específica con la utilización de una única imagen por clase en la base de datos" (22). Este tipo de procesamiento están inspirados en base a la habilidad de los seres humanos para aprender u obtener un conocimiento sin la necesidad de repetirlo más de 100 veces. En la presente, este constituye un campo de estudio bastante activo con diferentes publicaciones en el tema. Este método es útil en el caso del

reconocimiento facial, dónde los sistemas tienen que actuar de una forma ágil ya que las limita a recolectar cientos de imágenes y entrenar el modelo en periodos de varias horas. El proceso se centra en solicitarle al usuario que provea fotos de su rostro en varios ángulos, a diferentes edades, con cambios en su estilo o uso de accesorios, entre otros; no sólo sería ineficiente en términos de usabilidad sino probablemente iría en contra su derecho a la privacidad. Véase la Figura 18.

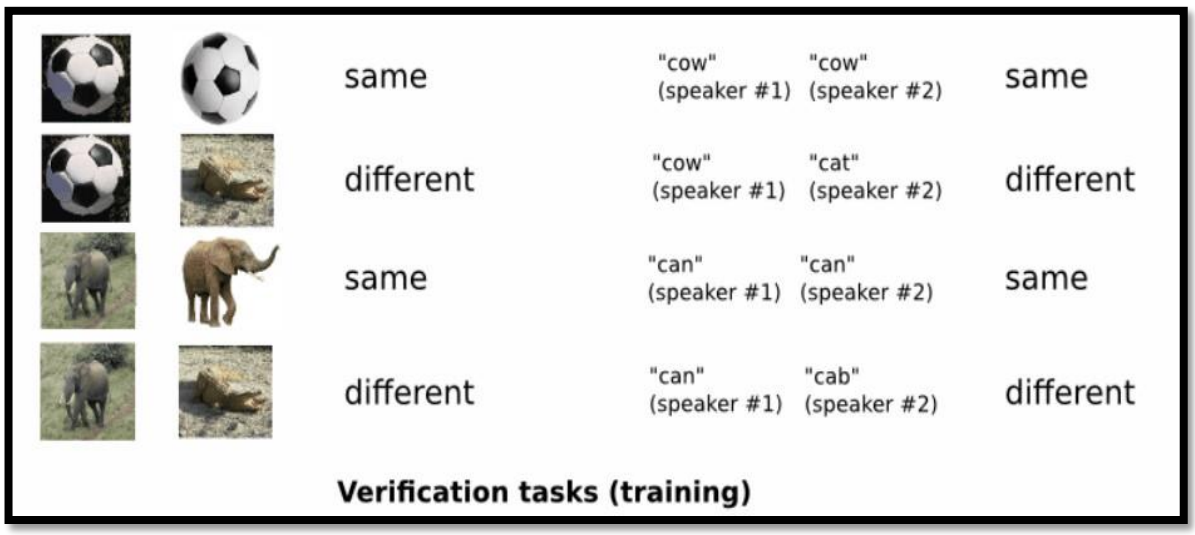


Figura 18. Red neuronal siamés, aprendizaje de una sola imagen

Fuente: (22)

“Los usos de las medidas de similitud donde se podría usar una red gemela son cosas como el reconocimiento de escritura, en la detección automática de rostros en flujos de video o imágenes y la coexistencia de consultas con documentos indexados. La aplicación más conocida de las redes gemelas es el reconocimiento facial, donde las imágenes conocidas de personas se calculan previamente y se comparan con una imagen de un torniquete o similar” (23). Una gran diferencia que se tiene que resaltar es que existen dos problemas ligeramente diferentes. Uno es identificar a una persona entre un gran número de otras registradas, ese es el problema del reconocimiento facial. DeepFace es un ejemplo de tal sistema. En su forma más compleja se busca reconocer a una sola persona en una estación de aeropuerto o tren. El otro es la verificación facial , es decir, verificar si la foto en un pase es

la misma que la persona que afirma ser la misma persona. La red gemela puede ser la misma, pero el desarrollo y su implementación es totalmente diferente.

El aprendizaje en redes gemelas se puede realizar con la función de pérdida de triplete o pérdida de contraste . Para el aprendizaje por pérdida de triplete, se compara un vector de referencia (imagen de anclaje) con un vector positivo (imagen positiva) y un vector negativo (imagen falsa). El vector negativo forzará el aprendizaje en la red, mientras que el vector positivo actuará como un normalizador y regulador de los parámetros de aprendizaje. Para el aprendizaje por pérdida contrastiva debe haber un decaimiento de peso o coeficiente para regularizar los resultados del modelo, o alguna operación parecida a una normalización (24). Véase la Figura 19.

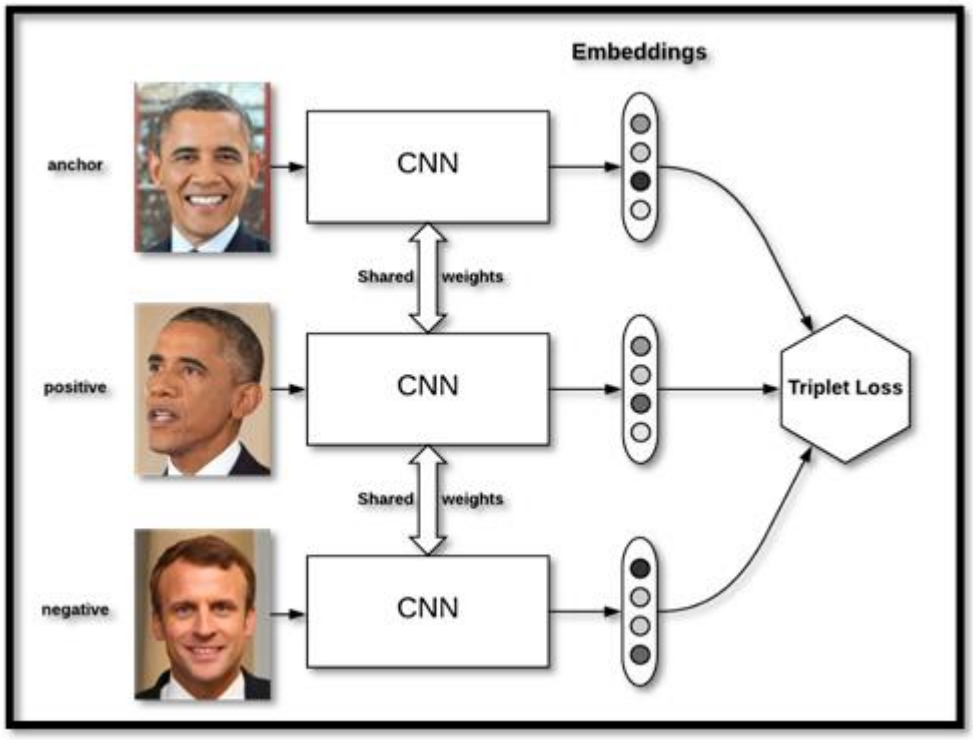


Figura 19. Función de pérdida triple

Fuente: (24)

La aplicación más reconocida en el tema es la de control de accesos, la cual permite que exista un ingreso seguro a instalaciones de edificios, oficinas, aplicaciones digitales, centros con información

sensible, entre otros. Este uso hace que se proteja a las personas y a sus diferentes tipos de información privada. Además, se pueden resaltar aplicaciones en la seguridad para la protección de estaciones de aeropuertos y trenes, alerta al identificar personas requisitoriadas; en la protección y generación de confianza entre los clientes del comercio electrónico o transacciones bancarias. (25).

2.2.2. Reconocimiento facial

El reconocimiento facial "Es una manera de identificar o confirmar la identidad de una persona mediante su rostro. Los sistemas de reconocimiento facial se pueden utilizar para identificar a las personas en fotos, videos o en tiempo real. Es una categoría de seguridad biométrica. Otras formas de software biométrico incluyen el reconocimiento de voz, el reconocimiento de huellas digitales y el reconocimiento de retina o iris. La tecnología se utiliza principalmente para la protección y las fuerzas de seguridad, aunque hay un creciente interés en otras áreas de uso." (26).

El procedimiento de reconocimiento facial se da mediante dispositivos de cualquier tipo y forma que posean de tecnología fotográfica digital como una cámara, para generar y obtener las imágenes que representan los datos necesarios para registrar los rasgos biométricos faciales de la persona a reconocer. A diferencia de otras soluciones de identificación como la huella digital, contraseñas, o la identificación con voz, el reconocimiento biométrico facial utiliza patrones matemáticos dinámicos de la persona resaltan a este sistema como uno de los más difíciles de evadir. (27).

El objetivo del reconocimiento facial "Es encontrar una serie de datos del mismo rostro en un conjunto de imágenes de entrenamiento en una base de datos. La gran dificultad reside en lograr que este proceso se realice en tiempo real, algo que no está al alcance de todos los proveedores de software de verificación de identidad" (27).

Al contrario de lo que ocurre con la tecnología biométrica de huella digital, que son inalterables durante toda la vida, el reconocimiento facial debe tener en cuenta distintos factores ambientales externos como los niveles de luz, ángulos, envejecimiento y la opacidad. Véase la Figura 20.

La buena calidad de la imagen, en un sistema de reconocimiento facial, es un aspecto esencial. Así, es probable que el sistema IFRS no pueda procesar imágenes de manera correcta, lo cual concluirá en identificaciones erróneas, y de la misma manera puede influir considerablemente tanto en la precisión de la búsqueda como en los propios resultados (28).

Figura 20. Mapeo de las características faciales

Fuente: (28)

Los sistemas de reconocimiento facial capturan una imagen a través de un flujo de video o se les brinda imágenes específicas como entrada. Las características de la imagen varían según la calidad del dispositivo mediante el cual se están recolectando las imágenes. Estos realizan una comparación dentro de una base de datos la información y características de la imagen entrante en tiempo real en foto o vídeo, siendo esta última una opción mucho más fiable y segura que la información obtenida en una imagen estática. Este procedimiento necesita de una conexión hacía una base de datos o a un sistema de archivos que puedan gestionar y guardar las imágenes capturadas. En esta comparación de rostros, se analizan los rasgos fiduciales matemáticamente. La imagen entrante y se verifica con los datos biométricos que se encuentran registrados en la base de datos al momento de solicitar un acceso a una aplicación, sistema o incluso edificio.

Se utiliza principalmente en sistemas de video vigilancia que protejan la seguridad y eficiencia del proceso de reconocimiento facial de usuarios. En estos sistemas se utiliza un lector que define las

características del rostro, y cuando este solicita el acceso, se verifica comparando los datos obtenidos con el sistema de archivos que tiene en posesión todas las imágenes. Sin embargo, estas aplicaciones no son útiles a largo plazo ya que, a medida que pasan los años, los rasgos faciales varían y al solicitar el acceso ya no coinciden con la imagen o reducen su precisión significativamente. Para solucionar este problema se recomienda realizar un mantenimiento a las imágenes capturadas del usuario en un plazo establecido. Véase la Figura 21.

Los sistemas de reconocimiento facial en general tienden a funcionar de la siguiente manera:

a. Detección de la cara: "Detecta que hay una cara en la imagen, sin identificarla.

Si se trata de un video, también podemos hacer un seguimiento de la cara. Proporciona la localización y la escala a la que encontramos la cara" (29).

b. Alineación de la cara: "Localiza los componentes de la cara y, mediante transformaciones geométricas, la normaliza respecto propiedades geométricas, como el tamaño y la pose, y fotométricas, como la iluminación. Para normalizar las imágenes de caras, se pueden seguir diferentes reglas, como la distancia entre las pupilas, la posición de la nariz, o la distancia entre las comisuras de los labios. También se debe definir el tamaño de las imágenes y la gama de colores. Normalmente, para disminuir la carga computacional del sistema, se acostumbra a utilizar imágenes pequeñas en escala de grises. A veces también se realiza una ecualización del histograma"

(29).

c. Extracción de características: "Proporciona información para distinguir entre las caras de diferentes personas según variaciones geométricas o fotométricas" (29).

d. Reconocimiento: "El vector de características extraído se compara con los vectores de características extraídos de las caras de la base de datos. Si encuentra uno con un porcentaje elevado de similitud, nos devuelve la identidad de la cara; si no, nos indica que es una cara desconocida" (29).

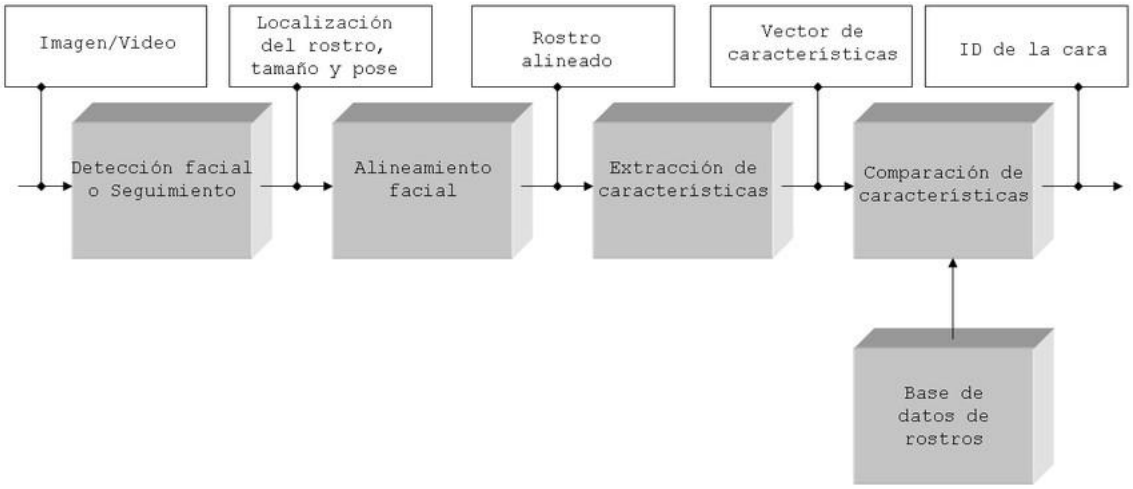


Figura 21. Mapa de proceso de un sistema de reconocimiento facial

Fuente: (29)

Los algoritmos más utilizados y populares para desarrollar soluciones de reconocimiento facial, debido a su ágil implementación, son Fisherfaces, Eigenfaces y LBPH que son ofrecidos por la librería OpenCV. Así, los proyectos biométricos basados en la identificación en base a las características faciales se desarrollaron en base a los presentes algoritmos que permitían una identificación en base a un flujo de video y de manera inmediata. A continuación, se pasan a definir cada uno de los algoritmos.

Eigenfaces: Es uno de los algoritmos en reconocimiento facial más antiguos desarrollado en base a los métodos más simples. Su funcionamiento es confiable en ambientes donde se controlan los factores externos como la luz u opacidad. Además, su precisión está condicionada por el uso de grandes datasets.

Su accionar se basa en la obtención de rostros que estén direccionados frontalmente para poder agregarle una escala de grises que resalte las características faciales. Luego, pasa a realizar una estandarización de los píxeles que facilitan su manejo e impacto. Finalmente, se extraen las características esenciales de un rostro.

Fisherfaces: Es un método que se enfoca en el reconocimiento de caras, teniendo en cuenta la posición facial y los niveles de iluminación. Su funcionamiento se basa en clasificar y dimensionar las

características de un rostro en base al algoritmo discriminante lineal de Fisher. Su estructura está basada en el algortimos Eigenfaces pero está orientada a mejorar la clasificación de clases. Este algoritmo es especialmente útil en imágenes o flujos de video que posean gran variación en la iluminación y en cambios continuos de las expresiones faciales.

LBPH: También conocido como patrón binario local que se diferencia del resto de algoritmos por su facilidad de implementación. Fue propuesto por primera vez en el año 1994 para su aplicación sobre la clasificación de texturas. Su característica estructura permite una simple puesta en marcha. El proceso inicia con la transformación de la imagen o video en una escala de grises para posteriormente aplicar un preprocesamiento que concluya en una matriz binaria de los píxeles de la imagen. De manera que, se aplica la distancia euclidiana que permite comparar los píxeles de dos imágenes y finaliza en su reconocimiento exitoso o fallido.

2.2.3. Control de acceso

Es un método que permite mantener un control de las personas que ingresan y las que no a las instalaciones de edificio u organización. Se refiere a su aplicación en lugares en los que debes mostrar tu documento de identidad para comprobar que efectivamente tienes dicha identidad o permiso de ingresar. El control de acceso es esencial dentro de una organización para que todos los usuarios tengan el acceso correspondiente a los datos necesarios y recursos de sistema (30). Ahora bien, un sistema de control de acceso consiste en una serie de restricciones que se van aplicando de acuerdo con la información, datos o recursos a los cuales se desea acceder. Se basa en los procesos de autorización e identificación.

El control de acceso consiste en la verificación de si una entidad (una persona, vehículo, ordenador, etc...) solicitando acceso a un recurso tiene los permisos o derechos que se requieren para brindarle permiso. Así, ofrece la posibilidad de acceder a recursos físicos (por ejemplo, a un edificio, a un local, a un país) o lógicos (por ejemplo, a un sistema operativo o a una aplicación informática específica).

"Es un sistema electrónico que restringe o permite el acceso de un usuario a un área específica validando la identificación por medio de diferentes tipos de lectura (clave por teclado, tags de proximidad o biometría) y a su vez controlando el recurso (puerta, torniquete o talanquera) por medio de un dispositivo eléctrico como un electroimán, cantonera, pestillo o motor" (31).

Citamos a continuación los diferentes tipos de control de acceso, los que más se han implementado en la actualidad. De acuerdo con los requerimientos de una organización o grupo de trabajo. En base a lo definido podemos resaltar diferentes tipos de control de acceso según el sistema de identificación que utilicen:

- Sistema de proximidad: "Permite la utilización de tarjetas u otros objetos que al acercarlos al terminal inicia la autenticación. En este tipo de control de acceso tenemos que destacar la innovadora tecnología RFID, que además de ofrecer una alta seguridad, es precisa, fiable y cuenta con una gran capacidad de almacenamiento de datos" (32).
- Sistemas biométricos: "Se basan en el reconocimiento de una característica física de la persona que solicita el acceso para que pueda verificarse de forma automática e instantánea. El más utilizado en las empresas es el lector de huella digital, que cuenta con numerosas ventajas como evitar la suplantación de identidad, acabar con los problemas de olvido de tarjetas, además de ser un sistema sencillo y eficaz" (32).
- Sistemas de reconocimiento de matrícula o TAG: "Controlan el acceso mediante la identificación de la persona, del vehículo o la combinación de ambas. Los vehículos pueden identificarse por tarjeta/TAG o por lectura de matrícula" (32).

2.3. Definición de términos básicos

- **Red neuronal artificial:** Es la representación de un modelo matemático que está desarrollado en base a las neuronas del cerebro humano, cuyo objetivo es imitar el razonamiento y aprendizaje de un ser humano (27).

- **Red neuronal siamés:** Es una red neuronal artificial que usa la misma configuración en dos ramas que permiten calcular el grado de similitud entre los datos de entrada gracias al cálculo de los vectores de salida comparables (27).
- **Reconocimiento facial:** Esta tecnología consiste en una manera de verificar, identificar o confirmar la identidad de una persona mediante sus características faciales (28).
- **Control de acceso:** El control de acceso es o son los mecanismos que permiten o restringen el acceso de una persona a los recursos o instalaciones de una organización o grupo de trabajo. Puede ser, que, dentro de una misma organización se pueda autorizar o denegar el ingreso a una determinada zona a determinados empleados (29).
- **Verificación de vida:** Es la medida aplicada sobre el reconocimiento facial con el objetivo de diferenciar entre una persona o una foto a través de características como el movimiento, tamaño o parpadeos (30).
- **Detección facial:** Es una manera de identificar en una imagen, video o en tiempo real uno o varios rostros humanos. La cuál, puede ser aplicada sobre medidas biométricas como el reconocimiento facial (31).
- **OpenCV:** Es una librería software open-source de machine learning que provee una infraestructura para aplicaciones de visión artificial. La librería consiste en más de 2500 algoritmos que permiten identificar objetos, caras, clasificar acciones humanas en vídeo o imágenes. Además, de hacer seguimiento de movimientos de objetos, extraer modelos 3D, etc (32).
- **Arduino:** Arduino es una plataforma de código abierto que sirve para el desarrollo electrónico. El cual, consiste tanto en hardware como en software libre, flexible y fácil de utilizar para desarrollar aplicaciones y sistemas completos (32).
- **Scrum:** Es una técnica de la metodología ágil que ofrece una forma de organizar y gestionar el trabajo de un grupo para ofrecer soluciones complejas. (33)

- **Función de pérdida de triplete:** Probablemente una de las mejores funciones para reconocimiento facial. Permite organizar las imágenes en tres carpetas: anclaje, positivas y negativas que posibilitan un entrenamiento a nivel de diferencias entre la comparación de píxeles a nivel de imágenes en anclaje positivas y anclaje-negativas.
- **Training loss:** Es una de las métricas durante el entrenamiento que es utilizada para evaluar como un modelo de deep learning se ajusta a los datos de entrenamiento. Es decir, calcula el error del modelo en su partición de entrenamiento.
- **Validation loss:** Es una métrica que permite evaluar el rendimiento de un modelo de deep learning. Además, permite prevenir el sobreajuste del modelo a través de la comparación de sus resultados con la partición de entrenamiento.
- **Descenso de gradiente:** Es un algoritmo de optimización que se caracteriza por ser iterativo. Su funcionamiento está en base a calcular el mínimo/máximo a través de una derivada que permita usualmente obtener el error mínimo del modelo de aprendizaje profundo.
- **Tensorflow:** Es el framework de código abierto basado en el lenguaje python que ofrece una gran cantidad de herramientas, librerías y recursos para la creación de Redes Neuronales Artificiales. (29)
- **Aplicativo Desktop:** Son los programas de computador, que pueden ser ejecutados sin la dependencia de internet o alguna tecnología externa (35).

CAPÍTULO III. METODOLOGÍA

3.1. Método y alcance de la investigación

3.1.1. Método de la investigación

El presente trabajo de investigación se desarrolla en base al marco de trabajo Scrum, teniendo en cuenta su manera ágil de gestionar las mejores prácticas para trabajar en grupo y obtener el mejor resultado de un proyecto en un plazo establecido. “En Scrum se realizan entregar parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Así, está especialmente aplicado para proyectos complejos, donde se necesita obtener un resultado pronto y la productividad son fundamentales. Su eficiencia e impacto está respaldada por su aplicación en gran magnitud de proyectos y su antigüedad” (33). Se fundamenta la aplicación de Scrum en el proyecto, por las siguientes razones:

- El sistema dinámico de la metodología Scrum permite desarrollar un esqueleto básico pero enfocado en las entregas funcionales. Así, es posible tener un producto funcional en un tiempo establecido que se optimizará durante el desarrollo del proyecto.
- Las entregas frecuentes de los requerimientos permitirán que la empresa y sus encargados puedan sugerir modificaciones. Así, dispongan de una funcionalidad correcta al terminar el proyecto.
- En base a su aplicación, se pueden prever diferentes cambios durante su desarrollo e implementación que podrán gestionarse. Lo cual, incrementa la retroalimentación y revisión del producto que reduce la posibilidad de la existencia de fallas en su versión final.

“El marco de trabajo Scrum se desarrolla en base a una secuencia de fases que garantiza la calidad de la versión final del producto teniendo en cuenta las correcciones y fallos durante su ejecución” (33) como se muestra en la Figura 22.

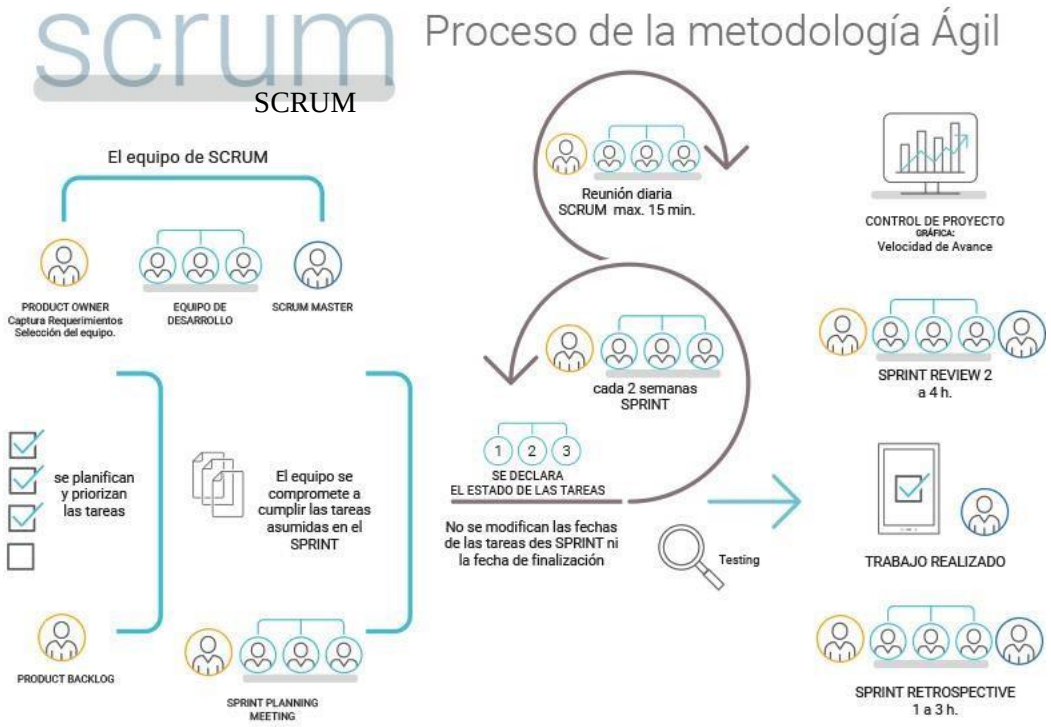


Figura 22. Ciclo de la metodología Scrum

Fuente: (33)

3.1.2. Personas y roles del proyecto

A continuación, en la Tabla 2, se establece el listado de los involucrados en el proyecto.

Tabla 2. Listado de involucrados

Persona	Rol
Wilmer Rojas	Product Owner
Jovanny Canchucaja	Scrum Master
Andrei Saavedra	Equipo

3.1.3. Fase de desarrollo y entregables

Como parte del uso de Scrum se detalla enseguida las Fases de desarrollo y entregables contempladas para este proyecto, además se especifica las historias de usuario asignadas a cada sprint, véase la Tabla 3, 4 y 5.

Tabla 3. Fases de desarrollo SCRUM

Fases de desarrollo Scrum	
Iniciación	
Project charter	Elaboración del documento
	Verificación del documento
	Entrega del documento
	Aprobación del documento
Planificación	
Plan de dirección del proyecto	Elaboración de la pila del producto
	Validación de pila del producto
Ejecución	
Historias de usuario	Identificación de historias de usuario
	Descripción de historias de usuario
	Priorización de historias de usuario
	Verificación de historias de usuario
Procesos funcionales	Identificación de procesos
	Diseño de procesos funcionales
	Verificación de procesos

Tabla 4. Distribución de requerimientos por sprint

Sprint	Historia de Usuario	Objetivo	Estado
Sprint 01	Como usuario final, requiero activar y desactivar el reconocimiento facial automático	Desarrollo de la funcionalidad del reconocimiento facial	Completado
Sprint 02	Como usuario final, requiero que el sistema de reconocimiento facial pueda hacer una detección de vida	Desarrollo de la funcionalidad verificación de vida	Completado
	Como usuario final, requiero que el sistema de reconocimiento facial sea automático	Automatización del proceso de reconocimiento facial	Completado
Sprint 03	Como usuario final, requiero registrar a un empleado al sistema de reconocimiento facial	Desarrollo de la funcionalidad de registro de empleados	Completado
	Como usuario final, requiero eliminar a un empleado del sistema de reconocimiento facial	Desarrollo de la funcionalidad de eliminación de empleados	Completado
	Como usuario final, requiero visualizar la lista actual de los empleados registrados	Desarrollo de la interface que permite la visualización de la lista actual de empleados registrados	Completado

Tabla 5. Seguimiento y control del proyecto

Seguimiento y control del proyecto	
Sprint 1	Reuniones con los miembros del área de seguridad de FUDEC Perú
Sprint 2	
Sprint 3	
Cierre – Memoria Final	
Memoria final	Elaboración de documentos
	Verificación de documentos
	Carta de aceptación del proyecto

3.1.4. Alcance de la investigación

La presente investigación se limitó al área gerencial, proceso de control de acceso a la oficina principal a través de un sistema de reconocimiento facial que capturará las imágenes a través de una cámara de seguridad que una vez que identifique correctamente al personal autorizado enviará una señal a un cerrojo de selenoide de Arduino que brindará el acceso a las instalaciones de la organización.

CAPÍTULO IV. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

4.1. Mapa de proceso del control de acceso

En la Figura 23, se muestra el proceso de control de acceso actual que se ejerce en las instalaciones de FUDEC Perú por parte del personal de seguridad.

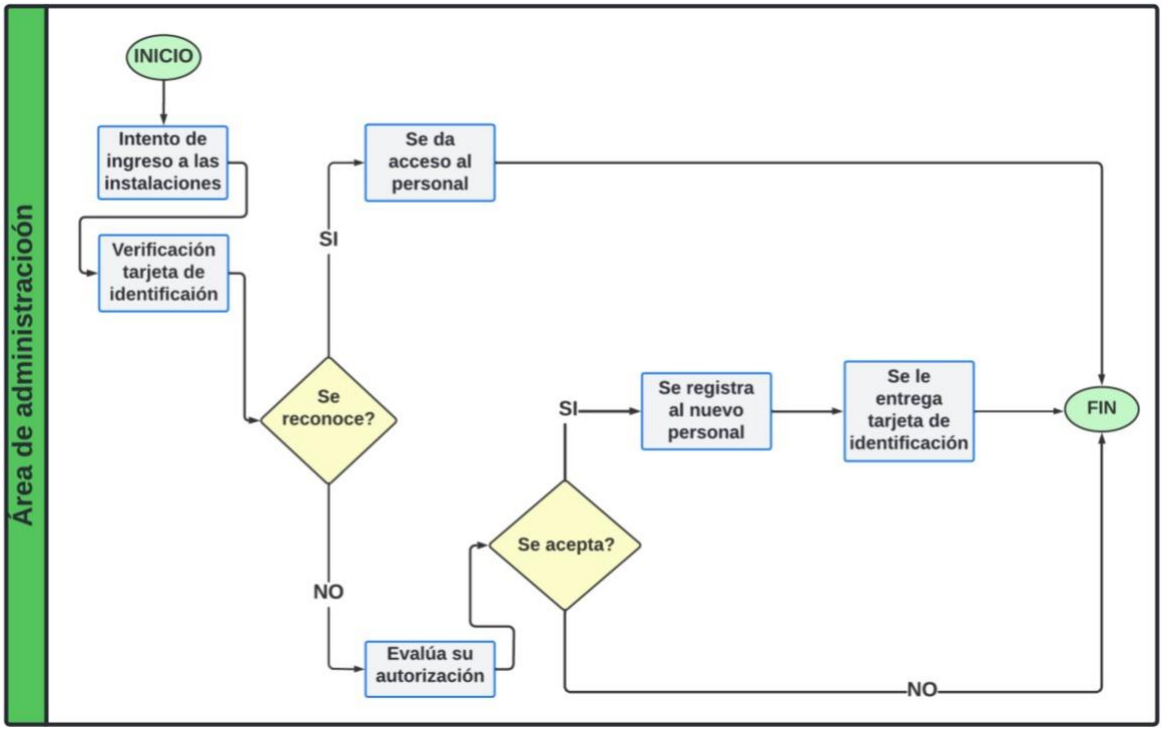


Figura 23. Proceso de control de acceso

4.2. Identificación de requerimientos funcionales

Para el análisis de la solución, se estableció como proyecto final un aplicativo de reconocimiento facial que controle el acceso a la oficina principal de la organización. Por ende, la aplicación deberá ser desarrollada de forma local es decir en una versión desktop que solo pueda ser accedida por las computadoras autorizadas de la organización.

En la Tabla 6, se plasmaron los requerimientos funcionales que se obtuvieron a partir de las técnicas de observación, encuestas y entrevistas.

Tabla 6. Requerimientos funcionales

ID	Enunciado de la historia	Alias	Estado	Esfuerzo	Iteración	Prioridad
HU-001	Como usuario final, requiero registrar un empleado al sistema de reconocimiento facial	Registrar Empleado	Planificado	Alto	2	Alta
HU-002	Como usuario final, requiero eliminar a un empleado del sistema de reconocimiento facial	EliminarEmpleado	Planificado	Medio	2	Alta
HU-003	Como usuario final, requiero activar y desactivar el reconocimiento facial cuando sea necesario	ControlReconocimiento	Planificado	Alto	1	Alta
HU-004	Como usuario final, requiero visualizar la lista actual del personal registrado	ListaPersonal	Planificado	Bajo	3	Media
HU-005	Como usuario final, requiero que el sistema de reconocimiento facial pueda hacer una detección de vida	Detección Vida	Planificado	Alto	3	Alta
HU-006	Como usuario final, requiero que el sistema de reconocimiento facial sea automático	Reconocimiento Automático	Planificado	Alto	1	

4.3. Especificación de requerimientos funcionales

Las especificaciones concernientes a cada uno de los requerimientos funcionales, es decir el contexto, eventos y resultados esperados son detalladas en la Tabla 7.

Tabla 7. Especificación de requerimientos funcionales

ID	Rol	Características	Razón	Criterio de Aceptación	Evento	Resultado
HU-RF001	Como usuario final	Requiero registrar a un nuevo empleado al sistema	Con la finalidad de brindarle la autorización	Brindar funcionalidad de registrar a un empleado	Cuando ingrese a las instalaciones de la organización	El sistema lo reconocerá como una persona con autorización para ingresar
HU-RF002	Como usuario final	Requiero eliminar a una persona del sistema	Con la finalidad de que no tenga autorización en el sistema para ingresar a las instalaciones de la organización	Brindar funcionalidad de eliminar a un empleado	Cuando el empleado renuncie o se decida quitarle la autorización	El sistema de reconocimiento facial no lo brindará acceso.
HU-RF003	Como usuario final	Requiero activar y desactivar el reconocimiento facial automático	Con la finalidad de gestionar el sistema de acuerdo con los requerimientos actuales	Brindar un botón de botón de encendido y apagado del sistema de reconocimiento facial	Cuando no se quiera mantener un control de acceso	El sistema solo controlará el acceso cuando este activado el sistema
HU-RF004	Como usuario final	Requiero visualizar la lista actual de todas las personas registradas en el sistema	Con la finalidad de saber que personas están registradas	Brindar una lista de los empleados registrados	Cuando se quiera informar de los empleados con autorización	El sistema brindará una lista de los empleados registrados actualmente
HU-RF005	Como usuario final	Requiero que el reconocimiento facial pueda hacer una detección de	Con la finalidad de prevenir el fraude o un acceso no	Brindar la funcionalidad de poder entrenar al	Cuando no se quiera permitir el ingreso con alguna de las	El sistema brindará la funcionalidad conteo de parpadeos para

		vida	autorizado	modelo	fotos del personal registrado	la detección de vida
HU-RF006	Como usuario final	Requiero que el reconocimiento facial sea automático	Con la finalidad de tener un control de acceso continuo	Brindar la funcionalidad de automatizar el proceso de reconocimiento	Cuando se quiera ejecutar el sistema sin alguna ayuda de un tercero	El sistema brindará la funcionalidad mediante la detección facial

4.4. Identificación de requerimientos no funcionales

Los requerimientos no funcionales se establecieron por requisito de los interesados envueltos en el proyecto. Estos describen características del proyecto que demuestren los indicadores de calidad como lo establece el estándar de calidad ISO 12207 en el proceso de análisis de requisitos del sistema. Estos se detallan en la Tabla 8.

Tabla 8. Requerimientos no funcionales

ID	Enunciado de la historia	Atributo
HU-RNF-001	El sistema de reconocimiento facial debe estar disponible las 24 horas del día y los 365 días del año, sujeto a la funcionalidad de la cámara de videovigilancia	Disponibilidad
HU-RNF-002	El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 2 horas.	Usabilidad
HU-RNF-003	La interfaz de usuario será implementada para los sistemas operativos Windows, MacOS y Linux.	Compatibilidad
HU-RNF-004	El sistema mostrará todas sus interfaces en el idioma español	Usabilidad

4.5. Conformación del equipo de trabajo

En el equipo de trabajo del proyecto participa el miembro representante de FUDEC Perú para validar y dar la conformidad de la finalización del proyecto. Asimismo, como Scrum Manager el encargado del área de TI. En la Tabla 9 se muestran los elementos participantes.

Tabla 9. Equipo de trabajo

Rol	Persona	Área
Product Owner	Wilmer Rojas	Gerencial
Scrum Manager	Jovanny Canchuchaja	TI
Development Team	Andrei Saavedra	TI

4.6. Análisis morfológico

En la HU-RF-001 se muestra la pantalla de registro de empleado, en ella se muestra las indicaciones para el registro, el campo de texto para el nombre y finalmente el botón de registro. Véase la Tabla 10.

Tabla 10. HU-RF-001 - RegistrarEmpleado

Código	Nombre Historia	Módulo	Criterio Aceptación
HU-RF-001	RegistrarEmpleado	Gestión del aplicativo	Brindar funcionalidad de registrar a un empleado
			El usuario deberá ingresar el primer nombre y apellido del empleado a registrar para posteriormente presionar el botón registrar fotos.

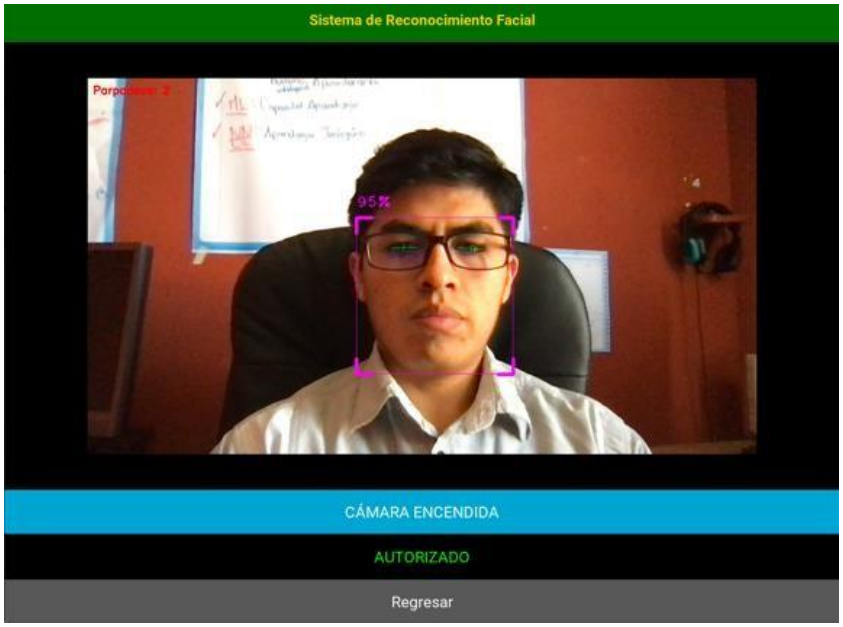
En la HU-RF-002 se muestra la pantalla de eliminación de empleado, en ella se muestra las indicaciones para una correcta eliminación, el campo de texto para el nombre y finalmente el botón de eliminar empleado. Véase la Tabla 11.

Tabla 11. HU-RF-002 - EliminarEmpleado

Código	Nombre Historia	Módulo	Criterio Aceptación
HU-RF-002	EliminarEmpleado	Gestión del aplicativo	Brindar funcionalidad de eliminar a un empleado
			El usuario deberá ingresar el nombre con el cual se registró el empleado para luego presionar el botón eliminar fotos del empleado

En la HU-RF-003 se muestra la pantalla de reconocimiento facial, en ella se muestra el botón para iniciar el reconocimiento, la detección y verificación facial y el texto del resultado del reconocimiento si autorizado o no autorizado. Véase la Tabla 12.

Tabla 12. HU-RF-003 - ControlReconocimiento

Código	Nombre Historia	Módulo	Criterio Aceptación
HU-RF-003	ControlReconocimiento	Reconocimiento Facial	Brindar un boton de botón de encendido y apagado del sistema de reconocimiento facial
			El usuario deberá presionar el botón de cámara encendida si quiere iniciar el reconocimiento y si quiere apagarla tendrá que volver a presionar en el mismo botón

En la HU-RF-004 mostrada en la Tabla 13, son mostrados todos los empleados que se encuentran actualmente registrados en el sistema de reconocimiento facial, con sus respectivos nombres y apellidos.

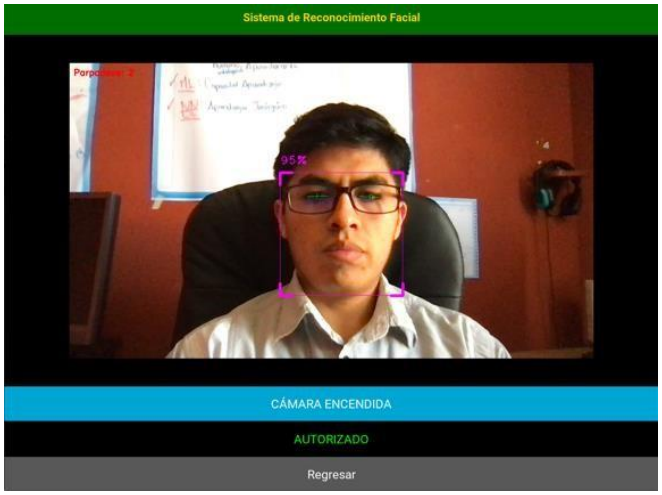
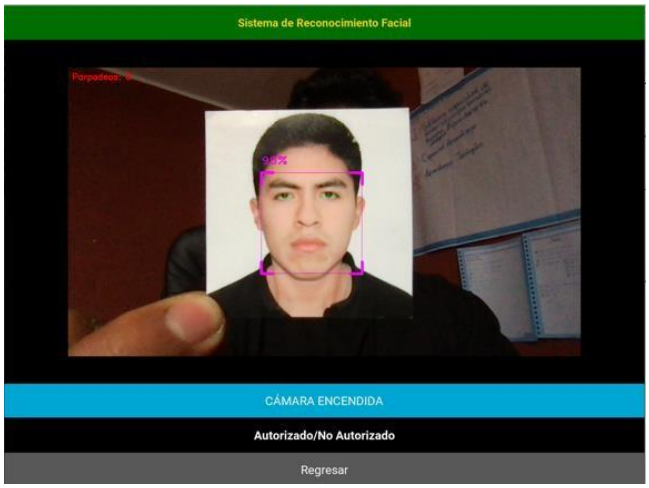
Tabla 13. HU-RF-004 - ListaPersonal

<table>
<tr><th>Código</th><th>Nombre Historia</th><th>Módulo</th><th>Criterio Aceptación</th></tr>
<tr><td>HU-RF-004</td><td>ListaPersonal</td><td>Gestión del aplicativo</td><td>Brindar una lista de los empleados registrados</td></tr>
<tr><td colspan="3"> 
</td><td>El usuario deberá presionar el botón lista del personal para obtener la lista de registrados</td></tr>
</table>

En la HU-RF-005 se muestra la pantalla de reconocimiento facial, en ella se muestra el flujo de video de la cámara de vigilancia donde se está realizando la verificación de vida a través del número de parpadeos del rostro facial detectado.

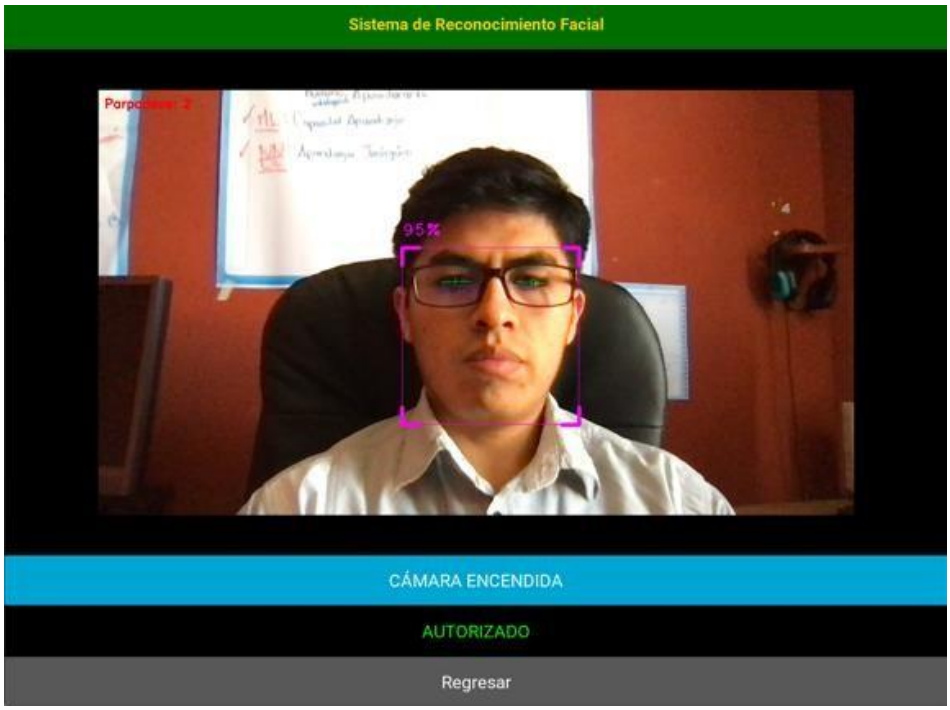
Véase la Tabla 14.

Tabla 14. HU-RF-005 - DetecciónVida

Código	Nombre Historia	Módulo	Criterio Aceptación
HU-RF-005	DetecciónVida	Reconocimiento facial	Brindar la funcionalidad la detección de vida en el flujo de video
 			El usuario deberá iniciar el proceso de reconocimiento facial. Así, la detección de vida se realizará automáticamente cada vez que detecte un rostro

En la HU-RF-006 se muestra la pantalla de reconocimiento facial, en ella se muestra el flujo de video de la cámara de vigilancia donde se está realizando la detección facial y se muestra el nivel de detección en un porcentaje sobre la parte superior derecha del cuadro de detección. Véase la Tabla 15.

Tabla 15. HU-RF-006 - ReconocimientoAutomático

Código	**Nombre Historia**	**Módulo**	**Criterio Aceptación**
HU-RF-006	ReconocimientoAutomático	Reconocimiento facial	Brindar la funcionalidad de realizar el reconocimiento facial automáticamente
			El usuario deberá iniciar el proceso de reconocimiento facial. Una vez encendido, el proceso será automático cada vez que se detecte un rostro.

4.7. Diagrama del proceso de reconocimiento facial

En base a la implementación del sistema de reconocimiento facial se desarrolla un diagrama del proceso en el cuál se plasman las diferentes etapas que forman parte del proceso de control de acceso. El proceso solo se realiza dentro del área de seguridad ya que el manejo del aplicativo y el control del acceso solo se desarrollan dentro de esta área. El proceso inicia con un intento de ingreso a las instalaciones, así se inician una seria de verificaciones para prevenir el fraude. Posteriormente, se evalúa el reconocimiento facial y su registro para brindar acceso o no a un empleado de la organización. Véase la Figura 24.

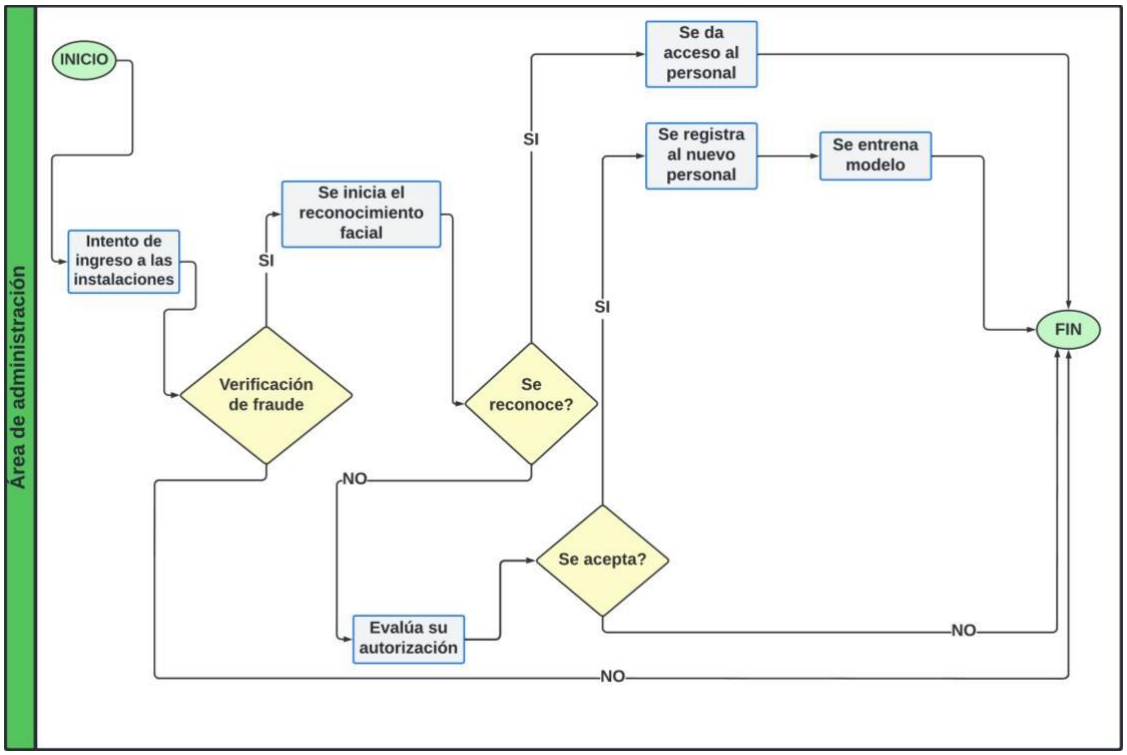


Figura 24. Proceso del sistema de reconocimiento facial y control de acceso

4.8. Análisis de solución

4.8.1. Arquitectura de solución física

La arquitectura de la solución fue desarrollada en base a los equipos y dispositivos en posición de la organización. Se muestra en la Figura 25, una computadora de escritorio con un sistema operativo

Windows 7 que poseerá el software de reconocimiento facial. El cual, esta conectado a una cámara de videovigilancia de la marca FDT del modelo FD7901W. Las cuales, al momento de ejecutar un reconocimiento facial exitoso se enviará una señal a la placa Arduino Uno, que a su vez esta conectado a un cerrojo de selenoide, a través del puerto serial COM3. Teniendo en cuenta esta información, se diseñó la arquitectura en base a los siguientes puntos:

- El aplicativo será ejecutado sobre un lenguaje de programación Python 3.6.4 y sobre el framework Kivy 2.0.
- El aplicativo podrá ser ejecutado en los sistemas operativos Windows, Linux y MacOS.
- La red neuronal siamés y los modelos de clasificación serán ejecutados sobre el framework Tensorflow 2.0.
- La conexión del aplicativo y la placa Arduino será ejecutada sobre la versión Arduino 1.8.19.

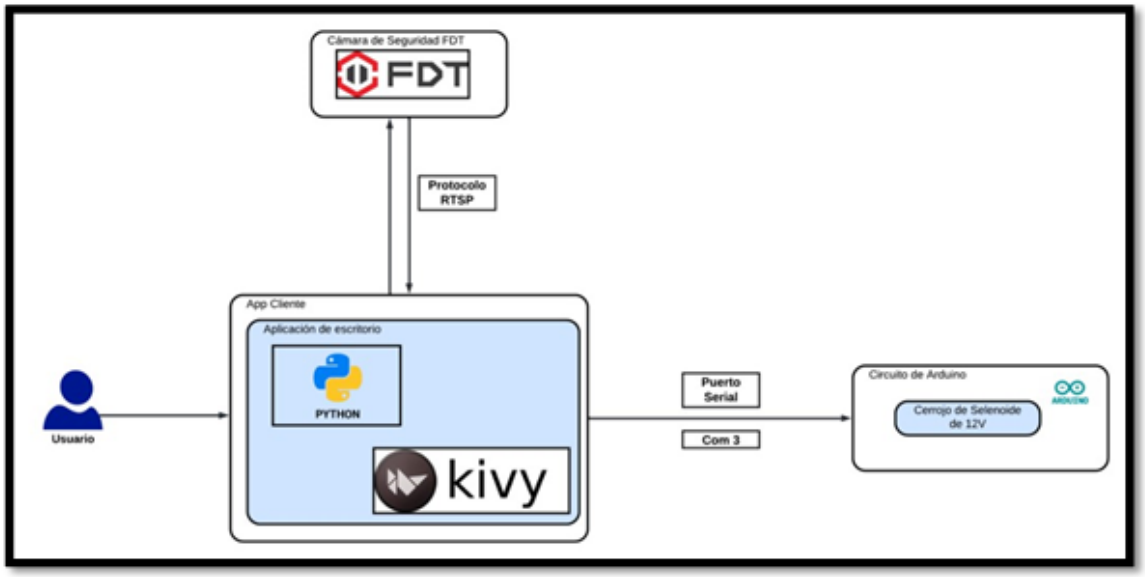


Figura 25. Arquitectura de hardware

4.8.2. Arquitectura de solución lógica

La arquitectura de la solución está conformada por el aplicativo de escritorio que agrupa a los modelos de clasificación, el reconocimiento facial y la conexión serial a través de Arduino. Véase la Figura 26.

El aplicativo de escritorio solo está compuesta por la capa vista, ya que en el presente trabajo de investigación no es necesaria el uso de una base de datos.

La capa vista hace referencia a todas las interfaces del aplicativo, las cuales controlan los modelos de clasificación, la red neuronal siamés y la conexión serial. La capa vista usa el caché temporal compartido para comunicarse con la red neuronal y los modelos de clasificación. De esta manera, los modelos y la red neuronal se mantienen inactivos hasta que la vista lo requiera.

La red neuronal siamés está conformada por dos redes neuronales convolucionales paralelas con los mismos parámetros que se encargan de realizar la comparación entre dos imágenes entrantes. De esta manera, a través de una función de similaridad se obtiene el porcentaje de similitud entre las dos imágenes.

Los modelos de clasificación se encargan de usar el modelo pre entrenado con la red neuronal siamés para obtener los valores de similitud. Posteriormente, se entrenan los modelos con los resultados obtenidos del modelo pre entrenado. Así, logrando un sistema de reconocimiento facial que pueda ser entrenado y reentrenado en caso de un requerimiento de brindar o quitar el acceso a un empleado de la organización.

La conexión serial está desarrollada en base a la conexión de puertos de Arduino. Al momento de recibir la señal de que el reconocimiento facial ha sido exitoso se activará la placa Arduino UNO con el objetivo de abrir el cerrojo de selenoide.

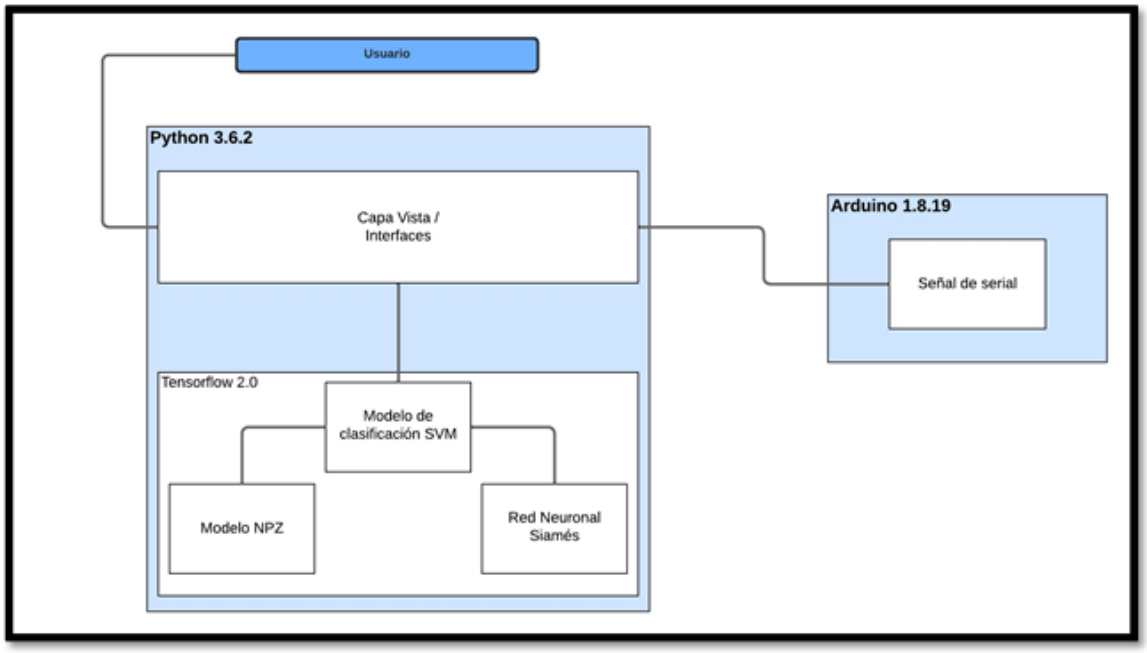


Figura 25. Arquitectura de la solución lógica

4.8.3. Análisis tecnológico

Se consideraron técnicas y herramientas para construir el producto.

4.8.3.1. Herramientas de hardware

Se enfoca en el hardware utilizado para el desarrollo del sistema de reconocimiento facial, véase la Tabla 16.

Tabla 16. Herramientas de hardware

Hardware	Especificación
PC	Memoria RAM: 12gb Disco duro: 1 tb Procesador: i5 9400 Sistema operativo: Windows 64 bits, x64
Cámara de videovigilancia	Resolución de 720 Visión nocturna Rotación

	Conexión inalámbrica y cableada
Arduino	Arduino UNO
Cerrojo de puerta de selenoide	Selenoide de 12V
Baterías	Baterías de 7200 mmap
Módulo de relay	Módulo de relay de 5v

4.8.3.2. Herramientas de software

Para el desarrollo del sistema de reconocimiento facial se usaron las siguientes herramientas de software que permitieron el buen desarrollo del proyecto. Véase la Tabla 17.

Tabla 17. Herramientas de software

Software	**Especificación**
Herramientas para codificar la solución	Arduino Visual Studio Code
Lenguaje del código de programación	Python 3.6
Framework de desarrollo	Kivy 2.0

4.9. Diseño de la solución

A continuación, se presentan las interfaces del software desarrollado. Las cuales, fueron validadas conjuntamente con el product owner y los stakeholders de la organización a través del proceso de UX/UI que se implementó durante el desarrollo de la metodología Scrum.

4.9.1. Interfaz de usuario de la página principal

El sistema de reconocimiento facial es un aplicativo de escritorio que solo estará instalado en las computadoras autorizadas por la organización. De esta manera, se controlará el acceso a la oficina principal a través de un reconocimiento facial. El cual, abrirá el cerrojo una vez que el reconocimiento sea exitoso. Además, el sistema diferencia entre una persona real y una foto, lo cual evita el fraude o un acceso no autorizado. Así, se busca tener un sistema de control de acceso automático y disponible cuando sea necesario.

El aplicativo inicia con una página principal en la cual se exige ingresar la contraseña proporcionada por el jefe de sistemas Jovanny Canchuchaja para mantener un ingreso al sistema por parte de las personas autorizadas a dicho proceso, considerando el escenario de más de un usuario del sistema. La interfaz está diseñada con los colores e imágenes de la organización. Véase la Figura 27.

Figura 27. Interfaz de bienvenida y login

4.9.2. Interfaces de usuarios de las páginas secundarias

Una vez que se ingrese en el sistema se muestra el menú principal que contiene la opción para iniciar el proceso de reconocimiento facial o para gestionar el aplicativo en sus aspectos de registros, listas, etc.

En el menú principal se puede apreciar las opciones: Reconocimiento facial, que esta encargado de activar/desactivar el proceso de reconocimiento cuando se desee. Esto, como plan de contingencia si en algún punto no se desea usar el control de acceso en ciertas situaciones extraordinarias y Gestionar

Aplicativo, que brinda funciones para que el sistema se pueda desarrollar de una forma óptima. Véase la Figura 28.

Figura 28. Interfaz del menú principal

Una vez que se ingrese al reconocimiento facial se visualiza la interfaz que permite activar o desactivar el reconocimiento automático y también se muestra un mensaje si la persona ha sido autorizada o no. Además, antes de realizar el reconocimiento, el aplicativo evalúa dos aspectos: la detección facial y la detección de vida. La detección facial es desarrollada con el objetivo de reconocer una cara en un flujo de video que nos permite evaluar el grado de la detección que nos permitirá realizar un reconocimiento una vez que la detección tenga un valor del 95% o mayor. Así se evita que cualquier factor externo del ambiente (luz, opacidad, ángulo) tenga un efecto negativo sobre la precisión del reconocimiento. Por otro lado, la detección de vida se da a través de un contador de parpadeos que evita el fraude o algún acceso no autorizado. Véase la Figura 29.

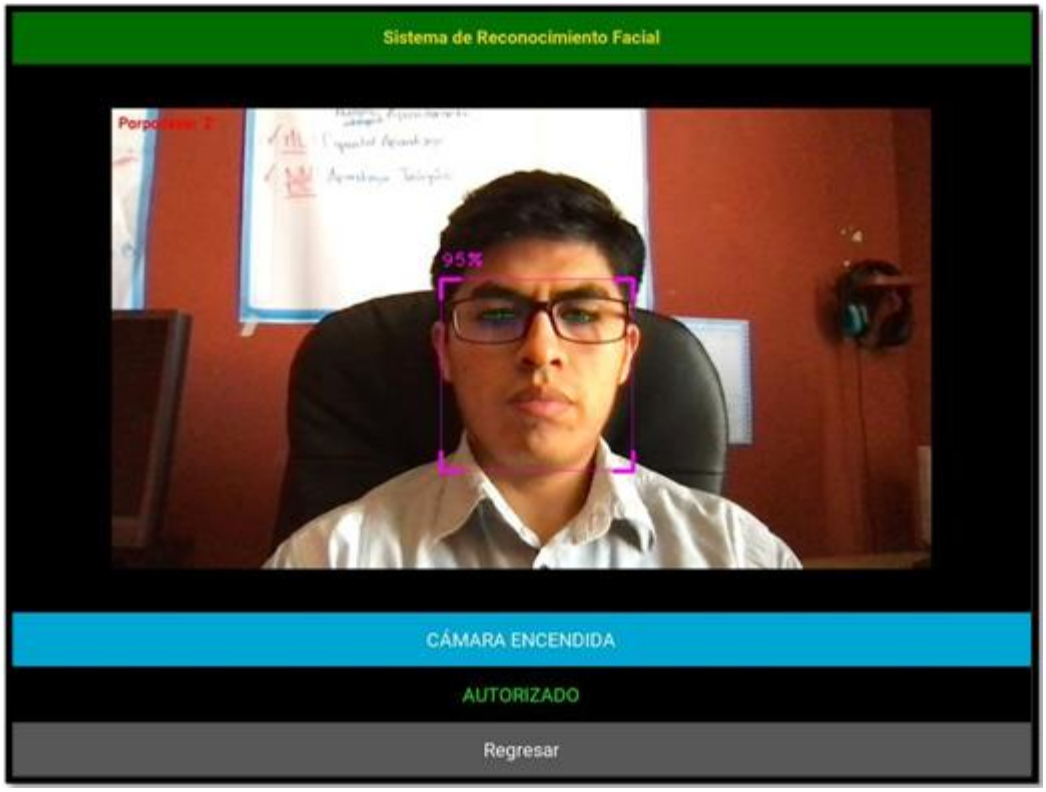


Figura 29. Interfaz de reconocimiento facial

Por otro lado, si se ingresa a la opción de gestión del aplicativo se muestra un menú secundario que contiene las funcionalidades de registrar personal, entrenar modelo, eliminar personal y lista del personal. Véase la Figura 30.

Figura 30. Interfaz del menú secundario

Así, dentro del menú secundario se puede ingresar al registro del personal. En la cual, se muestra una interfaz que permite al usuario registrar a una persona con su primer nombre y apellido a través de una serie de fotos que tomarán automáticamente. Véase la Figura 31.

Figura 31. Interfaz de registro del empleado

Dentro de la segunda opción de entrenar modelo, se puede entrenar el modelo, pero solo una vez que se haya realizado un registro o eliminación de un empleado del sistema. Así, se logrará actualizar el sistema reconocimiento facial solo con los empleados que se encuentren registrados. Véase la Figura 32.

Figura 32. Interfaz de entrenamiento

Dentro de la tercera opción de eliminar personal, se ingresa el nombre con el cual se registró al empleado y posteriormente presionar en el botón de eliminar personal. Así, consecuentemente se procede a entrenar el modelo para que los cambios tengan efecto sobre el sistema. Véase la Figura 33.

Figura 33. Interfaz de eliminación

Dentro de la cuarta opción de lista de personal, se visualiza a todos los empleados que se encuentran actualmente registrados en el sistema de reconocimiento facial. Véase la Figura 34.

Figura 34. Interfaz de empleados registrados

CAPÍTULO V. CONSTRUCCIÓN

5.1. Construcción de la red neuronal siamés

5.1.1. Modelo de la red neuronal

Para la creación de la red neuronal convolucional del tipo siamés se utiliza la mejor arquitectura para el reconocimiento facial según (34). La arquitectura inicia con la toma de imágenes de 105x105 píxeles como entradas (inputs). Después, se procede a la primera convolución que consiste en la aplicación de 64 filtros de kernel, que como resultado se obtiene 64 muestras matriciales de 96x96 píxeles. Así, se vuelve a aplicar 64 filtros de kernel con un resultado de 48x48 píxeles. Luego, se aplican, cuatro veces, 128 filtros de kernel que nos resulta en un 9x9 píxeles. Finalmente, la última convolución se realiza con 256 filtros para obtener una dimensión de 6x6 píxeles. En la arquitectura no se muestra la cola siamesa para evitar la redundancia, pero es justo en esta etapa donde la arquitectura siamesa se une en la capa de vectores de 4096 unidades calculando la distancia o similaridad entre las dos imágenes de entrada. Véase la Figura 35.

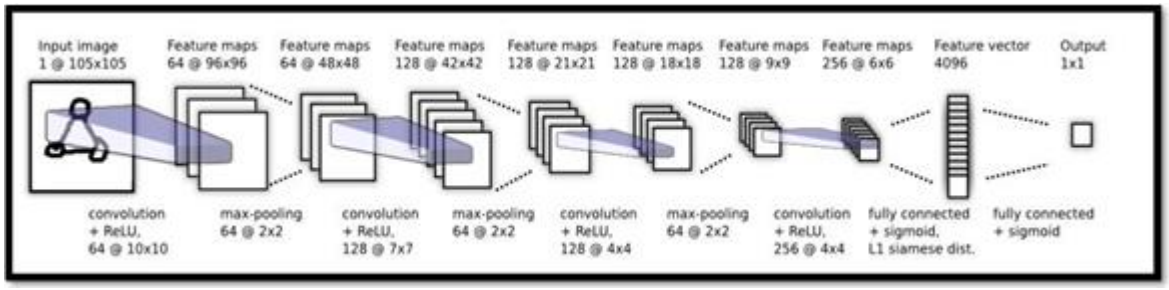


Figura 35. Arquitectura de una red neuronal siames

Fuente: (34)

Para realizar el entrenamiento de la red neuronal convolucional del tipo siamés se utiliza el dataset "Rostros etiquetados en la naturaleza (LFW)" el cual se desarrolló con el objetivo de aplicarlo sobre proyectos de reconocimiento facial. El dataset contiene más de 13000 imágenes de caras obtenidas desde la internet que están etiquetadas con el nombre de la persona. Existen de 3 a más fotos por

persona. Las imágenes recolectadas fueron analizadas y preprocesadas por el algoritmo de detección facial Viola-Jones. Véase la Figura 36.

Figura 36. Dataset LFW

5.1.2. Obtención de datos

Una vez terminado el desarrollo de la red neuronal siamés, se procedió a organizar y clasificar las imágenes que permitieron el entrenamiento de la red neuronal.

Según (35) la red neuronal siamés al ser un tipo de red neuronal convolucional diferente y única no se entrena de la misma forma que se entrena una red neuronal. Su entrenamiento se caracteriza por utilizar la función de pérdida de triplete. La cuál, consiste entrenar el modelo a través de la comparación de tres categorías: las imágenes positivas, negativas y de anclaje. Esta organización permitirá que el modelo aprenda que cuando una imagen positiva se compare con una de anclaje el resultado será positivo. Mientras que, cuando se compare una imagen negativa con la de anclaje será negativo. De esta manera, el modelo aprenderá que cuando el resultado de la función de similaridad o L1 es mínima las dos imágenes pertenecerán a la misma persona mientras que si la distancia es mayor no pertenecerán a la misma persona. Véase la Figura 37.

Figura 37. Carpetas de entrenamiento

Para la carpeta de imágenes positivas y de anclaje se consideraron 200 imágenes de la misma persona con el fin explícito de que durante el entrenamiento el modelo aprenda a diferenciar entre dos imágenes que pertenecen a la misma persona y cuáles no. Véase la Figura 38 y 39.

Figura 38. Carpeta de anclaje

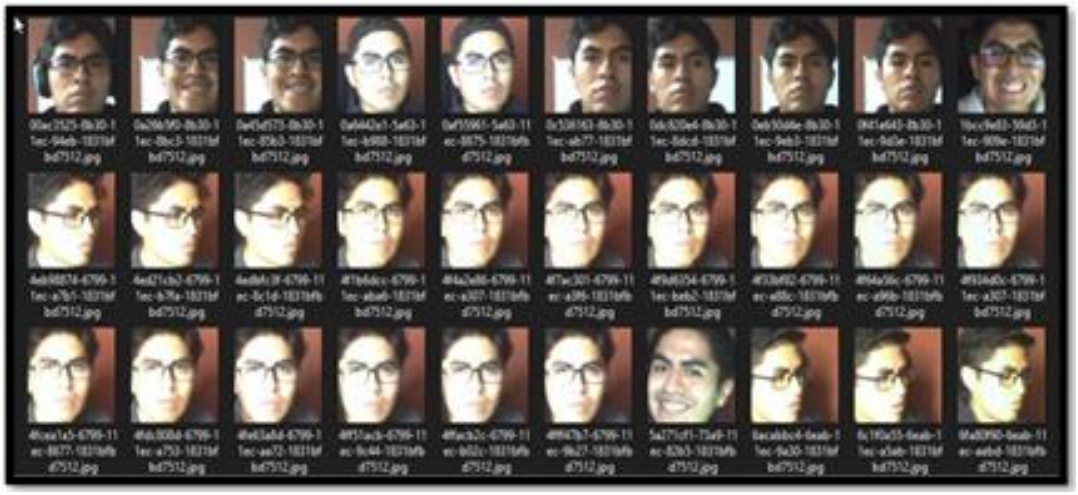


Figura 39. Carpeta de imágenes positivas

Y para la carpeta de imágenes negativas se insertaron las más 13000 del dataset LFW que permitirá un entrenamiento profundo gracias a la gran cantidad de datos de entrenamiento. Véase la Figura 40.

Figura 40. Carpeta de imágenes negativas

5.1.3. Construcción del modelo

Para el desarrollo del modelo fue necesario definir el dataset y sus respectivas etiquetas que se usaron para el entrenamiento del modelo. En una primera instancia se pasa a definir las variables que tendrán todas las imágenes de la función de pérdida de triplete. Véase la Figura 41.

```
anchor = tf.data.Dataset.list_files(ANC_PATH+'\*.jpg')
positive = tf.data.Dataset.list_files(POS_PATH+'\*.jpg')
negative = tf.data.Dataset.list_files(NEG_PATH+'\*.jpg')
```

Figura 41. Configuración de la función de pérdida de triplete

Posteriormente se realiza la unión de las imágenes de anclaje con las positivas, así como las imágenes negativas con las de anclaje. Así, definiendo que el primer grupo son los resultados positivos y el segundo grupo representan los resultados negativos. Véase la Figura 42.

```
positives = tf.data.Dataset.zip((anchor, positive, tf.data.Dataset.from_tensor_slices(tf.ones(len(anchor)))))
negatives = tf.data.Dataset.zip((anchor, negative, tf.data.Dataset.from_tensor_slices(tf.zeros(len(anchor)))))
data = positives.concatenate(negatives)
```

Figura 42. Creación de carpetas para la función de triplete

Por otro lado, para poder obtener esas imágenes de una dimensión de 105x105 píxeles que son necesarias para la arquitectura siamés fue necesario crear una función de preprocesamiento de cada una de las imágenes en nuestro dataset. Véase la Figura 43.

```python
def preprocess(file_path):

    # Lee la imagen desde la ruta
    byte_img = tf.io.read_file(file_path)
    # Carga la imagen
    img = tf.io.decode_jpeg(byte_img)

    # Modifica el tamaño a 105x105
    img = tf.image.resize(img, (105,105))
    # Escala la imagen en una valor de 0 a 1
    img = img / 255.0

    # Retorn la imagen
    return img
```

Figura 43. Función de preprocesamiento de imágenes

Finalmente, se divide el dataset en los grupos de entrenamiento, testeo y validación. De esta manera, se puede observan los resultados de nuestro modelo a través de gráficas que comparan los valores de los diferentes grupos. Además, se considera la división de validación que permite evitar el sobre ajuste de los valores del modelo. Véase la Figura 44.

```python
# Partición de entrenamiento
train_data = data.take(round(len(data)*.8))
train_data = train_data.batch(16)
train_data = train_data.prefetch(8)

# Partición de testeo
test_data = data.skip(round(len(data)*.8))
test_data = test_data.take(round(len(data)*.3))
test_data = test_data.batch(16)
test_data = test_data.prefetch(8)

# Partición de validación
val_data = test_data.skip(round(len(data)*.5))
val_data = val.take(round(len(data)*.3))
val_data = test_data.batch(16)
val_data = test_data.prefetch(8)
```

Figura 44. División del dataset en entrenamiento, testeo y validación

Una vez que se cuenta con el dataset, se pasa a construir la red neuronal siamés en base a la mejor arquitectura para el reconocimiento facial.

Véase la Figura 45.

```
inp = Input(shape=(100,100,3), name='input_image')

c1 = Conv2D(64, (10,10), activation='relu')(inp)

m1 = MaxPooling2D(64, (2,2), padding='same')(c1)

c2 = Conv2D(128, (7,7), activation='relu')(m1)
m2 = MaxPooling2D(64, (2,2), padding='same')(c2)

c3 = Conv2D(128, (4,4), activation='relu')(m2)
m3 = MaxPooling2D(64, (2,2), padding='same')(c3)

c4 = Conv2D(256, (4,4), activation='relu')(m3)
f1 = Flatten()(c4)
d1 = Dense(4096, activation='sigmoid')(f1)

mod = Model(inputs=[inp], outputs=[d1], name='embedding')
```

Figura 45. Creación de la red neuronal siamés

La cual, en una forma gráfica y conjunta tiene la siguiente estructura. Véase la Figura 46.

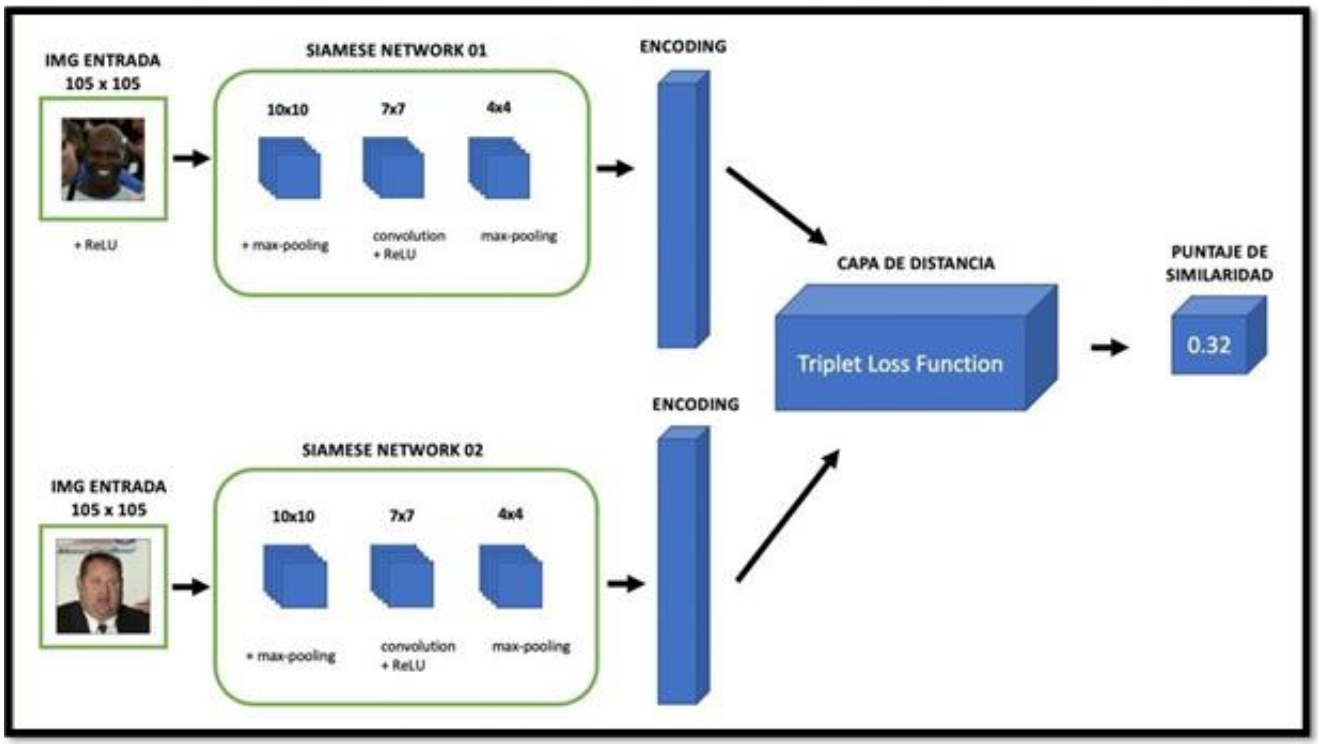


Figura 46. Estructura de una red neuronal

El entrenamiento de una red neuronal siamés es particular y único ya que no sigue el mismo comportamiento que una red neuronal convolucional común. En base, a su entrenamiento a través de una función de pérdida de triplete es necesario crear una función de entrenamiento personalizada que permita la arquitectura de la red neuronal siamés.

Primero, para realizar la creación de una función personalizada es necesario usar la la función reservada de Tensorflow "@tf.function". La cual, nos permite crear una función que define el dataset con el cual se entrenará el modelo. Además, se define el cálculo de la gradiente que le permitirá al modelo aprender y ajustas los parámetros de entrenamiento y retornando finalmente la pérdida del modelo. Véase la Figura 47.

```python
@tf.function
def train_step(batch):


    with tf.GradientTape() as tape:
        # Obtiene una imagen de anclaje con una positiva o negativa
        X = batch[:2]
        # Obtiene el etiquetado
        y = batch[2]


        yhat = siamese_model(X, training=True)
        # Se calcula la pérdida
        loss = binary_cross_loss(y, yhat)
    print(loss)

    # Calcula los gradientes
    grad = tape.gradient(loss, siamese_model.trainable_variables)

    # Calcula los parámetros actualizados
    opt.apply_gradients(zip(grad, siamese_model.trainable_variables))

    # Retorn la pérdida
    return loss
```

Figura 47. Configuración función entrenamiento personalizada

Una vez que se tiene listo la función que permite el entrenamiento a través de la función de pérdida de triplete, se pasa a definir la función de entrenamiento con sus respectivas configuraciones de las métricas, barras de progreso, el dataset tanto de entrenamiento como de validación y finalmente termina la función retornando todos los resultados obtenidos del entrenamiento. Véase la Figura 48.

```
def train(train,test, EPOCHS):

    metrics_names1 = ['loss_train', 'loss_test','acc','pr']
    metrics_names2 = ['loss_test']
    losses_train = []
    losses_test = []
    accs1 = []
    accs2 = []

    for epoch in range(1, EPOCHS+1):
        print("\n epoch {}/{}".format(epoch,EPOCHS))
        progbar1 = tf.keras.utils.Progbar(len(train), stateful_metrics=metrics_names1)
        progbar2 = tf.keras.utils.Progbar(len(test), stateful_metrics=metrics_names2)


        r1 = Recall()
        p1 = Precision()


        for idx, batch_train in enumerate(train):
            # Run train step here
            loss_train = train_step(batch_train)
            yhat = siamese_model.predict(batch_train[:2])
            r1.update_state(batch_train[2], yhat)
            p1.update_state(batch_train[2], yhat)
            values=[('loss_train',loss_train.numpy()),('acc',r1.result().numpy()), ('pr',p1.result().numpy())]
            progbar1.update(idx+1, values=values)


        r2 = Recall()

        for idx, batch_test in enumerate(test):

            loss_test = test_step(batch_test)
            yhat = siamese_model.predict(batch_test[:2])
            r2.update_state(batch_test[2], yhat)
            values=[('loss_test',loss_test.numpy()), ('acc_loss',r2.result().numpy())]
            progbar2.update(idx+1, values=values)


        losses_train.append(loss_train)
        losses_test.append(loss_test)
        accs1.append(r1.result())
        accs2.append(r2.result())


        if epoch % 10 == 0:
            checkpoint.save(file_prefix=checkpoint_prefix)

    return losses_train, losses_test, accs1, accs2
```

Figura 48. Función de entrenamiento

Así, se define la función de entrenamiento que recibe como parámetros el número de épocas y el dataset para entrenamiento y testeo. Además, se implementan los diferentes parámetros que nos servirán para evaluar el entrenamiento del modelo durante las 1000 épocas que se aplicarán al proceso.

El primer parámetro que se consideró fue el accuracy del entrenamiento el cuál dejaba de mejorar alrededor de la época 1100. Así, con el fin de garantizar un aprendizaje continuo y relevante se consideró las 1000 épocas. Véase la Figura 49.

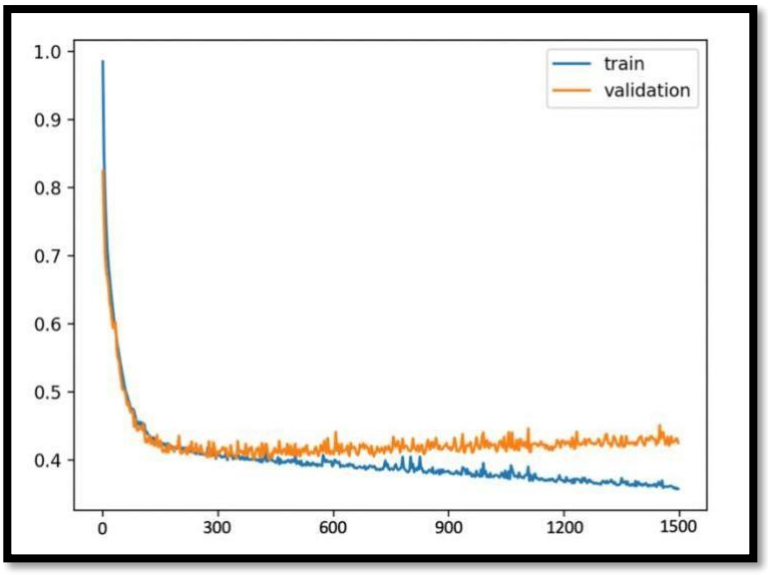


Figura 49. Comparación del error en la partición de entrenamiento y validación

Como segundo parámetro, en el artículo "A siamese long short-term memory architecture for human re-identification" y "Siamese neural network based gait recognition for human identification" se puede resaltar que durante el entrenamiento de la red neuronal, para una clasificación de imágenes y su aplicación sobre el reconocimiento de caracteres, llegaba al punto de sobre ajuste u overfitting alrededor de las épocas 769 y 972 respectivamente. Véase la Figura 50.

```
binary_cross_loss = tf.losses.BinaryCrossentropy()

opt = tf.keras.optimizers.Adam(1e-4) # 0.0001
```

Figura 50. Parámetros del entrenamiento

Para encontrar el número de épocas que se ajuste de manera correcta tanto al tamaño del dataset como a la función de pérdida de triplete se realizó un experimento entre los números 500, 1000, 2000 y 4000

tomando como referencia el antecedente (2) en el que se aplican 2000 épocas para un dataset de 70000 imágenes. Al realizar el experimento se evaluaron las primeras 10 épocas. De tal manera, que permita hacer un cálculo de la raíz cuadrada media para cada uno de los casos. Así, se alcanzó un puntaje menor durante las 1000 épocas que concluyeron en su selección para el entrenamiento. Véase la Figura 51.

```
EPOCHS = 1000
```

Figura 51. Número de épocas

Ya habiendo realizado las configuraciones requeridas, se inicia el entrenamiento del modelo a través del siguiente código que invoca a la función de entrenamiento con sus parámetros respectivos. Véase la Figura 52.

```
history = train(train_data, test_data, EPOCHS)
```

Figura 52. Sintaxis para el inicio del entrenamiento

Se inicia el proceso de entrenamiento en el cual se visualizan las métricas de precisión y pérdida haciendo una comparación para los datasets de entrenamiento y validación.

Según (36) , el Accuracy expresa el porcentaje de acierto obtenido por una

Red Neuronal Artificial durante su entrenamiento, tomando valores de 0 a 1, donde 1 significa una tasa del 100% de acierto y 0 indica una tasa del 0% de acierto. El Loss expresa el porcentaje de pérdida o porcentaje de error obtenido por una Red Neuronal Artificial, tomando valores de 0 a 1, donde 1 significa una tasa de error del 100% y 0 indica una tasa de error del 0%.

Y en cuanto a la pérdida del modelo, según (37), expresa el castigo por una mala predicción. Es decir, la pérdida es un número que indica qué tan mala fue la predicción del modelo en un solo ejemplo. Si la predicción del modelo es perfecta, la pérdida es cero; de lo contrario, la pérdida es mayor.

Durante el entrenamiento, la pérdida inicia con un valor de 0.1080 para el entrenamiento y 0.0725 para la validación. Por otro lado, la precisión del entrenamiento inicia con 0.9487 y la de testeo con 1.000. Véase la Figura 53.

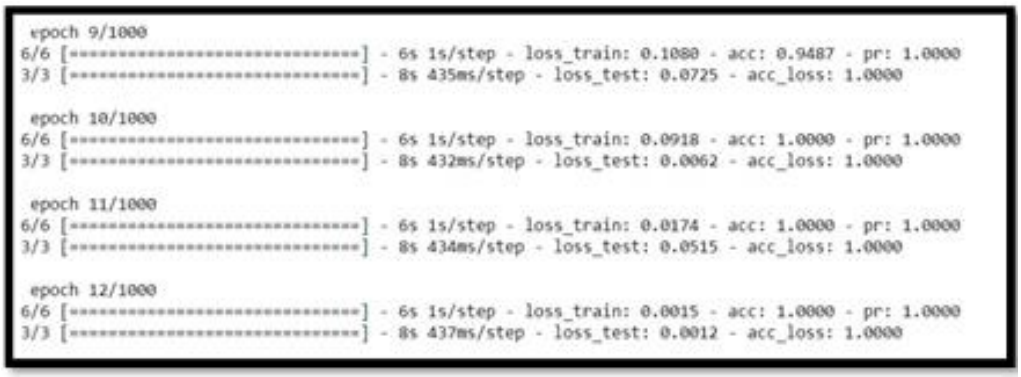


Figura 53. Entrenamiento, época 10

Según (38) el desarrollo del entrenamiento a través de las épocas las métricas de precisión aumentarán su valor gradualmente o se mantendrán en el valor de 1.000, que nos indica que la red ha logrado aprender a identificar las clases del dataset. Por otro lado, se espera que la pérdida tenga ciertas bajas y altas, pero se espera obtener el menor valor posible.

Véase la Figura 54.

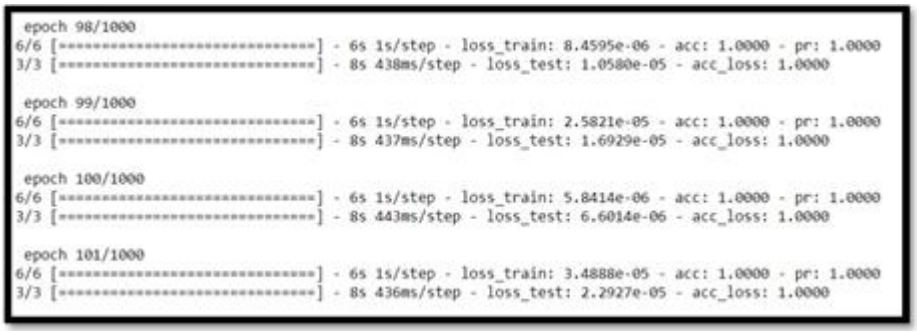


Figura 54. Entrenamiento, época 100

Al llegar a la época 400, el Accuracy se mantiene en 1, lo que significa que, por cada imagen analizada, acierta en cada una de ellas, sin embargo, el ajuste de pesos aún sigue cambiando. Véase la Figura 55.

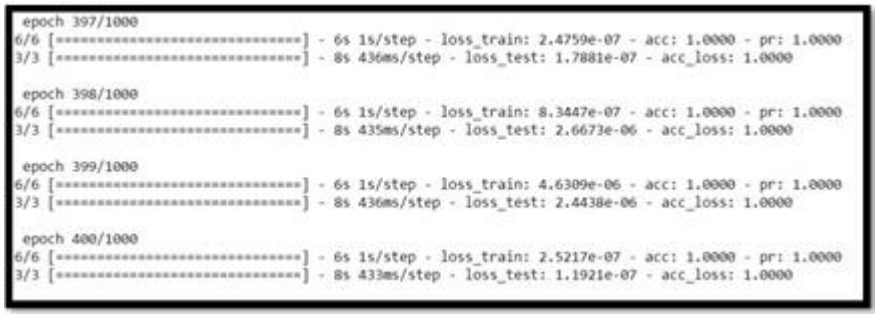


Figura 55. Entrenamiento, época 400

Al llegar a la época número 600, lo primero que se visualiza en que la precisión se sigue manteniendo en 1, lo cual significa que la precisión de la red es exacta al momento de reconocer las clases. Sin embargo, aún los pesos se siguen ajustando debido a su continuo valor cambiante.

Véase la Figura 56.

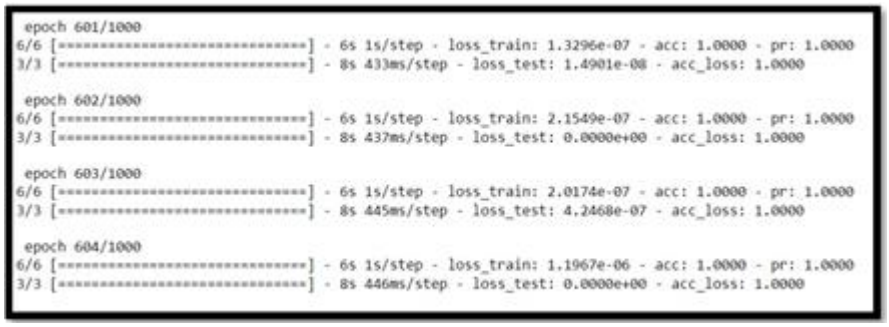


Figura 56. Entrenamiento, época 600

Durante la época número 800, se puede visualizar aún un valor de pérdida que cambia con cada época. Lo cual, aún sigue aprendiendo y modificando sus parámetros internos que permitirán reducir la pérdida del modelo hasta obtener el valor mínimo posible. Véase la Figura 57.

Figura 57. Entrenamiento, época 800

Finalmente, con el término del entrenamiento en la época número 1000, se puede visualizar que la pérdida mínima obtenida está en base a un 0.0041. Lo cual, demuestra que el error del modelo es mínimo. Mientras que, la precisión se mantuvo con un valor de 1. Así, obteniendo un proceso de entrenamiento de calidad para el modelo de la red neuronal siamés.

Véase la Figura 58.

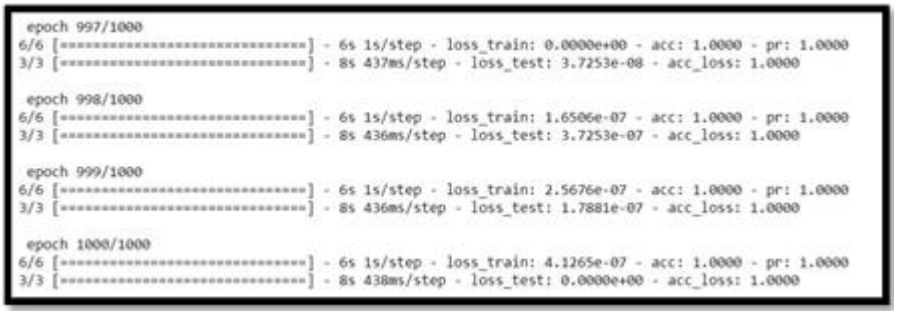


Figura 58. Entrenamiento, época 1000

Una vez que se obtuvo la red neuronal completamente entrenada se procedió a guardarla en el formato HDF5 que es usada para guardar modelos de aprendizaje automático en la librería Keras. La cual, permitió almacenar la arquitectura del modelo, pesos y la configuración de entrenamiento, todo en un solo archivo, de manera que posibilita una fácil implementación y carga del modelo. Véase la Figura 59.

```
# Guardar el modelo
siamese_model.save('siamese_network.h5')
```

Figura 59. Guardado del modelo en el formato HDF5

A continuación se muestra el código del software de escritorio que demuestra la carga del modelo y el envio de la imagen capturada, una vez que se haya realizado una detección facial correcta y un conteo de parpadeos correcto. Véase la Figura 60.

```
## Capturar imagen y enviarla a la red neuronal
SAVE_PATH = os.path.join('application_data', 'input_image', 'input_image.jpg')
ret, frame = capture.read()
cv2.imwrite(SAVE_PATH, frame)

in_encoder = Normalizer()
out_encoder = LabelEncoder()
out_encoder.classes_ = np.load('./models/classes.npy')
facenet_model = load_model('./models/siamese_network.h5')

with open('./models/trained_model.pkl', 'rb') as f:
    model = pickle.load(f)

random_face = extract_face(os.path.join('application_data', 'input_image', 'input_image.jpg'))
random_face_emd = in_encoder.transform([get_embedding(facenet_model, random_face)])[0]

samples = np.expand_dims(random_face_emd, axis = 0)
yhat_class = model.predict(samples)
yhat_prob = model.predict_proba(samples)
```

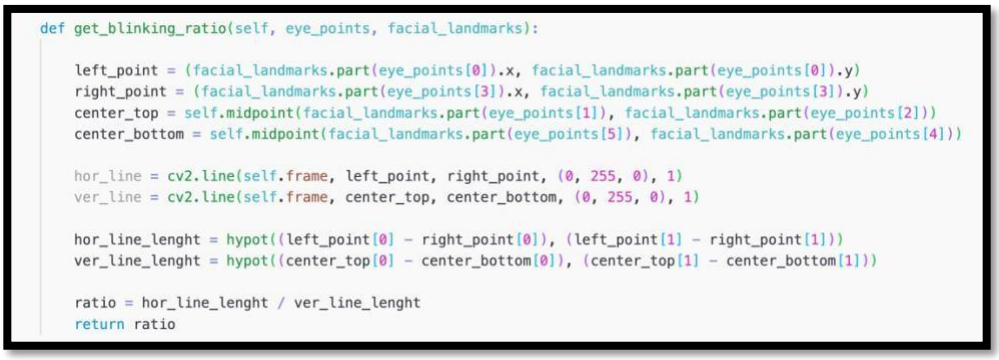
Figura 60. Código que ejecuta el reconocimiento facial con la imagen capturada

Asimismo, se presenta el código que hace posible la detección facial a través de la librería Mediapipe que realiza una detección basada en seis puntos clave del rostro detectado que propocionan un funcionamiento con múltiples rostros. Véase la Figura 61.

```
mpFaceDetection = mp.solutions.face_detection
faceDetection = mpFaceDetection.FaceDetection(0.75)
gray = cv2.cvtColor(self.frame, cv2.COLOR_BGR2GRAY)
frameRGB = cv2.cvtColor(self.frame, cv2.COLOR_BGR2RGB)
results = faceDetection.process(frameRGB)
faces = self.detector(gray)

if results.detections:
    for id, detection in enumerate(results.detections):

        bboxC = detection.location_data.relative_bounding_box
        ih, iw, ic = self.frame.shape
        bbox = int(bboxC.xmin * iw), int(bboxC.ymin * ih), int(bboxC.width * iw), int(bboxC.height * ih)
        x1, y1 = int(bboxC.xmin * iw) + int(bboxC.width * iw), int(bboxC.ymin * ih) +  int(bboxC.height * ih)
        cv2.rectangle(self.frame, bbox, (255,0,255), 1)
        # X, Y
        cv2.line(self.frame, (int(bboxC.xmin * iw), int(bboxC.ymin * ih)), (int(bboxC.xmin * iw) + 30,int(bboxC.ymin * ih)),
        cv2.line(self.frame, (int(bboxC.xmin * iw), int(bboxC.ymin * ih)), (int(bboxC.xmin * iw),int(bboxC.ymin * ih) + 30),
        #X1, Y1
        cv2.line(self.frame, (x1, int(bboxC.ymin * ih)), (x1 - 30,int(bboxC.ymin * ih)), (255,0,255), 5)
        cv2.line(self.frame, (x1, int(bboxC.ymin * ih)), (x1,int(bboxC.ymin * ih) + 30), (255,0,255), 5)
        # X, Y
        cv2.line(self.frame, (int(bboxC.xmin * iw), y1), (int(bboxC.xmin * iw) + 30,y1), (255,0,255), 5)
        cv2.line(self.frame, (int(bboxC.xmin * iw), y1), (int(bboxC.xmin * iw),y1 - 30), (255,0,255), 5)
        #X1, Y1
        cv2.line(self.frame, (x1, y1), (x1 - 30,y1), (255,0,255), 5)
        cv2.line(self.frame, (x1, y1), (x1,y1 - 30), (255,0,255), 5)

        score = int(detection.score[0]*100)
        cv2.putText(self.frame, f'{score}%', (bbox[0],bbox[1]-20), cv2.FONT_HERSHEY_PLAIN, 2, (255,0,255), 2)

        if(score >= 95):
            self.rostro = True
```

Figura 61. Código que permite la ejecución del proceso de detección facial

Finalmente, se evidencia el código que haciendo uso del algoritmo de detección facial previamente expuesto se evalúan los puntos que se situan en los ojos del rostro detectado para poder hacer el conteo de parpadeos en base a la distancia de estos puntos. Véase la Figura 62.



```
def get_blinking_ratio(self, eye_points, facial_landmarks):

    left_point = (facial_landmarks.part(eye_points[0]).x, facial_landmarks.part(eye_points[0]).y)
    right_point = (facial_landmarks.part(eye_points[3]).x, facial_landmarks.part(eye_points[3]).y)
    center_top = self.midpoint(facial_landmarks.part(eye_points[1]), facial_landmarks.part(eye_points[2]))
    center_bottom = self.midpoint(facial_landmarks.part(eye_points[5]), facial_landmarks.part(eye_points[4]))

    hor_line = cv2.line(self.frame, left_point, right_point, (0, 255, 0), 1)
    ver_line = cv2.line(self.frame, center_top, center_bottom, (0, 255, 0), 1)

    hor_line_lenght = hypot((left_point[0] - right_point[0]), (left_point[1] - right_point[1]))
    ver_line_lenght = hypot((center_top[0] - center_bottom[0]), (center_top[1] - center_bottom[1]))

    ratio = hor_line_lenght / ver_line_lenght
    return ratio
```

Figura 62. Código que ejecuta la verificación de vida por medio del conteo de parpadeos

5.1.4. Resultados y validación del modelo

A continuación, se muestran gráficamente los resultados obtenidos durante el entrenamiento del modelo.

En la siguiente figura se puede apreciar que la Red Neuronal Siamés requirió menos esfuerzo para la clasificación de cada Clase, dado que el Validation Loss y el Training Loss, a pesar de tomar mayor tiempo, logran alinear sus resultados. Un Training Loss y un Validation Loss con valores cercanos a 0, significa que se ha logrado valores de Pérdida muy bajos, consecuentemente se asume que el modelo es correcto. Véase la Figura 63.

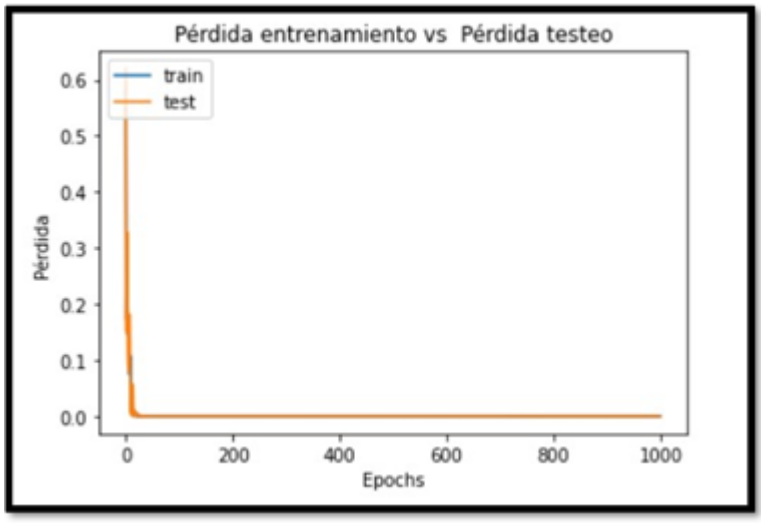


Figura 63. Pérdida, entrenamiento y validación

En la siguiente figura se muestra que el Validation Accuracy llega al valor de 1 a un tiempo parecido que la precisión de validación. Según (39) un Training Accuracy y un Validation Accuracy con valores cercanos a 1, indica que se ha logrado valores de Acierto muy altos, consecuentemente se asume que el modelo es correcto. Véase la Figura 64.

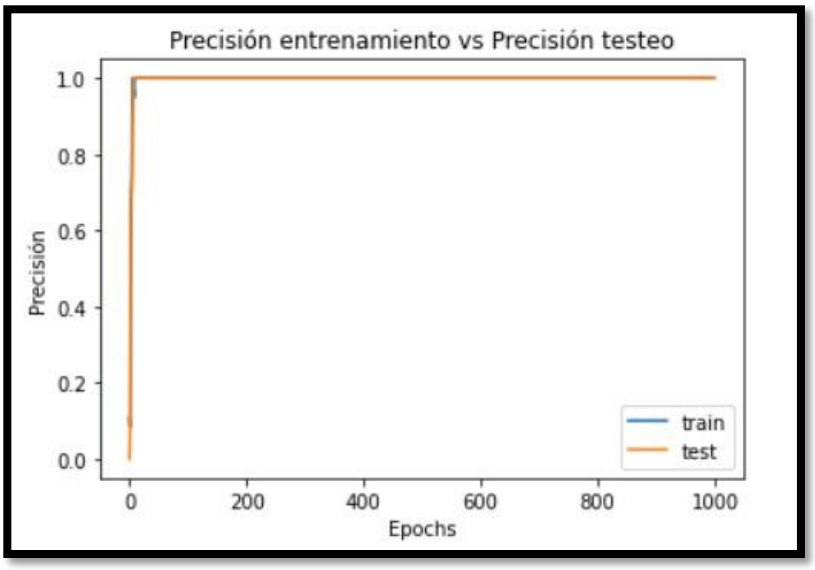


Figura 64. Precisión, entrenamiento vs validación

Ahora apreciaremos una comparación extra del Training Accuracy con el Training Loss, cuyo comportamiento se muestra óptimo, dado que el Training Accuracy logra obtener un valor de 1 y el Training Loss obtiene valores cercanos a 0.

El Training Loss cercano a 0 significa que se ha logrado valores de Pérdida muy bajos, consecuentemente se asume que el modelo es correcto. Y en el Training Accuracy el valor cercano a 1 significa que se ha logrado valores de Pérdida muy altos, consecuentemente se asume que el modelo es correcto. Véase la Figura 65.

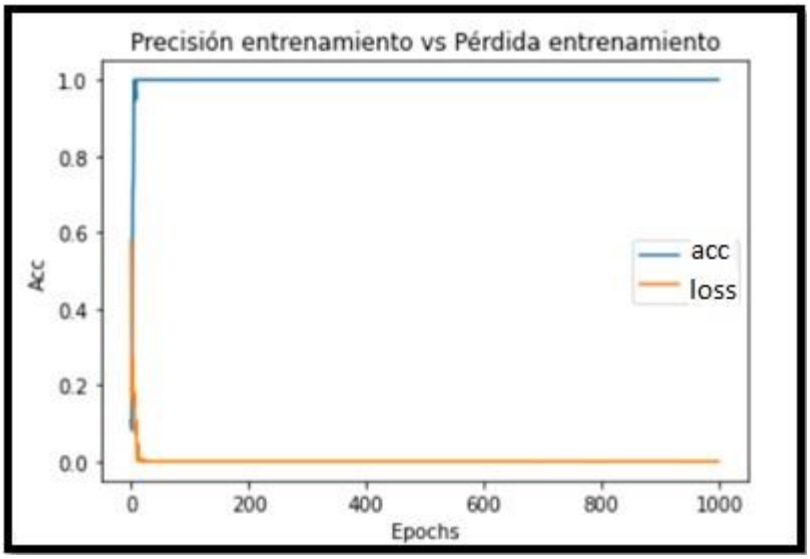


Figura 65. Entrenamiento, precisión vs pérdida

Si comparamos el Validation Accuracy con el Validation Loss, nos muestran comportamientos adecuados al igual que la figura 85, dado que el Validation Accuracy logra obtener valores de 1 y el Validation Loss logra obtener valores cercanos a 0.

El Validation Loss cercano a 0 significa que se ha logrado valores de pérdida muy bajos, consecuentemente se asume que el modelo es correcto. Así, también en el Validation Accuracy el valor cercano a 1 significa que se ha logrado valores de Pérdida muy altos, consecuentemente se asume que el modelo es correcto. Véase la Figura 66.

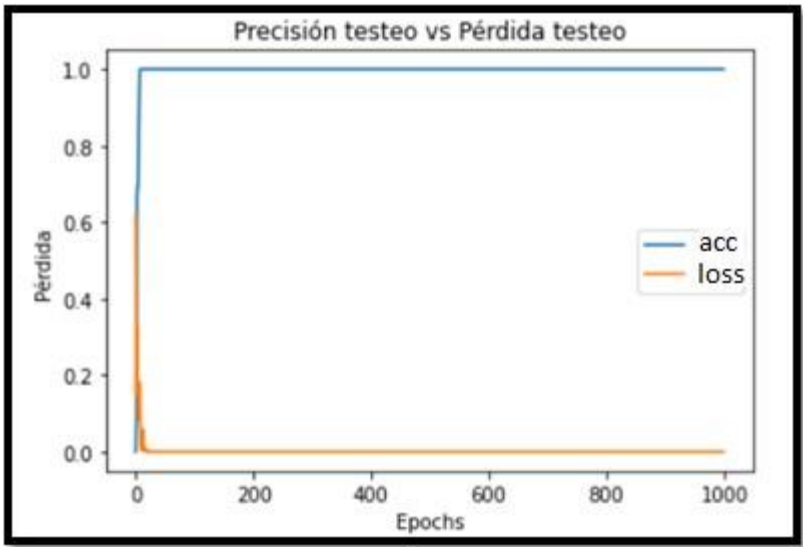


Figura 66. Validación, precisión vs pérdida

5.2. Pruebas y resultados

5.2.1. Pruebas de campo

Pruebas de la red neuronal siamés

En el apartado 5.1.4, ya se realizó la validación del modelo, sin embargo, el desarrollo del presente proyecto consideró realizar una prueba de campo durante una semana de trabajo, lo que se llevó a cabo del 11 al 15 de abril del 2022, donde se realizaron pruebas a los cuatro empleados registrados. Lo que sirvió para llevar a cabo las pruebas del modelo. De esta manera, se registraron todos los resultados dentro de una matriz de confusión que, según (40), en el campo del aprendizaje automático y específicamente el problema de la clasificación, como el de reconocimiento facial, una matriz de confusión, también conocida como matriz de error, es un diseño de tabla específico que permite la visualización del rendimiento de un algoritmo, típicamente uno de aprendizaje supervisado. Así, se obtendrán las métricas de accuracy, precision, recall y F1 medida.

En la presente investigación se opta por emplear las redes neuronales convolucionales del tipo siamés en el reconocimiento facial debido a sus ventajas sobre otros tipos de algoritmos. De esta manera, se

hizo una comparación entre los algoritmos histograma de patrón binario local(lbph), fisherfaces, eigenfaces y la red neuronal siamés en la función de reconocimiento facial a través de la matriz de confusión que nos permitirá obtener las métricas de recall, precision, accuracy y f1 score.

Las pruebas se desarrollaron en base al número total de empleados ya que el cálculo realizado en base a un nivel de confianza del 95% resultó en una muestra de 15 personas, de las cuales solo 12 de ellas se encontraban registradas en el sistema de reconocimiento facial, con el objetivo de realizar las pruebas del algoritmo. Para la red neuronal, se realizaron 40 pruebas. Estas pruebas se fueron desarrolladas bajo las mismas condiciones con el objetivo de tener un resultado sin ningún tipo de sesgo que represente la precisión de cada prueba.

Condiciones:

- Las pruebas se realizaron con la misma cámara
- Las pruebas se realizaron bajo una misma distancia
- Las pruebas se realizaron bajo las mismas condiciones ambientales externas (iluminación, pose, expresiones, y ángulo)

Con el objetivo de tener un resultado más realista y cerca al funcionamiento de la red neuronal siamés, se optó por realizar las pruebas dentro del dataset creado por la aplicación de escritorio. El cual, brindará resultados más cercanos a la realidad que al aplicar las pruebas en el dataset LFW que sirvió para el entrenamiento del modelo.

La matriz de confusión representa los resultados de los algoritmos puestos a prueba, en base a 4 métricas:

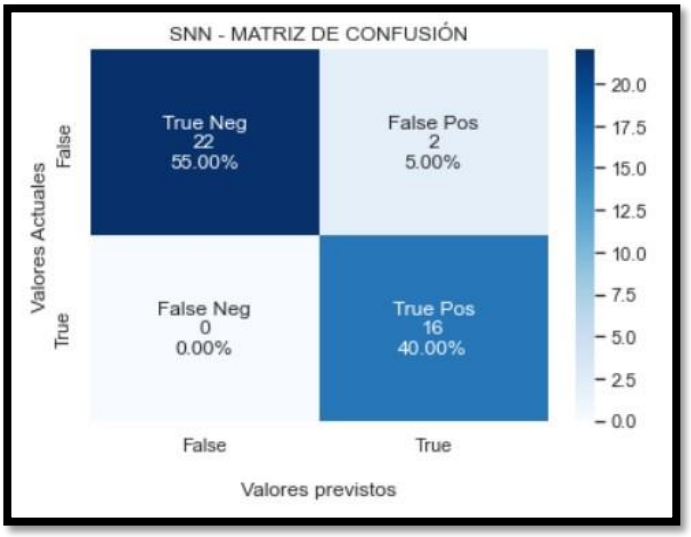
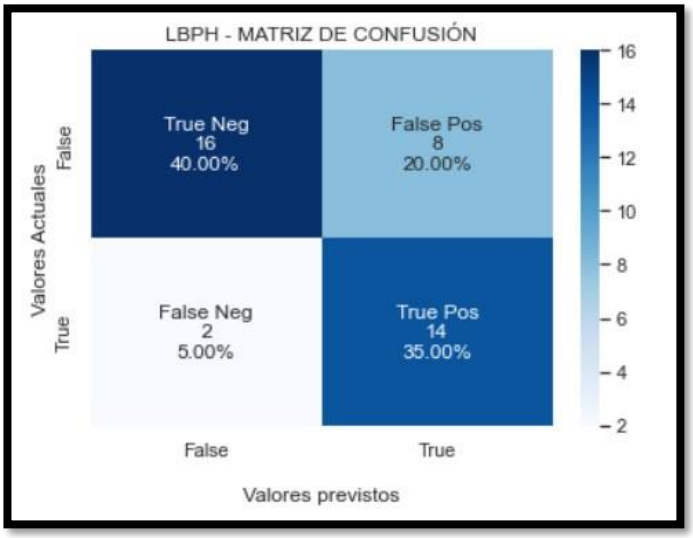
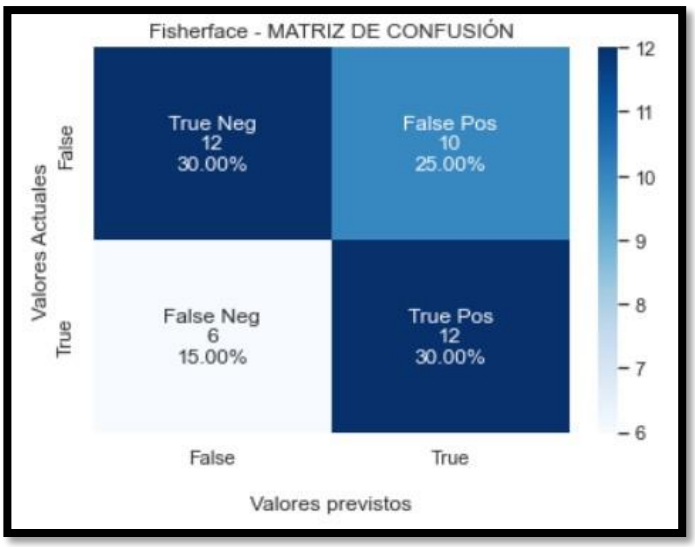
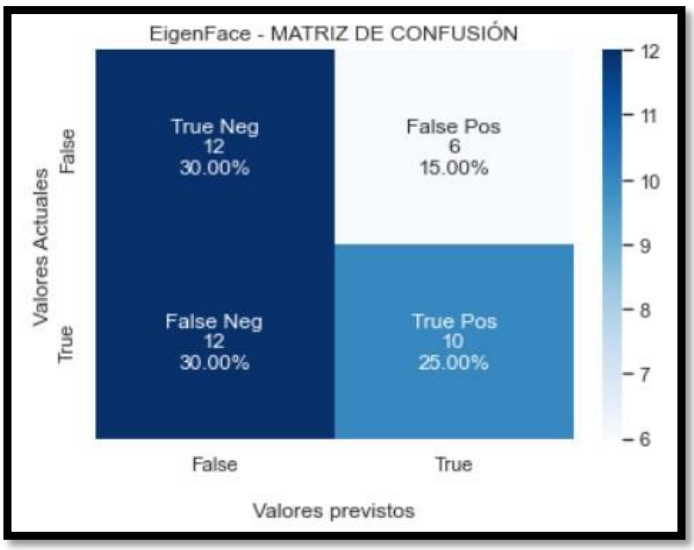
- Verdaderos positivos(tp): Las personas que se han reconocido correctamente y que estan registradas en el sistema.

- Verdaderos negativos(tn): Las personas que no han sido reconocidas y que no estan registras en el sistema.
- Falsos positivos(fp): Las personas que se han reconocido. Sin embargo, no se encontraban registras en el sitema.
- Falsos negativos(fn): Las personas que no se han reconocido. Sin embargo, si se encontraban registradas en el sistema.

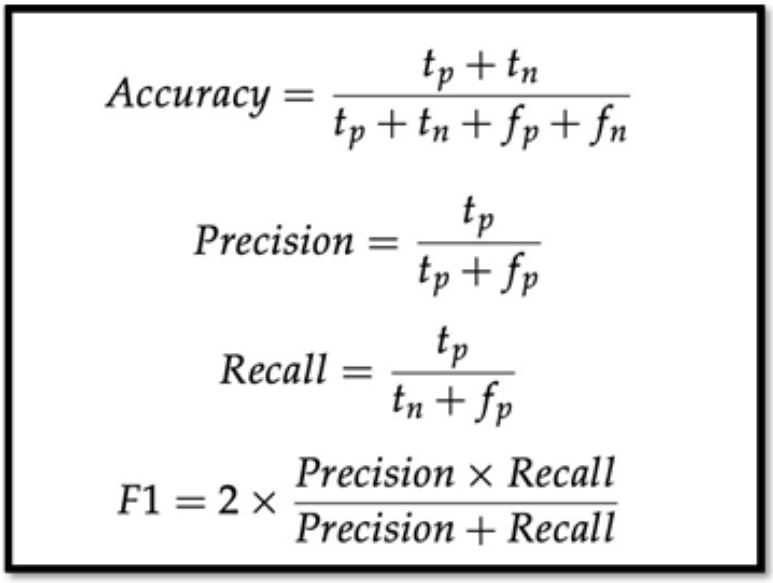
Una vez establecidas las condiciones de las pruebas de campo, se ejecutaron y plasmaron dentro de una matriz de confusión. Véase la Tabla 18.

Tabla 18. Comparación de algoritmos de reconocimiento facial

RED NEURONAL SIAMÉS	**Algoritmo LBPH**

<table>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>FISHERFACE</td><td>EIGENFACE</td></tr>
<tr><td> 
</td><td> 
</td></tr>
</table>

Al momento de obtener todos los resultados, la matriz de confusión nos permitirá evaluar los resultados de los algoritmos en base a las métricas de accuracy, precision, medida F1 y recall. Véase la Figura 67.



$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

$$Precision = \frac{t_p}{t_p + f_p}$$

$$Recall = \frac{t_p}{t_n + f_p}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Figura 67. Métricas de la matriz de confusión

Fuente: (40)

A continuación se pasan a plasmar los resultados de las pruebas de campo aplicadas a los algoritmos.

Tabla 19. Resultados de la matriz de confusión

Algoritmos	Accuracy	Precision	Recall	F1 Score
LBPH	0.75	0.67	0.72	0.69
FisherFaces	0.60	0.54	0.54	0.54
EigenFaces	0.55	0.67	0.75	0.71
SNN	0.95	0.92	1	0.96

Del análisis de la tabla podemos concluir que la red neuronal fue superior en los siguientes aspectos:

- La red neuronal siamés demostró a través de la métrica “precision” obtenida durante las pruebas fue de un 92%, lo cual significa que el modelo solo tiene 8 errores en pruebas de 100 intentos. En diferencia a las demás, que no superaron el 80%.

- La red neuronal siamés demostró a través de la métrica “accuracy” obtenida demuestra que tan exacto ha sido el modelo al momento de reconocer a clases tanto negativas como positivas. Así, obteniendo un valor del 95%. Marcando una gran diferencia frente a los algoritmos.
- La red neuronal siamés demostró a través de la métrica “recall” obtenida hace referencia a la exactitud del modelo al momento de reconocer a todas las clases positivas. Es decir, reconocer correctamente a todo el personal registrado. Se obtuvo el valor de 1, lo cual hace referencia a su precisión en cuanto a los demás.
- La red neuronal siamés demostró a través de la métrica “F1 Score” que permite visualizar el rendimiento que tuvo el modelo en base a un promedio estadístico. Se obtuvo un 96% del rendimiento que superó a los demás algoritmos hasta por un 20%.

Pruebas del dataset

Para obtener el número correcto de imágenes a utilizar en la aplicación de reconocimiento facial se hicieron pruebas prácticas para evaluar la efectividad con un número diferente de imágenes de entrenamiento y testeo. Se dividió el experimento en grupos de 10, 30 y 50 imágenes tomando como referencia la investigación (40) que resalta que para una red neuronal siamés se requieren un mínimo de imágenes de entrenamiento que varíen en un intervalo de 5 a 75 imágenes por cada clase. Las pruebas se registran con sus respectivos contadores de intentos correctos e incorrectos. Finalmente, se obtendrá el porcentaje de efectividad de cada grupo que permitirá su comparación.

Las pruebas se realizaron bajo las mismas condiciones con el objetivo de tener un resultado sin ningún tipo de sesgo que represente la precisión de cada prueba. Véase la Figura 68.

Condiciones:

- Las pruebas se realizaron con la misma cámara
- Las pruebas se realizaron bajo una misma distancia

- Las pruebas se realizaron bajo las mismas condiciones ambientales externas (iluminación, pose, expresiones, y ángulo)

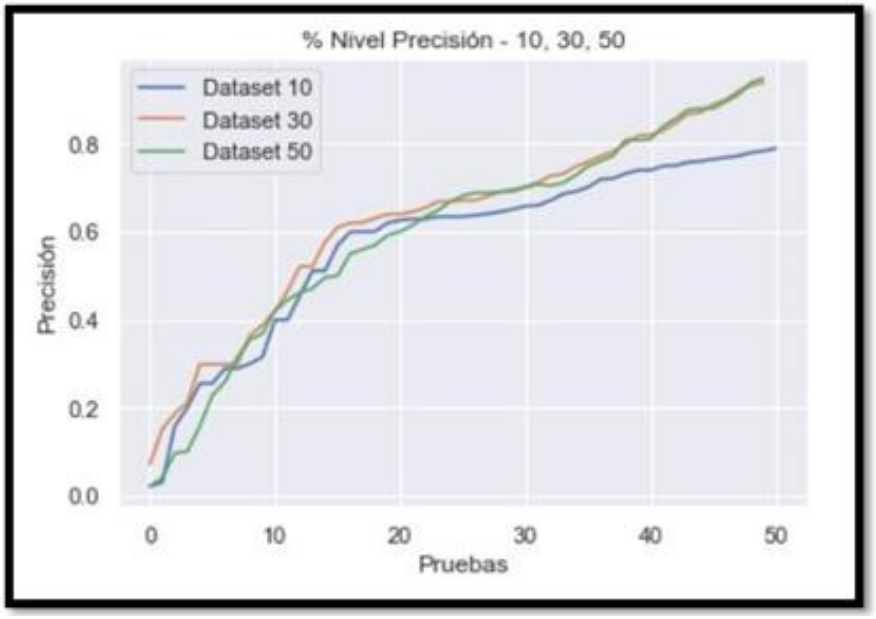


Figura 68. Precisión por el tipo de dataset

De esta manera se plasmaron todos los resultados dentro de una tabla que permitieron su fácil análisis.

Tabla 20. Pruebas de los datasets

	Número de pruebas: 50		
Dataset	**Correctos**	**Incorrectos**	**Resultado %**
Dataset de 10 imágenes	39	11	78
Dataset de 30 imágenes	47	3	94
Dataset de 50 imágenes	47	3	94

En base al presente resultado se eligió usar el dataset de 50 imágenes ya que representaba un mayor número lo cuál iba a contribuir con una mejor comparación entre los valores del grupo de entrenamiento y testeo. Además, se consideró que cada persona representaría 800kb en memoria lo que

demuestra que es el espacio de memoria a tomar por cada persona es menor a 1 megabyte. Por lo cuál, un gran número de personas registradas en el sistema no causará un inconveniente mayor.

5.2.2. Resultados

Objetivo 1: Determinar la influencia de la red neuronal siamés en la mejora del control de acceso a las instalaciones de FUDEC Perú

Para realizar un correcto reconocimiento facial, se implementó un nuevo tipo de red neuronal que permitió una agilidad de los procesos de registro y eliminación, de entrenamiento, verificación de vida y detección facial.

La red neuronal siamés, a través de su característica de aprendizaje de una sola oportunidad, facilitó el proceso de registro ya que solo se necesitaba 50 fotos del empleado para que pueda ser registrado en el sistema de reconocimiento facial. Por otro lado, para el proceso de eliminación solo hace falta borrar las fotos del empleado y volver a entrenar el modelo para que tenga efecto. Además, a través de la función de pérdida de triplete se logró un entrenamiento correcto que permitió garantizar la precisión del reconocimiento facial. Véase la Figura 69.

Los resultados obtenidos de las pruebas al reconocimiento facial, se obtuvo:

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

$$Precision = \frac{t_p}{t_p + f_p}$$

$$Recall = \frac{t_p}{t_n + f_p}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Figura 69. Red neuronal siamés matriz de confusión

Una vez plasmados los resultados, se procede a evaluar sus métricas en la siguiente Tabla 21.

Tabla 21. Resultados de la red neuronal siamés

Algoritmos	Accuracy	Precision	Recall	F1 Score
SNN	0.95	0.92	1	0.96

De esta manera, a través de las siguientes métricas obtenidas, se puede afirmar:

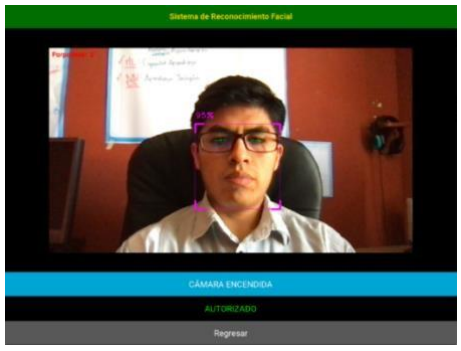
- La métrica "precision" obtenida durante las pruebas fue de un 92%, lo cual significa que el modelo solo tiene 8 errores en pruebas de 100 intentos.
- La métrica "accuracy" obtenida demuestra que tan exacto ha sido el modelo al momento de reconocer a clases tanto negativas como positivas. Así, obteniendo un valor del 95%.
- La métrica "recall" obtenida hace referencia a la exactitud del modelo al momento de reconocer a todas la clases positivas. Es decir, reconocer correctamente a todo el personal registrado.
- La métrica "F1 Score" permite visualizar el rendimiento que tuvo el modelo en base a un promedio estadístico. Se obtuvo un 96% del rendimiento, lo cual supone que el modelo ha sido desarrollado y entrenado de una manera correcta.
- La métrica de "precision" obtenida en la presente investigación es superior a otras que utilizaron otras técnicas para el reconocimiento facial.

Objetivo 2: Determinar la influencia de la detección facial en la mejora del control de acceso a las instalaciones de FUDEC Perú

Durante el desarrollo del sistema de reconocimiento facial, en la fase de implementación, surgió un requerimiento de automatización del proceso ya que no se podía tener un personal de seguridad que solo este controlando el acceso durante sus horas laborales. Además, se identificó que el reconocimiento facial tenía ciertas dificultados de identificar al empleado cuando se presentaban factores ambientales externos como la luz, la opacidad, o el ángulo. Lo cual, reducía la precisión obtenida del modelo.

En consecuencia, se aplica un algoritmo de la librería dlib de detección facial que mapea 67 puntos de un rostro. Así, se logra mantener una detección continua de los rostros en un flujo de video.

Tabla 22. Detección facial en el flujo de video

Una vez que se implemente el algoritmo, se puede obtener un porcentaje de la detección. La cual, está en base a que tan bien se puede visualizar el rostro en cuanto a la luz, la opacidad y el ángulo.

Durante los resultados del presente objetivo, se pudo identificar la precisión de la detección facial. La cual, consistió en una prueba de 50 intentos en los cuáles se busca encontrar la tasa de la detección facial. Cabe resaltar que todas las pruebas se realizaron bajo las mismas condiciones. A continuación, los resultados. Véase la Figura 70.

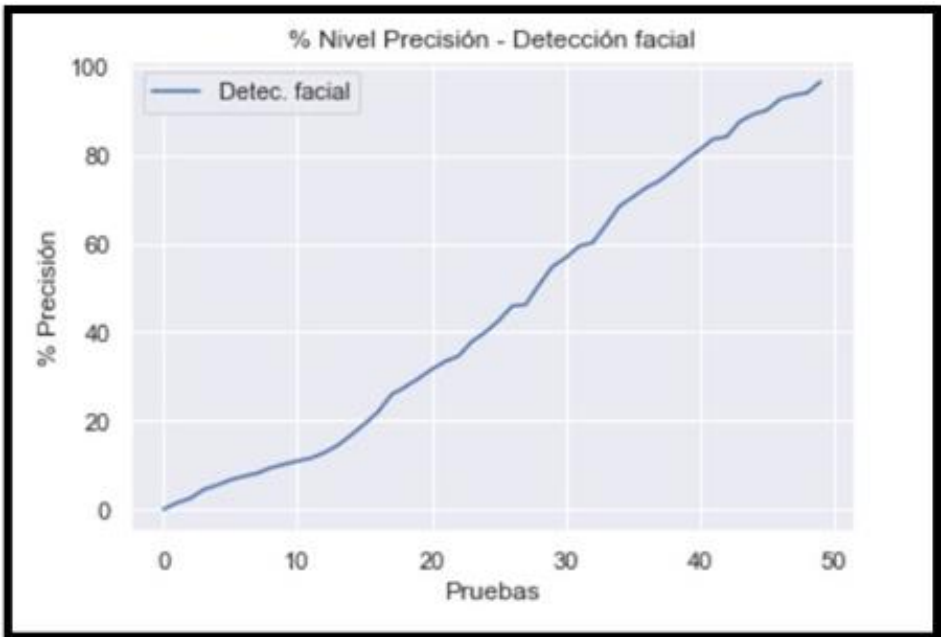


Figura 70. Porcentaje de precisión de detección facial

A continuación, se muestran los valores de las pruebas realizas y la tasa en un porcentaje:

Tabla 23. Tasa de detección facial.

	Correctos	**Incorrectos**	**Tasa de la detección facial**
Detección Facial	48	2	96%

Finalmente, se resalta que la detección facial contribuyó en los siguientes aspectos:

A través del porcentaje de detección se pudo reducir el impacto de los factores ambientales externos debido a que el reconocimiento facial se realizará cuando el nivel detección supere o sea igual al 94%.

Garantizando que el reconocimiento facial solo se ejecute en las mejoras condiciones ambientales que incrementen la precisión del modelo.

De la misma forma, haciendo uso del porcentaje de detección, se logró la automatización del proceso de reconocimiento facial ya que al momento de que se reconozca un rostro en el flujo de video de la cámara de vigilancia se realizará el reconocimiento facial.

Objetivo 3: Determinar la influencia de la verificación de vida en la mejora del control de acceso a las instalaciones de FUDEC Perú

Durante años la aplicación de las tecnologías biométricas representó una de las mejores opciones para control el acceso. Sin embargo, con el crecimiento de la delincuencia se presentaron casos, específicamente en la tecnología de reconocimiento facial, en los que se cometía fraude a través de la presentación de documentos o identificaciones que posean fotos de algún empleado autoriza. De esta manera, se lograba tener un acceso no autorizado que suponía un riesgo para la organización. En consecuencia, con el objetivo de desarrollar sistemas de reconocimiento facial más seguros se plantea la verificación de vida.

Para validar el correcto funcionamiento de la verificación de vida se realizaron pruebas. Las cuales, consistieron en una prueba de 50 intentos en los cuáles se busca encontrar la tasa de la verificación de vida. Cabe resaltar que todas las pruebas se realizaron bajo las mismas condiciones.

A continuación, los resultados. Véase la Figura 71.

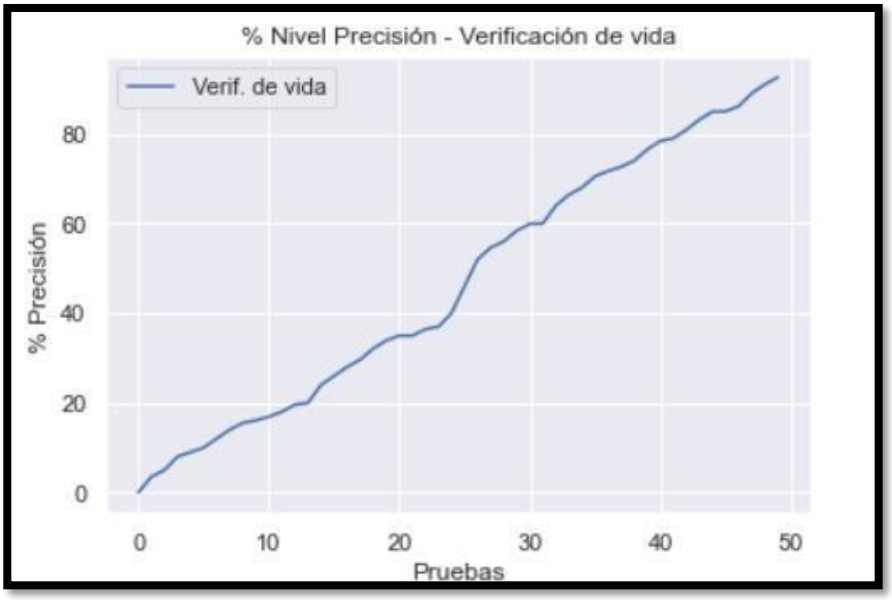


Figura 71. Porcentaje de precisión de la verificación de vida

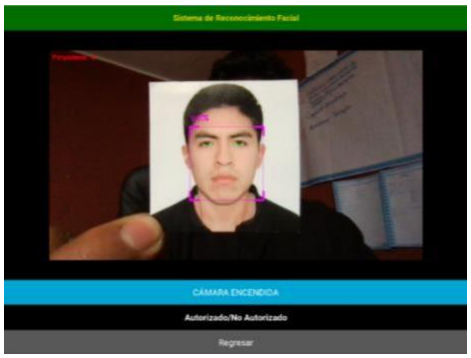
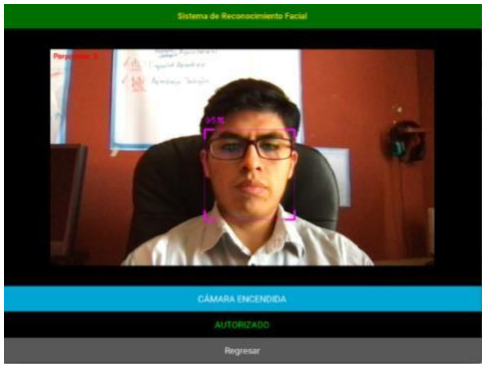
A continuación, se muestran los valores de las pruebas realizas y la tasa en un porcentaje:

Tabla 24. Tasa de verificación de vida

	Correctos	**Incorrectos**	**Tasa de la detección facial**
Verificación de Vida	46	4	92%

En el presente proyecto de investigación, se aplica la verificación de vida a través de la detección de los ojos y el conteo de parpadeos. Así, se logró verificar que el rostro detectado pertenezca a una persona real y no a una foto. Del mismo modo, se logró que solo se ejecute el reconocimiento facial una vez que se haya logrado tres parpadeos que verifiquen la autenticidad del rostro.

Tabla 25. Verificación de vida por el número de parpadeos

CONCLUSIONES

a. Con respecto al objetivo específico 1 "Determinar la influencia de la red neuronal siamés en la mejora del control de acceso a las instalaciones de FUDEC Perú", se concluye que la implementación del sistema en base a este tipo de red logró una precisión general del 94% que influye de manera positiva garantizando un control de acceso seguro.

b. Con respecto al objetivo específico 2 "Determinar la influencia de la detección facial en la mejora del control de acceso a las instalaciones de FUDEC Perú", se concluye que la implementación del sistema en base a este algoritmo logró una precisión de detección del 98% que influye de manera positiva garantizando un reconocimiento facial de calidad y su automatización

c. Con respecto al objetivo específico 3 "Determinar la influencia de la verificación de vida en la mejora del control de acceso a las instalaciones de FUDEC Perú", se concluye que la implementación del sistema en base a esta técnica logró una precisión general del 98% que influye de manera positiva evitando cualquier fraude que se presente en el sistema de reconocimiento facial.

d. Basándose en las conclusiones anteriores, se infiere que el uso de la red neuronal siamés en la tecnología de reconocimiento facial es la mejor opción frente a otros algoritmos, tomando en cuenta los resultados obtenidos a través de la matriz de confusión en las Pruebas de Campo del presente estudio.

RECOMENDACIONES

a. Con respecto al objetivo específico “Determinar la influencia de la red neuronal siamés en la mejora del control de acceso a las instalaciones de FUDEC Perú” se recomienda aplicar la tecnología de reconocimiento en tiempo real que agilice el proceso de identificación.
b. Con respecto al objetivo específico “Determinar la influencia de la verificación de vida en la mejora del control de acceso a las instalaciones de FUDEC Perú” se recomienda agregar un control adicional en 3D que permita verificar las dimensiones del rostro detectado.
c. Con respecto al objetivo específico 3 "Determinar la influencia de la verificación de vida en la mejora del control de acceso a las instalaciones de FUDEC Perú" se recomienda incorporar la tecnología de detección de movimiento para trabajar de una forma mucho más confiable.
d. Con respecto al proyecto en general, se recomienda la aplicación de un sistema de acceso electrónico de barras que garantice un control de acceso de alta seguridad.

REFERENCIAS BIBLIOGRÁFICAS

1. DAMMERT, Lucía; ARIAS, Patricia. El desafio en la delincuencia en latinoamerica: diagnóstico y respuestas de política. 2016.
2. FLORES MORENO, Diego Esteban; VILLACÍS FLORES, Santiago Alejandro. Implementación de un sistema de seguridad biométrica para la unidad de innovación tecnológica de la Universidad de las Américas. 2018.
3. RIOS, Alina. Seguridad y biometría en cuestión: el sistema federal de identificación biométrica (SIBIOS) en Argentina. *Aposta*, 2020.
4. MIER, Steeven Gustavo Romero. Inteligencia artificial como herramienta de estrategia y seguridad para defensa de los Estados. *Revista de la Escuela Superior de Guerra Naval*, 2019, vol. 16, no 1, p. 51-70.
5. VALENCIA, Jesús, et al. Detección de infracciones y matrículas en motocicletas, mediante visión artificial, aplicado a Sistemas Inteligentes de Transporte. *Revista Ibérica de Sistemas e Tecnologias de Informação*, 2020, no 37, p. 1-15.
6. TOVAR, Luis C.; ECHAVEZ, Martín E.; MARTELO, Raúl J. Diseño e implementación de un sistema de biometría facial para el control de acceso en instituciones de educación superior.
7. GUAYTA, Lucas Rogerio Garcés; SALINAS, Yasser Cesar Alvarado; ENRIQUEZ, Nixon Rafael Paladines. Procesamiento de video utilizando Apache Hadoop con OpenCV y JavaCV para reconocimiento facial. Infociencia, 2019, vol. 10, no 1, p. 25-31.
8. JACINTO, Marlene R. Paredes, et al. Sistema de vigilancia biométrico facial para el control delincuencial en la división policial Chimbote.
9. FILIO TORRES, Edgar Alfredo. Diseño de un sistema de seguridad física mediante Reconocimiento Facial a través del flujo de video, siguiendo las mejores prácticas de las normas ISO 80601, 13154, 19794 y el NISTIR 8238, para el área de seguridad de una empresa minera.

10. MUCHA, Jemima Elias, et al. Aplicación del deep learning para el reconocimiento facial con la presencia de oclusiones en el contexto de la pandemia covid 2021. Revista ECIPerú Volumen, 2021, vol. 18, no 1.
11. HUAMAN JULIAN, Zuly Milagros. Implementación de un sistema de gestión de seguridad electrónica con Machine Learning dirigido a Prosegur Perú para gestión de seguridad en viviendas de Lima Metropolitana. 2020.
12. MOREANO, José Augusto Cadena, et al. Reconocimiento facial con base en imágenes. *Revista Boletín Redipe*, 2017, vol. 6, no 5, p. 143-151.
13. VARIOR, Rahul Rama, et al. A siamese long short-term memory architecture for human re-identification. En European conference on computer vision. Springer, Cham, 2016. p. 135-153.
14. ORNA VILLALTA, Gustavo David, et al. Diseño e implementación de un sistema embebido de reconocimiento facial para el control de acceso usando deep learning. 2019. Tesis de Licenciatura. Quito.
15. KHAN, Suleman, et al. Facial recognition using convolutional neural networks and implementation on smart glasses. En 2019 International Conference on Information Science and Communication Technology (ICISCT). IEEE, 2019. p. 1-6.
16. ZHANG, Cheng, et al. Siamese neural network based gait recognition for human identification. En *2016 ieee international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016. p. 2832-2836.
17. KASAR, Manisha M.; BHATTACHARYYA, Debnath; KIM, T. H. Face recognition using neural network: a review. *International Journal of Security and Its Applications*, 2016, vol. 10, no 3, p. 81-100.
18. NGUYEN, Anh; YOSINSKI, Jason; CLUNE, Jeff. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

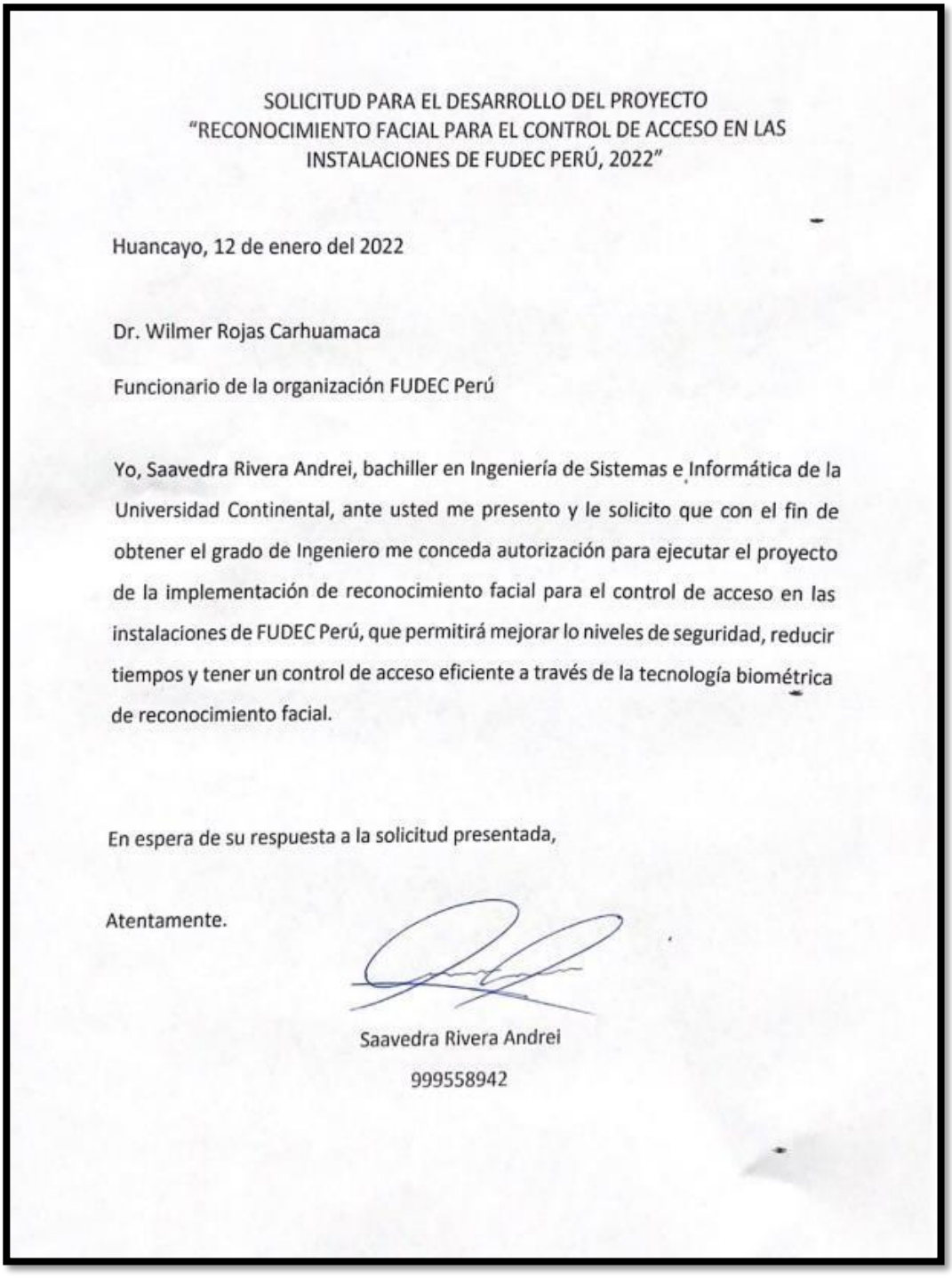
19. SCHMIDHUBER, Jürgen. Deep learning in neural networks: An overview. *Neural networks*, 2015, vol. 61, p. 85-117.
20. AGGARWAL, Charu C., et al. Neural networks and deep learning. *Springer*, 2018, vol. 10, p. 978-3.
21. ZHOU, Xiaokang, et al. Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 2020, vol. 17, no 8, p. 5790-5798.
22. DE BAETS, Leen, et al. Detection of unidentified appliances in non-intrusive load monitoring using siamese neural networks. *International Journal of Electrical Power & Energy Systems*, 2019, vol. 104, p. 645-653.
23. MEHMOOD, Atif, et al. A deep Siamese convolution neural network for multi-class classification of Alzheimer disease. *Brain sciences*, 2020, vol. 10, no 2, p. 84.
24. KOCH, Gregory, et al. Siamese neural networks for one-shot image recognition. En *ICML deep learning workshop*. 2015. p. 0.
25. DONG, Xingping; SHEN, Jianbing. Triplet loss in siamese network for object tracking. En *Proceedings of the European conference on computer vision (ECCV)*. 2018. p. 459-474.
26. BHATTACHARYA, Shubhobrata, et al. Smart attendance monitoring system (SAMS): a face recognition based attendance system for classroom environment. En *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2018. p. 358-360.
27. GÓMEZ, Tesillo; MAYUMI, Cynthia. Análisis comparativo de los algoritmos Fisherfaces y LBPH para el reconocimiento facial en diferentes condiciones de iluminación y pose, Tacna–2015. 2016.
28. HAGHIGHAT, Mohammad; ABDEL-MOTTALEB, Mohamed; ALHALABI, Wadee. Fully automatic face normalization and single sample face recognition in unconstrained environments. *Expert Systems with Applications*, 2016, vol. 47, p. 23-34.

29. PABIANIA, Maribelle Dequilla, et al. Face recognition system for electronic medical record to access out-patient information. *Jurnal Teknologi*, 2016, vol. 78, no 6-3.
30. ALONSO-SIERRA, Juan David, et al. Sistema de reconocimiento facial para control de acceso a viviendas. 2019.
31. RADZI, Syafeeza Ahmad, et al. IoT based facial recognition door access control home security system using raspberry pi. *International Journal of Power Electronics and Drive Systems*, 2020, vol. 11, no 1, p. 417.
32. COLE, Orane; EL-KHATIB, Khalil. A privacy enhanced facial recognition access control system using biometric encryption. En *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2017. p. 199-206.
33. ZAMBRANO GARCÍA, Stefany Tatiana, et al. Diseño del proceso de mejora continua de la organización basado en el marco de trabajo scrum. 2020.
34. JOSE, Edwin, et al. Face recognition based surveillance system using facenet and mtcnn on jetson tx2. En *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. IEEE, 2019. p. 608-613.
35. LIU, Daizong, et al. SAANet: Siamese action-units attention network for improving dynamic facial expression recognition. *Neurocomputing*, 2020, vol. 413, p. 145157.
36. CHEN, Jiasi; RAN, Xukan. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 2019, vol. 107, no 8, p. 1655-1674.
37. JIA, Xianyan, et al. Highly scalable deep learning training system with mixedprecision: Training imagenet in four minutes. *arXiv preprint arXiv:1807.11205*, 2018.
38. REN, Fuji; XUE, Siyuan. Intention detection based on siamese neural network with triplet loss. *IEEE Access*, 2020, vol. 8, p. 82242-82254.
39. DE GROOF, Albert J., et al. Deep-learning system detects neoplasia in patients with Barrett's esophagus with higher accuracy than endoscopists in a multistep training and validation study with benchmarking. *Gastroenterology*, 2020, vol. 158, no 4, p. 915-929. e4.

40. NAWWAR, NADIA MOSTAFA; KASBAN, Hany; SALAMA, May. Improvement of confusion matrix for Hand Vein Recognition Based On Deep-Learning multiclassifier Decisions. *Arab Journal of Nuclear Sciences and Applications*, 2021, vol. 54, no 4, p. 133-146.
41. BASAVEGOWDA, Hema Shekar; DAGNEW, Guesh. Deep learning approach for microarray cancer data classification. *CAAI Trans. Intell. Technol.*, 2020, vol. 5, no 1, p. 22-33.

ANEXOS

ANEXO 01. SOLICITUD PARA EL DESARROLLO DEL PROYECTO



SOLICITUD PARA EL DESARROLLO DEL PROYECTO
"RECONOCIMIENTO FACIAL PARA EL CONTROL DE ACCESO EN LAS INSTALACIONES DE FUDEC PERÚ, 2022"

Huancayo, 12 de enero del 2022

Dr. Wilmer Rojas Carhuamaca

Funcionario de la organización FUDEC Perú

Yo, Saavedra Rivera Andrei, bachiller en Ingeniería de Sistemas e Informática de la Universidad Continental, ante usted me presento y le solicito que con el fin de obtener el grado de Ingeniero me conceda autorización para ejecutar el proyecto de la implementación de reconocimiento facial para el control de acceso en las instalaciones de FUDEC Perú, que permitirá mejorar lo niveles de seguridad, reducir tiempos y tener un control de acceso eficiente a través de la tecnología biométrica de reconocimiento facial.

En espera de su respuesta a la solicitud presentada,

Atentamente.

Saavedra Rivera Andrei
999558942

ANEXO 02. ACTA DE ACEPTACIÓN DEL PROYECTO

ACTA DE ACEPTACIÓN DEL PROYECTO

Nombre del proyecto	Siglas del proyecto
Reconocimiento facial para el control de acceso en las instalaciones de FUDEC Perú, 2022	RFCA
Nombre del cliente	
Dr. Wilmer Rojas Carhuamaca	

Declaración de la aceptación formal	
Por el presente se deja constancia de que el proyecto "Reconocimiento facial para el control de acceso en las instalaciones de FUDEC Perú, 2022" ha sido aceptado y aprobado por el Dr. Wilmer Rojas Carhuamaca	
Listado de requerimientos	Estado
Como usuario final, requiero activar y desactivar el reconocimiento facial automático	Completado
Como usuario final, requiero que el sistema de reconocimiento facial pueda hacer una detección de vida	Completado
Como usuario final, requiero que el sistema de reconocimiento facial sea automático	Completado
Como usuario final, requiero registrar a un empleado al sistema de reconocimiento facial	Completado
Como usuario final, requiero eliminar a un empleado del sistema de reconocimiento facial	Completado
Como usuario final, requiero visualizar la lista actual de los empleados registrados	Completado
Observaciones adicionales	
La instalación y desarrollo de la aplicación cumplió con los requerimientos de la organización	
Aceptado por	
Nombre del funcionario	Fecha
Dr. Wilmer Rojas Carhuamaca	02/05/2022

Dr. Wilmer Rojas Carhuamaca
DNI 41601643

ANEXO 03. PROTOCOLO DE CAPTURA DE IMAGEN

PROTOCOLO

PARA LA CAPTURA DE IMAGEN DEL SISTEMA DE RECONOCIMIENTO FACIAL

Autor:

Bach. Andrei Saavedra Rivera

Versión:

1.0

Fecha de actualización:

21/05/2022

Revisor:

Dr. Daniel Gamarra

Resumen

El protocolo definido para la captura de imagen del sistema de reconocimiento facial está conformado por un conjunto de reglas e indicaciones que guían el correcto patrón o modelo de obtención de imagen. Este protocolo permite estandarizar las acciones e instrumentos que serán utilizados en el sistema para asegurar la precisión del reconocimiento.

La presencia de este protocolo cuenta con parámetros que permitirán dar a conocer el funcionamiento en las mejores condiciones para mejorar la eficiencia del modelo. El reconocimiento facial depende de factores como el ángulo, distancia, opacidad, iluminación y nitidez que influyen de forma positiva o negativa sobre el resultado final.

Este protocolo permite resolver esta variabilidad de factores, ya que precisa de manera puntual los parámetros y características de los instrumentos, así como las mejores acciones para lograr un reconocimiento facial de calidad.

Objetivos

- Promover la correcta utilización del sistema para conseguir el mayor rendimiento y efectividad en el reconocimiento facial. Asimismo, lograr una verificación de vida precisa y la detección facial sea la correcta para prevenir cualquier factor externo que impida la identificación.
- Homologar el proceso de reconocimiento facial y control de acceso para su correcto uso.

Ámbito de aplicación

Este protocolo está dirigido a todas las personas que instalarán y harán uso del sistema.

Terminología

- **Sistema de reconocimiento facial para el control de acceso:** Es un sistema de computación que cuenta con parámetros, métodos y técnicas de inteligencia artificial para lograr un reconocimiento facial exitoso y brindar acceso a las instalaciones de la organización.
- **Python:** Es un lenguaje de programación de propósito general. En este caso, es el lenguaje que se usó para desarrollar el sistema de reconocimiento facial.
- **Arduino:** Es una plataforma de creación de software y hardware. Se usó para codificar y construir el sistema de control de acceso.
- **TensorFlow:** Es una librería de código abierto para aprendizaje profundo. Es la librería usada para codificar el sistema de reconocimiento facial.
- **NumPy:** Es una librería de funciones matemáticas gestionadas a través de listas de alto nivel. Fue usada para codificar el sistema de reconocimiento facial.
- **OpenCV:** Es una librería de codigo abierto que brinda funciones para aplicaciones de visión artificial. Fue usa para codificar el sistema de reconocimiento facial.

Recursos

Para instalación:

- Espacio disponible en disco de 300 megabytes.
- Instalación de Python en su versión 3.6.
- Instalación de las librerías Tensorflow, OpenCV (no menor a 4.4.41), y Numpy
- Instalación de Arduino en su versión 1.8.19.

Para su uso:

- Cámara de videovigilancia que permita capturar el flujo de video con una resolución igual o mayor a 1920 x 1080 píxeles.
- Cerrojo de selenoide de 12V
- Placa Arduino UNO
- Relay 5V
- Baterías de 4200 mAh

Procedimiento

Antes de la captura de imagen

Dispositivo de captura

- La cámara de videovigilancia debe contar con la iluminación suficiente para que la detección facial pueda igualar o superar el 95% de detección.
- La cámara de videovigilancia debe estar a una distancia no mayor a 1.5 metros en relación con el rostro.
- Tiene que estar en una posición estratégica que permita obtener un ángulo frontal del rostro.
- Puede capturar el flujo de video a color o en escala de grises.

Rostro

- De preferencia la persona debe portar los accesorios faciales que utiliza de forma permanente.
- Se sugiere situar el rostro de forma frontal hacia la cámara de videovigilancia.

Durante el proceso de captura de rostro

Las siguientes indicaciones deben considerarse para un buen reconocimiento facial:

- Realizar parpadeos lentos y completos que permitan al sistema la verificación de vida.
- Posicionarse de tal manera que el rostro tenga un ángulo frontal a la cámara de videovigilancia.

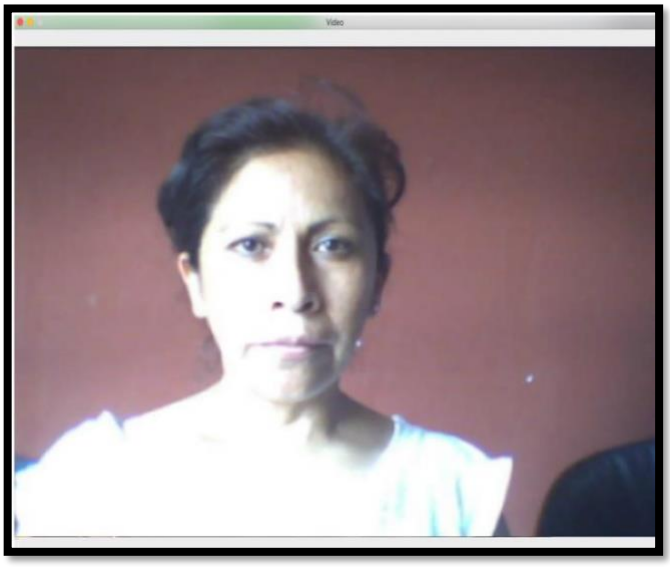
ANEXO 04. DETALLES DE PRUEBAS FUNCIONAL DEL SISTEMA DE RECONOCIMIENTO FACIAL

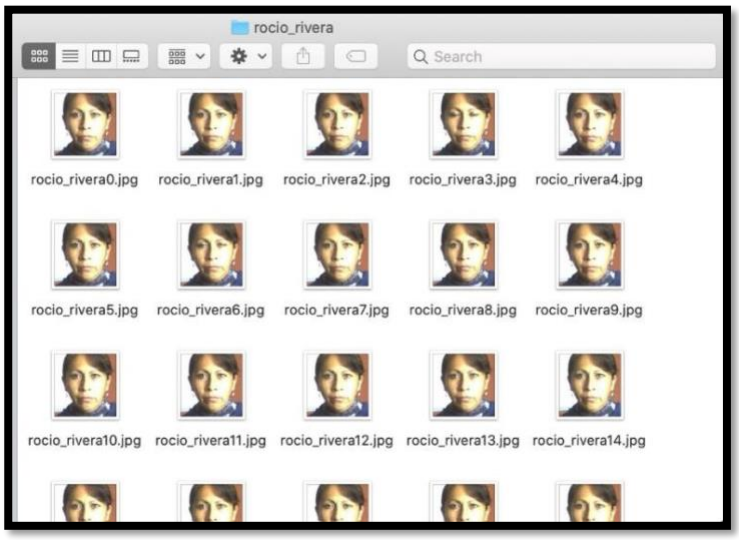
Registro de un nuevo empleado al sistema	**CP001**	**RF-001**
Descripción: El sistema permitirá el registro de un empleado nuevo al sistema a través de una captura de imágenes y entrenamiento del modelo.		
Técnica de prueba caja negra: Requerimiento funcional / Caso de uso		
Casos: Caso 1.1: Datos de entrada: Ingresar el primer nombre y apellido del empleado a registrar. Resultado esperado: El sistema empieza a capturar imágenes automáticamente una vez que se detecte un rostro en el flujo de cámara y las guarda en el sistema de archivos. Posteriormente entrenar el modelo para que el registro tenga efecto sobre el sistema y se pueda identificar al nuevo personal.		

Resultado obtenido:

1) Ingreso del nombre y apellidos

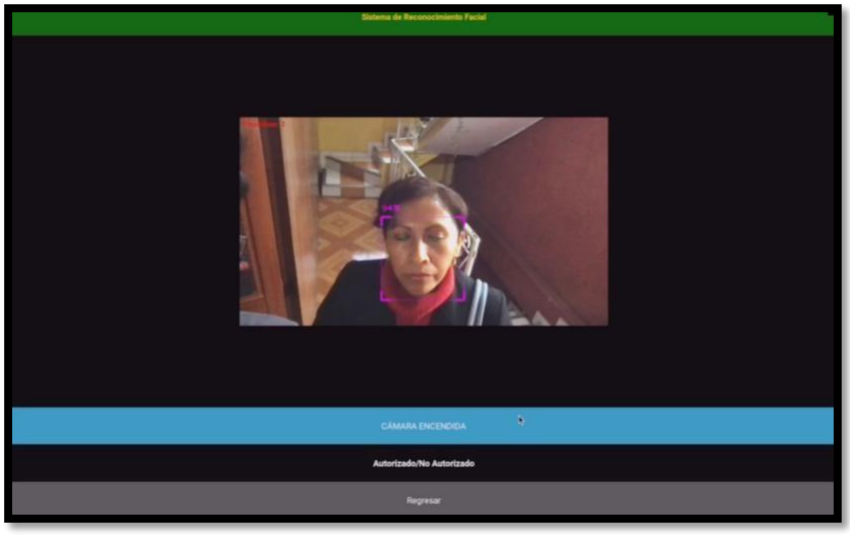
2) Captura de imágenes

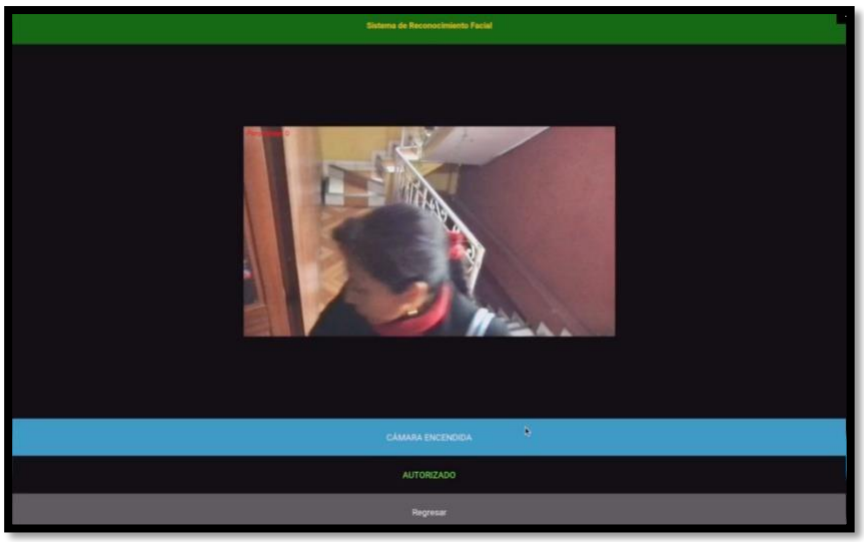




3) Entrenamiento del modelo

4) Identificación del nuevo empleado





Eliminación de un empleado del sistema	CP002	RF-002
Descripción: El sistema permitirá eliminar a un empleado del sistema para que ya no tenga acceso a las instalaciones de la organización. El proceso se da a través del ingreso del nombre y apellido del empleado para posteriormente entrenar el modelo para que los cambios surjan efecto.		
Técnica de prueba caja negra: Requerimiento funcional / Caso de uso		
Casos: Caso 1.1: Datos de entrada: Ingresar el primer nombre y apellido del empleado Resultado esperado: El sistema elimina la carpeta del empleado con sus capturas de imágenes respectivas. Al entrenar el modelo el sistema ya no será capaz de identificar al empleado.		

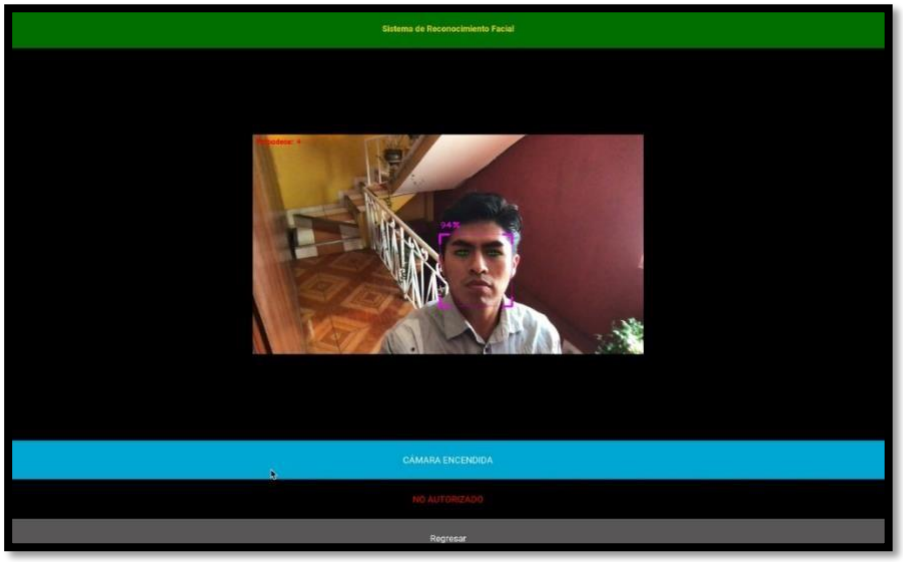
Resultado obtenido:

1) Ingreso del nombre y apellidos

2) Eliminación de la carpeta del empleado

3) Entrenamiento del modelo

4) El empleado no es identificado



Activación y desactivación del sistema de reconocimiento facial	**CP003**	**RF-003**
Descripción: El sistema permitirá la activación o desactivación del sistema para tener un control de acceso eficiente siempre y cuando sea necesario.		
Técnica de prueba caja negra: Requerimiento funcional / Caso de uso		

Casos:

Caso 1.1:

Datos de entrada:

Inicia el proceso de reconocimiento facial al presionar el botón “Iniciar Reconocimiento” que mostrará un mensaje de “cámara encendida ” y s i se vuelve a presionar se desactiva el proceso del sistema con un mensaje de “cámara desactivada”.

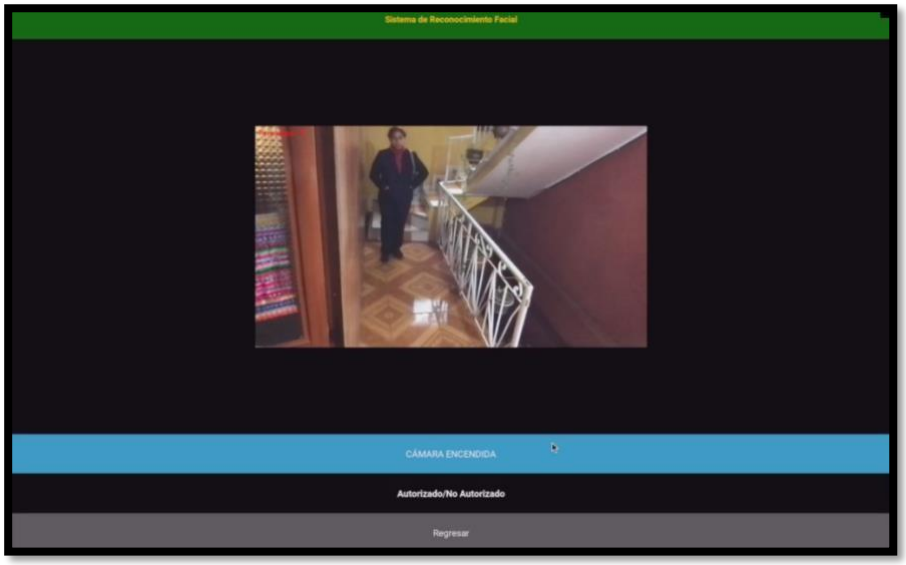
Resultado esperado:

El sistema inicia el proceso de reconocimiento facial con el click sobre el botón “Iniciar Reconocimiento” y si se vuelve a presionar se detiene el proceso.

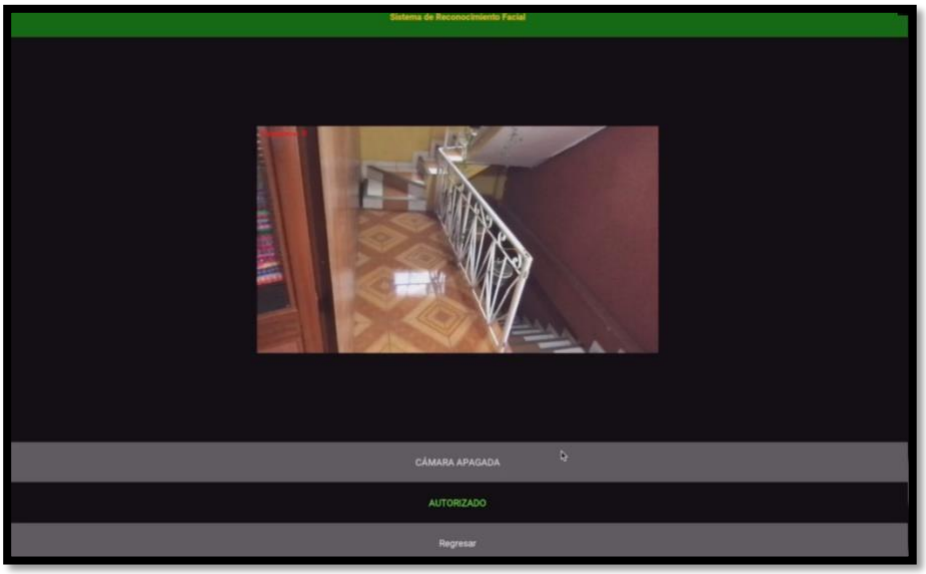
Resultado obtenido:

1) Inicio del proceso de reconocimiento facial

2) Mensaje de confirmación



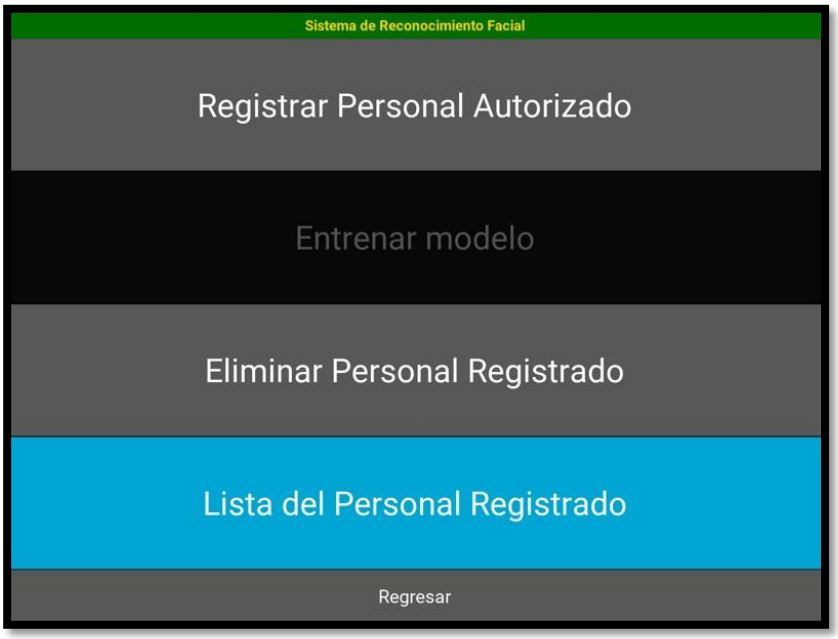
3) Suspensión del proceso de reconocimiento facial



Visualización de la lista actual de empleados registrados	CP004	RF-004
Descripción: El sistema permitirá visualizar a través de una interfaz a los empleados que se encuentran registrados en el sistema de reconocimiento facial.		
Técnica de prueba caja negra: Requerimiento funcional / Caso de uso		
Casos: Caso 1.1: Datos de entrada: Presionar el botón de “Mostrar Lista Personal” Resultado esperado: El sistema redireccionará a una interfaz en la cual se podrán visualizar a todos los empleados que están actualmente registrados en el sistema de reconocimiento facial		

Resultado obtenido:

1) Click en el botón “Mostrar Lista Personal”

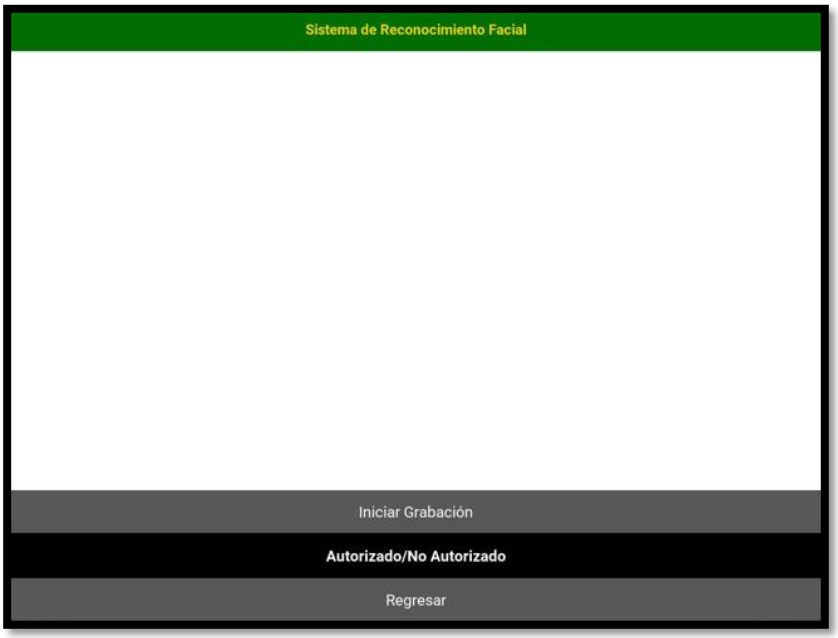


2) Visualización de la lista del personal actual registrado

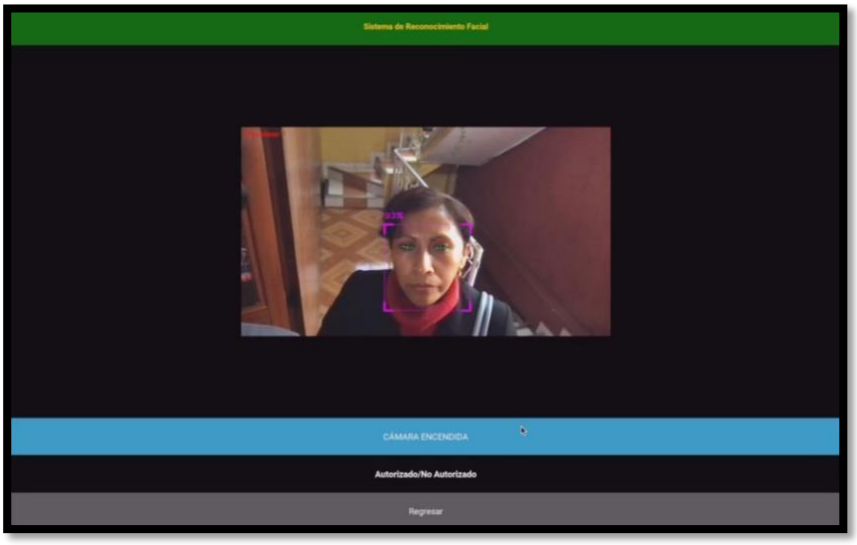
Verificación de vida	CP005	RF-005
Descripción: El sistema de reconocimiento antes de realizar la captura de imagen ejecutará una verificación de vida que da a través de un conteo de parpadeos para prevenir la suplantación de identidad.		
Técnica de prueba caja negra: Requerimiento funcional / Caso de uso		
Casos: Caso 1.1: Datos de entrada: Presionar el botón “Iniciar Reconocimiento” Resultado esperado: Una vez que se detecte un rostro en el flujo de video, se iniciará con el proceso de verificación de vida a través del conteo de parpadeos del		

Resultado obtenido:

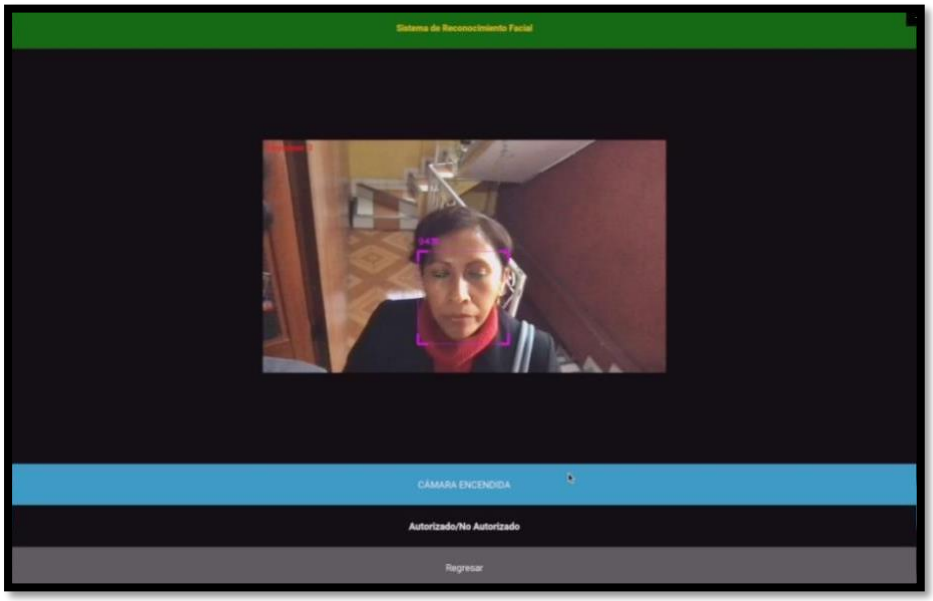
1) Inicio del proceso de reconocimiento facia l



2) Detección facial y conteo de parpadeos



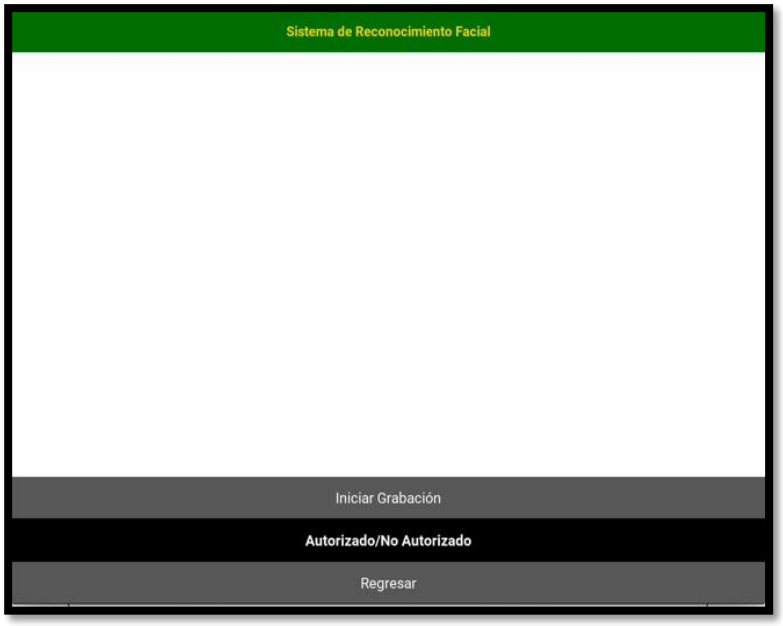
3) Captura de imagen



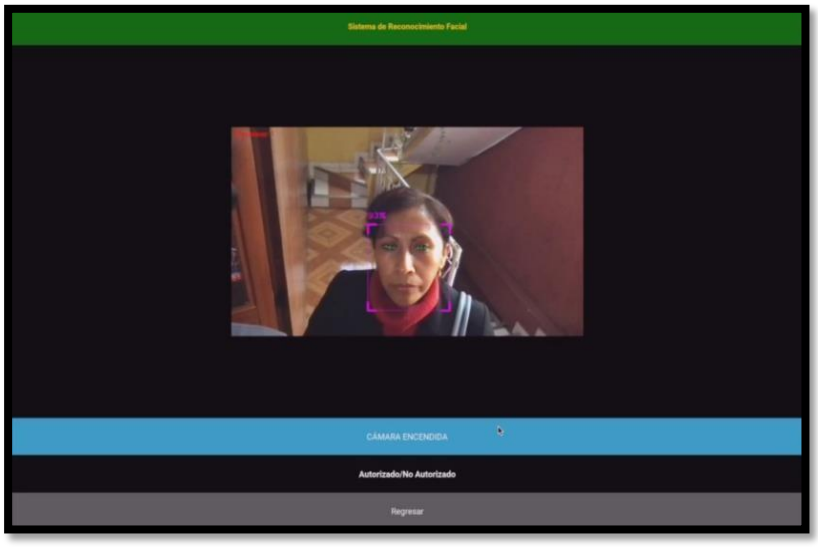
Automatización del sistema de reconocimiento facial	CP006	RF-006
Descripción: El sistema de reconocimiento facial a través de la detección facial localizará los rostros y le asignará un porcentaje de detección que está en base a la nitidez e iluminación de la imagen. Una vez que se alcance el 95% o más se capturará la imagen automáticamente.		
Técnica de prueba caja negra: Requerimiento funcional / Caso de uso		
Casos: Caso 1.1: Datos de entrada: Presionar el botón “Iniciar Reconocimiento” Resultado esperado: Se localizará el rostro y una vez que iguale o supere el 95% de detección se capturará la imagen automáticamente.		

Resultado obtenido:

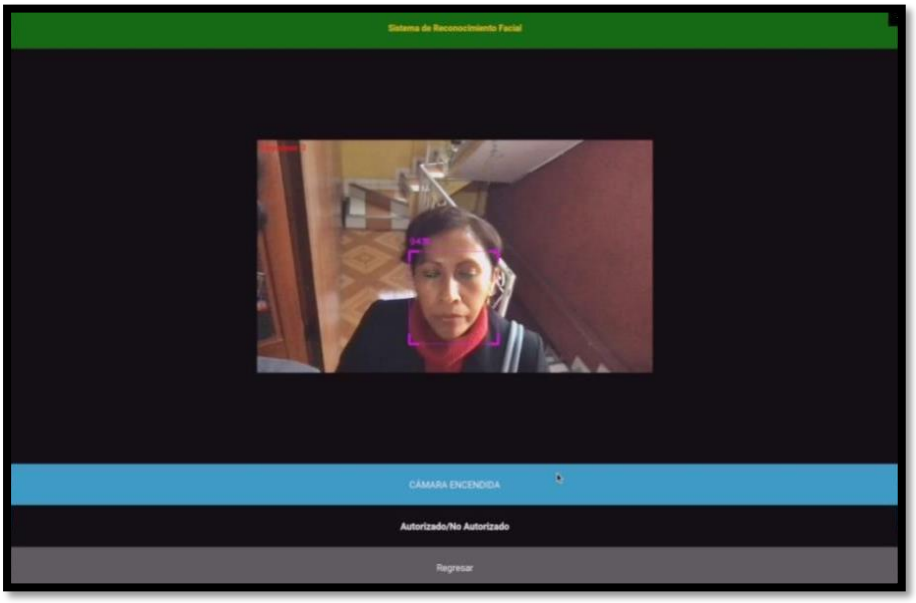
1) Inicio del proceso de reconocimiento facial



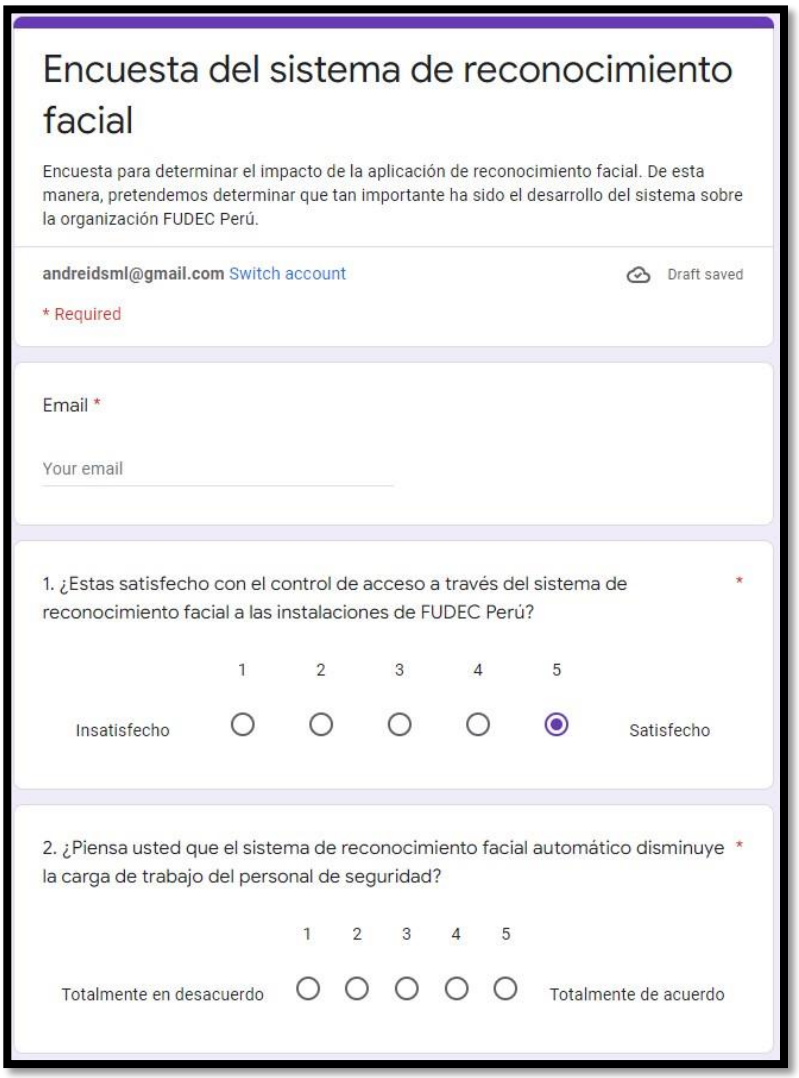
2) Detección facial y conteo de parpadeos



3) Captura de imagen



ANEXO 05. ENCUESTA DEL SISTEMA



Encuesta del sistema de reconocimiento facial

Encuesta para determinar el impacto de la aplicación de reconocimiento facial. De esta manera, pretendemos determinar que tan importante ha sido el desarrollo del sistema sobre la organización FUDEC Perú.

andreidsml@gmail.com Switch account

Draft saved

* Required

Email *

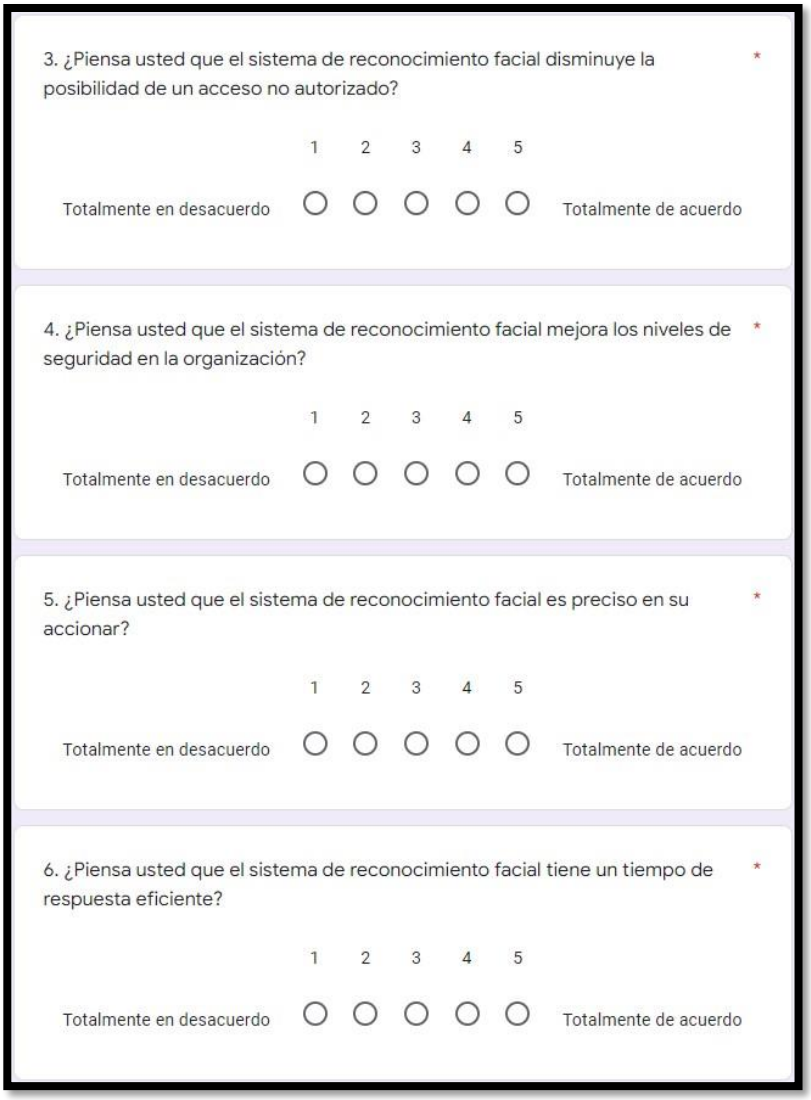
Your email

1. ¿Estas satisfecho con el control de acceso a través del sistema de reconocimiento facial a las instalaciones de FUDEC Perú? *

	1	2	3	4	5	
Insatisfecho	○	○	○	○	◉	Satisfecho

2. ¿Piensa usted que el sistema de reconocimiento facial automático disminuye la carga de trabajo del personal de seguridad? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo



3. ¿Piensa usted que el sistema de reconocimiento facial disminuye la posibilidad de un acceso no autorizado? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo

4. ¿Piensa usted que el sistema de reconocimiento facial mejora los niveles de seguridad en la organización? *

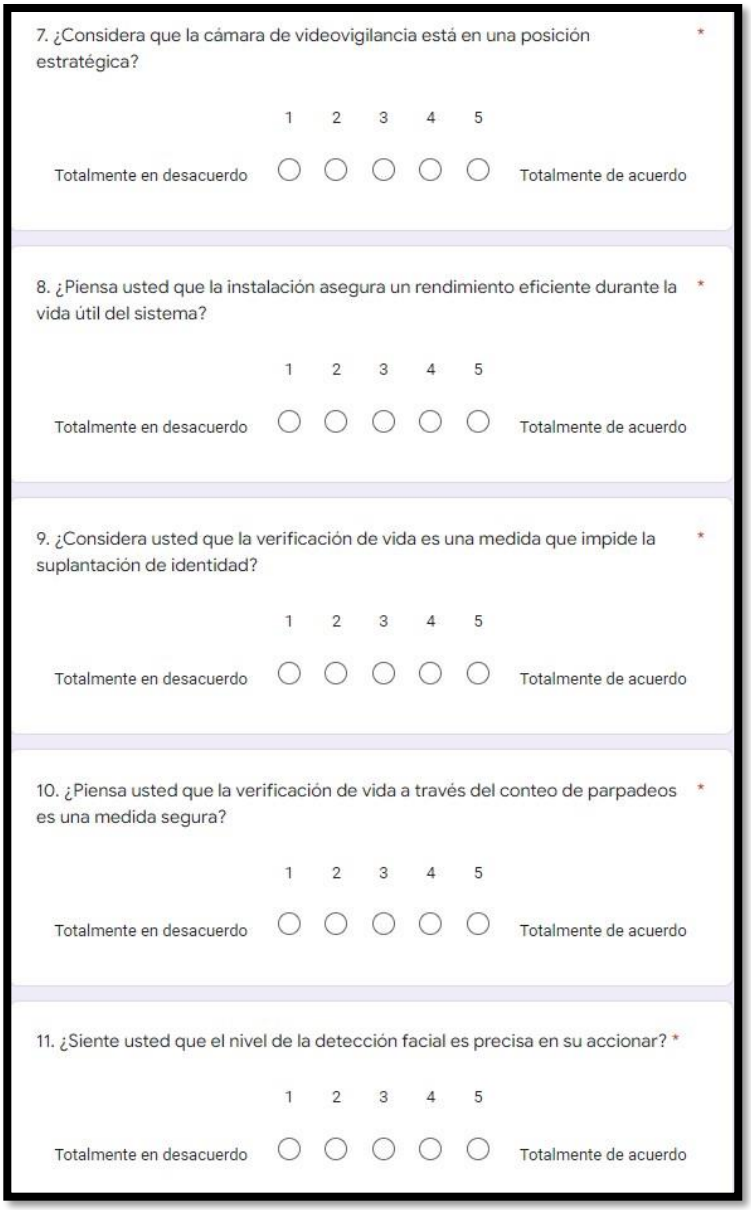
	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo

5. ¿Piensa usted que el sistema de reconocimiento facial es preciso en su accionar? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo

6. ¿Piensa usted que el sistema de reconocimiento facial tiene un tiempo de respuesta eficiente? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo



7. ¿Considera que la cámara de videovigilancia está en una posición estratégica? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo

8. ¿Piensa usted que la instalación asegura un rendimiento eficiente durante la vida útil del sistema? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo

9. ¿Considera usted que la verificación de vida es una medida que impide la suplantación de identidad? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo

10. ¿Piensa usted que la verificación de vida a través del conteo de parpadeos es una medida segura? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo

11. ¿Siente usted que el nivel de la detección facial es precisa en su accionar? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo

ANEXO 06. Validez y confiabilidad de la encuesta del sistema de reconocimiento facial

VALIDEZ

Coeficiente de Validez de Contenido (CVC)

Ítem	Juez 01	Juez 02	Juez 03	Sx_1	Mx	CVC_i	P_{ei}	CVC_{tc}
Ítem 01	20	20	20	60	3	1	0.03703704	0.96296296
Ítem 02	18	20	20	58	2.9	0.966666667	0.03703704	0.92962963
Ítem 03	20	19	20	59	2.95	0.983333333	0.03703704	0.9462963
Ítem 04	19	18	19	56	2.8	0.933333333	0.03703704	0.8962963
Ítem 05	20	19	19	58	2.9	0.966666667	0.03703704	0.92962963
Ítem 06	19	20	20	59	2.95	0.983333333	0.03703704	0.9462963
Ítem 07	19	20	19	58	2.9	0.966666667	0.03703704	0.92962963
Ítem 08	20	20	20	60	3	1	0.03703704	0.96296296
Ítem 09	20	20	19	59	2.95	0.983333333	0.03703704	0.9462963
Ítem 10	20	19	20	59	2.95	0.983333333	0.03703704	0.9462963
Ítem 11	19	20	19	58	2.9	0.966666667	0.03703704	0.92962963
							CVC_t	0.93872054

CONFIABILIDAD

Alfa de Cronbach

Estadísticos total-elemento

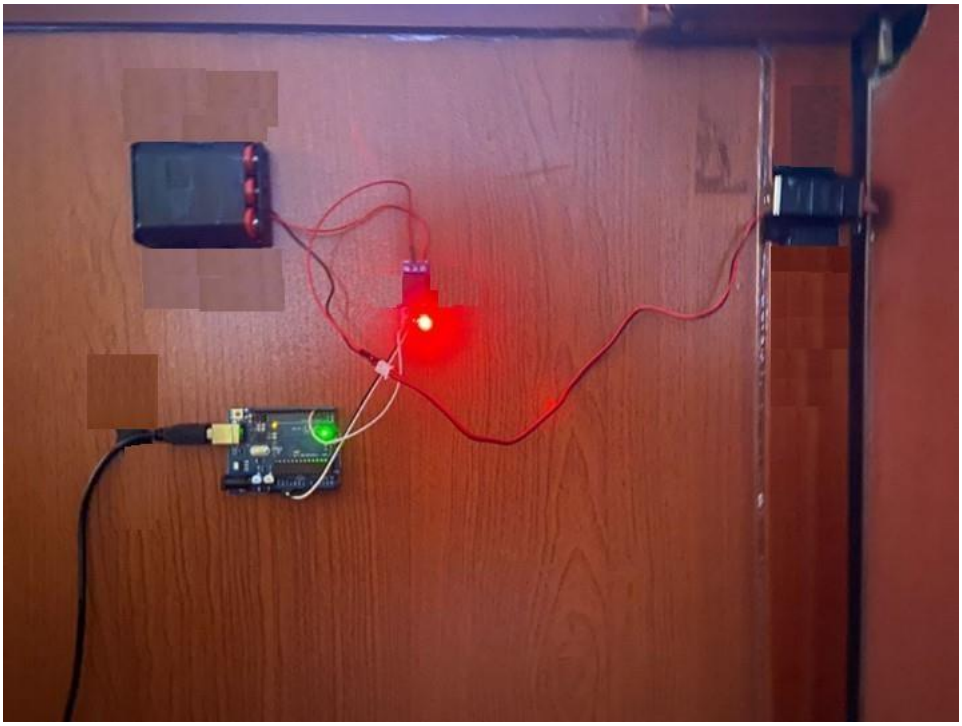
	Media de la escala si se elimina el elemento	Varianza de la escala si se elimina el elemento	Correlación elemento-total corregida	Alfa de Cronbach si se elimina el elemento
VAR00002	49,2000	2,886	,645	,766
VAR00003	49,2667	2,352	,648	,778
VAR00004	49,2000	2,886	,645	,766
VAR00005	49,1333	3,267	,480	,787
VAR00006	49,0667	3,781	,000	,812
VAR00007	49,1333	3,267	,480	,787
VAR00008	49,0667	3,781	,000	,812
VAR00009	49,2000	2,743	,784	,748
VAR00012	49,1333	3,410	,320	,801
VAR00011	49,1333	3,267	,480	,787
VAR00010	49,1333	3,410	,320	,801

Estadísticos de fiabilidad

Alfa de Cronbach	N de elementos
,804	11

ANEXO 07. Implementación del Sistemas de Reconocimiento Facial y Control de Acceso

Sistema de Control de Acceso



Cámara FDT-79001

Código del Software Arduino

arduino_03

```
int RELAY = 3;
int option;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(RELAY, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  digitalWrite(RELAY, HIGH);

  if (Serial.available() > 0)
  {
    option = Serial.read();
    if (option == 'p')
    {
      digitalWrite(RELAY, LOW);
      delay(5000);
      digitalWrite(RELAY, HIGH);
    }
    if (option == 'n')
    {
      digitalWrite(RELAY, LOW);
      delay(20000);
      digitalWrite(RELAY, HIGH);
    }

  }

}
```

Código del Software de Escritorio

```
def webcam(self):


    self.detector = dlib.get_frontal_face_detector()
    self.predictor = dlib.shape_predictor("./models/shape_predictor_68_face_landmarks.dat")

    self.counter = 0
    self.font = cv2.FONT_HERSHEY_PLAIN

    print(self.ids.btn_iniciar.state)


    if self.ids.btn_iniciar.state == "down":
        self.ids.btn_iniciar.text = "CÁMARA ENCENDIDA"
        self.clock_variable = Clock.schedule_interval(self.update, 1.0/33.0)
    elif self.ids.btn_iniciar.state == "normal":
        self.ids.btn_iniciar.text = "CÁMARA APAGADA"
        self.clock_variable.cancel()
```