



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**TITULO DEL TRABAJO DE TITULACIÓN
DETECCIÓN DE SIMILITUD FACIAL BASADO EN
DISTANCIAS EUCLIDIANAS Y DE DATOS ATÍPICOS PARA
DIFERENCIACIÓN DE GEMELOS MONOCIGÓTICOS**

AUTOR

MUÑOZ ALEJANDRO, LESLY STEPHANY

MODALIDAD: Proyecto UIC

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

Ing. Rosero Vásquez, Shendry Balmore, Msc.

Santa Elena, Ecuador

Año 2023



UPSE

**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TRIBUNAL DE SUSTENTACIÓN

**Ing. José Sánchez Aquino Mgt.
DIRECTOR DE LA CARRERA**

**Ing. Shendry Rosero Vasquez, Msc.
TUTOR**

**Ing. Walter Orozco Iguasnia. Mgt.
DOCENTE ESPECIALISTA**

**Ing. Mariorie Coronel S. Mgt.
DOCENTE GUÍA UIC**



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Muñoz Alejandro Lesly Stephany, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información,

La Libertad, a los 08 días del mes de agosto del año 2023

TUTOR

Ing. Shendry Balmore Rosero Vásquez, Msc.



UPSE
UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD
Yo, MUÑOZ ALEJANDRO LESLY STEPHANY

DECLARO QUE:

El trabajo de Titulación, “Detección De Similitud Facial Basado En Distancias Euclidianas Y De Datos Atípicos Para Diferenciación De Gemelos Monocigóticos” previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 08 días del mes de agosto del año 2023

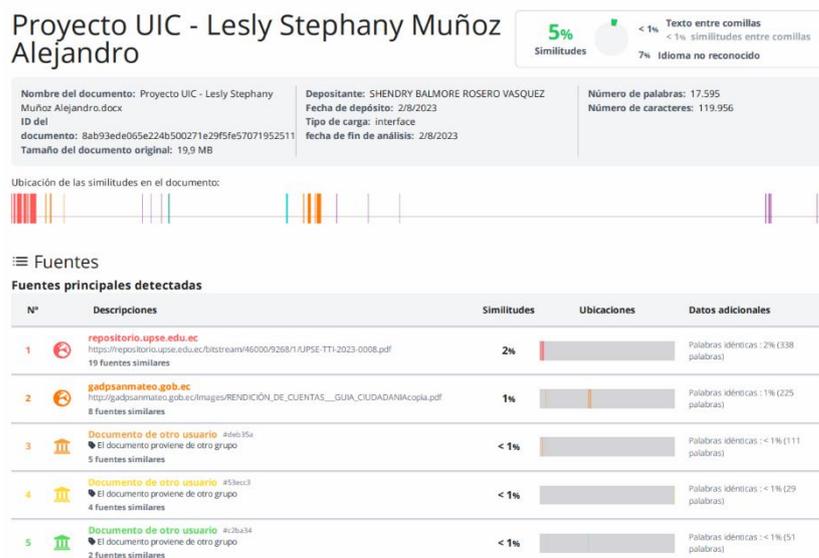
EL AUTOR

Lesly Stephany Muñoz Alejandro

CERTIFICACIÓN DE ANTIPLAGIO

En calidad de tutor del trabajo de titulación denominado “**DETECCIÓN DE SIMILITUD FACIAL BASADO EN DISTANCIAS EUCLIDIANAS Y DE DATOS ATÍPICOS PARA DIFERENCIACIÓN DE GEMELOS MONOCIGÓTICOS.**” elaborado por la estudiante, **Lesly Stephany Muñoz Alejandro**, estudiante de la Carrera de Tecnologías de la Información de la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, previo a la obtención del título de Ingeniera en Tecnologías de la Información y la Comunicación, me permito declarar que una vez analizado en el sistema anti plagio “COMPILATIO MAGISTER”, y luego de haber cumplido los requerimientos exigidos de valoración el presente proyecto ejecutado, se encuentra con 5% de similitud con respecto a otros documentos, por consiguiente se procede a emitir el presente informe.

Adjunto reporte de similitud.



Atentamente,


Ing. Shendry Rosero V. Ms.CC

TUTOR



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
AUTORIZACIÓN**

Yo, MUÑOZ ALEJANDRO LESLY STEPHANY

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 08 días del mes de agosto del año 2023

EL AUTOR

A handwritten signature in blue ink, appearing to read "Lesly Stephany Muñoz Alejandro", is written over a horizontal line.

LESLY STEPHANY MUÑOZ ALEJANDRO

ÍNDICE GENERAL	
TITULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
ÍNDICE GENERAL	VII
ÍNDICE DE TABLAS	X
ÍNDICE DE FIGURA	XI
RESUMEN	XII
ABSTRACT	XIII
INTRODUCCIÓN	1
CAPÍTULO I	2
1. Fundamentación	2
1.1. Antecedentes	2
1.2. Descripción Del Proyecto	5
1.3. Objetivos	10
1.3.1. Objetivo general	10
1.3.2. Objetivos específicos	10
1.4. Justificación	10
1.5. Alcance del proyecto	12
1.6. Metodología	13
1.6.1. Metodología de investigación	13
1.6.2. Técnicas de Investigación	13
1.7. Metodología de desarrollo	14
CAPÍTULO II	16
2.1. Marco Contextual	16
2.1.1. Provincia de Santa Elena	16
2.1.2. Misión	16
2.1.3. Visión	17

2.1.4. LEY ORGÁNICA DE TRANSPARENCIA Y ACCESO A LA INFORMACIÓN PÚBLICA	17
2.1.4. LEY ORGÁNICA DE PARTICIPACIÓN CIUDADANA Y LEY ORGÁNICA DEL CONSEJO DE PARTICIPACIÓN CIUDADANA Y CONTROL SOCIAL	18
2.2. MARCO CONCEPTUAL	19
2.2.1. Distancias Euclidianas	19
2.2.2. Datos Atípicos	19
2.2.3. Diferencia en detección rostro y reconocimiento facial	20
2.2.4. Entrenamiento del reconocedor	20
2.2.4.1. LBPH (Histograma de patrón binario local)	20
2.2.4.2. MTCNN (Redes Neuronales Convolucionales Multitarea)	20
2.2.4.3. Tensorflow-gpu	21
2.2.4.4. Keras	21
2.2.4. OpenCV 4.2	21
2.2.5. Python 3.6	22
2.2.8. MSVC (Microsoft Visual Studio):	22
2.2.9. PyQt5	22
2.2.10. SQLite3	22
2.2.11. ResNet	23
2.2.12. Moblenet	23
2.2.13. Google Colab	23
2.2.14. Anaconda	24
2.2.15. Compute Unified Architecture (CUDA)	24
2.2.16. Entornos de desarrollo	25
2.2.17. Entorno virtual de Python	25
2.2.18. Detección de objetos usando OpenCV	25
2.2.19. Componentes de un sistema de visión por computador	26
2.3. Marco Teórico	26
2.3.1. Identificación de rostros familiares y desconocidos a partir de características internas y externas	26
2.3.2. Psicometría, perfiles y sesgos. El caso del reconocimiento facial	27
2.3.3 Sistema de reconocimiento facial con Deep Learning	27
CAPÍTULO III	28
3. Propuesta	28

3.1. Requerimiento	28
3.1.1. Requerimientos funcionales	28
3.2. Componente de la Propuesta	31
3.2.1. Arquitectura del Sistema	31
3.2.1.1. Algoritmo basado en el modelo LBPH	31
3.2.1.2. Algoritmo basado en el MTCNN (Multi-task Cascaded Convolutional Networks).	35
3.2.1.3. Algoritmo basado en TensorFlow y Keras	37
3.2.1.4. Extracción de características Faciales	40
3.2.1.5. Cálculo de Distancias Euclidianas:	42
3.2.1.6. Identificación de Datos Atípicos:	42
3.3. Resultados	43
3.3.1. Pruebas	43
3.3.1.1. Prueba de tiempo de respuesta	43
3.3.1.2. Prueba de tasa de aciertos	48
Conclusiones	53
Recomendaciones	54
Bibliografía	56
ANEXOS	60

ÍNDICE DE TABLAS

Tabla 1.	Resultados de identificación de gemelos.	5
Tabla 2.	Artículos de Ley Orgánica De Transparencia Y Acceso A La Información Pública [21] .	17
Tabla 3.	Artículos De Participación Ciudadana y Ley Orgánica del CPCCS [21].	19
Tabla 4.	Requerimientos funcionales	30
Tabla 5.	Prueba de tiempo de respuesta gemelo 1 y 2.	45
Tabla 6.	Prueba de tiempo de respuesta gemelo 1.	46
Tabla 7.	Prueba de tiempo de respuesta gemelo 2.	47
Tabla 8.	Prueba de tasa de aciertos gemelo 1 y 2.	49
Tabla 9.	Prueba de tasa de aciertos gemelo 1.	51
Tabla 10.	Prueba de tasa de aciertos gemelo 2.	52

ÍNDICE DE FIGURA

Figura 1.	Tasa de aceptación falsa de una expresión natural-natural y expresión natural-sonriente tomado de Afaneh, Noroozi, & Toygar.	3
Figura 2.	Tasa de aceptación falsa para iluminación controlada y una iluminación no controlada tomado de Afaneh, Noroozi, & Toygar.	3
Figura 3.	Característica útil para distinguir la imagen de hermanos gemelos idénticos tomado de la revista Internacional de investigación científica y de ingeniería.	11
Figura 4.	Fases de la metodología cascada	14
Figura 5.	Ubicación de la Provincia de Santa Elena.	16
Figura 6.	Capturando rostro.	32
Figura 7.	Escala gris Mile	32
Figura 8.	Escala gris Stephany	33
Figura 9.	Etiqueta 0 y 1	33
Figura 10.	Detención con el modelo LBPH	35
Figura 11.	Dataset MySQL	36
Figura 12.	Detención con la biblioteca MTCNN (Multi-task Cascaded Convolutional Networks).	37
Figura 13.	Extracción de rostro.	39
Figura 14.	Detención con la librería TensorFlow	40
Figura 15.	Malla en la posición tridimensional.	41
Figura 16.	Malla de puntos faciales.	41
Figura 17.	Prueba del algoritmo de reconocimiento facial basado en LBPH	61
Figura 18.	Prueba del algoritmo de reconocimiento facial basado en MTCNN	62
Figura 19.	Prueba del algoritmo de reconocimiento facial basado en TensorFlow con Keras	64
Figura 20.	Certificado Antiplagió.	65

RESUMEN

El presente estudio se enfocó en investigar un sistema avanzado de detección de similitud facial para diferenciar gemelos monocigóticos de manera precisa. Se implementaron técnicas basadas en el cálculo de distancias euclidianas y la identificación de datos atípicos, empleando algoritmos de reconocimiento facial como TensorFlow con Keras, LBPH y MTCNN. Mediante pruebas exhaustivas bajo diversas condiciones de iluminación y distancias, se evaluaron el tiempo de respuesta y la tasa de aciertos de cada algoritmo. Los resultados revelaron que MTCNN obtuvo la tasa de aciertos más alta con un promedio del 97%, seguido por TensorFlow con Keras con un promedio del 94%, mientras que LBPH alcanzó un promedio del 78%. Los hallazgos sugieren que el sistema investigado es una solución prometedora para la detección precisa de gemelos monocigóticos, con posibles aplicaciones en campos como identificación, seguridad y vigilancia.

Palabras claves: Reconocimiento Facial, Distancias Euclidianas, Datos Atípicos, Gemelos.

ABSTRACT

The present study focused on investigating an advanced facial similarity detection system to accurately differentiate monozygotic twins. Techniques based on the calculation of Euclidean distances and the identification of atypical data were implemented, using facial recognition algorithms such as TensorFlow with Keras, LBPH and MTCNN. Through extensive testing under various lighting conditions and distances, the response time and hit rate of each algorithm were evaluated. The results revealed that MTCNN had the highest hit rate with an average of 97%, followed by TensorFlow with Keras with an average of 94%, while LBPH achieved an average of 78%. The results demonstrated that the investigated system is a promising solution for the accurate detection of monozygotic twins, with possible applications in fields such as identification, security, and surveillance.

Keywords: Facial Recognition, Euclidean Distances, Outliers, Twins.

INTRODUCCIÓN

En el contexto actual, la detección de similitud facial basada en distancias euclidianas y la identificación de datos atípicos para diferenciar gemelos monocigóticos se presenta como un área de investigación en el campo del reconocimiento facial. Es un campo tecnológico en constante crecimiento y desempeña un papel cada vez más relevante en nuestra vida diaria. Estos sistemas se utilizan en diversos establecimientos como aeropuertos y centros comerciales. El reconocimiento facial ofrece ventajas significativas, como la prevención de casos de robo de identidad, una mayor precisión en la identificación de personas y la eliminación de posibles manipulaciones por parte de los usuarios, entre otros beneficios.

En los últimos años, el desarrollo de sistemas de reconocimiento facial ha experimentado un notable aumento debido a su aplicación en áreas de gran relevancia, como la seguridad informática, pública, económica, asistencia laboral y el control y monitoreo ambiental, entre otros. Estos sistemas han sido implementados con el propósito de abordar problemas como la suplantación de identidad, que ha aumentado considerablemente incluso durante la pandemia. Sin embargo, a medida que la tecnología avanza, el reconocimiento facial se enfrenta a desafíos significativos, como las oclusiones que dificultan la precisión en la identificación y los cambios en la iluminación que pueden alterar drásticamente los rasgos faciales.

El aumento de sistemas biométricos y reconocimiento facial ha revolucionado la identificación y acceso a dispositivos y servicios. Sin embargo, la similitud física extrema de gemelos monocigóticos representa un desafío para el reconocimiento facial. La identificación errónea puede ocurrir debido a las características faciales prácticamente idénticas de estos gemelos. Se están empleando estrategias como el uso de características biométricas adicionales y técnicas avanzadas de aprendizaje automático para superar este obstáculo. Aun así, el reconocimiento facial sigue siendo un campo en desarrollo, y en aplicaciones críticas de seguridad, pueden ser necesarios métodos de autenticación adicionales para garantizar la precisión y confiabilidad del sistema.

En esta investigación, abordamos el desafío del reconocimiento facial en gemelos monocigóticos, cuya extrema similitud física dificulta su diferenciación precisa. Buscamos superar estas limitaciones mediante el uso de enfoques de aprendizaje profundo, conjuntos de datos más extensos y la incorporación de características

biométricas adicionales para lograr una identificación más confiable y robusta en aplicaciones de reconocimiento facial.

se explorarán enfoques que analicen la estructura de código en algoritmos de reconocimiento facial para evaluar el tiempo de respuesta y mejorar la eficiencia del sistema. Además, se validará el desempeño del algoritmo mediante la implementación de una técnica de extracción de características faciales para garantizar su precisión en la identificación de diferencias sutiles entre gemelos idénticos.

La evaluación comparativa de algoritmos de reconocimiento facial basados en TensorFlow con Keras, LBPH (Local Binary Pattern Histogram) y MTCNN (Multi-task Cascaded Convolutional Networks) se realizará para seleccionar el enfoque más adecuado, considerando su capacidad de detectar y manejar datos atípicos, lo cual es esencial en el contexto de gemelos monocigóticos. Con esta investigación, se busca avanzar en la mejora de la identificación biométrica y contribuir al desarrollo de sistemas de reconocimiento facial más precisos y confiables en aplicaciones que involucren la diferenciación de gemelos idénticos.

CAPÍTULO I

1. Fundamentación

1.1. Antecedentes

Durante varios años, se ha llevado a cabo un constante proceso de mejora del algoritmo de reconocimiento facial con el objetivo de poder distinguir entre caras que presentan una gran similitud, como es el caso de los gemelos idénticos. Esta dificultad radica en las sutiles diferencias existentes en los pequeños rasgos faciales que los caracterizan. Por consiguiente, resulta altamente inapropiado intentar identificar a personas idénticas mediante la extracción de características faciales comunes. Incluso hasta el momento actual, no se ha logrado desarrollar un método capaz de reconocer a gemelos similares basándose en su ADN [1].

La igualdad entre individuos se basa en diversas características biométricas, geométricas y fotométricas. En el ámbito biométrico, las huellas dactilares pueden llegar a ser idénticas en algunos casos. Desde una perspectiva geométrica, el rostro de los gemelos rara vez presenta diferencias significativas, lo que puede generar confusión entre las personas. Sin embargo, en cuanto a las características fotométricas, los métodos de

reconocimiento enfrentan grandes desafíos al intentar distinguir a los gemelos, ya que existen muy pocas posibilidades de lograr un éxito en dicho reconocimiento [2].

Los algoritmos desarrollados para el reconocimiento facial en gemelos idénticos presentan tasas de errores específicas en diferentes situaciones. Según los resultados obtenidos, se ha determinado que las tasas de error son del 2.07% para expresiones faciales naturales, del 0.0% para expresiones sonrientes y del 2.2% en condiciones de iluminación controlada. Estas tasas se comparan con el mejor algoritmo de última generación, el cual presenta tasas de error del 4.5%, 4.2% y 4.7% respectivamente bajo las mismas condiciones. En la figura 1 y 2 se muestran los porcentajes de reconocimiento facial para gemelos idénticos y no gemelos, considerando la variación en las expresiones faciales y en la iluminación. [3].

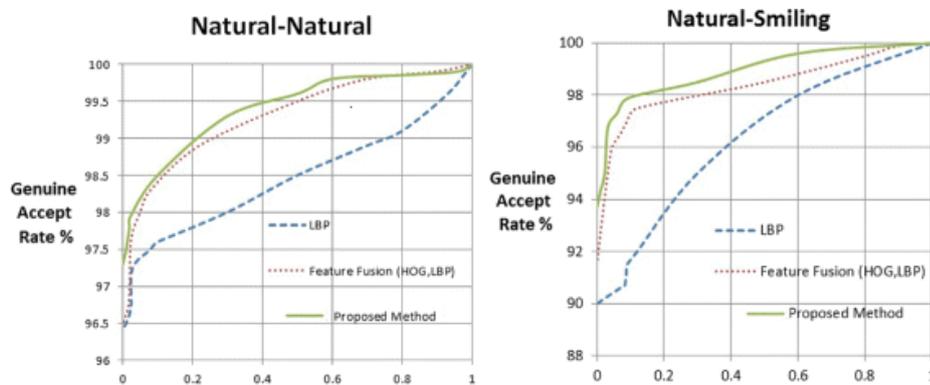


Figura 1. Tasa de aceptación falsa de una expresión natural-natural y expresión natural-sonriente tomado de Afaneh, Noroozi, & Toygar.

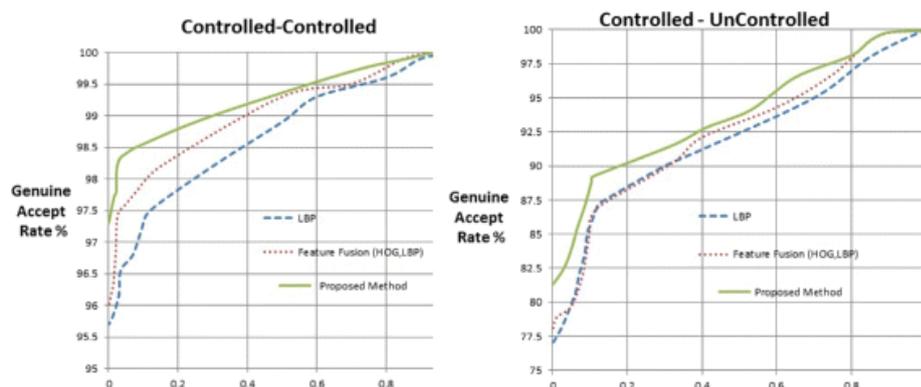


Figura 2. Tasa de aceptación falsa para iluminación controlada y una iluminación no controlada tomado de Afaneh, Noroozi, & Toygar.

La suplantación de identidad mediante la utilización de un hermano gemelo con el fin de acceder y manipular información confidencial dentro de una institución o dispositivo es uno de los principales desafíos en este ámbito. A simple vista, una persona puede

identificar pequeños rasgos que diferencian a los gemelos, pero al utilizar un algoritmo de reconocimiento facial, la tasa de éxito puede ser limitada. El sistema de reconocimiento facial presente en el SAMSUNG Galaxy A32 no puede considerarse completamente confiable, ya que, si registramos el rostro de un gemelo sin tener el rostro del otro, ambos podrían desbloquear el teléfono sin inconvenientes.

En el campo del procesamiento de imágenes, la identificación precisa entre gemelos idénticos representa un desafío considerable debido al alto grado de similitud en su apariencia facial general. La mayoría de los algoritmos desarrollados para el reconocimiento facial presentan un rendimiento deficiente al intentar determinar si se trata del hermano gemelo correcto. Por lo tanto, se requiere llevar a cabo un estudio exhaustivo sobre la viabilidad y utilidad de las características faciales como firmas biométricas para abordar esta problemática de manera más efectiva.

En el documento titulado "UN SISTEMA DISTINTIVO DE RECONOCIMIENTO FACIAL BASADO EN PUNTOS DE REFERENCIA PARA GEMELOS IDÉNTICOS MEDIANTE LA EXTRACCIÓN DE CARACTERÍSTICAS PONDERADAS NOVEDOSAS", se propone la utilización de un algoritmo genético modificado para optimizar los pesos de las características relevantes. Estas características ponderadas son luego clasificadas utilizando un clasificador de máquinas de vectores de soporte. En el marco de este experimento, se emplearon un total de cuatrocientas cuarenta imágenes faciales, tanto de gemelos idénticos como de no gemelos, con el fin de evaluar la efectividad del sistema propuesto [4].

El proyecto titulado "RECONOCIMIENTO FACIAL BASADO EN PUNTOS CARACTERÍSTICOS DE LA CARA EN ENTORNOS NO CONTROLADOS" se centra en el estudio, implementación y evaluación de un sistema automático para la detección y corrección de puntos característicos faciales que han sido mal marcados. Estos puntos se obtienen a través de un sistema comercial automático. Además, se emplea una base de datos que simula entornos tanto controlados como no controlados, y esta base de datos está disponible de forma gratuita para la comunidad científica. El objetivo principal del proyecto es mejorar la precisión y robustez del reconocimiento facial en entornos no controlados mediante la optimización de la detección y corrección de los puntos característicos faciales [5].

El proyecto titulado "AUTENTICACIÓN POR RECONOCIMIENTO FACIAL PARA APLICACIONES WEB, UTILIZANDO SOFTWARE LIBRE", se llevan a cabo pruebas de identificación de gemelos con el objetivo de determinar qué método podría distinguir de manera más precisa entre rostros similares en caso de que se registren gemelos como usuarios (dos usuarios con características faciales de rasgos similares). Para este propósito, se realizaron cien muestras de autenticación biométrica facial utilizando una resolución de webcam de 320x240 y se identificó al usuario llamado Carlos. Los resultados de estas pruebas se presentan en la tabla 1. El propósito de este proyecto es evaluar la efectividad de diferentes métodos de reconocimiento facial en escenarios donde la distinción entre gemelos es un desafío [6].

Método	Aciertos	Rechazos	Confusión
Eigenfaces	96%	3%	1%
Fisherfaces	13%	24%	63%

Tabla 1. Resultados de identificación de gemelos.

En este análisis se examinará el funcionamiento de cada algoritmo con el fin de determinar cuál de ellos es el más eficiente en la tarea de distinguir si una foto pertenece al gemelo número uno o si se trata de la misma persona. Además, se buscará identificar los pesos específicos que deben ser modificados para mejorar la precisión y seguridad del sistema. El objetivo es evaluar y comparar los algoritmos en términos de su capacidad de diferenciación y establecer las modificaciones necesarias para optimizar su desempeño en este contexto específico.

1.2. Descripción Del Proyecto

En la actualidad, los algoritmos de reconocimiento facial presentan una precisión del 100% en la identificación de personas que no son idénticas. Sin embargo, cuando se trata de individuos con rostros muy similares y diferencias mínimas en los rasgos faciales, la precisión de estos algoritmos se reduce más de lo esperado. Con el objetivo de evaluar y comparar diferentes algoritmos, se realizará un cuadro comparativo que permita analizar sus fortalezas y limitaciones en este contexto.

Un proyecto de investigación relevante en este campo es el denominado "RECONOCIMIENTO FACIAL BASADO EN PUNTOS CARACTERÍSTICOS DE LA CARA EN ENTORNOS NO CONTROLADOS". Esta metodología se compone de cinco fases, las cuales se han adaptado para esta investigación con el fin de evaluar el tiempo de respuesta y recomendar a los usuarios el mejor enfoque en términos de eficiencia y resultados. Estas fases proporcionarán una estructura sólida para el desarrollo del proyecto y ayudarán a evaluar la idoneidad de los algoritmos en entornos no controlados, donde la diferenciación de gemelos es un desafío [5].

En el desarrollo de esta investigación, se utilizarán las siguientes herramientas:

Pip (Versión 3.0): La herramienta en cuestión es un script desarrollado en Python que tiene como objetivo simplificar el proceso de descarga e instalación de paquetes del lenguaje Python desde el Python Package Index (PyPI). El script automatiza las tareas necesarias para buscar, descargar y configurar los paquetes requeridos, lo que ahorra tiempo y reduce posibles errores en la instalación manual de los mismos [7].

OpenCV (Open Source Computer Vision Library) (Versión 3.4 o más reciente): Es una biblioteca de software de visión por computadora y aprendizaje automático de código abierto. Fue desarrollada con el propósito de ofrecer una infraestructura compartida para aplicaciones relacionadas con la visión por computadora y para acelerar la implementación de tecnologías de percepción automática en productos comerciales. Al estar bajo la licencia Apache 2, OpenCV permite a las empresas utilizar y modificar el código fuente de manera flexible y adaptarlo a sus necesidades específicas [8].

Python (Versión 3.6 o más reciente): Es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con una semántica dinámica. Su diseño se centra en la legibilidad del código y en la facilidad de uso, lo que lo hace especialmente atractivo para el desarrollo rápido de aplicaciones y su utilidad como lenguaje de scripting o para la integración de componentes existentes [9].

Tensorflow-gpu (Versión 1.5.0 o posterior): Es un framework desarrollado y mantenido por Google que brinda una plataforma eficiente para ejecutar operaciones matemáticas en CPU (Central Processing Unit) o GPU (Graphics Processing Unit). Está diseñado

específicamente para el procesamiento de datos y tareas de aprendizaje automático (machine learning). [10].

Keras 2.1.3: Es una interfaz de programación de aplicaciones (API) diseñada específicamente para seres humanos, priorizando la usabilidad y facilitando el desarrollo de aplicaciones de aprendizaje automático. A diferencia de otras APIs más complejas y orientadas a máquinas, Keras se enfoca en seguir las mejores prácticas para reducir la carga cognitiva del usuario [11].

MSVC (Microsoft Visual Studio): El entorno de desarrollo integrado (IDE) denominado Microsoft Visual C++ (MSVC) es una herramienta proporcionada por Microsoft para el desarrollo de aplicaciones en los lenguajes de programación C, C++ y C++/CLI. MSVC es un software propietario que inicialmente se lanzó como un producto independiente, pero más tarde se integró como parte de Visual Studio, una suite de herramientas de desarrollo de Microsoft. [12].

El presente proyecto tiene como objetivo brindar al usuario información relevante sobre el manejo del algoritmo de reconocimiento facial. Se detallarán las fases del algoritmo, explicando en cada una los procesos, funcionamiento y estructura. Asimismo, se realizará una comparativa de los diferentes algoritmos investigados, evaluando características y rendimiento. Esto permitirá al usuario comprender y tomar decisiones fundamentadas sobre el uso del reconocimiento facial, teniendo en cuenta las mejores prácticas y opciones disponibles en el campo [5].

El presente documento se centrará en abordar las siguientes fases a realizar con el fin de alcanzar los objetivos establecidos en el proyecto:

Fase 1. Generar un dataset

- Identificación de fuentes de datos adecuadas para recopilar imágenes de gemelos monocigóticos y no monocigóticos, considerando la diversidad en condiciones de iluminación, expresiones faciales y cambios físicos.
- Adquisición y almacenamiento de imágenes de gemelos, asegurando la integridad y calidad de la información asociada.
- Obtener y almacenar las imágenes de gemelos, asegurando su calidad y la información asociada.

- Realización de un análisis exploratorio de los datos para comprender la distribución y características del dataset, como la variabilidad de las características faciales.

Fase 2. Preprocesamiento de imagen

- Aplicación de técnicas de normalización de imágenes para eliminar variaciones en el brillo, contraste y color, asegurando la consistencia en la apariencia visual de las imágenes.
- Implementación de correcciones geométricas y de deformación para alinear las características faciales en todas las imágenes, asegurando una alineación precisa y consistente.
- Aplicación de técnicas de filtrado y eliminación de ruido para mejorar la calidad visual de las imágenes y eliminar artefactos que puedan afectar el procesamiento posterior.
- División adecuada del dataset en conjuntos de entrenamiento y prueba, asegurando una distribución equilibrada de las clases para evitar sesgos en el entrenamiento y la evaluación del modelo.

Fase 3. Procesamiento de imágenes

- Implementación de algoritmos de reconocimiento facial basados en técnicas de aprendizaje automático, como redes neuronales convolucionales (CNN), para identificar y localizar automáticamente las regiones faciales de interés en las imágenes.
- Extracción de características faciales relevantes de las regiones identificadas, utilizando técnicas de procesamiento de imágenes y aprendizaje automático para capturar información discriminativa, como texturas, patrones o características geométricas.
- Almacenamiento eficiente de las características extraídas para su posterior uso en el cálculo de distancias euclidianas y en la detección de similitud facial.
- Validación y optimización de los algoritmos de procesamiento de imágenes mediante técnicas de evaluación y ajuste de hiperparámetros para mejorar la precisión y el rendimiento.

Fase 4. Extracción de características

- Aplicación de técnicas de procesamiento de imágenes y aprendizaje automático para extraer características específicas de las imágenes faciales, como descriptores de textura, histogramas de colores o características de forma.
- Utilización de algoritmos de extracción de características, como filtros espaciales o transformaciones de dominio, para obtener representaciones compactas y discriminativas de las características faciales.
- Realización de un análisis dimensional para reducir la dimensionalidad de las características extraídas y eliminar redundancias, utilizando métodos como Análisis de Componentes Principales (PCA) o selección de características basada en criterios estadísticos.
- Evaluación de la calidad y discriminabilidad de las características extraídas mediante técnicas de evaluación, como validación cruzada o medidas de separabilidad entre clases.

Fase 5. Resultado

- Cálculo de las distancias euclidianas entre las características faciales de las imágenes del dataset, utilizando técnicas de comparación y emparejamiento.
- Establecimiento de un umbral de similitud para diferenciar entre gemelos monocigóticos y no monocigóticos, basado en el análisis de las distancias calculadas.
- Evaluación del desempeño del sistema mediante métricas de evaluación para medir la capacidad de detección de similitud facial y diferenciación de gemelos.
- Realización de análisis estadísticos y generación de gráficos para visualizar los resultados.
- Presentación de conclusiones fundamentadas en los resultados obtenidos, destacando las fortalezas y limitaciones del enfoque propuesto, y sugerencias para mejoras futuras, como la incorporación de técnicas más avanzadas de aprendizaje automático o la exploración de enfoques multimodales.

El presente proyecto se enfoca en el campo de la inteligencia computacional, específicamente en la investigación de cuatro algoritmos de reconocimiento facial. El objetivo es identificar las posibles fallas en la capacidad de estos algoritmos para

distinguir al hermano intruso en diferentes condiciones, como expresión natural, sonriente, iluminación controlada e iluminación no controlada. Se lleva a cabo un análisis exhaustivo de cada parte del código de los algoritmos, con el fin de determinar cuál de ellos presenta una mayor precisión al momento de procesar y reconocer las imágenes en los escenarios mencionados [13].

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un sistema de detección de similitud facial basado en el cálculo de distancias euclidianas y la identificación de datos atípicos, para la diferenciación efectiva de gemelos monocigóticos.

1.3.2. Objetivos específicos

- Analizar la estructura de código en algoritmos de reconocimiento facial para la evaluación de tiempo de respuesta.
- Validar el desempeño del algoritmo mediante la implementación de una técnica de extracción de características faciales para el manejo del algoritmo.
- Evaluar la tasa de aciertos de algoritmos de reconocimiento facial basado en TensorFlow con Keras, LBPH (Local Binary Pattern Histogram) y MTCNN (Multi-task Cascaded Convolutional Networks) en la detección de anomalías en rostros humanos.

1.4. Justificación

En el ámbito de la evolución tecnológica, se han desarrollado sistemas inteligentes y automáticos que abarcan tanto el entorno virtual como el físico. Entre los avances más destacados se encuentran los sistemas basados en la simulación de capacidades humanas, particularmente en el campo de la visión [14]. En este contexto, se aborda el desafío de distinguir gemelos idénticos mediante el análisis de imágenes faciales. El objetivo es desarrollar técnicas y algoritmos que permitan una identificación precisa y confiable entre individuos que comparten una apariencia facial extremadamente similar [3].

La revista "Internacional de investigación científica y de ingeniería" llevó a cabo una encuesta enfocada en el reconocimiento facial de gemelos idénticos, con el objetivo de determinar qué características se consideraron más relevantes para las respuestas

correctas e incorrectas. Entre las opciones proporcionadas, los participantes seleccionaron con mayor frecuencia las características de lunares, cicatrices y pecas, considerándolas más importantes que otros tipos de rasgos faciales. Estos resultados se presentan en la Figura 4 [15].

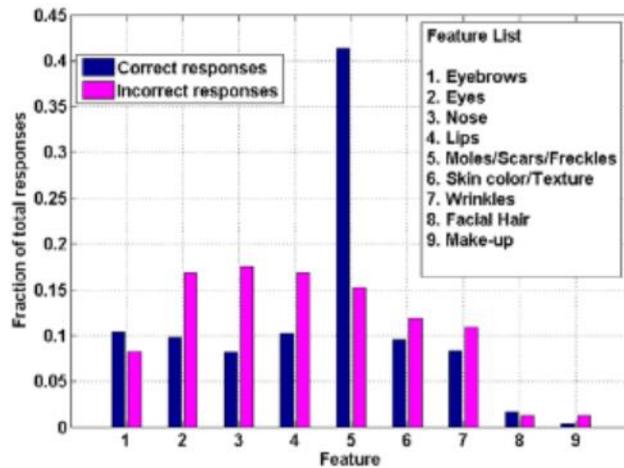


Figura 3. Característica útil para distinguir la imagen de hermanos gemelos idénticos tomado de la revista Internacional de investigación científica y de ingeniería.

El reconocimiento facial nos proporciona una forma precisa de distinguir entre gemelos. Para lograr esto, se realiza una extracción de puntos característicos presentes en el rostro, seguido de un proceso de tres etapas en el que la imagen es sometida a un análisis de reconocimiento. Si la imagen cumple con los criterios establecidos, se procede a verificar el correcto posicionamiento de los marcos de los ojos. Este proceso implica la modificación de la imagen original en escala de grises y el cálculo de los bordes utilizando el método Canny, lo cual permite determinar las coordenadas del centro de las pupilas.

La distancia entre las pupilas y la boca es un factor relevante, ya que se registra en un vector que nos proporcionará información sobre si el rasgo de los labios está correcta o incorrectamente marcado. Por otro lado, los ángulos formados por las cejas con respecto a los ojos nos permitirán calcular el ángulo entre el centro de ambas cejas y la línea que une el centro de las pupilas. Estos ángulos son de gran utilidad en el análisis de características faciales.

Con el fin de lograr una mayor simetría facial, se calculan las distancias verticales y horizontales entre los ojos, cejas, nariz, boca y barbilla. Esto permite determinar el lado interior y exterior del rostro, así como el punto central, lado izquierdo y lado derecho. Estos cálculos facilitan la visualización de la estructura facial. Sin embargo, en el caso de

imágenes normalizadas, la distancia horizontal entre las pupilas no aporta información relevante y, por lo tanto, no se considera en este análisis.

El tema propuesto se encuentra en línea con los objetivos del Plan Nacional de Desarrollo "Creación de Oportunidades", en particular, dentro del siguiente eje temático:

Eje 1.- Seguridad Integral.

Objetivo 9.- Garantizar la seguridad ciudadana, orden público y gestión de riesgos [16].

Política 9.1.- Fortalecer la protección interna, el mantenimiento y control del orden público, que permita prevenir y erradicar los delitos conexos y la violencia en todas sus formas, en convivencia con la ciudadanía en el territorio nacional y áreas jurisdiccionales. [16].

1.5. Alcance del proyecto

En este proyecto, se implementarán algoritmos de reconocimiento facial a través del análisis de su estructura de código, con el objetivo de verificar si cumplen con cada fase establecida. De esta manera, se evaluará su rendimiento y se determinará si se recomienda su uso, considerando la tasa de éxito en la identificación de gemelos intrusos en diferentes escenarios, incluyendo expresiones sonrientes, naturales, con iluminación controlada y con iluminación no controlada.

El presente proyecto se dividirá en las siguientes fases técnicas:

- Fase 1. Generar un dataset
- Fase 2. Preprocesamiento de imagen
- Fase 3. Procesamiento de imágenes
- Fase 4. Extracción de características
- Fase 5. Resultado

En este proyecto de detección de similitud facial basado en distancias euclidianas y datos atípicos para diferenciación de gemelos monocigóticos, se realizan diversas fases. En la Fase 1, se genera un dataset anotado con características faciales relevantes. La Fase 2 se enfoca en el preprocesamiento de imágenes, aplicando técnicas de normalización, corrección geométrica y eliminación de ruido. En la Fase 3, se emplean algoritmos de reconocimiento facial basados en aprendizaje automático para extraer características faciales.

La Fase 4 se centra en la extracción de características específicas y el análisis dimensional para reducir la dimensionalidad. Por último, en la Fase 5, se calculan las distancias euclidianas y se establece un umbral para diferenciar gemelos monocigóticos de no monocigóticos, evaluando el desempeño del sistema y presentando conclusiones basadas en los resultados obtenidos.

1.6. Metodología

1.6.1. Metodología de investigación

La presente investigación exploratoria tiene como objetivo obtener un enfoque universal del problema [17]. En este estudio, se llevará a cabo un análisis exhaustivo de cada algoritmo de reconocimiento facial para examinar su estructura y su funcionamiento en cada etapa del proceso de distinguir entre gemelos registrados, dado que existen escasos estudios que abordan de manera definida las características de los rasgos faciales idénticos. En consecuencia, se ajustará esta investigación para obtener detalles específicos sobre el tema, utilizando fuentes bibliográficas técnicas, encuestas y observaciones para lograr una extracción de características más detallada.

La presente investigación diagnóstica se realizará mediante encuestas dirigidas a los peninsulares con el objetivo de obtener un contexto amplio sobre las características que diferencian a los gemelos. Esta información será fundamental para otorgar mayor validez a los rasgos distintivos del rostro, los cuales son utilizados para determinar si una persona es la correcta o no en el proceso de reconocimiento facial.

Se llevará a cabo un análisis de la tasa de éxito en el reconocimiento facial de gemelos monocigóticos, tanto en su forma original sin la implementación de fases adicionales, como en su versión modificada con las fases que serán implementadas. El objetivo de este estudio es evaluar y comparar el rendimiento del algoritmo de reconocimiento facial en ambos casos, con el fin de determinar el impacto de las fases adicionales en la mejora de la precisión y la distinción entre los gemelos.

1.6.2. Técnicas de Investigación

A continuación, se detallarán las técnicas de recolección de información utilizadas.

➤ **Técnica:**

Encuestas, observaciones y fuentes bibliográficas

➤ **Instrumento:**

Se aplicarán cuestionarios a los peninsulares con el objetivo de identificar las características distintivas entre dos personas idénticas. Además, se llevarán a cabo observaciones para evaluar el tiempo de respuesta y el número de veces que el algoritmo permitió el acceso al gemelo intruso en situaciones sonrientes, normales, con iluminación controlada y no controlada. Las fuentes bibliográficas proporcionarán información relevante relacionada con la problemática abordada en este proyecto.

➤ **Población:**

La población objetivo de estudio estará constituida por individuos idénticos.

1.7. Metodología de desarrollo

Para el presente proyecto se utilizará la metodología en cascada, también conocida como modelo de desarrollo secuencial. Esta metodología se caracteriza por seguir un enfoque lineal y secuencial, en el cual cada fase del proyecto se lleva a cabo de manera secuencial, sin superposición entre ellas. [18]

El presente proyecto abarcara cinco fases que se describen en la figura 4.



Figura 4. Fases de la metodología cascada

➤ **Fase 1: Generar un dataset:**

Se seleccionan fuentes de datos adecuadas para recopilar imágenes de gemelos monocigóticos y no monocigóticos, se establecen criterios de selección y se adquieren las imágenes junto con la anotación precisa de las características faciales relevantes. Se realiza un análisis exploratorio del dataset para comprender su distribución y variabilidad.

➤ **Fase 2: Preprocesamiento de imagen**

Las imágenes se someten a técnicas de normalización para corregir variaciones en brillo, contraste y color. Se aplican correcciones geométricas y de deformación para alinear las características faciales. Se realiza filtrado y eliminación de ruido, y se divide el dataset en conjuntos de entrenamiento y prueba.

➤ **Fase 3: Procesamiento de imágenes**

Se implementan algoritmos de reconocimiento facial basados en aprendizaje automático para identificar y localizar las regiones faciales de interés en las imágenes. Se extraen características faciales relevantes utilizando técnicas de procesamiento de imágenes y aprendizaje automático, y se almacenan para su posterior uso.

➤ **Fase 4: Extracción de características**

Se aplican técnicas de procesamiento de imágenes y aprendizaje automático para extraer características específicas de las imágenes faciales, como descriptores de textura, histogramas de colores o características de forma. Se realiza un análisis dimensional para reducir la dimensionalidad y evaluar la calidad de las características extraídas.

➤ **Fase 5: Resultado**

Se calculan las distancias euclidianas entre las características faciales y se establece un umbral de similitud para diferenciar gemelos monocigóticos y no monocigóticos. Se evalúa el desempeño del sistema utilizando métricas de evaluación y se presentan conclusiones basadas en los resultados obtenidos, junto con posibles mejoras y recomendaciones para futuros trabajos.

CAPÍTULO II

2.1. Marco Contextual

2.1.1. Provincia de Santa Elena

Las pruebas se realizarán exclusivamente con personas gemelas residentes en la provincia de Santa Elena, una reconocida provincia costera a nivel nacional que cuenta con destinos turísticos populares como la playa de Salinas y la playa de Montañita. Esta provincia posee una conurbación urbana y actualmente tiene una población de 308,693 habitantes [19].

La provincia de Santa Elena se ubica en la costa de Ecuador, específicamente en la región litoral del país. Limita al norte con la provincia de Manabí, al este con la provincia de Guayas y al sur y oeste con el océano Pacífico. Su ubicación geográfica precisa es en las coordenadas $1^{\circ}56'00''S$ de latitud y $80^{\circ}45'00''O$ de longitud mostrado en la figura 5.

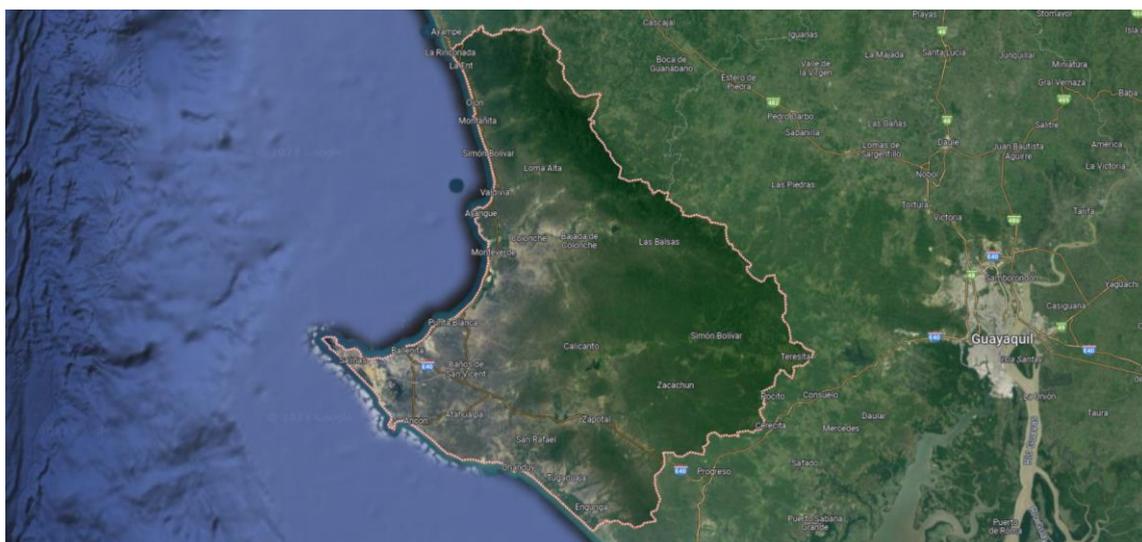


Figura 5. Ubicación de la Provincia de Santa Elena.

2.1.2. Misión

Nuestro Gobierno Autónomo Descentralizado Municipal, es administrador, gestionado, facilitador y regulador de bienes y servicios públicos permanentes, de calidad, con eficiencia, cobertura y acceso, mediante procesos, programas y proyectos inclusivos, participativos, transparentes para la sociedad, aplicando la solidaridad, el respeto, la responsabilidad y equidad [20].

2.1.3. Visión

El Gobierno Autónomo Descentralizado Municipal en el 2019 será una institución con capacidad administrativa, operativa y financiera, sólida e innovadora, generadora del desarrollo sostenible y sustentable del cantón, para los ciudadanos e inversionistas locales, nacionales y extranjeros, aplicando la gestión por resultados con transparencia, solidaridad, justicia y probidad [20].

2.1.4. LEY ORGÁNICA DE TRANSPARENCIA Y ACCESO A LA INFORMACIÓN PÚBLICA

La Ley Orgánica de Transparencia y Acceso a la Información Pública (LOTAIP) asegura el derecho de acceso a las fuentes de información para el ejercicio de la participación democrática en relación con el uso de archivos públicos, con la excepción de aquellas que, por razones de seguridad nacional, deben mantenerse confidenciales [21]:

Artículos	Contenido
Art. 1	Principios de Publicidad de la Información Pública
Art. 3	Ámbito de aplicación de la ley
Art. 4	Principios de aplicación de la ley
Art. 5	Información pública
Art. 7	Difusión de la información pública
Art. 8	Promoción del derecho de acceso a la información
Art. 9	Responsabilidad sobre la entrega de la información pública
Art. 23	Sanción a funcionarios y /o empleados públicos y privados

Tabla 2. Artículos de Ley Orgánica De Transparencia Y Acceso A La Información Pública [21].

2.1.4. LEY ORGÁNICA DE PARTICIPACIÓN CIUDADANA Y LEY ORGÁNICA DEL CONSEJO DE PARTICIPACIÓN CIUDADANA Y CONTROL SOCIAL

La Ley Orgánica de Participación Ciudadana y Control Social (LOPC) y la Ley Orgánica de Consejos Provinciales y Municipales de Planificación y Coordinación de la Competencia Sectorial (LOCPCCS) tienen como objetivo asegurar la implementación efectiva y la integración de los procesos diarios de participación ciudadana y gestión institucional. Esto se logra a través de los siguientes artículos que establecen los mecanismos, responsabilidades y lineamientos para fomentar la participación activa de la ciudadanía en la toma de decisiones y la planificación del desarrollo a nivel provincial y municipal [21]:

LOPC	LOCPCCS	CONTENIDOS
	Art. 5 N. 2	CPCCS establece mecanismos de Rendición de Cuentas.
Art. 88		Derecho ciudadano a la Rendición de Cuentas.
Art. 89	Art. 9	Definición de Rendición de Cuentas.
Art. 90	Art. 11	Sujetos obligados a rendir cuentas.
Art. 91		Objetivos de la Rendición de Cuentas.
Art. 92	Art.10	Contenidos de la Rendición de Cuentas que corresponde al nivel político.
Art. 93	Art.10	Contenidos de la Rendición de Cuentas que corresponde al nivel programático y operativo.
Art. 94		Mecanismos para rendir cuentas.
Art. 95		Periodicidad de la Rendición de Cuentas.
	Art. 11	Sanciones.

LOPC	LOPCCS	CONTENIDOS
	Art. 12	Monitoreo.
Art. 96		El libre acceso a la información pública.
Art. 97		Principios generales del acceso a la información.
Art. 98		Transparencia de la administración pública.
Art. 60, N. 4		Funciones de las asambleas locales: organizar Rendición de Cuentas.

Tabla 3. Artículos De Participación Ciudadana y Ley Orgánica del CPCCS [21].

2.2. MARCO CONCEPTUAL

2.2.1. Distancias Euclidianas

La distancia entre dos vectores puede ser medida utilizando valores de punto flotante o valores enteros. Si las columnas contienen valores numéricos en diferentes escalas, es común normalizar o estandarizar los valores en todas las columnas antes de calcular las distancias euclidianas. En casos donde se requiere calcular estas distancias miles o millones de veces, es posible evitar el cálculo de la raíz cuadrada y obtener la proporción relativa de los resultados. Esta técnica es especialmente útil en algoritmos de aprendizaje automático, donde se busca optimizar el rendimiento y la eficacia de los cálculos [22]:

$$\text{EuclideanDistance} = \text{suma de } i \text{ a } N (v1[i] - v2[i])^2$$

2.2.2. Datos Atípicos

La identificación de datos atípicos es de gran utilidad en el análisis de datos, ya que puede tener un impacto significativo en los resultados estadísticos. Estos datos atípicos tienen el potencial de revelar inconsistencias y detectar errores en todo el proceso de análisis. Un dato atípico se caracteriza por ser un punto de datos que se encuentra significativamente por encima o por debajo del patrón general observado en los grupos de datos o en los gráficos. Para ser considerado un dato atípico, debe cumplir al menos una de las siguientes dos condiciones [23]:

$$\text{Menor} < Q1 - 1.5(\text{IQR})$$

$$\text{Mayor} > Q3 + 1.5(\text{IQR})$$

2.2.3. Diferencia en detección rostro y reconocimiento facial

La detección de rostros se basa en el uso de tecnología de inteligencia artificial para identificar y analizar caras humanas dentro de imágenes mediante algoritmos especializados. Su objetivo principal es separar y distinguir los rostros del resto de los elementos presentes en la imagen. Por otro lado, el reconocimiento facial va más allá de la detección de rostros, ya que se basa en el uso de tecnología biométrica para determinar la identidad de una persona, no solo reconociendo su rostro, sino también considerando otras características y atributos únicos. Esta tecnología permite verificar la identidad de una persona mediante la comparación de características faciales con una base de datos previamente almacenada [24].

2.2.4. Entrenamiento del reconocedor

2.2.4.1. LBPH (Histograma de patrón binario local)

El algoritmo LBPH (Histograma de Patrón Binario Local) es utilizado para el reconocimiento facial, destacándose por su capacidad para reconocer rostros en diversas posiciones, ya sea de frente o de costado [25].

Para entender la intuición detrás del algoritmo, primero se explica cómo se representan las imágenes en forma de matrices compuestas por píxeles. Cada píxel tiene un valor que oscila entre 0 y 255, representando los colores rojo, verde y azul (RGB). El algoritmo LBPH se basa en aplicar una condición a una matriz de píxeles centrada en cada píxel para generar un valor binario. Luego, se calcula un valor decimal basado en los resultados binarios de los píxeles vecinos [25].

El algoritmo es robusto ante cambios en la iluminación, lo que le permite obtener buenos resultados en imágenes claras y oscuras. Utiliza histogramas para contar la frecuencia de aparición de valores binarios en diferentes áreas de la imagen. Con esto, puede identificar bordes y características clave de una cara, como los ojos o la boca [25].

2.2.4.2. MTCNN (Redes Neuronales Convolucionales Multitarea)

El modelo MTCNN (Multi-task Cascaded Convolutional Networks) es una arquitectura de red convolucional su objetivo es detectar rostros y facciones en imágenes. El modelo consta de tres etapas de redes convolucionales y utiliza el aprendizaje multi-task para

dividir las tareas de detección de rostros y facciones. Esto mejora la generalización al aprovechar la información específica del dominio en las señales de entrenamiento: [26].

- En la primera etapa, se utiliza una red tipo Fully Convolutional Network (FCN) para obtener los posibles candidatos de rostros y sus bounding boxes.
- En la segunda etapa, una red CNN llamada "The Refine Network (R-Net)" reduce los candidatos, ajusta las bounding boxes y realiza una supresión no máxima (NMS) para unir candidatos cercanos y determinar si la entrada contiene una cara.
- En la tercera etapa, otra red similar a la R-Net proporciona la posición de los ojos, nariz y boca de los candidatos detectados, completando la alineación facial.

2.2.4.3. Tensorflow-gpu

La herramienta en cuestión ofrece facilidad para la creación de modelos de aprendizaje automático en una amplia variedad de plataformas, incluyendo computadoras de escritorio, dispositivos móviles, la web y la nube. No importa si eres principiante o experto, ya que esta API de alto nivel está diseñada para brindar un enfoque accesible en el desarrollo de aplicaciones basadas en redes neuronales. Está basada en el estándar del API de Keras, lo que facilita el entendimiento y la implementación de las redes neuronales en tus proyectos [27].

2.2.4.4. Keras

La API en cuestión es una creación de Google que permite la implementación de redes neuronales. Se destaca por ser compatible con el cálculo de múltiples redes neuronales en su back-end. Proporciona una interfaz de alto nivel en Python, lo que facilita su aprendizaje y uso. Además, ofrece la flexibilidad de elegir entre varios back-ends para realizar los cálculos necesarios. Esto brinda a los desarrolladores la capacidad de utilizar la opción más adecuada para sus necesidades específicas [28].

2.2.4. OpenCV 4.2

La biblioteca de código abierto en cuestión es una herramienta ampliamente utilizada para el análisis de imágenes y videos, así como para la visión por computadora. Juega un papel crucial al permitir operaciones en tiempo real para identificar objetos, rostros y estructuras humanas a través de algoritmos especializados. Esta biblioteca proporciona capacidades para extraer y representar datos en un espacio vectorial, lo que facilita la realización de

operaciones matemáticas y el procesamiento de información visual de manera eficiente. En resumen, esta biblioteca es una poderosa herramienta que ofrece funcionalidades esenciales para el análisis y procesamiento de imágenes y videos en aplicaciones de visión por computadora [29].

2.2.5. Python 3.6

El lenguaje de programación mencionado es una herramienta ampliamente utilizada en el desarrollo de sitios web y software, así como en el análisis de datos y automatización de tareas. Permite la creación de variables y programas de diversa naturaleza, sin presentar dificultades específicas. Su versatilidad lo convierte en una opción beneficiosa tanto para principiantes como para programadores experimentados. Debido a su facilidad de uso y a su popularidad, este lenguaje de programación se ha convertido en uno de los más ampliamente utilizados en la industria actualmente [30].

2.2.8. MSVC (Microsoft Visual Studio):

El programa en cuestión es un entorno de desarrollo integrado (IDE, por sus siglas en inglés). Con esta herramienta, se pueden realizar tareas como la edición, desarrollo y escritura de código de diferentes lenguajes de programación, incluyendo C, C++, C++/CLI, .NET, JavaScript, TypeScript, XML, XSLT, HTML, CSS, Python, Ruby y Node.js, entre otros. Gracias a esta amplia lista de lenguajes soportados, ofrece a los usuarios una gran variedad de opciones para sus proyectos. Además, el IDE proporciona funcionalidades para trabajar en cuatro niveles clave: desarrollo, depuración, prueba y colaboración, facilitando así el proceso de desarrollo de software [31].

2.2.9. PyQt5

Tiene un kit de herramientas GUI multiplataforma, con un conjunto de enlaces de Python para QT v5. Se desarrollan aplicaciones de escritorio interactivo con una sencillez ya que tiene herramientas y la simplicidad de la biblioteca, ha hecho una herramienta llamada 'QtDesigné' para hacer el front-end por medio del método de arrastrar y soltar así el desarrollo tenga una rapidez y tener tiempo a las cosas de back-end [32].

2.2.10. SQLite3

La biblioteca C en cuestión es una solución de base de datos en disco que elimina la necesidad de un servidor separado. Permite acceder a la base de datos mediante una

variante no estándar del lenguaje de consulta SQL. Con esta biblioteca, es posible crear prototipos de aplicaciones utilizando SQL y luego transferir fácilmente el código a una base de datos de mayor escala. Esta característica resulta especialmente útil al desarrollar aplicaciones que requieren una base de datos más robusta y escalable en etapas posteriores del proceso de desarrollo. La biblioteca proporciona una manera eficiente de trabajar con datos y realizar consultas, sin la necesidad de configuraciones complejas de servidor [33].

2.2.11. ResNet

Aplica capas adicionales en las redes neuronales profundas con el objetivo de mejorar tanto la precisión de los resultados como el rendimiento general del modelo. A medida que se van agregando capas, la red neuronal es capaz de aprender características más complejas y abstractas. Tomando como ejemplo el reconocimiento de imágenes, en las primeras capas la red puede aprender a detectar bordes y texturas básicas, mientras que en capas más profundas es capaz de identificar objetos específicos. Este enfoque de añadir capas progresivamente permite que el modelo adquiera un mayor nivel de sofisticación y capacidad de reconocimiento a medida que se avanza en la arquitectura de la red neuronal [34].

2.2.12. Moblenet

El patrón de modelo de visión por computadora mencionado es una biblioteca de código abierto desarrollada por Google y diseñada específicamente para el entrenamiento de clasificadores. Utiliza técnicas de convolución profunda que permiten reducir el número de parámetros en comparación con otras arquitecturas de redes neuronales. Esto presenta una ventaja significativa, ya que nos brinda una red neuronal más limpia y profunda, al mismo tiempo que es más liviana en términos de recursos computacionales requeridos. Esta combinación de convoluciones profundas y reducción de parámetros nos permite obtener modelos de clasificación más eficientes y con un mejor rendimiento en tareas de visión por computadora [35].

2.2.13. Google Colab

Es una plataforma que ofrece la posibilidad de escribir y ejecutar código de Python de manera interactiva en un navegador web. Su objetivo principal es brindar a los usuarios un entorno accesible para proyectos de aprendizaje automático, análisis de datos y

educación. En el ámbito profesional, Colab se presenta como un servicio de notebooks de Jupyter alojado, lo que significa que no es necesario realizar la instalación del software en el dispositivo para utilizarlo. Además, se destaca por ser de acceso gratuito, lo que permite a los usuarios aprovechar sus funcionalidades sin incurrir en costos adicionales. Esta combinación de características convierte a Colab en una herramienta versátil y conveniente para aquellos que deseen colaborar, experimentar y desarrollar proyectos de Python en un entorno basado en la nube [36].

2.2.14. Anaconda

Es una plataforma diseñada especialmente para principiantes que deseen adentrarse en el aprendizaje de Python. Su instalación es sencilla y ofrece la ventaja de contar con el entorno Jupyter Notebook integrado, lo que facilita la escritura y ejecución de código. Además, Anaconda ofrece una amplia gama de funciones y bibliotecas adicionales, lo que la convierte en una herramienta versátil para proyectos diversos. Esta plataforma es compatible con los sistemas operativos Windows, macOS y Linux, lo que permite a los usuarios de diferentes sistemas operativos aprovechar sus funcionalidades. Cabe destacar que Anaconda es desarrollada por continuum.io, una empresa especializada en el desarrollo y soporte de herramientas basadas en Python, lo que garantiza su calidad y confiabilidad [37].

2.2.15. Compute Unified Architecture (CUDA)

La plataforma en cuestión presenta un impresionante potencial de cálculo, ideal para aplicaciones que demandan un alto rendimiento y requieren la capacidad de paralelización. Entre las áreas en las que se destaca su utilidad se encuentran los dominios de simulación, modelos científicos, física, análisis e investigación en ciencias médicas, así como también en el campo del aprendizaje automático, entre otros. Es importante destacar que esta plataforma está especialmente diseñada para el procesamiento mediante las GPU de Nvidia, lo que garantiza un rendimiento óptimo [38].

Las tareas que se ejecutan en esta plataforma no necesariamente deben realizarse de manera secuencial, ya que pueden llevarse a cabo simultáneamente con otras tareas a través de la tecnología CUDA. Además, se ofrece soporte para los lenguajes de programación C, C++ y Fortran, lo que permite una mayor flexibilidad para los desarrolladores. En resumen, esta plataforma brinda la facilidad de descargar y ejecutar

tareas de computación intensiva en las GPU de Nvidia utilizando CUDA, lo que la convierte en una herramienta de gran valor para aplicaciones de alto rendimiento [38].

2.2.16. Entornos de desarrollo

El desarrollo de software es un campo que engloba un conjunto de herramientas y procesos utilizados para crear aplicaciones y programas informáticos. Proporciona un entorno de trabajo donde los desarrolladores pueden crear, innovar y probar su código sin afectar el funcionamiento del entorno de producción. Existen tres tipos comunes de entornos de desarrollo: los centrados en un lenguaje de programación específico, los orientados a estructuras y los basados en kits de herramientas. Cada uno ofrece diferentes características y ventajas para los desarrolladores en su proceso de creación de software. [39].

2.2.17. Entorno virtual de Python

Una distribución de carpetas es un conjunto organizado de directorios que proporciona un entorno aislado y liviano para la ejecución de Python. Estos entornos virtuales permiten trabajar con diferentes versiones de paquetes y librerías, sin interferir con el entorno de Python global del sistema. La estructura de carpetas en un entorno virtual no requiere una comprensión detallada para obtener información sobre su contenido. Por ejemplo, mediante el uso de comandos y herramientas específicas, es posible crear, activar y desactivar entornos virtuales de forma sencilla y eficiente, permitiendo una gestión flexible de los requisitos de dependencias y facilitando el desarrollo de proyectos en Python [40].

2.2.18. Detección de objetos usando OpenCV

Es una técnica de visión artificial que permite localizar e identificar objetos en imágenes o videos. Mediante algoritmos y modelos de aprendizaje automático, esta técnica proporciona información precisa sobre la ubicación de los objetos en una escena, así como su etiquetado preciso. El modelo utilizado en la detección de objetos es capaz de analizar y reconocer diferentes objetos presentes en una imagen o video, brindando información detallada sobre la ubicación y la identidad de cada objeto detectado. Esto resulta invaluable en aplicaciones como reconocimiento facial, sistemas de vigilancia, conducción autónoma, entre otros. [41].

2.2.19. Componentes de un sistema de visión por computador

En el contexto de la visión por computadora, existen varios componentes esenciales que desempeñan roles clave en el proceso de captura y análisis de imágenes. Estos componentes incluyen la iluminación, que proporciona la luz necesaria para resaltar las características y permitir una visualización clara. El lente juega un papel importante al permitir el enfoque automático dentro de una distancia determinada, garantizando una imagen nítida y precisa [42].

El sensor de imagen desempeña un papel fundamental al convertir la luz en una imagen digital, que luego puede ser procesada y analizada. El procesamiento de visión es el componente encargado de extraer datos y características relevantes de la imagen digital, como bordes, colores o formas, permitiendo realizar análisis más avanzados [42].

Finalmente, las comunicaciones facilitan la coordinación y la conexión rápida entre los diferentes componentes del sistema, asegurando una sincronización eficiente y una interacción fluida entre los elementos de la máquina. Estos componentes trabajan en conjunto para permitir la captura, el procesamiento y el análisis de imágenes en aplicaciones de visión por computadora [42].

2.3. Marco Teórico

2.3.1. Identificación de rostros familiares y desconocidos a partir de características internas y externas

Se identifican subconjuntos de rasgos faciales críticos para los seres humanos mediante el uso de rostros familiares, desconocidos y el reconocimiento facial mediante redes neuronales profundas (DNN). Esta técnica proporciona un enfoque innovador para la ingeniería inversa al reemplazar diversas características faciales hasta lograr que los rostros sean reconocidos como identidades distintas. Se seleccionan de tres a cinco características de alta puntuación (PS), excluyendo aquellas con baja puntuación (PS) que se predicen como identidades diferentes. De esta manera, se logra obtener tanto características faciales externas como internas utilizando diferentes rasgos para el reconocimiento de rostros familiares y desconocidos [43].

2.3.2 Psicometría, perfiles y sesgos. El caso del reconocimiento facial

Muestra los problemas relacionados con la producción de sesgos en la inteligencia artificial (IA) y el aprendizaje automático. Se ha identificado la existencia de discriminación en términos de rasgos raciales, género y otros. En primer lugar, se llevó a cabo un estudio sobre el sesgo del mundo, el cual se refiere a las exclusiones presentes en la programación y el contexto computarizado, donde se evidencia una discriminación arraigada histórica y socialmente. Este análisis busca comprender y abordar los sesgos presentes en los sistemas de IA y el aprendizaje automático, con el objetivo de fomentar la equidad y la eliminación de sesgos en estas tecnologías. [44].

El segundo tipo se refiere a los sesgos introducidos por los programadores al entrenar los sistemas de IA, basándose en taxonomías hegemónicas que pueden llevar a interpretaciones erróneas. Por ejemplo, se podría asociar erróneamente que las personas que visten bien son conservadoras y tienen el cabello largo tienen una tendencia política específica [44].

El tercer tipo de sesgo, conocido como sesgo estadístico, está relacionado con los modelos de aprendizaje automático. En este caso, los algoritmos pueden dar más peso o importancia a variables que proporcionan información sobre los comportamientos humanos, lo que puede llevar a una pérdida de diversidad cultural en la sociedad [44].

Estos sesgos representan desafíos importantes en el campo de la IA y el aprendizaje automático, ya que pueden perpetuar discriminaciones y desigualdades. Es fundamental abordar estos sesgos y trabajar hacia modelos más equitativos y libres de sesgos, promoviendo la inclusión y considerando la diversidad cultural en el desarrollo de sistemas de IA [44].

2.3.3 Sistema de reconocimiento facial con Deep Learning

Se ha desarrollado un sistema de reconocimiento facial que tiene la capacidad de estimar la edad de un usuario a partir de su rostro. Para lograr esto, se utilizaron técnicas de aprendizaje profundo conocidas como redes neuronales, aprovechando las herramientas y recursos proporcionados por la comunidad de proyectos en este campo. Esta colaboración ha permitido acceder a una amplia información, ejemplos y discusiones provechosas que han favorecido la adopción de mejores prácticas [45].

El modelo de reconocimiento facial logró clasificar correctamente aproximadamente 2 de cada 3 imágenes, lo que representa un porcentaje de precisión del 65,92%. Este resultado demuestra la eficacia del sistema en la tarea de estimar la edad de los usuarios a partir de sus rostros. El desarrollo de este sistema es un avance significativo en el campo del reconocimiento facial y muestra el potencial de las técnicas de aprendizaje profundo para abordar tareas complejas como esta [45].

CAPÍTULO III

3. Propuesta

3.1. Requerimiento

3.1.1. Requerimientos funcionales

ID	REQUERIMIENTO	DESCRIPCIÓN DEL REQUERIMIENTO
RF-001	Captura de imágenes	El sistema debe consentir la captura de imágenes faciales con mayor calidad para el procesamiento y la comparación.
RF-002	Algoritmo de detección facial	Tener un algoritmo de detección facial preciso y confiable que pueda reconocer y extraer las características faciales.
RF-003	Extracción de características	El sistema debe tener la capacidad de extraer características faciales distintivas, como la forma de la cara, la disposición de los ojos, la nariz y la boca, para poder hacer las comparaciones.
RF-004	Cálculo de distancias euclidianas	El algoritmo tiene que calcular las distancias euclidianas entre las características faciales extraídas de dos imágenes diferentes. Esto permite cuantificar la similitud entre las caras y determinar si pertenecen a gemelos monocigóticos o no.

ID	REQUERIMIENTO	DESCRIPCIÓN DEL REQUERIMIENTO
RF-005	Umbral de similitud ajustable	El sistema debe permitir ajustar un umbral de similitud y establecer un límite para considerar dos caras como similares o no. Esto facilitará la detección de gemelos monocigóticos basada en las distancias euclidianas calculadas.
RF-006	Identificación de datos atípicos	El sistema debe identificar y manejar datos atípicos, como imágenes de menor calidad, expresiones faciales extremas o condiciones de iluminación adversas. Esto garantizará la precisión y confiabilidad del proceso.
RF-007	Gestión de la base de datos	El sistema debe contar con una base de datos que almacene las características faciales de referencia de gemelos monocigóticos conocidos. Esto permitirá realizar comparaciones eficientes y precisas con las nuevas imágenes capturadas.
RF-008	Interfaz de usuario intuitiva	El sistema debe contar con una interfaz de usuario fácil de usar que permita a los usuarios capturar imágenes, ajustar los parámetros de similitud y visualizar los resultados de la comparación.
RF-009	Escalabilidad	El sistema debe ser escalable para manejar un gran número de imágenes y gemelos monocigóticos en la base de datos. Esto garantizará que el sistema pueda manejar un crecimiento futuro y realizar comparaciones eficientes en tiempo real.
RF-0010	Precisión y rendimiento	El sistema debe ofrecer una alta precisión en la detección de similitud facial y un rendimiento eficiente, especialmente en términos de velocidad de procesamiento y tiempos de respuesta.

ID	REQUERIMIENTO	DESCRIPCIÓN DEL REQUERIMIENTO
RF-0011	Registro de gemelos monocigóticos	El sistema debe permitir el registro de nuevos gemelos monocigóticos en la base de datos, asegurando la asociación correcta de las características faciales con cada individuo.
RF-0012	Tolerancia a cambios en la apariencia	El sistema debe ser capaz de manejar cambios en la apariencia de los gemelos monocigóticos debido a factores como el envejecimiento, cambios de peso o estilo de cabello. Debe ser capaz de reconocer la similitud facial subyacente a pesar de estos cambios.
RF-0013	Manejo de variaciones étnicas	El sistema debe ser capaz de reconocer y comparar características faciales en gemelos monocigóticos que pertenezcan a diferentes grupos étnicos, considerando las diferencias inherentes a cada grupo.
RF-0014	Evaluación de confiabilidad	El sistema debe proporcionar una medida de confiabilidad para cada comparación de similitud facial realizada. Esto puede ser en forma de una puntuación o nivel de confianza que indique la certeza de la clasificación.
RF-0015	Actualización de la base de datos:	El sistema debe permitir la actualización de las características faciales de los gemelos monocigóticos en la base de datos cuando sea necesario, para reflejar cambios en su apariencia o agregar nuevas imágenes de referencia.

Tabla 4. *Requerimientos funcionales*

3.2. Componente de la Propuesta

3.2.1. Arquitectura del Sistema

3.2.1.1. Algoritmo basado en el modelo LBPH

Fase 1.- Generar un dataset

El presente dataset ha sido creado con el propósito de abordar el desafío de la diferenciación de gemelos monocigóticos, quienes comparten el mismo material genético y, por ende, presentan una apariencia física sumamente similar. Con el fin de garantizar la calidad y representatividad del dataset, se ha puesto especial atención en la diversidad de las imágenes incluidas.

El dataset abarca una amplia variedad de condiciones de captura, considerando distintas condiciones de iluminación, ángulos de visión, expresiones faciales, así como variaciones en el estilo de pelo y maquillaje. Esta diversidad busca proporcionar al modelo una experiencia de entrenamiento más realista y robusta, permitiéndole lidiar con las variaciones presentes en situaciones del mundo real.

Además, se han identificado y etiquetado datos atípicos dentro del dataset. Estas imágenes corresponden a casos donde los gemelos monocigóticos no se asemejan tanto como se esperaría, desafiando la tarea de diferenciación. La inclusión de estos datos atípicos resulta particularmente valiosa para el entrenamiento del modelo, ya que contribuyen a mejorar su capacidad para distinguir entre gemelos y detectar características faciales sutiles que podrían pasar desapercibidas.

El registro inicial del rostro de un gemelo en la base de datos se realizó mediante la captura automática de 300 imágenes para tener una mejor extracción de características, como se muestra en la figura 6. Se enfatizó la importancia de incluir rostros con una variedad de expresiones faciales y cambios de luz dentro del dataset. Para lograrlo, se emplearon técnicas de reconocimiento facial y algoritmos de procesamiento de imágenes, garantizando una captura precisa y coherente de los rasgos faciales.

Además, se instruyó a los gemelos para que realizaran diferentes gestos y expresiones emocionales durante el proceso de captura, enriqueciendo así la variabilidad presente en el dataset. Asimismo, se tuvo en cuenta una diversidad de condiciones de iluminación, incluyendo situaciones de luz controlada y desafiantes. Este enfoque asegura que el

dataset refleje condiciones reales y ayuda a fortalecer la capacidad del modelo para enfrentar diversas condiciones de iluminación en la diferenciación de gemelos monocigóticos.

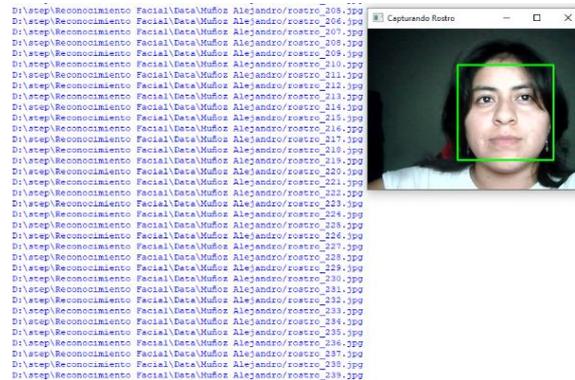


Figura 6. Capturando rostro.

Se utiliza el clasificador `haarcascade_frontalface_default.xml` para detectar los rostros en cada fotograma de video capturado. Una vez detectados, se extraen las regiones faciales del fotograma original, se redimensionan a 150x150 píxeles y se guardan en la carpeta correspondiente a la persona en el dataset.

El proceso de captura de imágenes se repite hasta que se presiona la tecla "Esc" o se alcanza el límite predefinido de imágenes (300 en este caso). Esto asegura que se capturen suficientes imágenes para entrenar y evaluar el sistema.

Al recorrer la lista de los rostros detectados, se recortó y redimensionó la región del rostro encontrado, con el objetivo de obtener un tamaño uniforme de 720x720 píxeles para cada rostro. Estos pasos se ilustran en las figuras 7 y 8. Este enfoque técnico garantiza una estandarización del tamaño de los rostros en el dataset, lo cual es beneficioso para el posterior entrenamiento y análisis. Además, la utilización del clasificador de Haar cascade y la conversión a escala de grises contribuyen a una detección y procesamiento eficiente de los rostros en las imágenes capturadas.



Figura 7. Escala gris Mile



Figura 8. Escala gris Stephany

Durante el proceso, se crearon dos listas vacías: una para almacenar las etiquetas de las imágenes del rostro y otra para almacenar los datos correspondientes. Para cada imagen, se incluyó una espera de 10 milisegundos antes de pasar a la siguiente, permitiendo así el correcto almacenamiento en los subdirectorios dentro del directorio principal, los cuales representan a cada persona de manera individual.

Se prestó especial atención a la organización y estructura adecuadas de las carpetas e imágenes dentro del dataset. Se asignaron etiquetas numéricas (0 y 1) para diferenciar las categorías, tal como se muestra en la figura 9.

```
labels= Squeezed text (80 lines).  
Numero de etiquetas 0: 169  
Numero de etiquetas 1: 300
```

Figura 9. Etiqueta 0 y 1

Este enfoque técnico de creación de listas y organización de imágenes y etiquetas en subdirectorios garantiza una gestión eficiente y ordenada del dataset, facilitando su posterior uso en tareas de entrenamiento y análisis de datos.

Fase 2.- Preprocesamiento de imagen

En la fase de preprocesamiento de imágenes, se preparan los datos para entrenar el reconocedor de rostros. Para ello, se realiza una serie de operaciones en cada imagen capturada durante la fase 1.

1. Lectura de imágenes y creación de etiquetas:

- Se especifica la ruta donde se almacenan las imágenes capturadas durante la fase 1, utilizando la variable `dataPath`.
- Se obtiene una lista de los nombres de las personas presentes en el dataset utilizando la función `os.listdir(dataPath)` y se almacenan en `peopleList`.

- Se inicializa una lista vacía llamada labels para almacenar las etiquetas de cada imagen de acuerdo con la persona a la que pertenece.
- Se inicializa otra lista vacía llamada facesData para almacenar los datos de las imágenes en escala de grises.

2. Conversión a escala de grises:

- Se recorre cada persona en la lista peopleList.
- Se crea la ruta completa a la carpeta de la persona actual utilizando personPath.
- Para cada imagen dentro de la carpeta de la persona actual, se realiza lo siguiente:
 - Se almacena la etiqueta de la persona actual en la lista labels.
 - Se lee la imagen en escala de grises utilizando `cv2.imread(personPath+'/'+fileName, 0)`.
 - La imagen leída se agrega a la lista facesData.

3. Entrenamiento del reconocedor de rostros:

- Se crea el reconocedor de rostros utilizando el clasificador LBPH (Local Binary Patterns Histograms) mediante `cv2.face.LBPHFaceRecognizer_create()`.
- Se inicia el entrenamiento del reconocedor utilizando `face_recognizer.train(facesData, np.array(labels))`, donde facesData son las imágenes en escala de grises y labels son las etiquetas correspondientes a cada imagen.
- Luego del entrenamiento, el modelo resultante se almacena en un archivo XML mediante `face_recognizer.write('modeloLBPHFace.xml')`.

En resumen, la fase de preprocesamiento de imágenes se encarga de leer las imágenes capturadas durante la fase 1, convertirlas a escala de grises y preparar los datos para entrenar el reconocedor de rostros utilizando el algoritmo LBPH. Una vez que se ha realizado el entrenamiento, el modelo resultante se almacena en un archivo XML para su uso posterior en el proceso de reconocimiento facial durante la fase 3.

Fase 3: Procesamiento de imágenes

Se comienza leyendo las imágenes almacenadas en la ruta especificada (dataPath). Para cada imagen, se convierte a escala de grises y se utiliza el clasificador **haarcascade_frontalface_default.xml** para detectar los rostros presentes en la imagen. Cada rostro detectado se redimensiona a 150x150 píxeles y se pasa por el reconocedor de rostros LBPHFace previamente entrenado.

El reconocedor de rostros LBPHFace utiliza un modelo entrenado previamente, almacenado en el archivo modeloLBPHFace.xml, para predecir la identidad de cada rostro. Se obtiene un resultado que incluye una etiqueta de identificación y una medida de similitud entre el rostro y el modelo.

El código muestra el resultado obtenido en la imagen original, incluyendo el nombre de la persona reconocida y un cuadro delimitador alrededor del rostro identificado como se muestra en la figura 10. El proceso de detección y reconocimiento de rostros se realiza en un bucle continuo hasta que se presiona la tecla "Esc".



Figura 10. Detención con el modelo LBPH

3.2.1.2. Algoritmo basado en el MTCNN (Multi-task Cascaded Convolutional Networks).

Fase 1.- Generar un dataset

En esta fase, se procede a la generación de un dataset de rostros faciales capturados en tiempo real de usuarios que son gemelos monocigóticos. Los gemelos monocigóticos son aquellos que comparten el mismo material genético, lo que los convierte en candidatos ideales para evaluar la precisión del sistema de reconocimiento facial.

1. Función register_capture():
 - Esta función permite capturar el rostro de un gemelo monocigótico utilizando la cámara en tiempo real.

- Se accede a la cámara utilizando la biblioteca OpenCV y se capturan los fotogramas para visualizarlos.
- Se utiliza el clasificador Haar Cascade (faceClassif) de OpenCV para detectar los rostros en los fotogramas capturados.
- Para cada rostro detectado, se procede a almacenarlo en una carpeta específica asociada al nombre del gemelo monocigótico.
- Las imágenes de los rostros capturados se guardan en formato .jpg junto a la descripción de texto y tipo binario como se muestra en la figura 11, en la carpeta correspondiente hasta alcanzar un límite de 300 imágenes por gemelo.

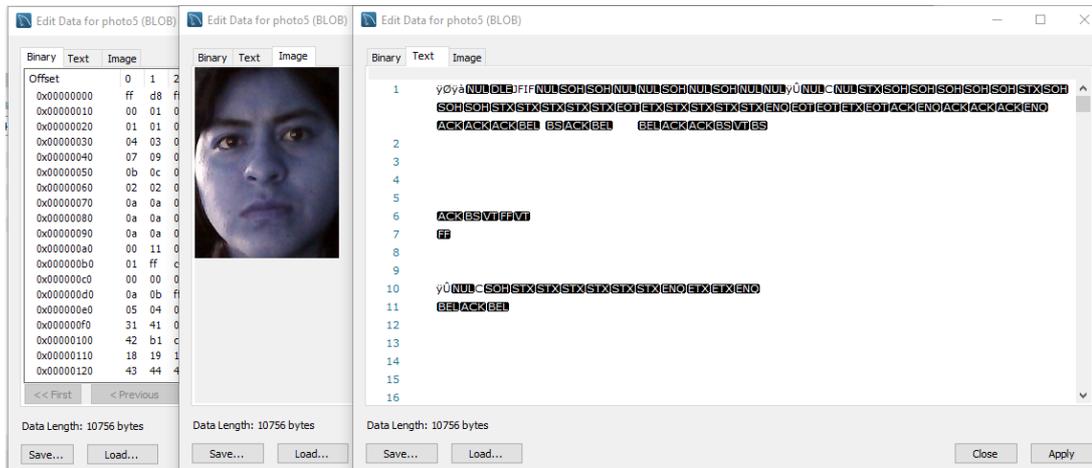


Figura 11. Dataset MySQL

Durante esta fase, se capturan los rostros de los gemelos monocigóticos para construir un dataset que se utilizará en las siguientes etapas del sistema de reconocimiento facial. El hecho de que los gemelos monocigóticos compartan un material genético idéntico nos permitirá evaluar cómo el sistema puede distinguir y reconocer sus rasgos faciales únicos. Esto ayudará a evaluar la eficacia y precisión del sistema en un escenario donde hay una gran similitud entre los usuarios, lo que representa un desafío adicional para el reconocimiento facial.

Fase 2: Preprocesamiento de imagen

Esta fase no está incluida en el código proporcionado. Sin embargo, en el preprocesamiento de imagen se pueden aplicar diversas técnicas para mejorar la calidad y la consistencia de las imágenes faciales capturadas. Algunas técnicas comunes incluyen el ajuste de contraste, la normalización de la iluminación, la eliminación de ruido y la alineación facial.

Fase 3: Procesamiento de imágenes

1. En el código algoritmo.py, la función login_capture() se encarga de capturar una imagen facial para el proceso de inicio de sesión. Los pasos en esta fase son los siguientes:
2. Se utiliza OpenCV para capturar imágenes en tiempo real desde la cámara.
3. La imagen capturada se guarda en formato JPG con el nombre del usuario y se almacena en una ruta específica.
4. Se utiliza la biblioteca MTCNN para detectar los rostros en la imagen capturada.
5. La función face() muestra los rostros detectados en una ventana gráfica utilizando Matplotlib.
6. Se llama a la función getUser() del archivo database.py para obtener los datos del usuario correspondiente a la imagen capturada.
7. Si se encuentra un usuario válido en la base de datos, se realiza la comparación entre la imagen capturada y la imagen de referencia del usuario utilizando el algoritmo de comparación compatibility().
8. Si la similitud entre las imágenes es mayor o igual a un umbral predefinido, se considera un inicio de sesión exitoso y se muestra un mensaje de éxito como se muestra en la imagen 12.



Figura 12. Detención con la biblioteca MTCNN (Multi-task Cascaded Convolutional Networks).

3.2.1.3. Algoritmo basado en TensorFlow y Keras

Fase 1.- Generar un dataset

En esta fase, se lleva a cabo la generación del dataset necesario para entrenar el modelo de reconocimiento facial específico para gemelos monocigóticos. El objetivo es capturar y almacenar imágenes de rostros faciales de gemelos idénticos para crear un conjunto de

datos que permita entrenar el modelo en la identificación precisa de estas personas con características físicas muy similares.

En el archivo data.py:

1. Se utiliza la biblioteca OpenCV (cv2) para acceder a la cámara web del ordenador y capturar fotogramas en tiempo real.
2. Se carga el clasificador Haar Cascade (facedetect) para la detección de rostros en los fotogramas capturados.
3. Se solicita al usuario que ingrese su nombre, que servirá como identificador para la carpeta en la que se guardarán las imágenes de su rostro.
4. Se crea una carpeta con el nombre del usuario en la ruta "images/" para almacenar las imágenes de su rostro.
5. Se inicia un bucle infinito para capturar continuamente fotogramas y detectar rostros en ellos.
6. Por cada rostro detectado, se recorta la región del rostro y se guarda como una imagen separada en la carpeta del usuario.
7. El proceso continúa hasta que se hayan capturado al menos 500 imágenes de rostros para cada gemelo monocigótico, lo que asegura que el dataset sea lo suficientemente grande para entrenar el modelo específico para gemelos.

En el archivo entrenamiento.py:

1. Se utiliza la biblioteca Keras para cargar el modelo previamente entrenado (keras_Model.h5) que se utilizará para realizar predicciones de reconocimiento facial.
2. Se carga un archivo de texto (labels.txt) que contiene los nombres de las clases o gemelos monocigóticos a las que se asignarán las predicciones.
3. Se accede a la cámara web del ordenador utilizando la biblioteca OpenCV (cv2).
4. Se inicia un bucle infinito para capturar continuamente fotogramas de la cámara web y realizar el proceso de reconocimiento facial en tiempo real.
5. Por cada fotograma capturado, se redimensiona la imagen a un tamaño de 224x224 píxeles para que coincida con el tamaño de entrada requerido por el modelo.

6. Se normaliza la imagen dividiendo cada valor de píxel por 127.5 y luego restando 1, lo que asegura que los valores de los píxeles estén en el rango [-1, 1].
7. Se realiza una predicción utilizando el modelo cargado y se obtiene el índice de la clase con mayor probabilidad de ser el gemelo monocigótico reconocido.
8. Se muestra la imagen capturada en una ventana y se imprime en la consola la clase o nombre del gemelo monocigótico reconocido junto con la puntuación de confianza obtenida en la predicción.
9. El bucle continúa hasta que se presione la tecla "Esc" (ASCII 27) en el teclado para salir del programa.

Esta fase se encarga de crear y preparar el dataset específico para gemelos monocigóticos y también de realizar el proceso de entrenamiento en tiempo real para reconocer a los gemelos a medida que aparecen frente a la cámara. El modelo entrenado será altamente especializado para lograr la distinción precisa entre gemelos idénticos.

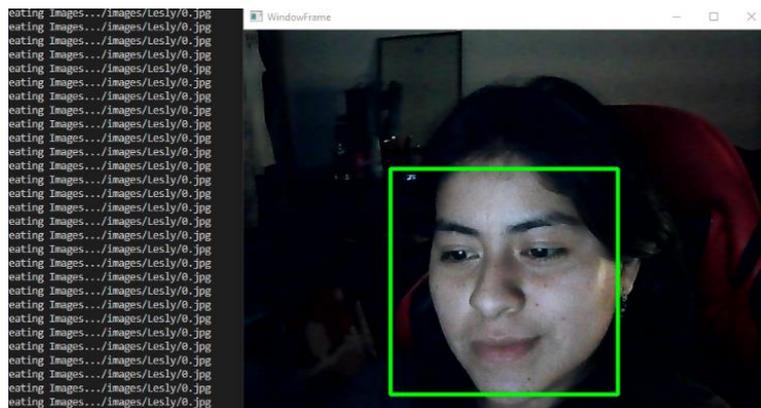


Figura 13. Extracción de rostro.

Fase 2: Preprocesamiento de imagen

1. Se utiliza la biblioteca OpenCV para capturar imágenes en tiempo real a través de la cámara.
2. En cada iteración del bucle, se lee un fotograma del video capturado.
3. La imagen capturada se redimensiona a un tamaño de 224x224 píxeles utilizando la función `cv2.resize()`.
4. Se muestra la imagen en una ventana de visualización.

Fase 3: Procesamiento de imágenes

1. Se utiliza la biblioteca OpenCV para capturar imágenes en tiempo real a través de la cámara.

2. En cada iteración del bucle, se lee un fotograma del video capturado.
3. La imagen capturada se redimensiona a un tamaño de 224x224 píxeles utilizando la función `cv2.resize()`.
4. La imagen se convierte en una matriz numpy y se remodela para que coincida con la forma de entrada requerida por el modelo.
5. Se normaliza la imagen dividiendo cada píxel por 127.5 y restando 1.
6. Se realiza la predicción utilizando el modelo cargado. La salida es un vector de probabilidades para cada clase.
7. Se selecciona la clase con la mayor probabilidad como la clase predicha.
8. Se muestra en la consola la clase predicha y el puntaje de confianza como se muestra en la figura 14.

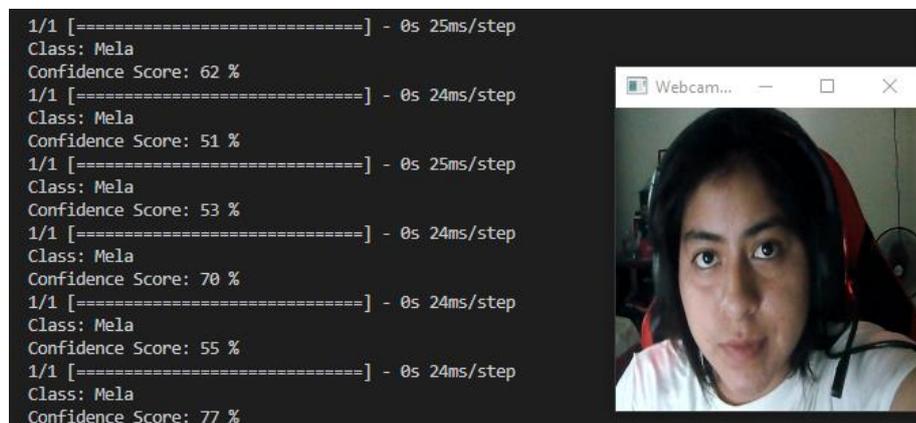


Figura 14. Detección con la librería TensorFlow

9. Se espera la entrada del teclado. Si se presiona la tecla "Esc" (ASCII 27), se interrumpe el bucle y se cierra la ventana.
10. Se liberan los recursos de la cámara y se cierra la ventana.

3.2.1.4. Extracción de características Faciales

En esta fase del algoritmo, se emplea la biblioteca MediaPipe para realizar la extracción de características faciales clave de los gemelos monocigóticos. La malla de puntos faciales proporciona una representación detallada y precisa de las estructuras anatómicas de cada gemelo. Cada rostro detectado en las imágenes es procesado mediante la función "`face_mesh.process()`", lo que resulta en un conjunto de puntos clave representativos (x_i , y_i , z_i) que describen la posición tridimensional de las estructuras faciales. Señalando en forma de malla como se muestra en la figura 15 y 16.

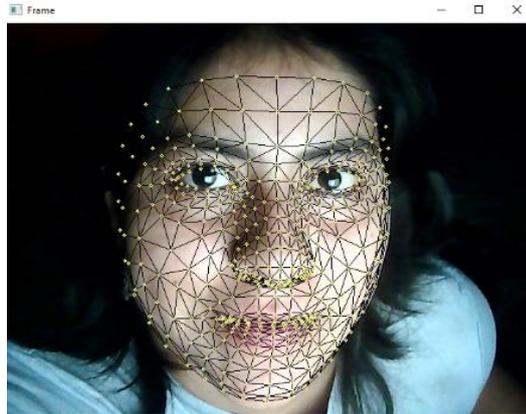


Figura 15. Malla en la posición tridimensional.

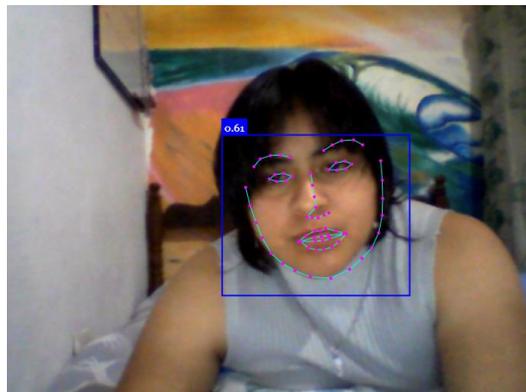


Figura 16. Malla de puntos faciales.

Ejemplo de extracción de características faciales utilizando MediaPipe:

```
# Cargar el módulo de detección de rostros
face_mesh = mp.solutions.face_mesh.FaceMesh(static_image_mode=False,
max_num_faces=2, min_detection_confidence=0.5,
min_tracking_confidence=0.5)
# Leer el fotograma desde la cámara
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
# Procesar la imagen para obtener los puntos faciales
results = face_mesh.process(image_rgb)

# Verificar si se detectaron rostros
if results.multi_face_landmarks:
    for face_landmarks in results.multi_face_landmarks:
        # Acceder a las coordenadas de los puntos faciales
        for landmark in face_landmarks.landmark:
            x, y, z = landmark.x, landmark.y, landmark.z
            # Procesar las coordenadas de los puntos faciales...
```

3.2.1.5. Cálculo de Distancias Euclidianas:

Una vez obtenidos los puntos clave representativos de las estructuras faciales de los gemelos, se procede a calcular las distancias euclidianas entre los puntos faciales correspondientes de cada gemelo monocigótico. Las distancias euclidianas cuantifican la separación y similitud entre las características faciales de los gemelos, permitiendo identificar patrones de variación y semejanza.

Para dos puntos faciales $P1 = (x1, y1, z1)$ y $P2 = (x2, y2, z2)$, la distancia euclidiana se calcula mediante la fórmula:

$$d(P1, P2) = \sqrt{((x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2)}$$

Estas distancias euclidianas entre los puntos faciales son calculadas para diferentes estructuras anatómicas, como cejas, ojos, labios, etc. La comparación de estas distancias entre gemelos permitirá determinar la similitud y diferencias entre sus características faciales.

Ejemplo de cálculo de distancia euclidiana entre dos puntos faciales:

```
def euclidean_distance(p1, p2):
    return np.sqrt((p2[0] - p1[0])**2 + (p2[1] - p1[1])**2 + (p2[2] -
p1[2])**2)
# Ejemplo de cálculo de distancia entre dos puntos faciales
point1 = (x1, y1, z1)
point2 = (x2, y2, z2)
distance = euclidean_distance(point1, point2)
```

3.2.1.6. Identificación de Datos Atípicos:

Para mejorar la robustez del sistema y lograr una diferenciación precisa de gemelos, se identificarán y etiquetarán datos atípicos dentro del conjunto de imágenes. Estos datos atípicos corresponden a gemelos monocigóticos que presentan características faciales que se desvían significativamente de lo esperado.

Durante la fase de generación del dataset, se detectan gemelos con configuraciones inusuales o atípicas en sus estructuras faciales. Por ejemplo, si un gemelo presenta una alineación atípica de los ojos o una forma inusual de los labios, se considerará un dato atípico.

La inclusión de estos datos atípicos en el proceso de entrenamiento es esencial para que el modelo aprenda a reconocer y diferenciar características sutiles que podrían pasar desapercibidas en casos más comunes.

Ejemplo de identificación de datos atípicos utilizando técnicas de aprendizaje no supervisado:

```
# Se tienen las distancias euclidianas entre puntos faciales de varios gemelos
distances_data = [...]

# Se utiliza el algoritmo DBSCAN para identificar los datos atípicos
outlier_detector = DBSCAN(eps=0.5, min_samples=5)
outlier_labels = outlier_detector.fit_predict(distances_data)

# Los datos atípicos se identifican como aquellos con etiqueta -1
```

3.3. Resultados

3.3.1. Pruebas

En este proyecto se llevó a cabo una evaluación exhaustiva de los algoritmos de reconocimiento facial basados en TensorFlow con Keras, LBPH y MTCNN, con el objetivo de analizar su eficacia en la detección de anomalías en rostros humanos. Para medir el rendimiento de cada algoritmo, se utilizaron métricas clave, que incluyeron la tasa de aciertos y el tiempo de respuesta.

Para llevar a cabo la evaluación, se empleó un conjunto de datos diverso que incluía tanto rostros normales como rostros con diversas anomalías, como cambios en la iluminación, variaciones en la expresión facial, oclusiones parciales y otras condiciones adversas que pueden afectar la precisión del reconocimiento facial.

Para realizar las pruebas, se empleó una cámara Web HD de 1080P con 5 MP máximos de píxeles como fuente de imágenes. Esta cámara proporcionó una alta calidad de imágenes faciales, lo que permitió una evaluación precisa y confiable de los algoritmos de reconocimiento facial bajo condiciones realistas.

3.3.1.1. Prueba de tiempo de respuesta

Los valores presentados en la tabla 5, 6 y 7 representan el tiempo promedio de ejecución, en milisegundos, obtenidos para cada algoritmo bajo condiciones específicas. El Algoritmo LBPH, el Algoritmo MTCNN y el Algoritmo TensorFlow con Keras fueron

sometidos a pruebas en tres diferentes escenarios: con iluminación, sin iluminación y a una distancia lejana de dos metros.

Cada algoritmo fue sometido a 30 ejecuciones, y se extrajeron los siguientes promedios en milisegundos (ms) para cada escenario específico:

Tiempo de Respuesta									
N° Prueba	Con iluminación			Sin iluminación			Lejos (Dos metros)		
	ms			ms			ms		
	LBPB	MTCNN	TensorFlow con Keras	LBPB	MTCNN	TensorFlow con Keras	LBPB	MTCNN	TensorFlow con Keras
1	0,029	0,026	0,030	0,033	0,020	0,028	0,028	0,020	0,029
2	0,027	0,020	0,026	0,032	0,020	0,030	0,028	0,022	0,030
3	0,035	0,021	0,029	0,031	0,022	0,027	0,030	0,024	0,029
4	0,033	0,020	0,026	0,032	0,018	0,029	0,030	0,021	0,031
5	0,035	0,021	0,029	0,031	0,021	0,029	0,030	0,022	0,029
6	0,032	0,021	0,030	0,035	0,021	0,029	0,030	0,020	0,029
7	0,023	0,022	0,030	0,034	0,020	0,029	0,029	0,019	0,031
8	0,035	0,022	0,027	0,033	0,017	0,030	0,029	0,018	0,029
9	0,034	0,035	0,029	0,033	0,021	0,030	0,029	0,021	0,028
10	0,031	0,030	0,026	0,032	0,033	0,029	0,030	0,032	0,031
11	0,033	0,024	0,030	0,032	0,030	0,030	0,030	0,026	0,030
12	0,033	0,023	0,025	0,033	0,023	0,031	0,027	0,024	0,027
13	0,033	0,024	0,030	0,034	0,022	0,029	0,030	0,023	0,030
14	0,034	0,021	0,028	0,033	0,020	0,030	0,027	0,021	0,030
15	0,033	0,020	0,024	0,035	0,016	0,025	0,027	0,021	0,025
16	0,034	0,019	0,030	0,037	0,020	0,031	0,026	0,017	0,029
17	0,034	0,021	0,030	0,037	0,019	0,030	0,026	0,021	0,029
18	0,032	0,020	0,030	0,036	0,022	0,031	0,027	0,020	0,030
19	0,033	0,020	0,029	0,035	0,021	0,027	0,027	0,021	0,029
20	0,032	0,024	0,025	0,035	0,022	0,028	0,027	0,022	0,025
21	0,034	0,022	0,030	0,035	0,030	0,030	0,029	0,029	0,032
22	0,034	0,031	0,031	0,034	0,030	0,028	0,029	0,033	0,030
23	0,032	0,030	0,026	0,032	0,028	0,027	0,029	0,026	0,030

Tiempo de Respuesta									
Nº Prueba	Con iluminación			Sin iluminación			Lejos (Dos metros)		
	ms			ms			ms		
	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras
24	0,032	0,025	0,031	0,034	0,026	0,030	0,029	0,025	0,029
25	0,030	0,025	0,028	0,037	0,022	0,037	0,029	0,026	0,030
26	0,031	0,022	0,028	0,034	0,022	0,030	0,029	0,021	0,030
27	0,032	0,021	0,027	0,036	0,021	0,028	0,028	0,019	0,030
28	0,031	0,021	0,031	0,036	0,019	0,032	0,030	0,020	0,030
29	0,033	0,019	0,029	0,036	0,017	0,029	0,029	0,027	0,030
30	0,032	0,021	0,029	0,037	0,019	0,030	0,029	0,019	0,031
Promedio:	0,032	0,023	0,028	0,034	0,022	0,029	0,028	0,022	0,029

Tabla 5. Prueba de tiempo de respuesta gemelo 1 y 2.

El método MTCNN mostró el tiempo de ejecución más bajo 0.023 ms, destacando su alta velocidad y eficiencia en el reconocimiento facial de gemelos. TensorFlow con Keras tuvo un tiempo de ejecución de 0.028 ms, también rápido y eficiente. LBPH obtuvo 0.032 ms, ligeramente más alto, pero aún aceptable. MTCNN y TensorFlow con Keras son opciones destacadas para aplicaciones de alta velocidad, mientras que LBPH es adecuado para aplicaciones que priorizan precisión sobre velocidad.

MTCNN es el método más rápido y eficiente con un tiempo de ejecución promedio de 0.022 ms en la detección y reconocimiento facial de gemelos en condiciones de poca luz. TensorFlow con Keras sigue de cerca con 0.029 ms, también mostrando un rendimiento rápido. LBPH tiene un tiempo de ejecución aceptable de 0.034 ms y puede ser adecuado para aplicaciones que no requieran alta velocidad en situaciones de iluminación nula.

MTCNN es el método más rápido y eficiente con un tiempo de ejecución de 0.022 ms en el reconocimiento facial de gemelos a dos metros de distancia. LBPH y TensorFlow con Keras tienen tiempos de ejecución ligeramente mayores (0.028 ms y 0.029 ms,

respectivamente), pero siguen siendo opciones rápidas y eficientes para aplicaciones de reconocimiento facial a larga distancia.

Tiempo de Respuesta									
N° Prueba	Con iluminación			Sin iluminación			Lejos (Dos metros)		
	ms			ms			ms		
	LBP	MTCNN	TensorFlow con Keras	LBP	MTCNN	TensorFlow con Keras	LBP	MTCNN	TensorFlow con Keras
1	0,027	0,02	0,026	0,032	0,02	0,03	0,028	0,022	0,03
2	0,033	0,02	0,026	0,032	0,018	0,029	0,03	0,021	0,031
3	0,032	0,021	0,03	0,035	0,021	0,029	0,03	0,02	0,029
4	0,035	0,022	0,027	0,033	0,017	0,03	0,029	0,018	0,029
5	0,031	0,03	0,026	0,032	0,033	0,029	0,03	0,032	0,031
6	0,033	0,023	0,025	0,033	0,023	0,031	0,027	0,024	0,027
7	0,034	0,021	0,028	0,033	0,02	0,03	0,027	0,021	0,03
8	0,034	0,019	0,03	0,037	0,02	0,031	0,026	0,017	0,029
9	0,032	0,02	0,03	0,036	0,022	0,031	0,027	0,02	0,03
10	0,032	0,024	0,025	0,035	0,022	0,028	0,027	0,022	0,025
11	0,034	0,031	0,031	0,034	0,03	0,028	0,029	0,033	0,03
12	0,032	0,025	0,031	0,034	0,026	0,03	0,029	0,025	0,029
13	0,031	0,022	0,028	0,034	0,022	0,03	0,029	0,021	0,03
14	0,031	0,021	0,031	0,036	0,019	0,032	0,03	0,02	0,03
15	0,032	0,021	0,029	0,037	0,019	0,03	0,029	0,019	0,031
Promedio:	0,032	0,023	0,028	0,034	0,022	0,030	0,028	0,022	0,029

Tabla 6. Prueba de tiempo de respuesta gemelo 1.

Los resultados son similares a los de la tabla 6 en cuanto a iluminación y distancia lejana (dos metros). Sin embargo, varían en condiciones sin iluminación para el algoritmo basado en TensorFlow con Keras.

MTCNN muestra el tiempo de ejecución más bajo (0.022 ms) y alta eficiencia en el reconocimiento facial del gemelo 1 en condiciones de iluminación nula. TensorFlow con

Keras tiene un tiempo de ejecución de 0.030 ms, también rápido y eficiente. LBPH tiene un tiempo de ejecución aceptable de 0.034 ms y es adecuado para aplicaciones que no requieran alta velocidad en condiciones de iluminación escasa o nula.

Tiempo de Respuesta									
Nº Prueba	Con iluminación			Sin iluminación			Lejos (Dos metros)		
	ms			ms			ms		
	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras
1	0,029	0,026	0,03	0,033	0,02	0,028	0,028	0,02	0,029
2	0,035	0,021	0,029	0,031	0,022	0,027	0,03	0,024	0,029
3	0,035	0,021	0,029	0,031	0,021	0,029	0,03	0,022	0,029
4	0,023	0,022	0,03	0,034	0,02	0,029	0,029	0,019	0,031
5	0,034	0,035	0,029	0,033	0,021	0,03	0,029	0,021	0,028
6	0,033	0,024	0,03	0,032	0,03	0,03	0,03	0,026	0,03
7	0,033	0,024	0,03	0,034	0,022	0,029	0,03	0,023	0,03
8	0,033	0,02	0,024	0,035	0,016	0,025	0,027	0,021	0,025
9	0,034	0,021	0,03	0,037	0,019	0,03	0,026	0,021	0,029
10	0,033	0,02	0,029	0,035	0,021	0,027	0,027	0,021	0,029
11	0,034	0,022	0,03	0,035	0,03	0,03	0,029	0,029	0,032
12	0,032	0,03	0,026	0,032	0,028	0,027	0,029	0,026	0,03
13	0,03	0,025	0,028	0,037	0,022	0,037	0,029	0,026	0,03
14	0,032	0,021	0,027	0,036	0,021	0,028	0,028	0,019	0,03
15	0,033	0,019	0,029	0,036	0,017	0,029	0,029	0,027	0,03
Promedio:	0,032	0,023	0,029	0,034	0,022	0,029	0,029	0,023	0,029

Tabla 7. Prueba de tiempo de respuesta gemelo 2.

MTCNN tiene el tiempo de ejecución más bajo, 0.023 ms, en el reconocimiento facial del gemelo 2 en condiciones de iluminación normal. LBPH y TensorFlow con Keras tienen tiempos de ejecución ligeramente más altos, 0.032 ms y 0.029 ms, respectivamente, pero aún ofrecen velocidades notables y son opciones eficientes.

MTCNN tiene el tiempo de ejecución más bajo, 0.022 ms, en el reconocimiento facial del gemelo 2 en condiciones de iluminación nula. LBPH y TensorFlow con Keras tienen tiempos de ejecución ligeramente más altos, 0.034 ms y 0.029 ms, respectivamente, pero aún ofrecen velocidades notables y son opciones eficientes.

Tanto MTCNN como TensorFlow con Keras tienen tiempos de ejecución bajos y muestran alta velocidad y eficiencia en el reconocimiento facial del gemelo 2 a dos metros de distancia. MTCNN es especialmente eficiente en situaciones lejanas, siendo una excelente opción para aplicaciones de seguridad y vigilancia. LBPH también tiene un buen rendimiento en condiciones de larga distancia.

3.3.1.2. Prueba de tasa de aciertos

La tasa de aciertos, definida como el porcentaje de casos en los que los algoritmos clasificaron correctamente los rostros con anomalías y los rostros normales, fue objeto de estudio en el presente análisis. Con el objetivo de obtener resultados precisos y confiables, cada algoritmo fue sometido a 30 ejecuciones, y se calcularon los siguientes porcentajes promedio para cada escenario específico:

Tasa de aciertos									
N° Prueba	Con iluminación %			Sin iluminación %			Lejos (Dos metros) %		
	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras
1	81	98	91	87	98	98	87	97	76
2	83	99	95	88	99	96	84	99	94
3	75	98	99	89	99	96	79	99	90
4	77	98	99	88	99	98	72	98	76
5	75	97	97	89	98	98	72	98	54
6	78	99	97	85	97	97	70	97	53
7	76	98	98	86	96	97	70	98	68
8	79	99	95	87	94	97	70	98	57
9	77	99	96	87	92	97	70	97	81

Tasa de aciertos									
N° Prueba	Con iluminación %			Sin iluminación %			Lejos (Dos metros) %		
	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras
10	77	99	94	88	95	98	71	96	85
11	77	98	93	88	92	98	71	98	87
12	76	98	96	93	93	99	71	97	85
13	77	99	97	87	93	99	70	96	91
14	76	97	95	86	93	99	70	96	94
15	76	97	91	87	93	91	73	97	92
16	78	99	95	85	99	96	70	96	95
17	77	99	95	83	95	99	70	98	97
18	77	95	95	83	99	92	73	98	94
19	78	95	97	84	96	100	73	98	91
20	76	92	95	85	94	90	74	97	88
21	76	95	93	85	98	96	74	98	93
22	78	97	93	85	99	94	73	98	93
23	78	92	88	86	97	98	73	96	94
24	80	96	91	88	98	94	73	96	93
25	79	97	90	86	97	94	71	97	93
26	78	96	90	83	97	93	71	97	91
27	82	90	93	86	93	94	71	98	90
28	79	92	94	83	87	98	71	97	91
29	78	91	96	82	97	96	72	99	90
30	79	99	90	84	98	98	70	98	84
Promedio:	78	97	94	86	96	96	73	97	85

Tabla 8. Prueba de tasa de aciertos gemelo 1 y 2.

Se observó que el método MTCNN obtuvo el promedio más alto de 97%, demostrando una mayor efectividad en la detección de rostros bajo diversas condiciones de iluminación. Por su parte, TensorFlow con Keras alcanzó un promedio de 94%, mostrando también un buen desempeño, aunque ligeramente inferior al método MTCNN. Mientras que LBPH obtuvo un promedio de 78%, lo que indica que este método puede requerir mejoras para enfrentar desafíos de iluminación más complejos.

Los resultados revelan que tanto MTCNN como TensorFlow con Keras obtuvieron promedios iguales de 96%, demostrando un rendimiento sobresaliente y consistente en la detección de rostros de gemelos monocigóticos en entornos de poca iluminación. Estos métodos han demostrado su capacidad para distinguir con precisión entre los gemelos, lo que es fundamental para aplicaciones de identificación y seguridad. Por otro lado, el método LBPH alcanzó un promedio de 86%, mostrando un rendimiento aceptable pero ligeramente inferior a los otros dos métodos evaluados.

MTCNN fue el método más destacado con una precisión promedio del 97%, idóneo para detección y reconocimiento facial a largas distancias en aplicaciones de seguridad y vigilancia. TensorFlow con Keras obtuvo una precisión promedio del 85%, adecuado para situaciones lejanas sin requerir extrema precisión. LBPH obtuvo un promedio del 73%, mejor para distancias cortas y con menor rendimiento en situaciones lejanas.

Tasa de aciertos									
N° Prueba	Con iluminación			Sin iluminación			Lejos (Dos metros)		
	%			%			%		
	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras
1	83	99	95	88	99	96	84	99	94
2	77	98	99	88	99	98	72	98	76
3	78	99	97	85	97	97	70	97	53
4	79	99	95	87	94	97	70	98	57
5	77	99	94	88	95	98	71	96	85

Tasa de aciertos									
N° Prueba	Con iluminación			Sin iluminación			Lejos (Dos metros)		
	%			%			%		
	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras
6	76	98	96	93	93	99	71	97	85
7	76	97	95	86	93	99	70	96	94
8	78	99	95	85	99	96	70	96	95
9	77	95	95	83	99	92	73	98	94
10	76	92	95	85	94	90	74	97	88
11	78	97	93	85	99	94	73	98	93
12	80	96	91	88	98	94	73	96	93
13	78	96	90	83	97	93	71	97	91
14	79	92	94	83	87	98	71	97	91
15	79	99	90	84	98	98	70	98	84
Promedio:	78	97	94	86	96	96	72	97	85

Tabla 9. Prueba de tasa de aciertos gemelo 1.

Para los gemelos número uno, los resultados obtenidos con iluminación y sin iluminación, al igual que los de la tabla 8, arrojan similitudes significativas. La variación relevante se manifiesta al analizar la distancia de dos metros, donde se observa un impacto en el rendimiento del algoritmo LBPH.

En particular, el método MTCNN sigue destacándose con una precisión promedio del 97%, incluso a una distancia de dos metros. Por otro lado, TensorFlow con Keras logra un promedio del 85%, manteniendo un desempeño aceptable en situaciones lejanas. Sin embargo, el método LBPH muestra un promedio del 72% a dos metros de distancia, lo cual indica una disminución significativa en su rendimiento en esta situación. Por

consiguiente, se sugiere que LBPH puede ser más adecuado para aplicaciones que requieran detección facial a distancias más cortas, en contraste con situaciones lejanas.

Tasa de aciertos									
N° Prueba	Con iluminación			Sin iluminación			Lejos (Dos metros)		
	%			%			%		
	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras	LBPH	MTCNN	TensorFlow con Keras
1	81	98	91	87	98	98	87	97	76
2	75	98	99	89	99	96	79	99	90
3	75	97	97	89	98	98	72	98	54
4	76	98	98	86	96	97	70	98	68
5	77	99	96	87	92	97	70	97	81
6	77	98	93	88	92	98	71	98	87
7	77	99	97	87	93	99	70	96	91
8	76	97	91	87	93	91	73	97	92
9	77	99	95	83	95	99	70	98	97
10	78	95	97	84	96	100	73	98	91
11	76	95	93	85	98	96	74	98	93
12	78	92	88	86	97	98	73	96	94
13	79	97	90	86	97	94	71	97	93
14	82	90	93	86	93	94	71	98	90
15	78	91	96	82	97	96	72	99	90
Promedio:	77	96	94	86	96	97	73	98	86

Tabla 10. Prueba de tasa de aciertos gemelo 2.

MTCNN obtuvo el promedio más alto 96% en la detección precisa del gemelo 2 en diversas condiciones de iluminación. TensorFlow con Keras alcanzó un promedio de 94%, también mostrando un buen rendimiento en la detección del gemelo 2. En cambio,

LBPH obtuvo un promedio de 77%, indicando la necesidad de mejoras para el reconocimiento facial de gemelos monocigóticos en condiciones de iluminación variadas. MTCNN y TensorFlow con Keras lograron destacados promedios de 96 y 97, respectivamente, en la detección de rostros del gemelo 2 en condiciones de iluminación nula. Esto es crucial para identificar gemelos monocigóticos en escenarios con poca o ninguna iluminación. LBPH obtuvo un promedio de 86, un rendimiento aceptable, aunque ligeramente inferior. Sin embargo, sigue siendo una opción viable, especialmente en condiciones de iluminación no extremadamente desfavorables.

MTCNN obtuvo un promedio excepcional de 98 en la detección de rostros del gemelo 2 a dos metros de distancia, destacando su alta capacidad para este desafío. TensorFlow con Keras alcanzó un promedio de 86, con buen rendimiento en distancia lejana. LBPH obtuvo un promedio de 73, más adecuado para distancias cortas y necesitando mejoras para reconocimiento a larga distancia.

Los resultados revelaron que MTCNN obtuvo la tasa de aciertos más alta entre los tres algoritmos evaluados. Su capacidad para detectar y clasificar rostros con anomalías fue destacable, lo que sugiere que es una elección prometedora para aplicaciones que requieren detección precisa de características faciales inusuales.

Por otro lado, TensorFlow con Keras mostró un desempeño sólido, aunque ligeramente inferior al de MTCNN. Sin embargo, sigue siendo una opción viable para aplicaciones de reconocimiento facial.

LBPH, por su parte, demostró un rendimiento aceptable pero menor en comparación con los otros dos algoritmos, lo que sugiere que podría no ser la mejor opción para tareas de detección de anomalías en rostros humanos.

Conclusiones

- Se analizó la estructura de código para diferenciar gemelos monocigóticos mediante reconocimiento facial. El dataset diverso fue fundamental para entrenar los modelos basados en LBPH, MTCNN y TensorFlow-Keras, que han demostrado resultados prometedores. El sistema presenta aplicaciones en seguridad, control de accesos y autenticación biométrica. Se sugiere continuar mejorando el dataset y explorar nuevos algoritmos para futuras versiones.

- Al realizar la evaluación de los tres algoritmos de reconocimiento facial para diferenciar gemelos monocigóticos en diversas condiciones. Los resultados obtenidos proporcionan información valiosa sobre la eficiencia y rendimiento de cada algoritmo. MTCNN se destaca como el más rápido y eficiente, seguido de cerca por TensorFlow con Keras, mientras que LBPH prioriza la precisión sobre la velocidad.
- Los resultados obtenidos en el estudio revelaron que el algoritmo MTCNN destacó notablemente, presentando la tasa de aciertos más alta y una mayor precisión en la detección de rostros con anomalías en gemelos monocigóticos. TensorFlow con Keras también mostró un rendimiento sólido, aunque levemente inferior al de MTCNN. Por otro lado, LBPH, si bien obtuvo un rendimiento aceptable, mostró un desempeño inferior en comparación con los otros dos algoritmos, sugiriendo que puede no ser la opción más adecuada para tareas de detección de anomalías en rostros humanos.

Recomendaciones

- Ampliar y diversificar el conjunto de datos: Recopilar un conjunto de datos que incluya imágenes de gemelos monocigóticos y no monocigóticos, con variaciones en la iluminación, expresiones faciales y ángulos. Esto asegurará una mayor generalización del modelo y mejorará la capacidad para identificar diferencias sutiles entre los gemelos.
- Implementar técnicas de preprocesamiento de imágenes: Aplicar técnicas de preprocesamiento, como normalización de la iluminación, eliminación de ruido y alineación de rostros, para mejorar la calidad de las imágenes y garantizar una representación estandarizada.
- Incorporar técnicas de reducción de dimensionalidad: Utilizar técnicas de reducción de dimensionalidad, como PCA o t-SNE, para reducir la alta dimensionalidad de los datos y mejorar la eficiencia del modelo sin comprometer la precisión.
- Evaluar y comparar diferentes algoritmos de detección de datos atípicos: Analizar y comparar algoritmos de detección de datos atípicos, como Isolation Forest, One-Class SVM y Local Outlier Factor, para identificar y manejar eficazmente

gemelos con características atípicas que puedan afectar el rendimiento del sistema.

- Realizar pruebas rigurosas y validación cruzada: Realizar pruebas exhaustivas y utilizar validación cruzada con diferentes conjuntos de datos para evaluar el rendimiento del modelo en términos de precisión, sensibilidad y especificidad.
- Proporcionar un mecanismo de retroalimentación y corrección: Implementar un mecanismo de retroalimentación que permita a los usuarios corregir posibles errores de identificación, lo que permitirá mejorar continuamente la precisión y robustez del sistema.
- Garantizar la privacidad y seguridad de los datos: Cumplir con las normativas y regulaciones de privacidad de datos, implementar técnicas de cifrado y protección para salvaguardar la información biométrica de los usuarios y garantizar la confidencialidad de los datos.

Bibliografía

- [1] S. Mousavi, M. Charmi y H. Hossein, Artists, *Recognition of identical twins based on the most distinctive region of the face: Human criteria and machine processing approaches*. [Art]. *Multimedia Tools and Applications* 80, 15765–15802, 2021.
- [2] K. Nahar, B. Abul-Huda, A. Al.Batined y R. AI-Khatib, Artists, *Twins and Similar Faces Recognition Using Geometric and Photometric Features with Transfer Learning*. [Art]. *International Journal of Computing and Digital System*, 2022.
- [3] A. Afaneh , F. Noroozi y O. Toygar, Artists, *Recognition of identical twins using fusion of various facial feature extractors*. [Art]. *EURASIP Journal on Image and Video Processing*, 2017.
- [4] S. Mousavi, M. Charmi y H. Hassanpoor, Artists, *A distinctive landmark-based face recognition system for identical twins by extracting novel weighted features*. [Art]. *Computers & Electrical Engineering*, 2021.
- [5] L. Blazquez Perez, Artist, *Reconocimiento Facial Basado en Puntos Característicos de la Cara en entornos no controlados*. [Art]. Universidad Autonoma De Madrid Escuela Politecnica Superior, 2013.
- [6] C. G. Noguera Andrade, Artist, *Autenticacion por Reconocimiento Facial para Aplicaciones Web, Utilizando Software Libre*. [Art]. Universidad Pontificia Bolivariana, 2012.
- [7] «Recursos python,» 19 abril 2014. [En línea]. Available: <https://recursospython.com/guias-y-manuales/instalacion-y-utilizacion-de-pip-en-windows-linux-y-os-x/>.
- [8] «OpenCV,» 2022. [En línea]. Available: <https://opencv.org/about/>.
- [9] «Python,» 2022. [En línea]. Available: <https://www.python.org/doc/essays/blurb/>.
- [10] R. KeepCoding, «Keepcoding Tech School,» 8 Agosto 2022. [En línea]. Available: <https://keepcoding.io/blog/para-que-sirve-tensorflow/#:~:text=TensorFlow%20es%20un%20framework%20desarrollado,en%20una%20CPU%20o%20GPU..>
- [11] «Keras,» 2022. [En línea]. Available: <https://keras.io/>.
- [12] «Microsoft,» 2022. [En línea]. Available: <https://visualstudio.microsoft.com/es/vs/features/cplusplus/>.
- [13] L. N. 1. R.O, Artist, *Reglamento del Centro de Investigacion de Sistema y Telecomunicacion*. [Art]. Universidad Estatal Peninsula de Santa Elena, 2019.
- [14] J. D. Alonso Sierra, Artist, *Sistema de Reconocimiento Facial para Control de Acceso a viviendas*. [Art]. Universidad Catolica de Colombia, 2019.
- [15] P. Kanchan, Artist, *A Survey on Face Recognition of Identical Twins*. [Art]. *International Journal of Scientific & Engineering Research*, 2015.

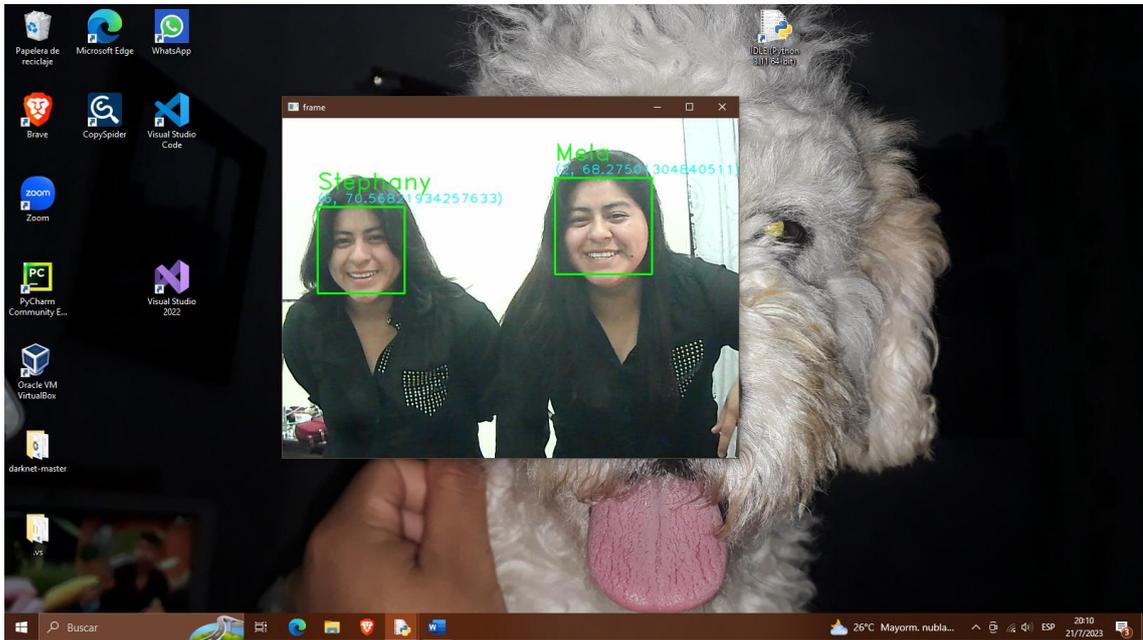
- [16] S. N. d. Planificacion, Artist, *Plan de Creacion de Oportunidades*. [Art]. 2021-2025.
- [17] N. Mohammad Naghi, Metodologia de la investigacion, Mexico: Limusa, S.A, 2005.
- [18] L. Hoory y C. Bottorff, «forbes,» 25 Marzo 2022. [En línea]. Available: <https://www.forbes.com/advisor/business/what-is-waterfall-methodology/>. [Último acceso: 25 Junio 2023].
- [19] I. n. d. e. y. censos, «Fasciculo Provincial Santa Elena,» pp. 1-8, 2010.
- [20] A. d. S. Elena, «Alcaldía de Santa Elena,» [En línea]. Available: <http://www.gadse.gob.ec/gadse/index.php/mision-y-vision-2/>.
- [21] C. D. P. C. Y. C. SOCIAL, Artist, *GUÍA ESPECIALIZADA DE RENDICIÓN DE CUENTAS PARA LOS GOBIERNOS AUTÓNOMOS DESCENTRALIZADOS*. [Art]. CONSEJO DE PARTICIPACIÓN CIUDADANA Y CONTROL SOCIAL.
- [22] J. Brownlee , «Machine Learning Mastery,» Python Machine Learning, 25 Marzo 2020. [En línea]. Available: [https://machinelearningmastery.com/distance-measures-for-machine-learning/#:~:text=Euclidean%20distance%20is%20calculated%20as,v2%5Bi%5D\)%5E2](https://machinelearningmastery.com/distance-measures-for-machine-learning/#:~:text=Euclidean%20distance%20is%20calculated%20as,v2%5Bi%5D)%5E2).
- [23] D. Lemonaki, «freeCodeCamp,» 24 Agosto 2021. [En línea]. Available: <https://www.freecodecamp.org/news/what-is-an-outlier-definition-and-how-to-find-outliers-in-statistics/>.
- [24] «honorlock,» 29 Octubre 2021. [En línea]. Available: <https://honorlock.com/blog/face-detection-vs-face-recognition-in-online-proctoring/>.
- [25] P. Singh, «analyticsvidhya,» 12 Julio 2021. [En línea]. Available: <https://www.analyticsvidhya.com/blog/2021/07/understanding-face-recognition-using-lbph-algorithm/>.
- [26] M. Vega, «medium,» 12 Mayo 2021. [En línea]. Available: <https://medium.com/lcc-unison/aplicaci%C3%B3n-de-reconocimiento-facial-en-tiempo-real-para-identificar-a-alumnos-de-la-licenciatura-en-2e2b53870995>.
- [27] TensorFlow, «TensorFlow,» 2023. [En línea]. Available: <https://www.tensorflow.org/overview?hl=es-419>.
- [28] simplilearn, «simplilearn,» 1 Abril 2023. [En línea]. Available: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras>.
- [29] GeeksforGeeks, «GeeksforGeeks,» 26 Octubre 2022. [En línea]. Available: <https://www.geeksforgeeks.org/opencv-overview/>.
- [30] Coursera, «Coursera,» 14 Abril 2023. [En línea]. Available: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>.
- [31] softwarekeep, «softwarekeep,» 2023. [En línea]. Available: <https://softwarekeep.com/help-center/what-is-microsoft-visual-studio-where-can-i-download-it>.

- [32] abhilkhnathdas, «geeksforgeeks,» 19 Diciembre 2022. [En línea]. Available: <https://www.geeksforgeeks.org/python-introduction-to-pyqt5/>.
- [33] docs.python, «Python,» 10 Mayo 2023. [En línea]. Available: <https://docs.python.org/3/library/sqlite3.html>.
- [34] G. Learning, «Great Learning,» 22 Marzo 2022. [En línea]. Available: <https://www.mygreatlearning.com/blog/resnet/>.
- [35] A. Pujara, «builtin,» 26 Enero 2023. [En línea]. Available: <https://builtin.com/machine-learning/mobilenet>.
- [36] Google, «Google research,» 2023. [En línea]. Available: <https://research.google.com/colaboratory/faq.html#:~:text=linkdone,learning%2C%20data%20analysis%20and%20education..>
- [37] D. Ellis, «hubspot,» 20 Enero 2023. [En línea]. Available: <https://blog.hubspot.com/website/anaconda-python>.
- [38] Incredibuild, «Incredibuild,» 2023. [En línea]. Available: <https://www.incredibuild.com/integrations/cuda>.
- [39] A. Zola, «Techtarget,» Abril 2022. [En línea]. Available: <https://www.techtarget.com/searchsoftwarequality/definition/development-environment>.
- [40] M. Breuss, «Realpython,» 2022. [En línea]. Available: <https://realpython.com/python-virtual-environments-a-primer/#what-is-a-python-virtual-environment>.
- [41] H. Sutaria, «Medium,» 31 Enero 2022. [En línea]. Available: <https://medium.com/@hastisutaria.ict19/object-detection-using-opencv-f94f61e88b23#:~:text=Object%20detection%20is%20a%20computer,all%20while%20accurately%20labeling%20them..>
- [42] Roboticsbiz, «Roboticsbiz,» 7 Febrero 2022. [En línea]. Available: <https://roboticsbiz.com/five-key-components-of-a-machine-vision-system/>.
- [43] N. Abudarham, L. Shkiller y G. Yovel, Artists, *Critical features for face recognition*. [Art]. 2019.
- [44] J. C. GÓMEZ BARRERA, Artist, *Psicometría, perfiles y sesgos. El caso del reconocimiento facial*. [Art]. Inmediaciones de la Comunicación, 2021.
- [45] R. Gonzalez Lozano, Artist, *Sistema de reconocimiento facial con Deep Learning*. [Art]. 2022.
- [46] «Techtarget,» 2022. [En línea]. Available: [//www.techtarget.com/whatis/definition/dark-web](https://www.techtarget.com/whatis/definition/dark-web).
- [47] «Google Maps,» [En línea]. Available: <https://www.google.com.ec/maps/place/Santa+Elena/@-2.1153681,-80.6019671,132761m/data=!3m1!1e3!4m6!3m5!1s0x902de9c595c373f9:0x7320449f38731f76!8m2!3d-2.2267105!4d-80.859499!16s%2Fm%2F0463b0w!5m1!1e4?hl=es>.
- [48] M. Build, «Microsoft,» 25 Mayo 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/cognitive-services/computer-vision/how-to/specify-recognition-model>.

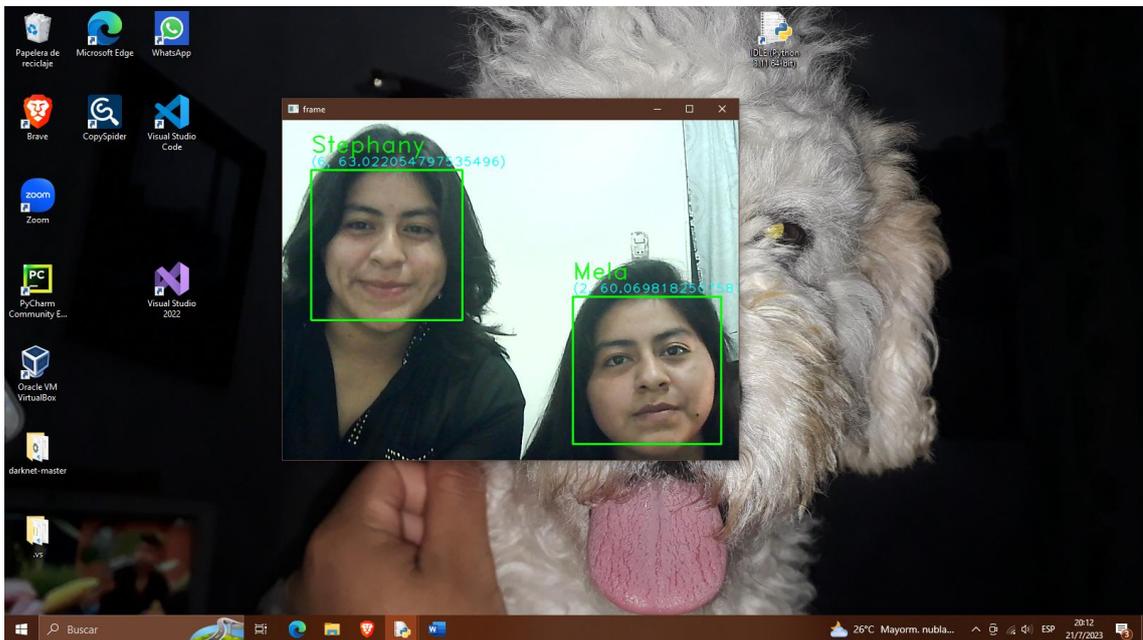
- [49] N. M. RAHARJA, M. A. FATHANSYAH y A. N. N. CHAMIM, Artists, *Sistema de seguridad de estacionamiento de vehículos con detección de reconocimiento facial basado en algoritmo eigenface.* [Art]. Revista de Robótica y Control (JRC), 2022.
- [50] I. Aliyu, M. Maishanu y M. A. Bomoi, Artists, *A Comparative Study of Eigenface and Fisherface Algorithms Based on OpenCV and Sci-kit Libraries Implementations.* [Art]. Department of Mathematical Sciences Abubakar Tafawa Balewa University Bauchi, 2022.
- [51] M. Atri, A. A. Falou, M. Jridi y Y. Kortli, Artists, *Face Recognition Systems: A Survey.* [Art]. University of Monastir, 2022.
- [52] L. A. Nieto, «DETECCIÓN DE PERSONAS / INTELIGENCIA ARTIFICIAL - OpenCV (HOG + SVM),» 17 Septiembre 2020. [En línea]. Available: <https://es.linkedin.com/pulse/detecci%C3%B3n-de-personas-inteligencia-artificial-opencv-nieto>.
- [53] Y. L. B. Y. Y. L. H. S. J. & C. J. Li, Artist, *3E-LDA: three enhancements to linear discriminant analysis.* [Art]. Transactions on Knowledge Discovery from Data (TKDD), 2021.
- [54] F. Tandia, Artist, *Dimensionality Reduction: PCA VS LDA for Face Recognition.* [Art]. Software Engineer at Trend Power Technology Co, 2020.

ANEXOS

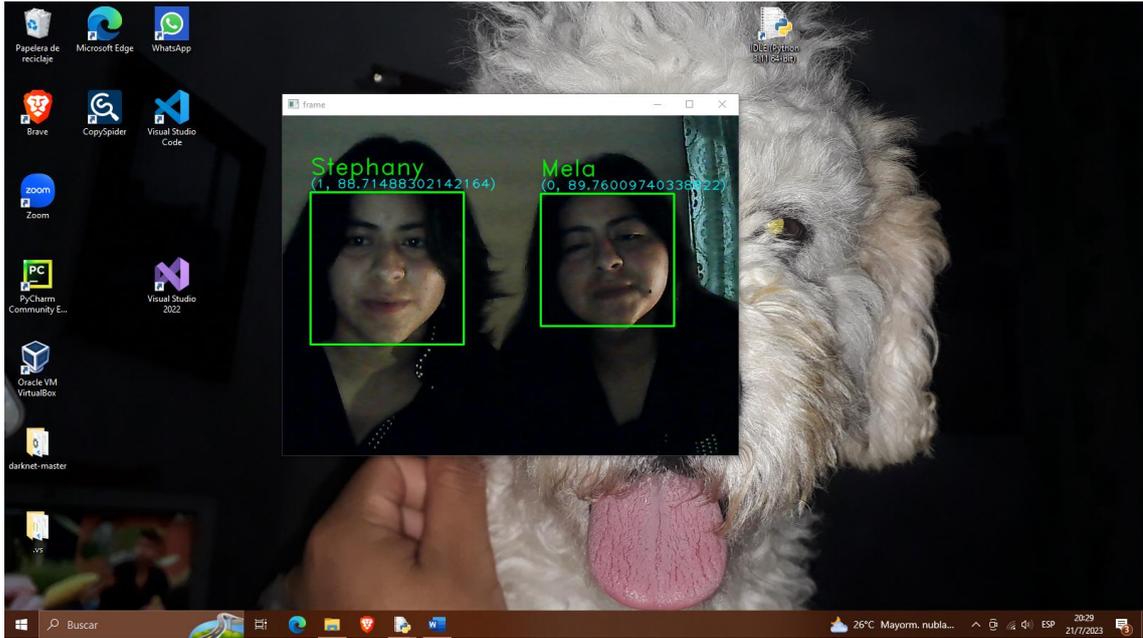
Anexos 1. Prueba del algoritmo de reconocimiento facial basado en LBPH



(a)



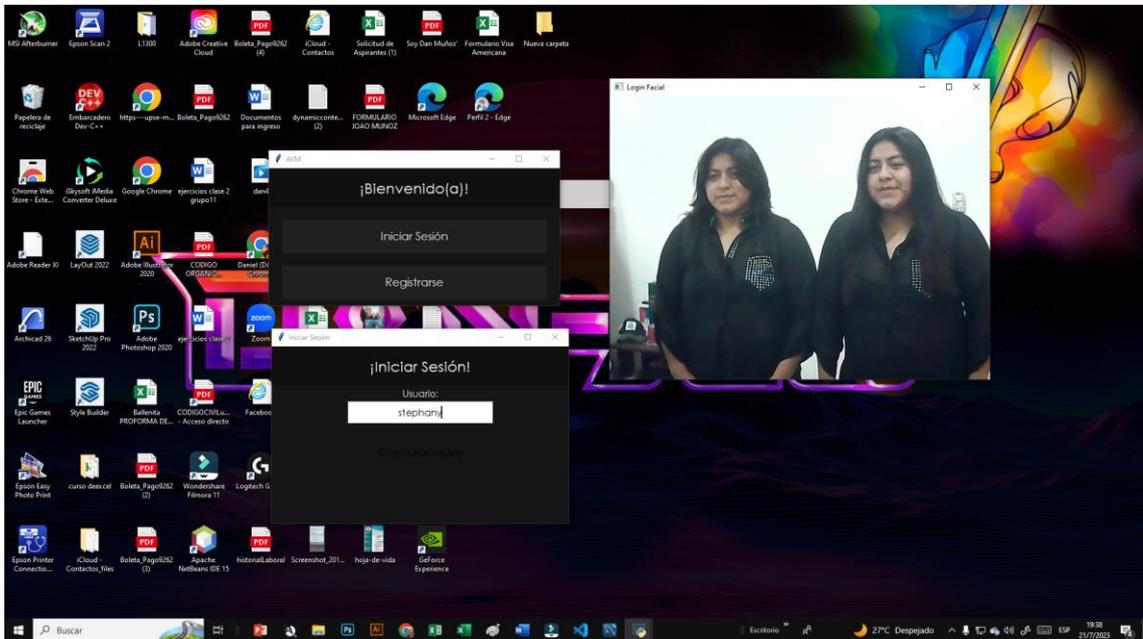
(b)



(c)

Figura 17. Prueba del algoritmo de reconocimiento facial basado en LBPH
 Anexo 1. Prueba del algoritmo de reconocimiento facial basado en LBPH

Anexos 2. Prueba del algoritmo de reconocimiento facial basado en MTCNN



(a)



(b)

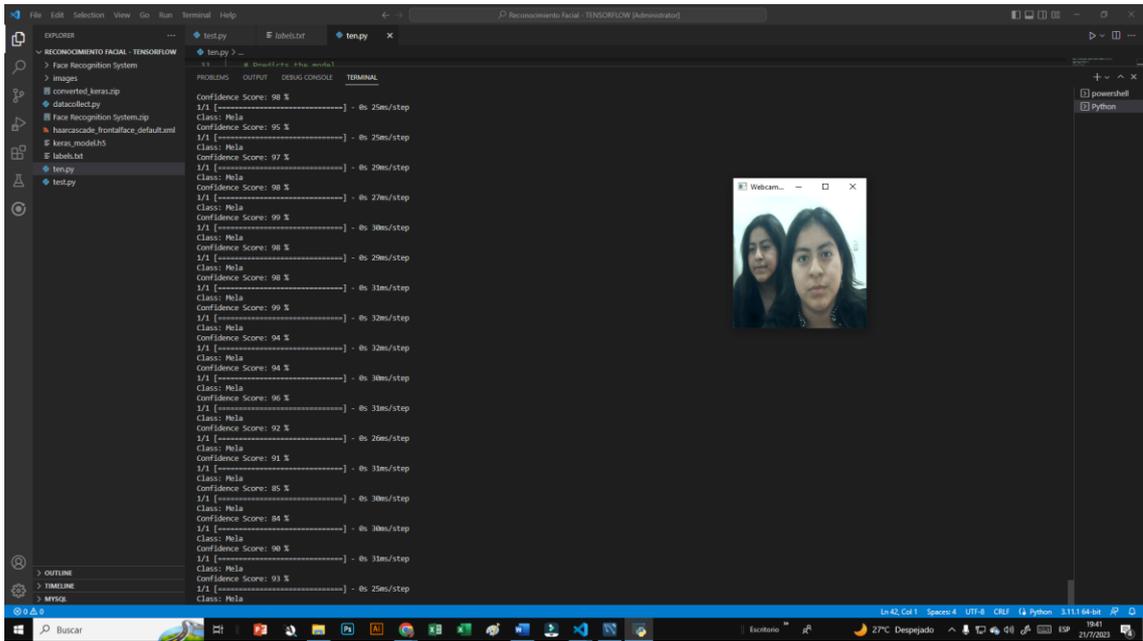


(c)

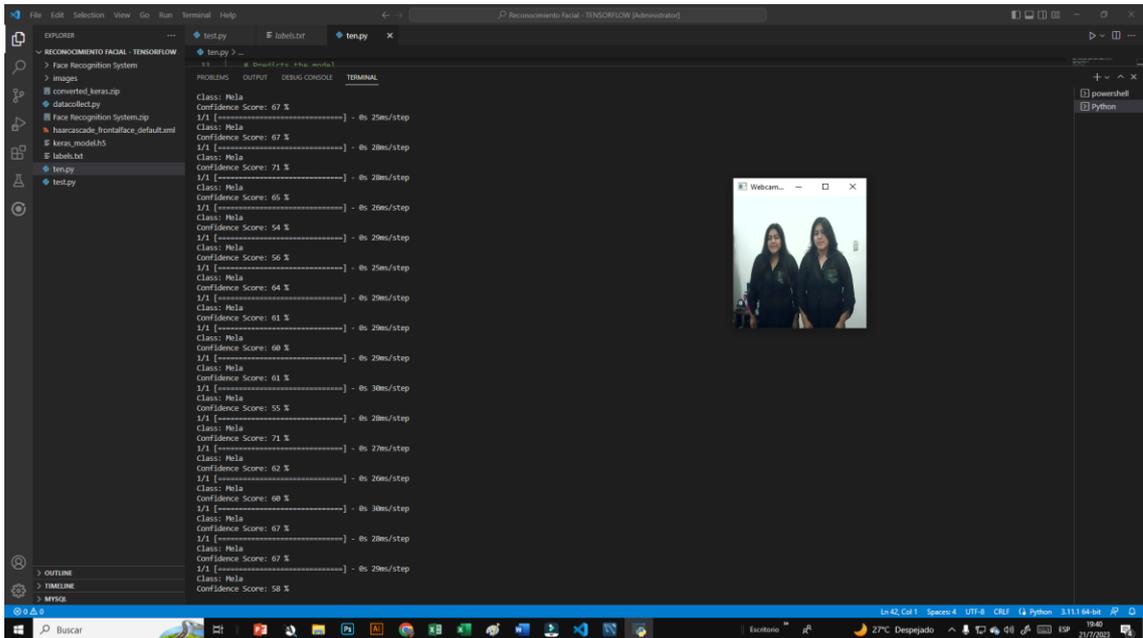
Figura 18. Prueba del algoritmo de reconocimiento facial basado en MTCNN

Anexo 2. Prueba del algoritmo de reconocimiento facial basado en MTCNN

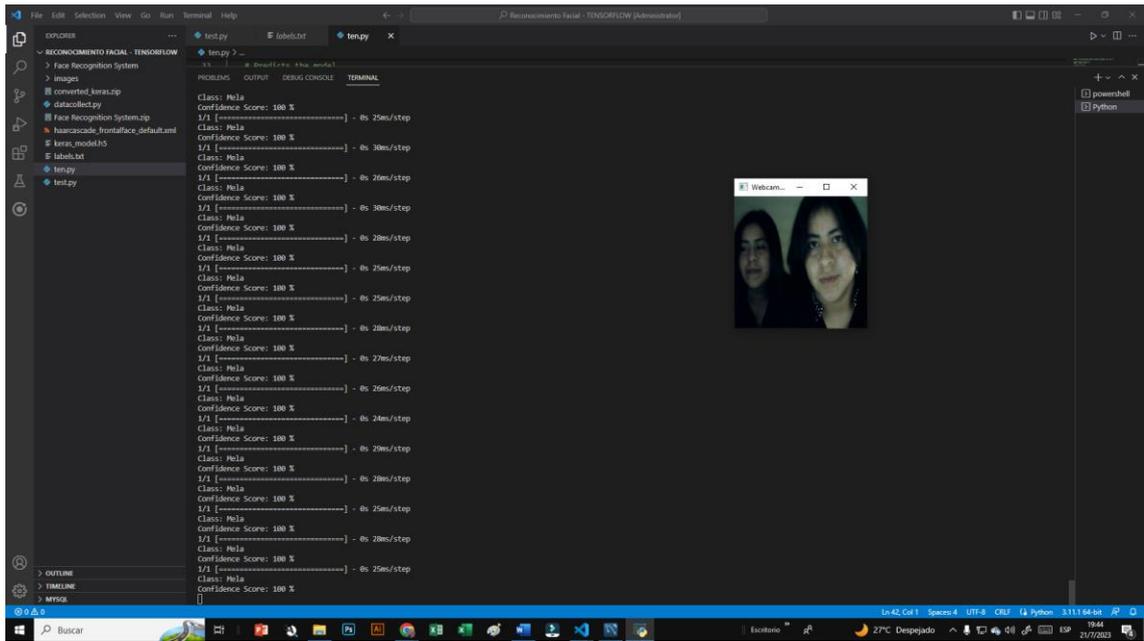
Anexos 3. Prueba del algoritmo de reconocimiento facial basado en TensorFlow con Keras



(a)



(b)



(c)

Figura 19. Prueba del algoritmo de reconocimiento facial basado en TensorFlow con Keras

Anexo 3. Prueba del algoritmo de reconocimiento facial basado en TensorFlow con Keras



Proyecto UIC - Lesly Stephany Muñoz Alejandro

5% Similitudes **Texto entre comillas**
< 1% < 1% similitudes entre comillas
7% Idioma no reconocido

Nombre del documento: Proyecto UIC - Lesly Stephany Muñoz Alejandro.docx	Depositante: SHENDRY BALMORE ROSERO VASQUEZ	Número de palabras: 17.595
ID del documento: 8ab93ede065e224b500271e29f5fe57071952511	Fecha de depósito: 2/8/2023	Número de caracteres: 119.956
Tamaño del documento original: 19,9 MB	Tipo de carga: interface	
	fecha de fin de análisis: 2/8/2023	

Ubicación de las similitudes en el documento:



Fuentes

Fuentes principales detectadas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	repositorio.upse.edu.ec https://repositorio.upse.edu.ec/bitstream/46000/9268/1/UPSE-TTI-2023-0008.pdf 19 fuentes similares	2%		Palabras idénticas : 2% (338 palabras)
2	gadpsanmateo.gob.ec http://gadpsanmateo.gob.ec/images/RENDICIÓN_DE_CUENTAS___GUIA_CIUDADANIAcopia.pdf 8 fuentes similares	1%		Palabras idénticas : 1% (225 palabras)
3	Documento de otro usuario #deb35a El documento proviene de otro grupo 5 fuentes similares	< 1%		Palabras idénticas : < 1% (111 palabras)
4	Documento de otro usuario #53ecc3 El documento proviene de otro grupo 4 fuentes similares	< 1%		Palabras idénticas : < 1% (29 palabras)
5	Documento de otro usuario #c2ba34 El documento proviene de otro grupo 2 fuentes similares	< 1%		Palabras idénticas : < 1% (51 palabras)

Fuentes con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	arxiv.org Twin identification over viewpoint change: A deep convolutional neural n... http://arxiv.org/abs/2207.05316	< 1%		Palabras idénticas : < 1% (38 palabras)
2	Documento de otro usuario #8095d0 El documento proviene de otro grupo	< 1%		Palabras idénticas : < 1% (19 palabras)
3	Documento de otro usuario #4c138d El documento proviene de otro grupo	< 1%		Palabras idénticas : < 1% (16 palabras)
4	shushufindi.gob.ec http://shushufindi.gob.ec/portal/wp-content/uploads/2021/06/no_cpccs-ple-sg-078-2021-561_-confor...	< 1%		Palabras idénticas : < 1% (29 palabras)
5	repositorio.uam.es https://repositorio.uam.es/bitstream/handle/10486/12092/59237_20130206LuisBlazquezPerez.pdf?se...	< 1%		Palabras idénticas : < 1% (22 palabras)

Fuentes mencionadas (sin similitudes detectadas)

Estas fuentes han sido citadas en el documento sin encontrar similitudes.

- <https://recursospython.com/guias-y-manuales/instalacion-y-utilizacion-de-pip-en-windows-linux-y-os-x/>
- <https://opencv.org/about/>
- <https://www.python.org/doc/essays/blurb/>
- <https://keras.io/>
- <https://visualstudio.microsoft.com/es/vs/features/cplusplus/>

Figura 20. Certificado Antiplagio.