TESIS UCH

UCH UNIVERSIDAD DE CIENCIAS Y HUMANIDADES

FACULTAD DE CIENCIAS E INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA CON MENCIÓN EN TELECOMUNICACIONES

TESIS

Para optar el título profesional de Ingeniero Electrónico con Mención en Telecomunicaciones

Diseño e implementación de un sistema de reconocimiento facial utilizando software e interfaces con alerta remota para dispositivos móviles

PRESENTADO POR

Huaraca Tadeo, Jhonny Betto

ASESOR

Zamora Villaorduña, Eduardo Manuel
0009-0001-8565-9933

Lima, Perú, 2023

Tesis publicada con autorización de los autores
no olvide citar esta tesis

www.uch.edu.pe

INFORME DE ORIGINALIDAD ANTIPLAGIO TURNITIN

Mediante la presente, Yo:

1. Nombres y Apellidos; JHONNY BETTO HUARACA TADEO con DNI 42591762

Soy egresados de la Escuela Profesional de Ingeniería Electrónica con mención en telecomunicaciones el 11 de octubre del año 2021, y habiendo realizado la[1] Tesis para optar el Título Profesional de [2]ingeniero electrónico, se deja constancia que el trabajo de investigación fue sometido a la evaluación del Sistema Antiplagio Turnitin el 17 de febrero del 2024, el cual ha generado el siguiente porcentaje de similitud de [3]: 15% de coincidencias

En señal de conformidad con lo declarado, firmo el presente documento a los 19 días del mes de abril del año 2024.

JHONNY BETTO HUARACA TADEO
DNI: 42591762

EDUARDO MANUEL ZAMORA VILLAORDUÑA
DNI:10581328

[1] Especificar qué tipo de trabajo es: tesis (para optar el título).

[2] Indicar el titulo o grado académico: Ingeniero Electrónico.

[3] Se emite la presente declaración en virtud de lo dispuesto en el artículo 8°, numeral 8.2, tercer párrafo, del Reglamento del Registro Nacional de Trabajos conducentes a Grados y Títulos – RENATI, aprobado mediante Resolución de Consejo Directivo N° 033-2016-SUNEDU/CD, modificado por Resolución de Consejo Directivo N° 174-2019-SUNEDU/CD y Resolución de Consejo Directivo N° 084-2022-SUNEDU/CD.

TESIS 15-02

INFORME DE ORIGINALIDAD

15%	14%	2%	6%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	repositorio.uch.edu.pe Fuente de Internet	4%
2	Submitted to Universidad de Ciencias y Humanidades Trabajo del estudiante	1%
3	dspace.ups.edu.ec Fuente de Internet	1%
4	elperuanolegal.blogspot.com Fuente de Internet	1%
5	hdl.handle.net Fuente de Internet	1%
6	Submitted to Escuela Politecnica Nacional Trabajo del estudiante	<1%
7	repositorio.uees.edu.ec Fuente de Internet	<1%
8	repositorio.ucsg.edu.ec Fuente de Internet	<1%
9	pki.reniec.gob.pe Fuente de Internet	

COMPROMISO

A mi madre, FELICITA SERAPIA TADEO ALBORNOS ORE, quien es madre y padre para mi. Persona que me brindó su apoyo constante e incondicional en esta etapa de mi carrera profesional, a pesar de las adversidades. Es a quien dedico mi tesis.

AGRADECIMIENTO

Especialmente quiero agradecer a la Universidad de Ciencias y Humanidades por abrirme los brazos para que pudiera recibir una formación profesional, Gracias.

Resumen

Entre marzo del 2016 y el 24 de enero 2019 se registraron a nivel nacional 2.539 delincuentes con orden de captura por el Ministerio del interior (MININTER), donde solo fueron capturados 1413 y 1126 se encontraron en libertad generando inseguridad ciudadana en el Perú, siendo el 25.5% de la población entre 15 y más años en algún momento han sido víctimas por aquellos criminales que se encuentran libres por alguna modalidad delictiva. Debido a las razones anteriores, es necesario reducir el índice de inseguridad ciudadana en la sociedad.

Así mismo, el estado destina dinero en programa de recompensa por delincuentes con orden de captura, generándose comunicación de diversa índole, inclusive información inexacta y por ende movilizando a la policía de manera innecesaria, impactando en el gasto de recursos.

La presente investigación tiene como objetivo realizar Diseño e Implementación de un Sistema de Reconocimiento Facial utilizando Software e Interfaces con alerta remota para dispositivos móviles. Así mismo, se realizará comparativas de imágenes faciales en tiempo real con las imágenes faciales almacenadas, detectando si existe similitud con la base de datos, donde podrá hacer uso del lenguaje de programación Python, también empleándose el algoritmo Local Binary Patterns Histogram (LBPH). Aplicado en aprendizaje profundo estando incluido en la librería CV2 utilizando la Jetson nano como placa base para el procesado del software y como hardware se implementará la Cámara Raspberry PI REV 1.3 para el reconocimiento facial de delincuentes con orden de captura que tendrá como alerta remota mediante un mensaje de texto a un dispositivo vinculado.

Palabras claves: Reconocimiento Facial, Algoritmo, Lenguaje de programación, detección de facial.

ABSTRACT

Between March 2016 and January 24, 2019, 2,539 criminals were registered nationwide with an arrest warrant by the Ministry of the Interior (MININTER), where only 1,413 were captured and 1,126 were found free, generating citizen insecurity in Peru, being the 25.5% of the population between 15 years of age and older have at some point been victims of those criminals who are free for some type of crime. Due to the above reasons, it is necessary to reduce the rate of citizen insecurity in society.

Likewise, the state allocates money in a reward program for criminals with an arrest warrant, generating communication of various kinds, including inaccurate information, and therefore mobilizing the police unnecessarily, impacting the expenditure of resources.

The objective of this research is to carry out the Design and Implementation of a Facial Recognition System using Software and Interfaces with remote alert for mobile devices. Likewise, it will make comparisons of facial images in real time with the stored facial images, detecting if there is similarity with the database, where it can use the Python programming language, also using the LOCAL BINARY PATTERNS HISTOGRAM (LBPH) algorithm. Applied in deep learning, being included in the CV2 library using the Jetson Nano as a motherboard for software processing and as hardware, the Raspberry PI REV 1.3 Camera will be implemented for the facial recognition of criminals with an arrest warrant that will be used as a remote alert through a text message to a linked device.

Keywords: Facial Recognition, Algorithm, Programming Language, Facial Detection.

Índice

LISTA DE FIGURAS

LISTA DE TABLAS

INTRODUCCIÓN

En América del Norte, el servicio de Investigación y control de Aduanas (ICE) donde se capturan a 162 criminales prófugos de la justicia. La institución destaco que aproximadamente el 90% de apresados, donde 143 criminales tenían orden de captura. En el Perú Entre marzo del 2016 y el 24 de enero 2019 se registraron a nivel nacional 2.539 delincuentes con orden de captura por el Ministerio del Interior (MININTER), donde solo fueron capturados 1413 y 1126 se encontraron en libertad generando inseguridad ciudadana. En el Perú, se viene incrementando de forma exponencial el aumento de delincuentes con orden de captura. Ante la ausencia de seguridad social, el proyecto de investigación realizará el diseño e implementación de un sistema de reconocimiento facial mediante software e interfaz con alerta remota para dispositivos móviles que permitirá el reconocimiento facial. Para el presente proyecto, se utilizará como placa la Jetson nano para el procesado del software y como hardware se implementará la cámara (Raspberry PI REV 1.3), que es el modelo de cámara de la Raspberry y también puede emplearse en la Jetson nano.

En el año 2020, Manuel Briones de Ecuador, realizó un sistema que permite acceder al control autentificado en un centro comercial con el fin de capturar criminales que tienen persecución con la justicia. Como solución propuesta, el grupo de investigación preestableció una base de datos, en donde utilizó la técnica de Machine Learning, donde se procesa los datos en forma automática, y un sistema de capacidad de conjunto de dato por medio de extracción de acierto de patrones. De esa forma estar anexada al centro de operación base de alerta, también se utilizó el lenguaje de programación Python y la librería OpenCV haciendo un reconocimiento facial del 99.38% eficaz. Para el presente proyecto de investigación, se utilizará el lenguaje de Python y la librería libre OpenCV.

En su estudio Velarde (2018), desarrolló un sistema de reconocimiento facial para el control de trata de personas en el Perú. Utilizó el algoritmo de clasificación en cascada centrado en el tipo Haar que permitió detectar el rostro de personas desaparecidas, utilizando el algoritmo Local Binary Patterns Histogram (LBPH) mediante la librería OpenCV, colocando parámetros del algoritmo en cascada, haciendo uso del archivo CSV con listado de imagen de

identificador de cámara web. Efectuando la petición de muestra de reconocimiento facial a desarrollarse como es el (LBPH), permitiendo predecir el rostro que desea identificar. Resultado reconocimiento facial mediante el (LBPH) con una efectividad de 86.75%. El presente trabajo de investigación tiene como objetivo realizar el Algoritmo Local Binary Patterns Histogran (LBPH) aplicado en el aprendizaje profundo incluido en la librería CV2.

La solución propuesta en el presente trabajo es la tecnología de algoritmo (LBPH) de reconocimiento facial para la detección de delincuentes con orden de captura. El método propuesto es diseño e implementación de un sistema de reconocimiento facial mediante software e interfaz con alerta remota para dispositivos móviles que permitirá el reconocimiento facial. Un software capaz de detectar el rostro del criminal, utilizando herramienta la Jetson Nano como placa base para el procesado de software y como hardware se implementará la PI cámara (Raspberry PI REV 1.3), que es el modelo de cámara de la RASPBERRY y también puede emplearse en Jetson Nano. En la placa base se incorpora el lenguaje de programación Python de aprendizaje profundo mediante la librería CV2, permitiendo el reconocimiento facial en tiempo real.

CAPÍTULO I: PLANTEAMIENTO Y FORMULACÍON DEL PROBLEMA

1.1. DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN

1.1.1. Planteamiento y descripción del problema

La ausencia de un mecanismo para identificar a los infractores de la ley que tienen órdenes de detección, pero eluden el castigo que causa problemas para la seguridad nacional. La sociedad se ha visto afectada por el aumento exponencial en los últimos años del número de personas prófugos que tienen órdenes de detención por sus delitos.

Ante la falta de una herramienta de identificación de delincuentes, estos se fugan con facilidad por los terminales terrestres, fronteras, aeropuertos etc. De esta forma, el estado para reducir el impacto de los delincuentes prófugos ofrece recompensa que oscila entre S/10.000 a S/150.000 soles por brindar información de delincuentes con orden de captura por diversos delitos cometidos.

Ante esta situación, es fundamental desarrollar una tecnología de reconocimiento facial para identificar a los delincuentes con órdenes de captura activas y elaborar un índice de seguridad para la población civil. Según el Instituto Nacional de Estadística e Informática (INEI), el 85.6% de la población siente que no está segura concluyendo que la comunidad se sienta incómoda en el país.

1.1.2. Formulación del problema general

¿Cómo realizar el reconocimiento facial con alerta remota para dispositivos móviles?

1.1.3. Formulación de los problemas específico

PE1. ¿Cómo reconocer los rostros de los prófugos de la justicia?

PE2. ¿Cómo integrar el módulo de captura, procesamiento, almacenamiento e interfaz de reconocimiento facial?

PE3. ¿Cómo reducir el error en el reconocimiento de rostros?

PE4. ¿Cómo verificar el funcionamiento del sistema mediante alerta remota hacia un dispositivo móvil?

1.2. DEFINICIÓN DE LOS OBJETIVOS DE LA INVESTIGACIÓN

1.2.1. Objetivo General

Diseñar e implementar un sistema de reconocimiento facial utilizando software e interfaces con alerta remota para dispositivos móviles

1.2.2. Objetivo Específicos

OE1. Desarrollar un sistema de reconocimiento de rostros en función de un algoritmo.

OE2. Desarrollar un prototipo de sistema software que integre el módulo de captura, procesamiento, almacenamiento e interfaz de reconocimiento facial

OE3. Implementar técnicas de procesamiento digital de imágenes que brinden un alto nivel de confiabilidad.

OE4. Desarrollar pruebas y validar el sistema de alerta remota hacia un dispositivo móvil

1.3. JUSTIFICACION DE LA INVESTIGACION

1.3.1. Justificación técnica

En la actualidad, el estudio de los sistemas de reconocimiento facial se está ampliando considerablemente. La detección e identificación de la región rostral ayuda a abordar el problema social de la inseguridad ciudadana. Esta tecnología de investigación ayuda a la resolución de problemas y permite implantar sistemas de reconocimiento facial en tiempo real, con márgenes de error y asertividad escasos o reducidos. El presente proyecto de investigación contribuye a la innovación tecnológica; los sistemas de reconocimiento facial permiten detectar el rostro de un individuo, lo que permite identificarlo y localizarlo. Se dispone de un sistema de reconocimiento facial con un valor medio de mercado de S/32 441 701 soles.

El presente trabajo de investigación economiza debido a que utiliza la tecnología actual, además de conocimientos de programación que potencian el desarrollo tecnológico. Este proyecto se trata básicamente diseño e implantación de un sistema de reconocimiento

facial utilizando software e interface con alerta remota al dispositivo móvil mediante un mensaje de texto. Así mismo, haciendo la detección facial del delincuente con orden de captura registrado en la base de dato en un periodo autentificado a tiempo real.

Utilizando el algoritmo de Local Binary Patterns Histogram (LBPH) de aprendizaje profundo mediante el módulo CV2 de la librería OpenCV que permite una interfaz para trabajar con funciones de procesamiento de imágenes y video.

1.3.2. Justificación económica

Según Briceño (2019), existe un déficit económico en el Estado Peruano sobre captura de criminales prófugos que evaden la justicia por diferentes delitos. Así mismo el Ministerio del Interior (MININTER) desarrolló el programa casa recompensa generando gasto de forma exponencial. Recompensas fallidas, en marzo del año 2016, (MININTER), desembolsó en gasto económico la suma de S/ 1 725 000 en el programa recompensa “Que ellos se cuiden”. Así mismo el año 2017 el (MININTER) pagó la cifra de S/7 025 000, posteriormente el año siguiente 2018 desembolsó el monto S/6 405 000. Sumando las cifras llevó a un total de 15 millones de soles para el año 2017 y 2018. El presente proyecto de investigación permitirá reducir el índice de inseguridad en la población y el gasto que se asigna por el tema de recompensas.

1.3.3. Justificación social

En el Perú se vive una realidad de forma cuantificada mediante el acto criminal, accediendo a dato estadístico actualizado, donde especifica que la mayor preocupación ciudadana es la delincuencia. La inseguridad ciudadana se viene incrementando de forma acelerada en el primer semestre de noviembre de 2015 a abril 2016, más del 45% de la población siendo mayor de edad, porcentaje que va aumentando en preocupación en años anteriores, sin hacer una rigurosa comparación en los primeros meses del 2012, donde el problema delincuencial en el país solo era de 28.5%. Como afirma

Sánchez (2016), “Asimismo, el 42.5% de los ciudadanos señalaron que otro de los grandes problemas es la corrupción, prima hermana de la criminalidad, y que de igual manera ha tenido un comportamiento en ascenso desde el 2012 que fue 25.9%.” (p.4). El problema de la población peruana es la falta de seguridad ciudadana, resumiendo que el principal motivo es la delincuencia el país.

Debido a la inseguridad ciudadana en los últimos años, se registraron a nivel nacional el incremento de forma exponencial de la delincuencia. El porcentaje de encuestados de 15 años o más que declararon sentirse incómodos es superior al 70% en casi todos los sectores. Las cifras de Ancash y Tumbes son inferiores: 64% y 65%, respectivamente (INEI, 2017). La presente investigación tiene como objetivo diseñar e implementar un sistema de reconocimiento facial utilizando software e interfaces con alerta remota para dispositivos móviles para mejorar la seguridad ciudadana contra fugitivos delincuentes que tienen orden de captura a nivel nacional.

1.4. ALCANCES Y LIMITACIONES DE LA INVESTIGACIÓN

1.4.1. Alcances

- El presente proyecto llegará a la fase de implementación.
- El proyecto de investigación simulará pruebas de funcionamiento mediante una base de datos de supuestos delincuentes con orden de captura para realizar el reconocimiento facial.
- El proyecto propuesto se podrá aplicar para delincuentes con orden de captura.
- La presente investigación podrá acoplarse a brindar acceso de seguridad en instituciones.
- El presente proyecto de investigación es un prototipo, donde permitirá que futuras investigaciones aporten mejoría e innovación.

1.4.2. Limitaciones

- El proyecto de investigación no tiene acceso a datos de la RENIEC.
- La presente investigación no realizará pruebas de funcionamiento de delincuentes con orden de captura del programa recompensa del MININTER ya que se requiere dos imágenes diferentes de la misma persona, porque el modelo tiene que ponerse a prueba con data distinta a data de entrenamiento (comparación de foto o video), para mostrar su confiabilidad. El MININTER solamente brinda una imagen.
- El presente proyecto de investigación no determinará el lugar de prueba, ya que será un prototipo de supuestos delincuentes con orden de captura en una base de datos propuesto.
- Dado que no se puede utilizar una base de datos con dos fotografías de personas con orden de captura, se realizará la validación de la eficacia del proyecto mediante supuestos delincuentes y con ello verificar el buen funcionamiento del modelo propuesto.

CAPÍTULO II: FUNDAMENTO TEÓRICO

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1. Internacionales

En el estudio de Manga (2019), la trata de personas es un problema social. La persona desaparece por diferentes motivos, las causas son: voluntaria, involuntaria, también muerte accidentada en personas desaparecidas en Colombia. La solución propuesta que ofrece el grupo de investigación es la creación y despliegue de un sistema biométrico facial para la identificación y búsqueda de personas, para ello realizó la técnica de biometría facial “Simulación de envejecimiento facial aplicada a los desaparecidos problemas de Persona”, que presenta un método de figura, el rostro de la persona en tres dimensiones espaciales, alto, largo, ancho. Simulación del prototipo de imagen facial de envejecimiento de los padres de la persona desaparecida, esta técnica consistió en lo siguiente, seleccionar la imagen fotográfica de rostro desaparecido, reconstruir del espacio tridimensional la foto del individuo, construcción de estructura de datos genómicos (EDG) de los papás, inicialización de bancos de gametos (meiosis),escoger la similitud de rostro más parecido al tipo de imagen fabricado partiendo de la fotografía, construcción de modelo paterno en la edad objetivo, configuración actual de estructura del dato genómico paternal, actualizar la estructura de dato genómico del desaparecido a partir del dato revelador paternal y el espacio tridimensional. Luego se obtuvo como resultado de prueba de biometría facial en personas desaparecida, capturando imagen de rostro y video en un total de 500 imágenes, 10 horas de prueba de video, obteniendo asertividad de 69% en imagen, 60% de video. Esta aplicación también es retribuyente para el uso de la detección de personas con orden de captura, que andan desaparecido de la justicia por diferente hecho delictivo. Se desea utilizar en la presente tesis la técnica comparativa o similitud de rostro almacenado en la data, también el estado aplicación en espacio tridimensional, ancho, alto, largo de una imagen de rostro.

En el estudio de Briones (2020), permite acceder al control autentificado de un centro comercial en Ecuador, con el fin de capturar a delincuentes prófugos de justicia. Teniendo una base de dato preestablecido, se utilizó la técnica facial como Machine Learning, tiene como fin, convertir los datos procesados en información. Teniendo un sistema de capacidad mediante un conjunto de dato por medio de una extracción de ciertos patrones, Proponiendo un algoritmo comparativo a través de una interfaz ya anexada alertamente con el centro operacional. Así mismo se realizó el uso de librería OpenCV, también un diagrama de funcionalidades para el sistema, teniendo como resultado obtenido la utilización de imagen a usar, mediante la técnica de reconocimiento facial de normalización para el alineamiento de imagen con el algoritmo de Deep Learning y entrenamiento de una Redes Neuronales Convolucionales (CNN). Este proyecto se realizó con el reconocimiento facial que determina un patrón Davis 360 donde permitió un asertividad de 99.38% y su implementación de 98% eficaz en la detección de rostro utilizando librerías con entrenamiento ya establecido. Para la presente investigación, se tendrá como utilidad el uso de la librería OpenCV, permitiendo la utilidad del lenguaje de programación Python. La presente investigación permite analizar lo robusto que puede ser la técnica de aprendizaje profundo del algoritmo Deep Learning.

En el estudio de López (2020), identifica la inseguridad por delincuencia común en la ciudad de Bogotá. Realizó diversos métodos para el sistema de implementación de reconocimiento facial con propósito de identificación de posibles delincuentes gracias al análisis aplicado por métodos seleccionado, identificación, extracción de características para luego dar el reconocimiento de rostro. Siendo detectado por la cámara video vigilancia y vinculada a la base de dato de criminales para utilidad de la Institución Policial de Bogotá. Realizo su óptimo funcionamiento del algoritmo de reconocimiento, como resultado de proyecto realizo un prototipo de

prueba, con almacenamiento en la base de dato, resultado métrica y velocidad de procesamiento. Resultado adquirido de 99,3% aciertos del algoritmo. Donde realizo la investigación de distancia entre rostros pueden alcanzar un umbral de error por la configuración de variable pueden ser errónea.

La investigación de González, et al.(2020), concluyeron que el país de Ecuador manifestó incremento e índice de inseguridad por diversos delitos. El trabajo de investigación busca reducir individuos requerido por justicia. El grupo de investigación plantea el desarrollo de plataforma de sitio web, utilizando el uso de método de detección de rostro OpenCV, combinando técnicas pre-entrenado Deep Metric Learning. El trabajo diseño para la plataforma de reconocimiento facial orientado a Redes Neuronales y técnica de libre acceso de Open Source. Resultado esperado de proyecto de investigación donde incorpora técnica y método de herramienta de detección de rostro. El reconocimiento de rostro tuvo una efectividad óptima para el sistema web. La plataforma tiene como porcentaje de acierto de 92,2%. Mediante este trabajo de investigación optare en utilizar en mi proyecto técnica de OpenCV.

2.1.2. Nacionales

Del trabajo de Barreto, et al. (2019), propone ayudar a disminuir en la trata de personas, utilizando arquitectura OPENFACE basada en aprendizaje máquina en la selva del Perú. El grupo de investigación realiza su propio modelo de sistema, creando una base de dato de personas desaparecidas en redes sociales e instituciones del estado, hacen uso de tecnología redes neuronales y aprendizaje profundo de OPENFACE e ingresando extracciones de características en imagen de rostro de personas desaparecidas, siendo las imágenes de rostro de redes sociales entrenadas con datos públicos de plataforma CASIA-Web y FaceScrub que trabajan de la mano de OPENFACE. El grupo de investigación propone la evaluación y desempeño del nuevo sistema de reconocimiento facial elaborando en condiciones normales y reales. Se estima tener un

resultado del 95% de aciertos, así como el mismo OpenCV, Torch, Python, utilizaremos como herramienta de OPENFACE, en mi proyecto de investigación utilizare lenguaje de programación Python.

En el estudio de Velarde et al. (2018), creó un sistema de reconocimiento facial en Cuzco (Perú) para luchar contra la trata de seres humanos. Adquirió información en la página web del Ministerio del Interior (MININTER) para la extracción de imágenes. Se efectuó el uso de algoritmo de clasificación de cascada centrado en características tipo Haar para la detección de rostro. En consecuencia, se utiliza el algoritmo Local Binary Pattern Histogram (LBPH) mediante la librería OpenCV, colocando parámetro del algoritmo de cascada, haciendo uso del archivo CSV con listado de imagen de identificador de cámara web. Efectuando una petición de muestra de reconocimiento facial a desarrollarse como es el algoritmo LBPH, permitiendo predecir el rostro que desea identificar, mediante la compilación del algoritmo para su determinada ejecución, indicando la detección de rostro con el clasificador en cascada, comparando el rostro, para luego ser almacenado en la base de dato por el (LBPH). La interfaz de historia de usuario "almacena rostros detectados" se inicia una comparativa de imagen facial por el algoritmo, atreves de la cámara web e imágenes de rostros guardados en la base de dato, si el algoritmo después de la comparativa tiene similitud entre las imágenes capturadas, por siguiente se efectúa el guardado de aquella imagen junto a su ID, hora y fecha de su identificación de rostro. Se presenta el resultado obtenido, donde se obtiene 22 imágenes de 32 pruebas con 50 rostros de persona, se reconoció el rostro de la persona detectada. Un buen resultado de reconocimiento facial mediante el algoritmo (LBHP) con efectividad de (86,75%). Se desea utilizar el tipo de algoritmo de reconocimiento facial del (LBPH) del presente artículo científico, así mismo la detección facial, utilizar el algoritmo de clasificación en basados de tipo Harr.

2.2. MARCO TEÓRICO

2.2.1. Conceptos

Este proyecto de investigación permite la creación de un sistema que puede utilizar el teléfono móvil para identificar a distancia rostros humanos en tiempo real a fugitivos en una base de datos. Para llevar a cabo este desarrollo e implementación, se obtiene un sistema de instrumentos y procedimientos técnicos. Al mismo tiempo, se desarrollan las ideas fundamentales que se aplicarán a lo largo de este proyecto a modo de introducción.

2.2.2. Reconocimiento Facial

El reconocimiento facial, inicio en los años sesenta mediante el sistema semiautomático que consistía en realizar marcas en la fotografía de esa forma delimitar la característica principal, ojos, nariz, boca, oído. El sistema de reconocimiento facial necesita un AMINISTRADOR para ubicar aquellos parámetros de rostro. Así mismo, “se calcularon las distancias medias y relaciones de estas marcas a un punto de referencia común, finalmente se comparaban con los datos de referencia”. Ya en los años setenta en una de sus obras toca expone, menciona color de cabello, el grosor de labios, narices entre otras características (Toca, 2011 citado por Gutiérrez, 2021, p.16).

“Los sistemas de reconocimiento biométrico utilizan características fisiológicas o de comportamiento propias de cada individuo para identificarlo, es decir, se reconoce al usuario por lo que es en lugar de por lo que posee o sabe” (Rodríguez, Giménez, & Molina, 2022).

Actualmente, la elaboración de nuevas tecnologías incorpora técnicas biométricas basados en huella digital, Iris, patrones de retina, destacando el sistema de identificación y autenticidad del individuo. Así mismo la descripción de aquellas técnicas mencionadas perciben diversos tipos de variabilidad, desempeño, aceptación del usuario y durabilidad, permanencia constante en los parámetros de sistema reconocimiento biométrico.

2.2.3. Proceso de detección y reconocimiento de rostros

El proceso de la detección de rostros y el reconocimiento de rostros son dos algoritmos muy diferentes: la detección de rostros utiliza la detección de rasgos faciales para identificar un rostro detectado, mientras que el reconocimiento de rostros utiliza la detección de rasgos faciales para identificar el propio (Flores, 2020).

2.2.4. Detección de rostro

Como afirma Briones (2020), para la detección de rostro depende del computador, debido a que debe tener la capacidad de descifrar lo que está detectando y adquirir la interpretación de la imagen facial. Para definir la región de interés, debe utilizarse la aplicación conceptual conocida como detectores, que permite extraer puntos importantes que se situarían dentro de la imagen recogida. Los detectores de bajo nivel de puntos, se localizan en esquinas o color ante la aplicación de filtros.

- Cara: Anchura
- Nariz: Anchura y altura de la nariz y ubicación y oposición de la nariz.
- Ojos: Distancia entre los ojos, anchura y altura de los ojos.
- Boca: Anchura de la boca., posición vertical de la boca, y grosor de los labios.
- Cejas: posición vertical desde la ceja hasta la localización vertical central de los ojos y grosor de la ceja.

2.2.5. Lenguaje de programación

Es un conjunto de instrucciones que permiten el tratamiento seguro y eficaz de la información por el ordenador y facilitan la comunicación hombre-máquina mediante algoritmos que el ordenador puede comprender. (Altamirano & Untuña, 2022).

Lenguaje formal que realizan estructuras de cadenas que permiten diversos tipos de resultados de código máquina. Estos lenguajes de programación son pertenecientes a la familia de lenguaje informático, que a su vez se aprovecha en elaboración de algoritmos,

mediante la programación (Argente, 2021 citado en Flores, 2022). Seguidamente, se mencionan diferentes tipos de lenguaje de programación el cual se empleará en esta investigación, siendo compatibles con OpenCV.

2.2.6. Tipo de lenguaje de programación

- **Lenguaje Python:** Lenguaje de programación de tipo Python es un software libre para utilizar donde su utilización es fácil de aprender y ayuda al sistema o proyecto de reconocimiento facial. Su metodología de programación orientada a objetos es sencilla pero eficaz, y sus estructuras de datos de alto nivel son eficientes. (Rossum, 2017 citado por Ibarra, 2020).
- **Lenguaje C:** El lenguaje de tipo C es un grupo de órdenes para efectuar un computador. Estas características de ordenes cumplen un protocolo de lenguaje de programación, para que el computador pueda identificar e interpretar. El problema es que a los humanos nos cuesta entender los lenguajes que realmente comprenden los ordenadores, ya que difieren mucho de los que hablamos habitualmente. (Domínguez, 2019).

Donde escribir programas internamente, dentro del computador ya sea el momento de dialogar es complicado, así como en la programación

- **Algoritmo de Viola-jones:** Algoritmo de Viola-Jones es un código que permite detectar la iluminación de rostro a tiempo real, a diferencia de otros tipos de algoritmos.

Como señala Gonzales (2019), el algoritmo utiliza una imagen integral para calcular las intensidades luminosas de porciones rectangulares de las imágenes, o lo que se conoce como características Haar-Like. Esta comparación constituye la base de la técnica, haciendo uso de su clasificación al azar y utilidad probabilística para luego agruparse en forma de cascada y trabaja

de la mano del algoritmo de aprendizaje basado en AdaBoost busca mejorar la identificación.

- **Algoritmo Machine Learning:** Este algoritmo también conocido como aprendizaje autonómico tiene como objetivo en conversión de datos a una determinada información.

Así también Briones (2020), el aprendizaje automático es un sistema de aprendizaje autónomo que aprende por sí solo utilizando diversos algoritmos y sin la ayuda de una persona. Permitiendo la extracción de parámetros a si dar solución a la incógnita. Este tipo de tecnología en actualidad es usado por diferentes tipos de instituciones, policiales, hospitalario, y seguridad.

2.2.7. Descriptores y clasificadores

Medir el tiempo eficaz de desarrollo de los clasificadores de entrenamiento, así mismo pronosticar a quien pertenecer las nuevas imágenes de rostro facial, cuyo procedimiento es valioso para la movilidad en dispositivos electrónicos de modo transitables, el rendimiento predictivo de quien puede corresponder una imagen de rostro es fundamental, no hay retraso notado mediante un usuario visualiza a un individuo (Flores, 2022).

2.2.8. Extractores de características o descriptores

Asi como lo menciona López (2020), el reconocimiento facial tiene como desarrollo de extracción características destacadas de una imagen requerida de una base de datos. Mediante técnicas para la extracción de características, el sistema recibe información biométrica de cualidades de rostro, transformando figuras geométricas, ejecutando técnicas y cálculos matemáticas, luego tener el procesamiento de los componentes, de esa manera seleccionar clases o patrones seleccionados de una imagen, los parámetros faciales seleccionados de las figuras de rostros se pueden diferenciar en dos tipos: las transitorias detallan de algún desperfecto facial del rostro del individuo, líneas de expresión, orejas. Así mismo, no transitorias, son parámetros faciales que no

pueden faltar, entre ellos son los ojos, la boca y la nariz. Para adjuntar métodos precisos, se debe considerar en las imágenes manipuladas o procesadas, si estas son de tipo dinámicas o estáticas, a sí mismo haciendo el uso del reconocimiento por características de entorno donde se solicita hacer el entrenamiento de identificación, poca iluminación, deficiencia del ambiente y los equipamientos tecnológicos de procesamiento útil. En esta etapa del algoritmo, la extracción debe cumplir tres pasos a seguir: reducción de la dimensionalidad, extracción de características y selección de características obteniendo como resultado a este sistema de división en clases o patrones positivos o negativos, proporcionando el incremento de probabilidad de identificación de las siguientes fases, antes de explayar la etapa de comparación es preciso enfatizar la clasificación de los métodos de extracción están conformados en dos grandes categorías: Holísticas o basadas en características.

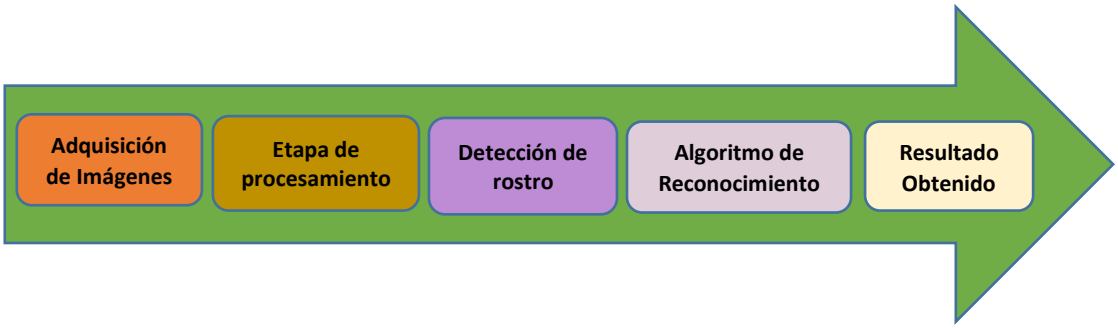
2.2.9. Clasificadores

Según Cabello (2021, como se citó en Gutiérrez, 2021), usar clasificadores es una manera precisa de obtener representaciones de cantidad de información siendo útil para la representación, produciendo la posibilidad de efectuar extracción de características e identificar distintos grupo o clase. La finalidad del clasificador es determinar una dirección a un conjunto de datos respecto a una detección facial.

2.2.10. Descripción de la arquitectura del proyecto

Figura 1

Etapa del reconocimiento facial



Nota. La figura muestra como proceso de adquisición de imagen de supuestos delincuentes con orden de captura, mediante etapa de procesamiento de imagen, así mismo, detección de rostro que realizara una comparación entre imágenes modelo e imágenes de entrenamiento para luego realizar el algoritmo de reconocimiento facial en función de la persona a identificar. Finalmente obtener un resultado esperado.

Por medio del diagrama, se realizaron ejecuciones donde se ha podido desarrollar e implementar sistemas de funcionamiento que desarrolle cada etapa, un software favorable y familiarizado para el usuario en cuanto acceso de datos solicitado. En este tipo de investigación, el sistema de reconocimiento facial derivado a la detección de supuestos delincuentes con orden de captura. En este proyecto de investigación, se realizará mediante un sistema embebido y software para adquirir resultados cuantificados. Mediante la Figura 1, se puede ver la arquitectura desarrollada.

2.2.11. Librería OpenCV

Según Sotomayor et al. (Minichino & Howse, 2015 citado en, 2020) fue elaborado mediante lenguaje de programación C++, el cual también es un lenguaje hibrido. Esta librería incluye variedades de algoritmos donde también se encuentran Python. Esta librería presenta multivariado de algoritmos, se observa que entre sus destacados el algoritmo de aprendizaje automático. Aquella librería, presenta técnica sofisticadas donde realiza aplicaciones permitiendo la detección de algún individuo en video y siendo accesible en diferentes tipos sistemas operativos al ser multiplataforma.

Tabla 1

Ventajas y desventajas Opencv

OPENCV		
FUNCIONES	**VENTAJAS**	**DESVENTAJAS**
Identifica, detecta, clasifica objetos y rostros.	Es Multiplataforma.	No se recupera la Información perdida.
Utilizado en áreas de robótica y realidad aumentada.	Se puede usar otros Lenguajes de programación.	Tiene compresiones/ Descompresiones Complejas y costosas.
Examina escenarios.	Es gratuita Es portable.	Documentación débil, no Siempre explica lo que significa cada parámetro.
-Extrae modelo 3D	Bajo uso de RAM	Pequeña biblioteca de aprendizaje automático

Nota. Dentro de la tabla 1. Muestra las ventajas e inconvenientes de la capacidad de la biblioteca OpenCV para reconocer y trabajar con diversos tipos de lenguajes de programación. Esta biblioteca se utiliza en varios proyectos de investigación, tiene buenas compresiones, así como otras sofisticadas y costosas, no recupera la información perdida, investiga el entorno de detección de forma gratuita, su documentación es pobre y no explica el significado de cada opción. Utiliza poca RAM, extrae modelos 3D y contiene una modesta biblioteca de aprendizaje automático. (Altamirano & Untuña, 2022).

2.2.12. Reconocimiento facial con OpenCV

OpenCV tiene una clase abstracta llamada face_recognition que se utiliza con los siguientes algoritmos y está especialmente pensada para el reconocimiento del rostro facial:

- Eigenfaces(PCA), EigenFaceRecognizer()
- Fisherfaces(LDA), createFisherFaceRecognizer()
- Histogramas de patrones binarios locales (LBPH), createLBPHFaceRecognizer ().
- Haarcascade_frontalface_default.xml

Se necesitan dos librerías más para hacer reconocimiento facial usando OpenCV y Python: face_recognition y dlib.

2.2.13. Librería dlib

Según Bastidas (como se citó en Briones, 2020), es un conjunto de herramientas C++ modelo que incluye herramientas para crear aplicaciones C++ complicadas y métodos de aprendizaje automático para abordar situaciones del mundo real (p.12 & 13).

2.2.14. Librería Face Recognition

Según como lo menciona Carmilema & Pérez (2019), para realizar un análisis de características de rostro, se extrae la captura de imagen facial. Este método de análisis es desarrollado por librería face_recognition. Este tipo de extracción facial hace una comparativa en la base de dato. Basado con esta técnica es factible la ejecución de identificación facial en tiempo real.

Esta biblioteca, desarrollada en Python, utiliza el marco dlib, que es un software multiplataforma con un objetivo genérico para el aprendizaje autónomo. Siendo el software independiente quien realiza fuertemente el diseño de componentes. Siendo un código libre determinado por una licenciatura de software Boost.

Es posible utilizar este programa con cualquier sistema operativo. Software que funciona en varias plataformas y puede utilizarse con cualquier sistema operativo. Siendo la librería mencionada que obtiene una efectividad cuantificada de 99.38%. Según la prueba de funcionamiento de LFN (Labeled Faces in the wild).

2.2.15. Redes neuronales convolucionales

Las redes neuronales artificiales (ANN), son compuestos de redes neuronales convolucionales (CNN), con características de perceptrón multicapa, método aplicable para aplicación de visión artificial, mediante etapas de reconocimiento facial.

Frecuentemente, se incorpora funciones después de cada capa, determinando un mapeo temporal lineal (CNN) conformada por diferentes medidas en múltiples capas de filtros convolucionales constituidos. Se deduce el trabajo desarrollado en cuanto a las capas son filtros para preprocesar imagen facial y requiriendo similitud en

parámetro de rostro. En su totalidad, los filtros adquieren numerosidad capacidad e identificación de características; bordes, esquinas, figuras geométricas círculos cuadrados.

Las aplicaciones robustas que utilizan filtros complejos permiten seleccionar, tipos de automóviles, tipos de animales, etc. El filtraje de selección, generalmente están al termino de las capas de red.

Los patrones y características que frecuentemente busca son:

- Objetos
- Texturas
- Curvas
- Colores
- Bordes
- Formas

Conformados por redes convolucionales se empieza la extracción de características, luego el resultado por muestreo es menor para la obtención del método de red convolucionales más accesible para hacer la selección final en cuanto a parámetros incorporados (Obando, 2019 citado en Muñoz, 2021).

2.2.16. Descripción de herramientas

- **Sistema de video vigilancia**

  Como lo menciona Gutiérrez (2021), el dispositivo tecnológico, facilita primordialmente la visualización o reproducción, de un grupo reducido de personas, imágenes de rostros faciales identificadas por video cámara. El desarrollo o ejecución, se efectúa de forma local o remota, desde cualquier punto, mediante el internet. Así mismo para optar en visualizar ese tipo de tecnología debería utilizar un celular tipo Android, computador o Tablet.

- **Sistema embebido**

  El tipo de tecnología NVIDIA Jetson Nano, tiene como resultado de mayor eficaz encuentro a la imagen. Raspberry PI tiene un retraso mayor en comparación de la primera tecnología mencionada, pero retribuye con la facilidad que se realiza al momento de la ejecución de la instalación de las dependencias necesarias, mayormente el tiempo que tarda en instalación es de ciento y veinte segundos. Este tipo de tecnología embebida es recomendado para tipos de trabajos que necesiten hacer una implementación de forma rápida perjudicando el rendimiento.

  Siendo el tipo de instalación de las dependencias presenta diversos tipos de compilaciones, el problema presenta diversos modos de instalación de las dependencias, retardo en tiempo de instalación, por ejemplo, periodo de tiempo de instalación de la librería OpenCV el cual tuvo una durabilidad de dos horas. Así mismo con NVIDIA Jetson Nano se obtiene un mayor rendimiento.

  Una de las causas por el cual se debe interior de la instalación para el sistema embebido se realiza la recopilación de la librería para tarjeta gráfica, adquiriendo un mayor rendimiento. Este tipo de sistema embebido es dirigido para trabajos de investigación que necesiten de forma rigurosa una alta capacidad de procesamiento, así mismo tengan el tiempo para hacer las determinadas instalaciones de sus dependencias, este proceso de ejecución en mayoría de los casos tarda más de un día sin detenerse (Zuluaga & Hernandez, 2022).

- **Raspberry PI**

  Es un computador reducido o computador simple. Sirve para fomentar y hacer estudio de investigación computacional en las escuelas. Fue desarrollado en el Reino Unido. Así mismo no hace énfasis si el Hardware es libre o con derecho de marca, pero se determina que tiene contrato con empresas

distribuidoras. También cualquier persona puede revender, a su vez se deduce que es de propiedad privada, permitiendo el uso libremente para diversos proyectos de investigación. “En todas sus versiones incluye un procesador Broadcom, una memoria RAM, una GPU, puertos USB, HDMI, Ethernet, 40 pines GPIO y un conector para cámara. El sistema operativo se almacena en una tarjeta microSD externa” (CENTURIÓN & ALMEIDA , 2022, p.18).

Todo lo contrario, realizan mediante el software de fuente abierta. Teniendo un sistema original llamada Raspbian, también nos proporciona el otorgamiento de otros tipos de sistemas operativos.

- **Web Services**

  Según Vega (2022, como se citó en Gutiérrez, 2021), es donde el computador o programa solicita y a la vez recibe información de diferentes computador o programa, transportando información e intercambio de sobre la red, “Internamente el web service procesa la información e interactúa con otros componentes, por ejemplo, base de datos, aplicaciones móviles, dispositivos físicos, centros de procesamiento de información, incluso otros webs services” (p.38).

2.3. MARCO METODOLÓGICO

2.3.1. Tipo de investigación

El presente trabajo de investigación tiene una finalidad tecnológica e innovadora, debido a que primero se investigará información sobre temas relacionados al proyecto propuesto, después se diseñará e implementará un sistema de reconocimiento facial para la identificación de delincuentes con orden de captura, permitiendo la seguridad ciudadanía. Esta tesis brinda una alternativa de acceso a seguridad de reconocimiento facial mediante la detección de delincuentes con orden de captura o podría acoplarse en brindar acceso de seguridad en instituciones. La investigación permitirá reducir el hecho delictivo o delito mayor del delincuente prontuariado, mediante el aporte de seguridad para prevenir la delincuencia.

2.3.2. Metodología de la investigación

La presente tesis desarrollará ante la inseguridad ciudadana que viene incrementando de forma exponencial, pruebas de funcionamiento mediante una base de datos de supuestos delincuentes con orden de captura. El presente proyecto de investigación no determinará el lugar de prueba, ya que será un prototipo de supuestos delincuentes con orden de captura en una base de datos propuesto para realizar el reconocimiento facial.

El diseño e implementación de la presente investigación tiene como objetivo desarrollar un sistema de reconocimiento facial utilizando software e interfaces con alerta remota para dispositivo móvil con el propósito de mejorar la calidad de vida en la sociedad y reducir el índice de inseguridad, para ello utilizaremos una serie imágenes de supuestos delincuentes con orden de captura almacenados en una base de dato donde se podrá realizar la detección facial mediante un sistema embebido y una Cámara Raspberry PI REV 1.3. Aplicado en aprendizaje profundo de librería OpenCV, utilizando la Jetson nano como placa base para el procesado del software para el reconocimiento facial de delincuentes con orden de captura que

tendrá como alerta remota mediante un mensaje de texto inalámbrico a un dispositivo vinculado. La identificación de rostros se evidenciará a tiempo real ya que las imágenes serán registradas en la base de dato.

2.4. MARCO LEGAL

- Mediante ley N° 30336 el congreso de la Republica ha delegado en el poder Ejecutivo la facultad de legislar en materia de fortalecimiento de la seguridad ciudadana, lucha contra la delincuencia y el crimen organizado, por un plazo de noventa (90) días calendario;

  Que, el literal a) del artículo 2 de la ley N°30336 faculta al Poder Ejecutivo para fortalecer la lucha contra la delincuencia y el crimen organizado;

  De conformidad con lo establecido en el artículo 104 de la Constitución Política del Perú;

  Con el voto aprobatorio del Consejo de Ministros; y,

  Con cargo a dar cuenta al congreso de la República;

  Ha dado el Derecho Legislativo siguiente:

  ESTABLECER BENEFICIO DE RECOMPENSA PARA PROMOVER Y LOGRAR LA CAPTURA DE MIENBROS DE ORGANIZACIONES CRIMINALES ORGANIZACIONES TERRORISTAS
  Y RESPONZABLES DE DELITOS DE ALTA LESITIVIDAD
- Artículo 2. numeral 6 de la constitución política del Perú a que los servicios informáticos, computarizados o no, públicos o privados, no suministren informaciones que afecten la intimidad personal y familiar.
- LEY DE PROTECCION DE DATOS PERSONALES (Ley N°29733)

  Clase de Datos:

  Datos personales: Es aquella información numérica, alfabética, grafica, fotográfica, acústica, sobre hábitos personales, o de cualquier otro tipo concerniente a las personas naturales que las identifica o las hace identificables a través de medios que puedan ser razonablemente utilizados.

  ARTICULO 2, numeral 8 de la constitución Política del Perú la libertad de creación intelectual, artística y científica, así como a la propiedad

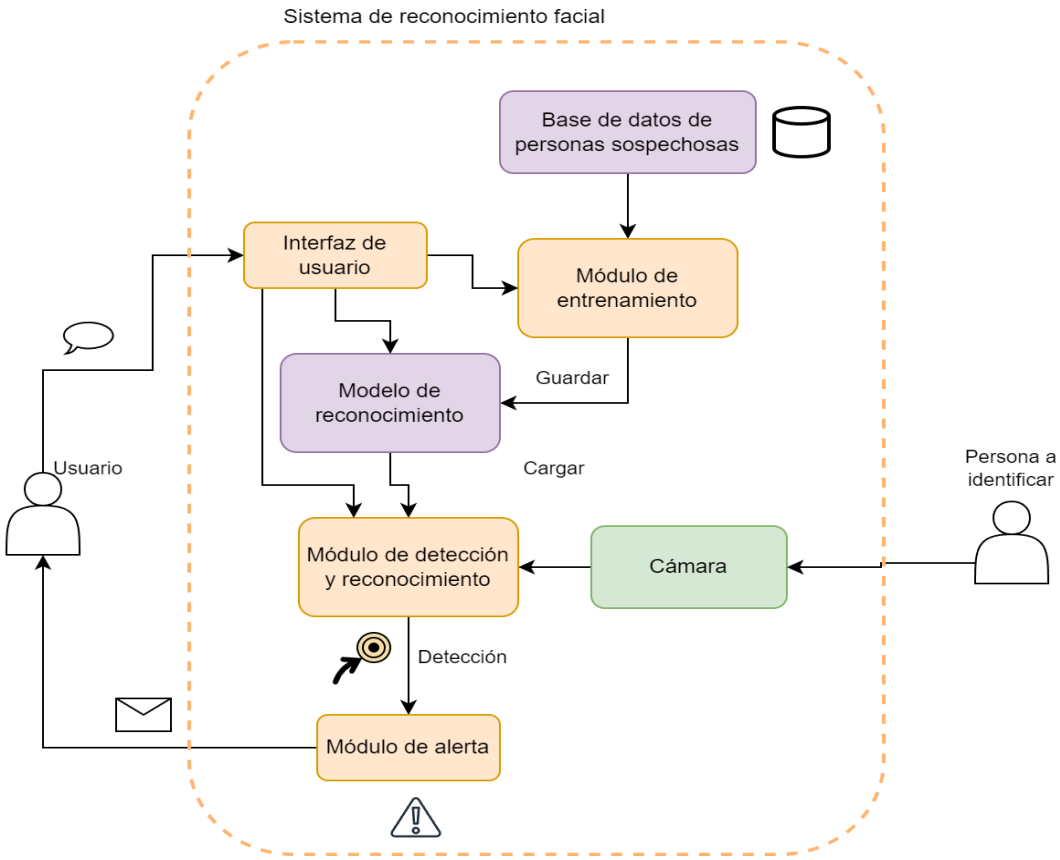
sobre dichas creaciones y a su producto. El estado propicia el acceso a la cultura y fomenta su desarrollo y difusión.

- LEY DE PROTECCION DE DATOS PERSONALES (Ley N°29733) Disposición de recursos: Todo titular de datos personales debe de contar con vías administrativas o jurídicas para reclamar y hacer valer sus derechos, cuando estos sean vulnerados por el tratamiento de sus personales.

2.5. ARQUITECTURA DEL SISTEMA

Figura 2

Arquitectura del sistema de diseño e implementación de un sistema de reconocimiento facial



Nota. En la figura 2. Muestra Arquitectura del sistema de diseño e implementación de un sistema de reconocimiento facial se estructura bajo los siguientes componentes:

Usuario:

Los usuarios interactúan con el sistema a través de una interfaz de acceso que puede ser una aplicación web o una aplicación móvil.

Interfaz de usuario:

El medio incluye todos los puntos de contacto entre el usuario y el equipo y permite la comunicación entre una máquina, equipo, ordenador o dispositivo y su usuario.

Base de datos de personas sospechosas:

Una base de datos es una colección de datos relativos a un sospechoso. Para este trabajo de investigación se realiza la adquisición de la base de 15 personas, datos Rulf face data.base. Para lo cual los rostros se capturan a partir de los videos de la base de datos el cual es dividido en data de entrenamiento y data de validación.

Módulo de entrenamiento:

Es el componente responsable de entrenar y mejorar el modelo de reconocimiento facial. Utiliza conjuntos de datos etiquetados para aprender y ajustar sus parámetros, de modo que pueda mejorar su precisión y rendimiento.

Modelo de reconocimiento:

Se refiere al algoritmo o conjunto de algoritmos utilizados para identificar y verificar la identidad de las personas basándose en las características faciales extraídas de las imágenes capturadas por la cámara.

Cámara:

Esta pequeña cámara puede capturar imágenes de 5 MP y vídeo HD 1080p a 30 fotogramas por segundo. Puede capturar imágenes y películas con Raspberry Pi 1, 2, 3 y 4. La cámara se puede utilizar con la versión más reciente de Raspbian, que es el sistema operativo recomendado de Raspberry Pi. Basta con conectarla directamente con su cable y reiniciarla para activar la función de vídeo.

Módulo de detección y reconocimiento:

Este módulo se encarga de capturar imágenes de las personas a través de la cámara, detectar rostros en estas imágenes y luego utilizar el modelo de reconocimiento para identificar a las personas en base a sus características faciales.

Módulo de alerta:

El módulo de alerta se desarrolla con parámetros de la plataforma twilio con vinculo de programación Python de modo virtual para identificación de rostro en el proyecto de investigación.

CAPÍTULO III: DESARROLLO DE LA APLICACIÓN

3.1. DESARROLLAR UN SISTEMA DE RECONOCIMIENTO DE ROSTRO EN FUNCIÓN ALGORITMO

El primer objetivo tiene dos etapas importantes:

La primera etapa inicia la captura de video mediante la librería OPENCV cv2.video capture (source), inicia el detector fr rostro mediante librería dlib quien detecta el bounding box en el rostro facial mediante el entorno. Se convierte la imagen en escala de grises gray=cv2.cvtcolor(img., cv2.color_BGR2GRAY). Detecta los bounding box de cada rostro en imagen x, y, w, h, que son variables de dimensiones del bounding box, para cada rostro realiza un preprocesamiento (redimensionamiento del bounding box).

La segunda etapa obtiene el reconocimiento de identidad mediante el LBPH (Histograma de patrón binario local), id, confidence=model.predict(face_roi). Además, consiste en comparar los histogramas y selección de personas en donde exista la menor distancia euclidiana para así tener una mejor similitud a la persona a reconocer. Dependiendo de la distancia, si ésta es lo suficientemente pequeña, se le asigna a la persona una identidad, caso contrario se le determina una identidad desconocida.

reconocimiento.py > faceDetection

```
import cv2
import dlib
def faceDetection(models, id_to_name):
  face_detector = dlib.get_frontal_face_detector()
  cam= cv2.videoCapture(0)
  font = cv2.FONT_HERSHEY_SIMPLEX
  font_scale = 1
  font_thickness =2
  lastName = “ “
  lastTime = time.time ()
  while True:
```

```
 ret, img = cam.read()
 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
 faces = face_detector(gray)
 for face in faces:
 x, y, w, h = face.left(), face.top(), face.width(), face.height()
 best_prediction = ("Unknown", -1, 100) # (Name, ID, Confidence)
 predictions = [ ] # List to store all predictions
 # Ensure the detected face region has valid dimensions
 if x >= 0 and y >= 0 and w >0 and h > 0 and (x+w)< gray.shape[1] and (y +
h)
 face_roi = cv2.resize(gray[y:y + h, x:x + w], (100, 100))
 for name, model in models:
 id, confidence = model.predict(face_roi)
 predictions.append((name, id, confidence))
 if confidence < best_prediction[2]:
 best_prediction = (name, id, confidence)
  name, id, confidence = best_prediction

  if confidence <= 80: # check if confidence is below 80%
cv2.rectangle(ing, (x, y), (x + w, y + h), (0, 255, 0), 2)
  # Get the name from the id_to_name dictionary if the key exists
  if id in id_to_name:
 name = id_to_name[id]
 confidence_text = " {0}%".format(round(confidence))
 cv2.putText(img, str(name), (x + 5, y - 5), font, font_scale, (255, 255, 255),
font_thickness)
 cv2.putText(img, str(confidence_text), (x + 5, y + h - 5), font, font_scale,
(255, 255, 0), font_thickness)

# si la persona fue reconocida y es una nueva persona o han pasado 5
minutos
if name != lastName or time.time () – lastTime > 300:
TimeAnDate = datetime.datetime.fromtimestamp(time.time()).strftime('%y-
%m-%d %H:%M:%S')
```

```
# Enviar mensaje: nombre ha sido reconocido fecha y hora
SendSMS(f"{name} ha sido reconocido/a a las {timeAndDate}")
   else:
  cv2.putText(img, "Unknown", (x + 5, y - 5), font, font_scale, (255, 255,
255), font_thickness)
  cv2.putText(img, str(confidence), (x +5, y + h - 5), font, font_scale, (255,
255, 0), font_thinckness)
   else:
   cv2.rectangle(img, (x,y), (x + w, y + h), (0, 0, 255), 2)
cv2.imshow('camera', img)

 k = cv2.waitkey(10) & 0xff # Press 'ESC' to exit
 if k == 27:
 break
# Release the video capture
cam.release()
cv2.destroyAllWindows()
```

El presente código se estructura bajo la siguiente jerarquización:

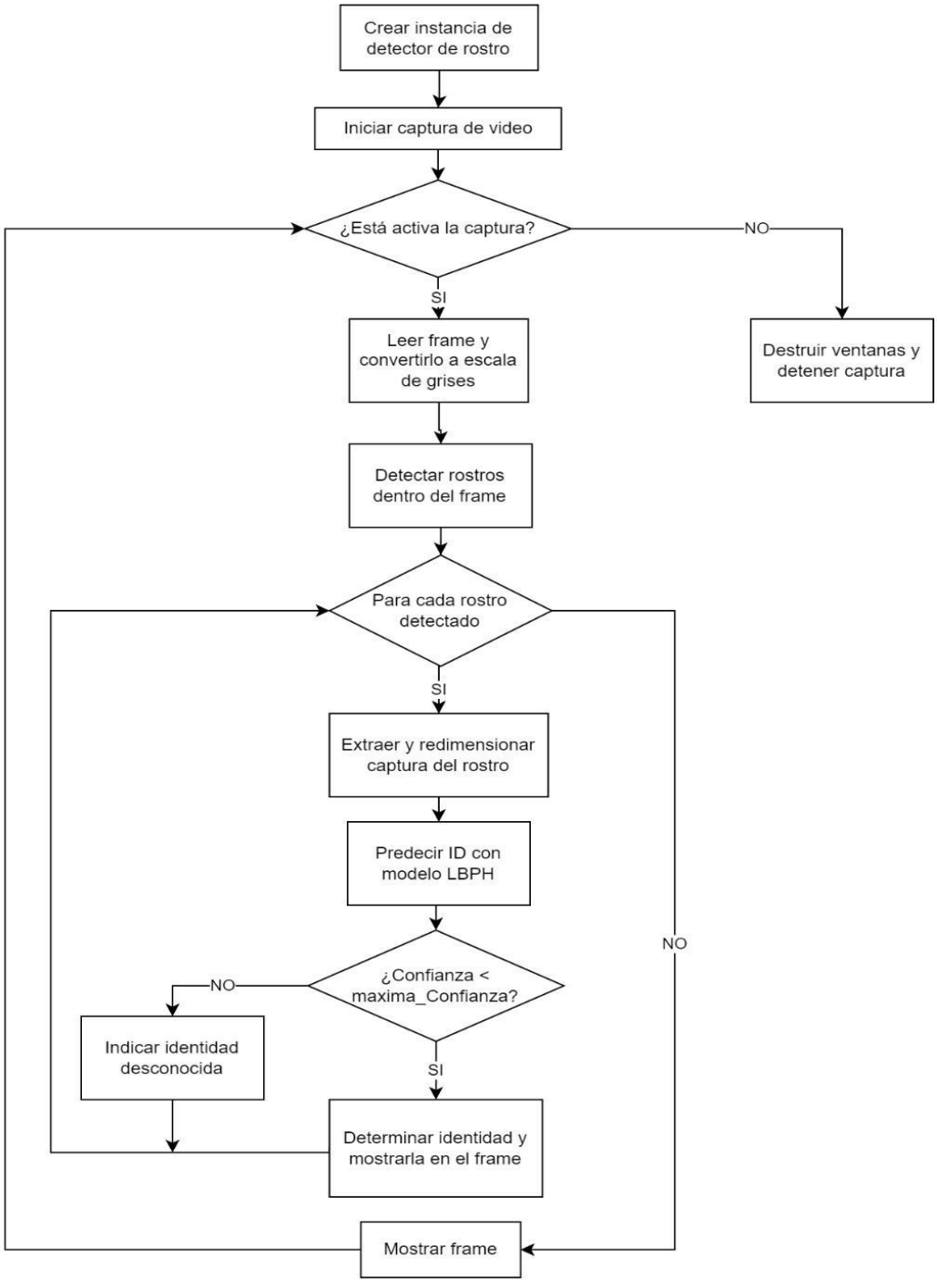
- **Importación de Bibliotecas:** Se importan las bibliotecas necesarias, cv2 (OpenCV) y dlib, para el procesamiento de imágenes y el reconocimiento facial, respectivamente.
- **Definición de la Función faceDetection:** Esta función toma dos argumentos, models y id_to_name. models es una lista de tuplas que contienen nombres y modelos de reconocimiento facial asociados, y id_to_name es un diccionario que mapea identificadores a nombres.
- **Inicialización del Detector de Caras:** Se inicializa el detector de caras frontal de dlib (face_detector).
- **Inicialización de la Cámara:** Se inicializa la captura de video desde la cámara (cam) utilizando cv2.videoCapture(0).

- **Configuración de Fuente y Bucle Principal:** Se configuran parámetros para la fuente de texto y se inicia un bucle principal que captura imágenes continuamente desde la cámara.
- **Procesamiento de la Imagen:** Se captura una imagen (img) de la cámara en cada iteración y se convierte a escala de grises (gray).
- **Detección de Caras:** Se utiliza el detector de caras para identificar las ubicaciones de las caras en la imagen.
- **Reconocimiento Facial:** Para cada cara detectada, se realiza el reconocimiento facial utilizando modelos previamente entrenados. Se almacenan las predicciones de cada modelo en la lista predictions.
- **Mejor Predicción:** Se determina la mejor predicción (mayor confianza) de todas las predicciones obtenidas.
- **Visualización de Resultados:** Si la confianza es mayor al 80%, se dibuja un rectángulo alrededor de la cara con el nombre y nivel de confianza correspondientes. Si la confianza es menor o igual al 80%, se marca la cara como "Desconocida".
- **Bucle de Presentación:** Se muestra la imagen procesada en una ventana llamada 'camera'. El bucle se mantiene hasta que se presiona la tecla 'ESC' (código 27).
- **Liberación de Recursos:** Después de salir del bucle, se liberan los recursos de la cámara y se cierran todas las ventanas.
- **Diferencia entre predicción y confianza:** La predicción significa tener un modelo de acuerdo con un conjunto de datos que puedan predefinir un nicho de conjuntos futuros. La confianza es lo la predicción hecha.

3.1.1. Etapa de Detección de rostro

Figura 3

Algoritmo de detección o reconocimiento



Nota. Elaboración propia, en la figura 3 se presenta el algoritmo de reconocimiento de rostro

Figura 4

Importar video captura de imágenes e importar la captura frontal de imágenes

Nota. Elaboración propia, en la figura 4 se importa las librerías video captura y adquisición frontal de imágenes, será el apartado que tendrá que aplicar como código de la línea 1, la librería cv2 de la cual nos va a permitir el video captura de imágenes del entorno, (una clase de videos por decirlo así). Así mismo en la línea 2 del código se importa la librería dlib el cual nos va a permitir hacer la captura frontal de imágenes.

Figura 5

Detección de rostro mediante modelo

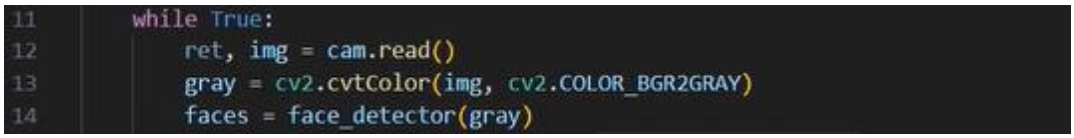
```
def face_detection(self, model, id_to_name, source=0, file_name=""):
    face_detector = dlib.get_frontal_face_detector()
    cam = cv2.VideoCapture(source)
    font = cv2.FONT_HERSHEY_SIMPLEX
    font_scale = 1
    font_thickness = 2
    lastName = ""
    lastTime = time.time()
```

Nota. Elaboración propia, en la figura 5 se muestra, para realizar la Detección de rostros, se va integrar el modelo, modelo que ya está entrenado previamente y el directorio, al ingresar la variable id_ to_name|, el cual nos va permitir ubicar la biblioteca mediante su índice correspondiente al nombre de la persona. Entonces con el nombre, dirección y models entrenado vamos a obtener la función faceDetection.

Luego la librería dlib, con la edición de la función get_frontal_face_detector () que hace la captura del rostro frontal y eso se va a almacenar en la variable face_detector. Para la parte de la cámara se hace el video captura, (0) es un argumento que siempre se coloca para el video captura que toma la cámara principal del dispositivo, en este caso el dispositivo normal, computador sería el valor cero (0) pertinente utilizando la Jetson nano, este valor puede cambiar dependiendo el ocupado de la cámara ya que si es uno o dos cámaras. Estas tres variables para la fuente que se va a mostrar dentro del recuadro que detecta el objeto, este mostrando una fuente tipo FONT_HERSHEY_SIMPLEX, el tamaño de la escala manteniendo font_scale en el valor de uno (1), no va a ver mayor escalamiento y el Font_thickness es igual a dos (2) que es igual al engrosamiento.

Figura 6

Proceso (while true), toma capturas infinitamente, capturas de video y captura de imagen



Nota. Elaboración propia, en la figura 6 se muestra el bucle, while este será el iterativo infinito siempre se va a dar (true). En el siguiente apartado tenemos en el almacenamiento en la variable imagen img, la lectura de la cámara, cam = cv2.videoCapture (0) video captura a tiempo real, img = cam.read() toma una imagen, por eso se condiciona, while true: infinitamente para que img = cam.read() toma una imagen repetidamente en todo instante de tiempo, no solo uno imagen, sino en todo infinitamente de tiempo siga tomando fotos, fotos, fotos. Se hace el escalamiento en escala de grises con la función cv2 y color BGR2GRAY y va a convertir RGB a escala de grises pasa de un vector unidimensional de tres capas RGB (rojo, verde y azul) lo va a pasar a una sola capa que a escala de grises, blanco y negro. Luego la variable faces se almacena dónde nos carga la imagen de la persona frontalmente face_detector = dlib.get_frontal_face_detector() y eso lo va convertir a escala de grises faces = face_detector(gray), la detección de rostro se realiza en escala de colores, no en escala de grises.

Figura 7

Desplazamiento y dimensiones de rostro

Nota. Elaboración propia, en la figura 7 se muestra el bucle for, donde vemos las variables de la imagen x, y, w, h

Bounding Box; cuadro delimitador, que es un sistema de coordenadas mediante el desplazamiento rectangular, utilizado para la detección de rostro.

w: ancho de la imagen del bounding box

h: altura de la imagen bounding box

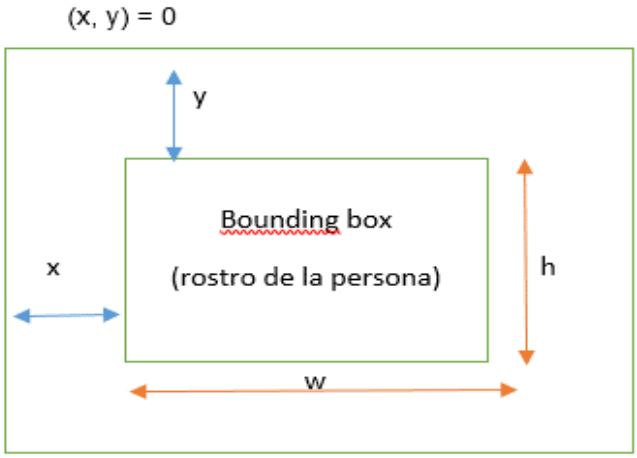
y: distancia entre el borde de la imagen superior y el bounding box

x: distancia entre la parte izquierda de la imagen y el bounding box

Se realizará la detección automáticamente; faces = fase-detector(gray)

Figura 8

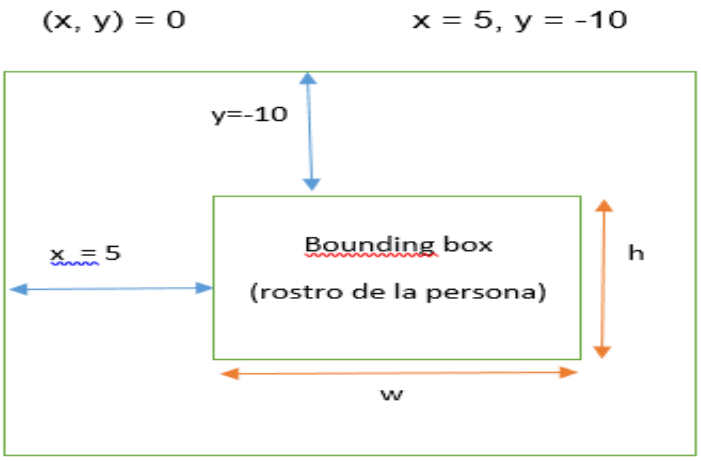
Un giro a la imagen completa, no al bounding box, sino a la imagen del entorno



Nota. Elaboración propia, en la figura 8 se muestra el recuadro del bounding box, se delimita mediante el desplazamiento, teniendo como variables para realizar la detección de rostro x, y, w, h.

Figura 9

Dimensiones del bounding box para la detección de rostro y posicionamiento



Nota. Elaboración propia, en figura 9 se muestra posición inicial del bounding box (x, y) =0. Así mismo sus dimensiones (w, h) del bounding box y el desplazamiento en su posición inicial, la derecha y de forma descendente para detección de rostro.

Figura 10

Nombre, índice, confidencia; lista para almacenar todas las predicciones

```
19        best_prediction = ("Unknown", -1, 100)  # (Name, ID, Confidence)
20        predictions = []  # List to store all predictions
21
```

Nota. Elaboración propia, en la figura 10 se muestra lista para mejor predicción y lista para almacenar todas las predicciones. Seleccionamos la mejor predicción con una variable de nombre desconocido “Unknown” que detecte a la otra persona que no está contenido en la base

de dato, entonces no reconoce el nombre, se denomina desconocido “Unknown” en este caso hay una predicción, un rostro con tal porcentaje de precisión que no es conocido, “que es desconocido”, por ello se asigna un valor de cien por ciento de precisión, para que un valor tenga cien por ciento de precisión es muy complicado y los casos reales casi nunca sedan, por eso siempre se pone el valor de la predicción de cien por ciento para luego hacer una comparación. En prediction = [ ], aquí vamos a necesitar, predicciones, por eso se coloca una variable nula donde vamos a ingresar los valores de las predicciones que están hechas en una lista [ ].

Figura 11

Detección de región de rostro con dimensiones válidas

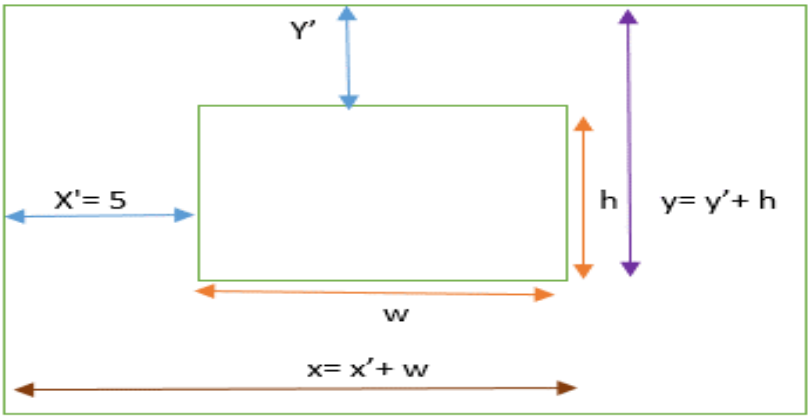
```
22          # Ensure the detected face region has valid dimensions
23          if x >= 0 and y >= 0 and w > 0 and h > 0 and (x + w) < gray.shape[1] and (y + h) < gray.shape[0]:
24              face_roi = cv2.resize(gray[y:y + h, x:x + w], (100, 100))
```

Nota. Elaboración propia, en la figura 11 se muestra la detección de las dimensiones del rostro tengan dimensiones válidas, por eso aquí se pone la condicional if, regresando al cuadro anterior del Bunding Box, la variable (x) no puede ser menor que cero, (x)<0, no sería posible, desde inicio seria mayor o igual a cero (x)≥0 para (y)≥0 también. Para (w) sea válido tiene que ser mayor que cero (w)>0, no puede ser igual a cero, ∉ w=0, ya que es contenido de la imagen, una imagen no puede tener un ancho de cero no existiría, de igual manera para la altura h. Así mismo, se realiza la suma respectiva (x+w), el valor del (x) de la imagen hacia el Bunding Box, el ancho del Bunding box donde está contenida la imagen, ese valor tiene que ser menor (x+w)< gray.shape[1] toma dos valores, el primer rango es [0] y el otro es [1] por la posición que esta, gray.shape[a b] esto cambia porque tiene una posición grayshape=[a b]; a=0, b=1, {0,1,2,3,4….}

Por ello se asigna a esta variable cero [0] con su lista dentro del cero, luego la posición [1] que sería gray.shape [1]. Esta suma de (w+x), es menor, luego (y+h)<gray.shape[0], nos aseguramos que tenga dimensiones válidas para aplicar un resize y se almacena en variable face_roi.

Figura 12

Imagen del bounding box con borde para detección de rostro



Nota. Elaboración propia, en la figura 12 se muestra imagen del bounding box con borde superior como imagen del entorno, para detección de rostro. Se realiza escala de grises para utilizar gray, así tener un rango y representar; ejemplo, gray [4 ,5], teniendo esos valores, donde (x)^(y) pasara a ser todo esos nuevos valores como se ve en la figura 12, gray [y:y+h , x : x+w], es el nuevo re-escalamiento que se tiene, ya que siempre es necesario para cualquier tipo de procesamiento de imagen.

Figura 13

Modelo entrenado para extracción de nombre de la persona

```
26                  for name, model in models:
27                      id, confidence = model.predict(face_roi)
28                      predictions.append((name, id, confidence))
29
```

Nota. Elaboración propia, en la figura 13 se muestra modelo entrenado para extracción de nombre del supuesto delincuente con orden de captura. En la variable for tenemos el models principal, que se ha entrenado previamente y a ese modelo se le extrae el name, model como variable para extraer el nombre de la persona. Para id determina el índice, confidence (la confianza en la mejor predicción de cien por ciento, esta variable le asignamos que contenga model.predict(face_roi), es la función para realizar las predicciones de nuestro modelo para la iteración models, del bucle for, para hacer la predicción del modelo (face_roi), seria cv2.resize(gray[y:y+h, x:x+w], (100,100)) donde ya está el re-escalamiento, la validación de las dimensiones y los valores de las variables (x), (y), (w), (h), donde face_roi=cv2.resize(gray[y:y+h, x:x+w], (100,100)) este es el bounding box que ha sido detectado un cuadro interno, que se realiza modelo de predicción para que nos dé el porcentaje de precisión. Hay una persona con un porcentaje del noventa y ocho por ciento, si digiera cuarenta por ciento, significa, que está mal el programa y no detecta bien, ósea el modelo está mal entrenado, (model.predict), la confidence y el id, está el índice de la persona. Así mismo se aplica predictions.append para que poner los argumentos del name que se extrae del models id, que es lo que se extrae, de la prediction se extrae todos los nombres; ejemplo, Karla, Carlos con estos dos nombres va a ser la predicción predict(face_roi), se detecta el bounding box de Karla, relacionado con su índice respectivo id (1), no es (2), porque de Carlos es (2), su porcentaje de confidence en un ochenta u noventa por ciento. Entonces con ese nombre que se extrajo al modelo y con la predicción para enlazarlo y estar seguro que se relacionan (name, id) con ello se afirma que el modelo está bien entrenado, porque relaciona bien el índice con el nombre y su confianza, eso es lo que hace predictions.append((name, id , confidence))

Figura 14

Predicción y valor de confianza

```
29
30                  if confidence < best_prediction[2]:
31                      best_prediction = (name, id, confidence)
```

Nota. Elaboración propia, en la figura 14 se muestra la predicción y el valor de confianza. parte del bet_prediction que se da un valor de confianza, del bet_prediction=(“Unknown”, -1, 100) número de nombre, índice, confianza que dimos un valor de 100 por ciento de confianza con un

valor desconocido se le asigna así queremos que esa variable se compare, entonces no se encuentra en la lista de la base de datos, es por ello no puede tener ningún nombre, Karla, Carlos, se hace la condicional (2) en if, confidence < best_prediction [2]: en best_prediction=("Unknown", -1, 100) # (name,Id,confidence), donde 2 es el valor 100%. La, best_prediction[2] me dara 100% y confidence < 100%.La variable best_prediction actualiza al append ((name, id, confidence)) anterior mente ya hemos realizado, unido, el nombre.

Tabla 2

Comparativa de índice, confianza y predicción

APPNEND			
NOMBRE	**ID(ÍNDICE)**	**NIVEL DE CONFIANZA**	**PREDICCIÓN**
Karla	1	98%	Buena
Carlos	2	60%	Mala

Nota. Elaboración propia, en la tabla 3 se muestra parámetros de índice, confianza y predicción. La confianza siempre será mejor si es del 100%, no existe un objeto que va a tener un 100% de precisión, porque siempre if confidence <best_prediction[2]: llegara a tener un porcentaje al 99,999…%, < 100%.

De la Tabla 2, para Karla, tendrá la mejor predicción, ya no tendrá nombre desconocido ("Unknown", -1, 100), su id, no será -1, porque ese valor no existe en los id, no existe valores negativos, solo valores positivos incluido el cero (0 1 2 3…). El nombre de Karla tendrá su posición como índice 1.

Tabla 3

Indica la mejor predicción

INDICA LA MEJOR PREDICCIÓN			
NOMBRE	**ID (ÍNDICE)**	**NIVEL DE CONFIANZA**	**PREDICCIÓN**
Karla	1	98%	% confidence, ya no es el 100%, es en un caso hipotético
Carlos	2	60%	Mala predicción

*Nota***.** Elaboración propia, en la tabla 4 se muestra la mejor predicción best_prediction=(name, id, confidence)

Figura 15

Mejor Predicción

Nota. Elaboración propia, en la figura 15 se muestra que la mejor predicción se basa en nombre de la persona, el índice de posición de orden y la confianza como resultado de predicción.

Tabla 4

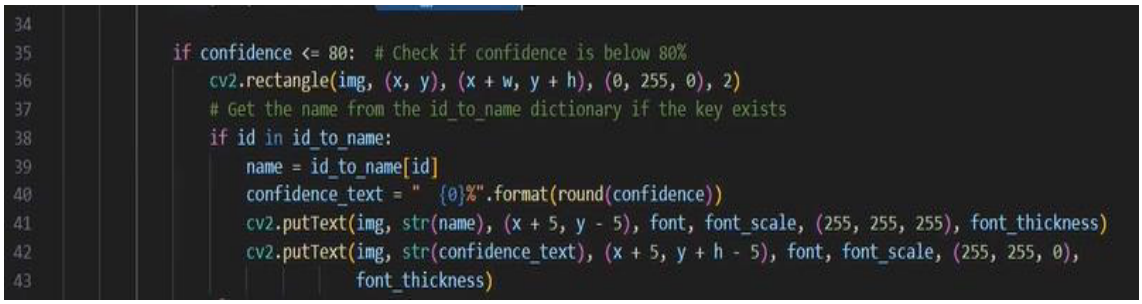
Porcentaje de confianza

NOMBRE (NAME)	INDICE (ID)	CONFIANZA (CONFIDENCE)
Karla	1	98%
Carlos	2	60%

Nota. Elaboración propia, Para la verificación del nivel de confianza de acuerdo a la tabla 5, indica de la mejor predicción best_prediction el name, id, y la confidence en determinado porcentaje de confiabilidad.

Figura 16

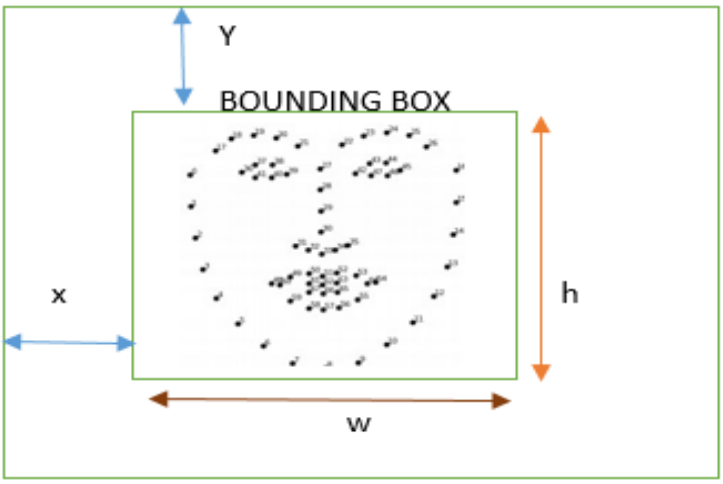
Comprobar si la confianza está por debajo del 80%, luego obtener el nombre del diccionario id_to_name si la clave existe



Nota. Elaboración propia, en la figura 16 se muestra todo el apartado se realiza una condicional, para verificar el nivel de confianza se alto, mayor ≥ 80 por eso se condiciona la confianza, si es menor o igual a 80 # como es el caso de Carlos 60%, entonces se dibuja un rectángulo en el rostro de la persona en el dibujo del bounding box.

Figura 17

Detectando la imagen del bounding box, la función rectángula



Nota. Elaboración propia, en la figura 17 se muestra el cv2.rectangle (img, (x,y), (x+w, y+h), (0, 255, 0),2) este apartado para hacer el dibujo rectángulo es el bounding box, antes no se usaba esta función cv2.rectangle, anteriormente, se solía utilizar estas variables x, y, w, h =face.let().face.top(),face with(), face.height(), toma esos valores, detecta la imagen en este apartado cv2.rectangle (img, (x,y), (x+w, y+h), (0,255.0),2). La función rectángular del bounding box.

Tabla 5

Relación Índice y nombre de la persona

ORDEN		BIBLIOTECA	
I	N	Índice Nombre	Índice Nombre
1	K	({1 , K}	{ 2 , C})
2	C	Id_to_name	id_to_name

Nota. Elaboración propia, en la tabla 6 se muestra para hallar el porcentaje de Carlos que es el 60%, que es su imagen, su recuadro del rostro, donde estará y luego se realiza la condicional, si el índice está en el orden, índice a nombre, si el índice es 1, está en la relación de índice a nombre se detecta.

Tabla 6

Biblioteca determina índice y nombre

ORDEN		BIBLIOTECA		
I	N	Índice Nombre		Índice Nombre
1	K	({1 , }	K	{ 2 , C})
2	C	0	1	id_to_name

> Posición (nombre de la persona)

Nota. Elaboración propia, en la tabla 7 se muestra la biblioteca, índice con nombre y otro índice, nombre, biblioteca, se selecciona índice 1 uno, nos va a votar la variable nombre k, Karla, el nombre siguiente, se selecciona el índice 2, se obtiene lo que está contenido en índice 2, que es el nombre de Carlos, con el cual se detecta que el 1 es el id, está dentro del id_to_name. Si esta detectado, entonces se realiza a colocar el nombre; ejemplo Karla. Este caso Karla seria, id =1 entonces id_to_name. A la disposición 1 seria porque este apartado está, biblioteca {1, k} -> posición cero 0.

Tabla 7

Confianza a porcentaje decimal

N	I	C
Karla	1	98%
Carlos	2	60%

Nota. Elaboración propia, en la tabla 8 se coloca la variable de confianza donde está nuestro porcentaje de confidence_text “(0)%”.format(round(confidence)), permitiendo redondear el valor de confianza ejemplo, de 93,999%, entonces con el (round(confidence)) redondeara al decimal más próximo 94% con el valor de porcentaje “{0}%”. De todo este confidence_texT= “{0}%”.format(round(confidence)) va a salir el 60% de Carlos.

Tabla 8

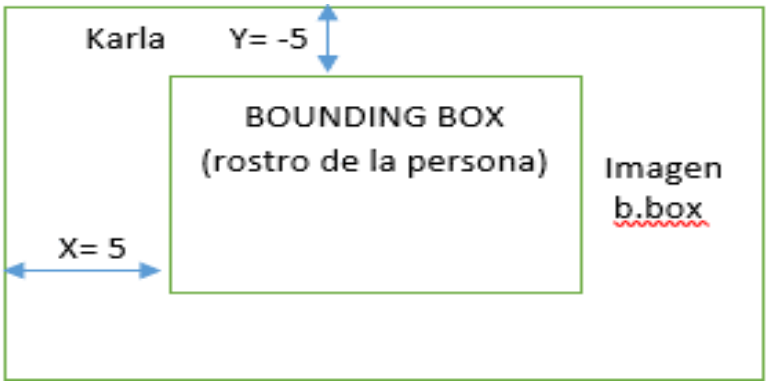
Posición desplazamiento del bounding box, respecto a la imagen del entorno

BOUNDING BOX	
X+5	y-5
right	below

Nota. Elaboración propia, en la tabla 9 se muestra la posición del desplazamiento del bounding box, cuando x=5 se desplaza a la derecha, cuando y = -5 desciende hacia bajo el bounding box para realizar la detección.

Figura 18

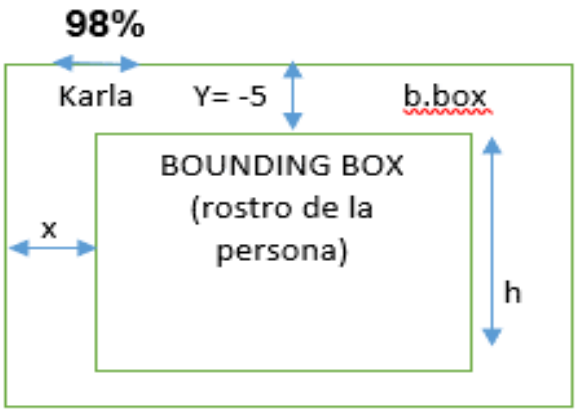
Reconocimiento del nombre de la persona



Nota. Elaboración propia, en la figura 18 se muestra el nombre de la persona, se realiza cv2.putText que en el video se muestre mientras que la persona está movimiento, digamos que el rostro sea detectado entonces ahí se va a colocar un texto es cual va estar contenido dentro de la imagen de este apartado cv2.putText(img,str(name), (x + 5, y-5), font, font_scale, (255, 255, 255), font_thickness), el cual se rige de la imagen, en este apartado de tipo nombre, nos va a dar el nombre de la persona name= id_to_name [id] en qué posición.

Figura 19

Nivel de confianza de precisión



Nota. Elaboración propia, en la figura 19 se muestra nivel de confianza de precisión. Así mismo, se declaró anteriormente font=cv2.FONT_HERSHEY_SIMPLEX, el tipo de fuente que ya se declaró, puede ser con arial, calibri diversos nombres. Apartados fondo-tamaños. Cv2.putText(img,str(confidence_text), (x+5, y+h -5), font,font_scale,(255,255,0), font_thickness), este apartado dentro de la imagen, pero este texto es el nivel de confianza ´por el porcentaje, el (x) ubicado a la derecha en ese mismo rango donde se visualiza, pero en el (y) si va a variar superpuesto se le suma y+h-5, el mismo (y-5), (donde está Karla en la imagen , 5 posiciones abajo + h).

Tabla 9

Número desconocido, índice no incluido en lista, número no encontrado

I	N	Base de dato		No en lista
1	K			
2	C			
3	J	{1 , k }	{ 2 , C}	{ 3 , J }

Nota. Elaboración propia, en la tabla 10 se muestra índice no incluido. En la biblioteca, en nuestra base de dato para que no tengamos otro tipo de dato que no es, especifico a lo que se busca, entonces si se detecta que tiene un id=3 o número desconocido entonces el índice no está incluido en la lista, en este caso no se encontraría.

Figura 20

Si la persona fue reconocida y es una nueva persona o han pasado 5 minutos

```
if name != lastName or time.time() - lastTime > 300:
    timeAndDate = datetime.datetime.fromtimestamp(time.time()).strftime('%Y-%m-%d %H:%M:%S')
```

Nota. En figura 20. Muestra a realizar si la persona fue reconocida, esto ocurre en la parte del algoritmo de detección y reconocimiento, si la persona ha sido reconocida existe dentro de esta lista entonces envía el mensaje.

Figura 21

Enviar mensaje: nombre ha sido reconocido fecha y hora

```
        sendSMS(f"{name} ha sido reconocido/a a las {timeAndDate}")
        lastName = name
        lastTime = time.time()
```

Nota. En la figura 21. esto se logra mediante la función datetime.datetime, pasamos al tiempo actual y lo convertimos en un texto, año, mes y la hora, pero se realiza mediante un if, si solo ha sido reconocido la persona.

Figura 22

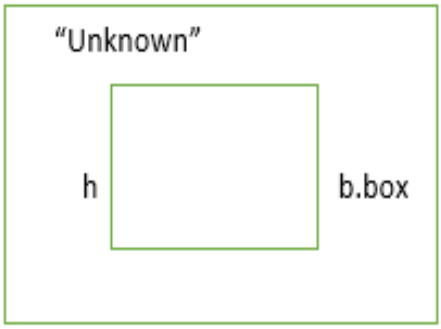
Texto desconocido

```
44                  else:
45                      cv2.putText(img, "Unknown", (x + 5, y - 5), font, font_scale, (255, 255, 255), font_thickness)
46                      cv2.putText(img, str(confidence), (x + 5, y + h - 5), font, font_scale, (255, 255, 0),
47                                  font_thickness)
```

Nota. En la figura 22. Muestra nivel de confianza con detección, pero con origen desconocido

Figura 23

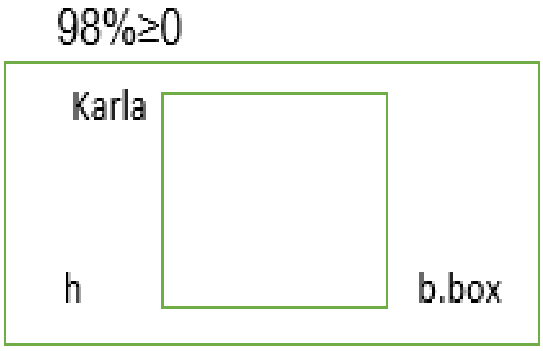
Nivel de confianza menor a 60%



Nota. Elaboración propia, en la figura 23 se muestra el nivel de confianza menor al 60 por ciento. En este caso se iría por el else, iba a poner un texto que diga "Unknown", si tiene nivel de confianza con la detección por ejemplo de 50%, con origen desconocido.

Figura 24

La predicción del porcentaje



Nota. Elaboración propia, en la figura 24 se muestra porcentaje de predicción, la confianza se basa en predicción del rostro capturado y la predicción no involucra el origen que sea desconocido, por ello puede detectar el porcentaje, pero no significa reconocer el nombre, no lo hará porque no se encuentra en la base de dato y no saldría algún nombre del listado de base de dato sino "Unknown". En caso detecte el nombre del listado, también saldrá el porcentaje, en este caso Karla 98%.

Figura 25

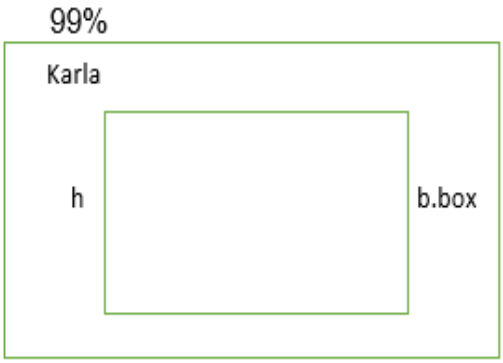
Porcentaje mayor a 80 porciento

```
48          else:
49              cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)
```

Nota. En la figura 25. Muestra en caso que el porcentaje sea mayor al 80 por ciento sería más preciso la confianza, entonces se describir el dibujo del rectángulo a la detección de rostro, ya no será necesario pasar por el proceso iterates condicional (if, else), porque el porcentaje es mayor a 80%.

Figura 26

El caso, donde el porcentaje de predicción es mayor a 80%



Nota. Elaboración propia, en la figura 26 se muestra, si el porcentaje sea mayor ≥ 80%, simplemente realiza el dibujo en el recuadro, simplemente ya no pasaran por todo este proceso de incondicional interactúes, ya que va ser preciso. Esto se hace por, cv2.rectangle(img,(x, y),(x+w, y+h),(0,0,255),2). Menor a 80% ya no están preciso, si fuera mayor a 80% simplemente se hace figura del recuadro.

Figura 27

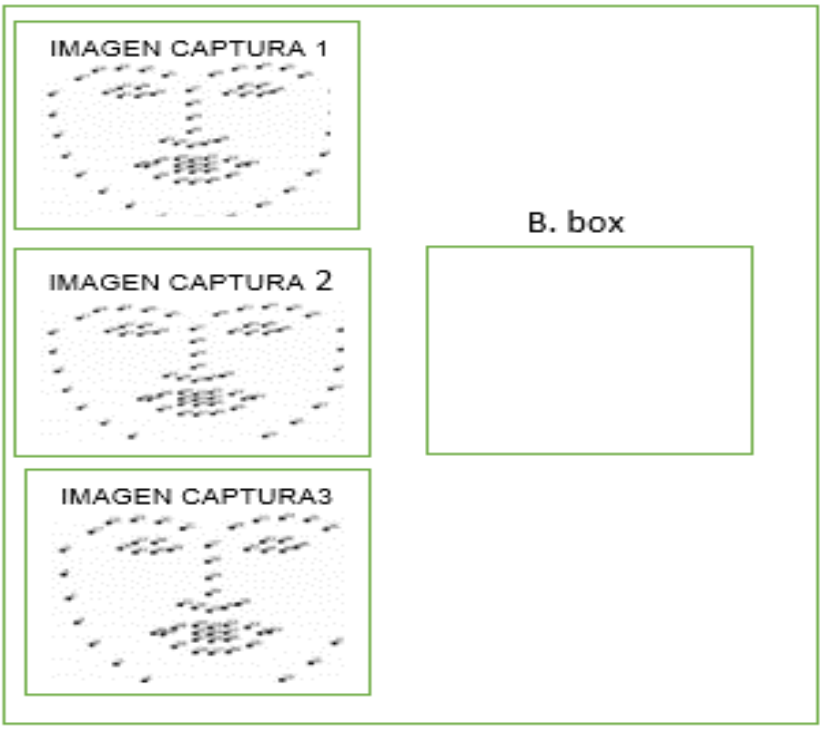
Recuadro de imagen de capturas de rostro

```
51          cv2.imshow('Camera', img)
```

Nota. Elaboración propia, en la figura 27 se muestra cv2.imshow que hace las capturas, entonces se hará la muestra.

Figura 28

Detección de captura de imágenes diferentes



Nota: Elaboración propia, en la figura 28 se muestra donde la interfaz aparte mostrara la imagen capturada imagen1, imagen 2, imagen 3 o desconocido. Resultado del bounding box y la rotación completa de la imagen en 10 grados.

Figura 29

Presione escape para salir del proceso y soltar la captura de video

```
53          k = cv2.waitKey(10) & 0xff  # Press 'ESC' to exit
54          if k == 27:
55              break
56
57      # Release the video capture
58      cam.release()
59      cv2.destroyAllWindows()
```

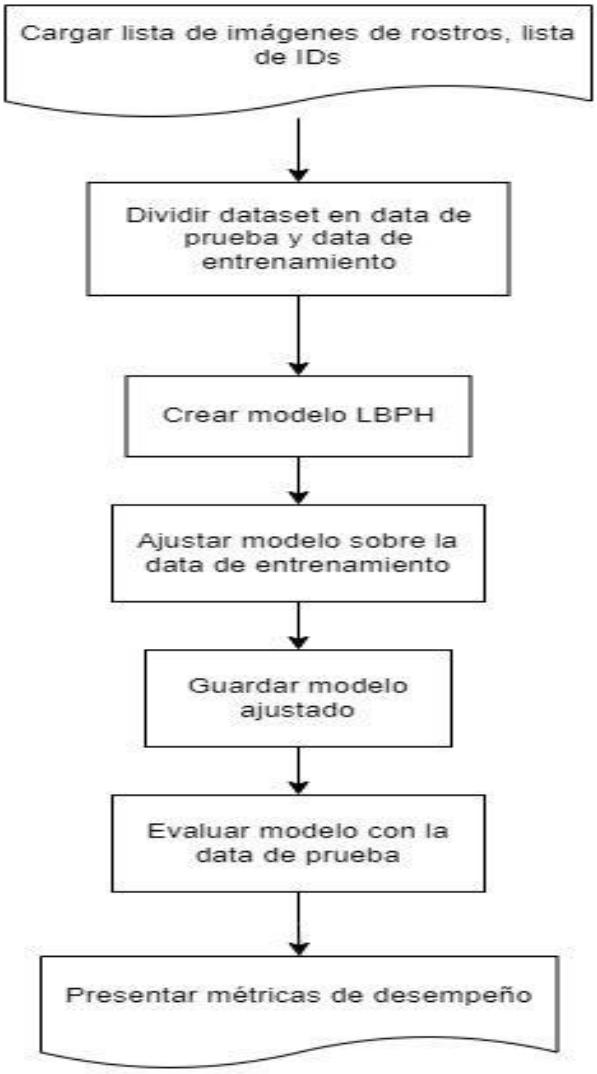
Nota. Elaboración propia, en la figura 29 se muestra escape para salir del proceso, hexadecimal porque en axff (escape, en el teclado), entonces con esta variable se espera 10segundos y la detección de la tecla escape. luego la condicional, que si la variable K asignada == 27, es el valor que Luego se hace una Variable waitkey el valor que quiere, este caso es de 10 ml/seg. Este es un código tantas veces se va a presionar escape, entonces con este número finaliza y termina

todo el problema, realizando haciendo todas aquellas capturas, el proceso del bucle(while true) es seguir tomando capturas infinitamente, capturas en el video y no dejara de tomar capturas, sobre todo este es un proceso rápido, entonces con este break rompemos este bucle infinito aplastando 27 veces, se puede poner 1ml/segundo, 0ml/segundo, pero esta para la parte del procesamiento 27, para que no se haga mucha carga computacional, por eso le asigna estos pocos valores 27, unas pausas más naturales y luego al final se tiene realizar estas dos funciones. La limpieza de la cámara que no tenga ninguna imagen tal cual como en el inicio y cv2.destroyALLwindows() es para todo el imagen o apartado que haga cv2 que se ha exportado o mostrado ya no siga, prácticamente estos dos limpiar el video captura.

3.1.2. Etapa de Entrenamiento

Figura 30

Algoritmo de entrenamiento



Nota. Elaboración propia, en la figura 30 se muestra el diagrama de flujo del entrenamiento

Figura 31

Función realiza el entrenamiento

```
148
149     # Función que realiza el entrenamiento del modelo con la data, lo guarda como en el disco duro, y lo valida para determinar su confiabilidad
150     def face_learning(self):
```

Nota. Elaboración propia, en la figura 31 se muestra dicho modelo redice un rostro a que persona predice dicho rostro, la función también debe guardar en el disco duro un archivo modelo para usar en esta de detección de reconocimiento en tiempo real, y finalmente determina la confiabilidad y la precisión del modelo sobre una data de prueba.

Figura 32

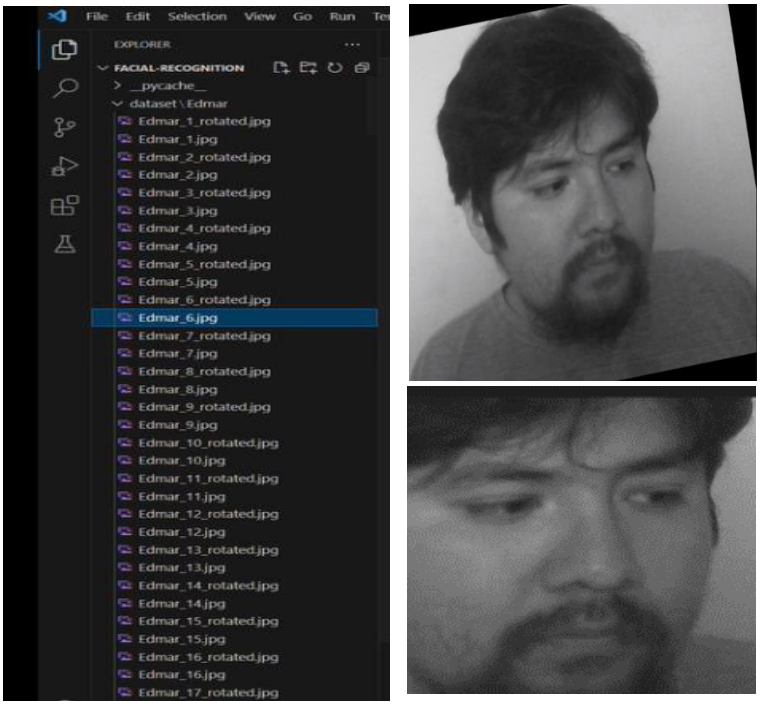
Folder donde están guardadas las imágenes

```
151
152         # El folder donde están guardadas las imágenes
153         path = 'dataset'
154
```

Nota. Elaboración propia, en la figura 32 se define mediante la función del algoritmo, donde se guardan imágenes, es el folder donde están guardadas, imágenes recolectadas, y luego mediante otra función def face_sanpling(self source-B) se dice que aquí están guardadas las imágenes path='dataset'.

Figura 33

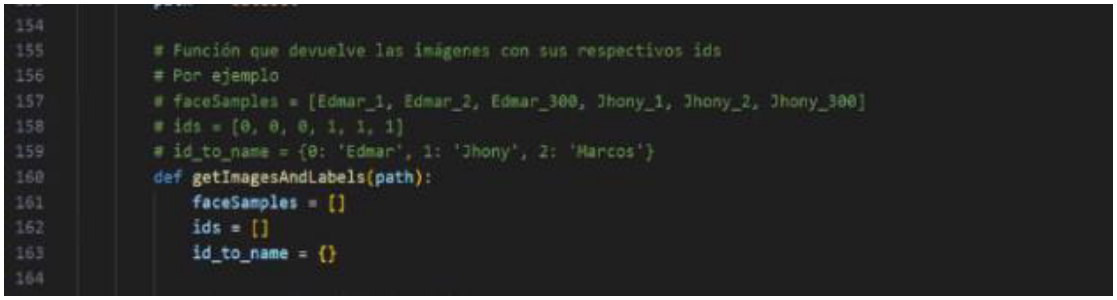
Forma que están guardadas las imágenes



Nota. Elaboración propia, en la figura 33 se muestra el folder diferentes archivos con respecto a su rostro de la persona. Por cada persona hay un directorio para diferentes personas.

Figura 34

Función para extraer archivo

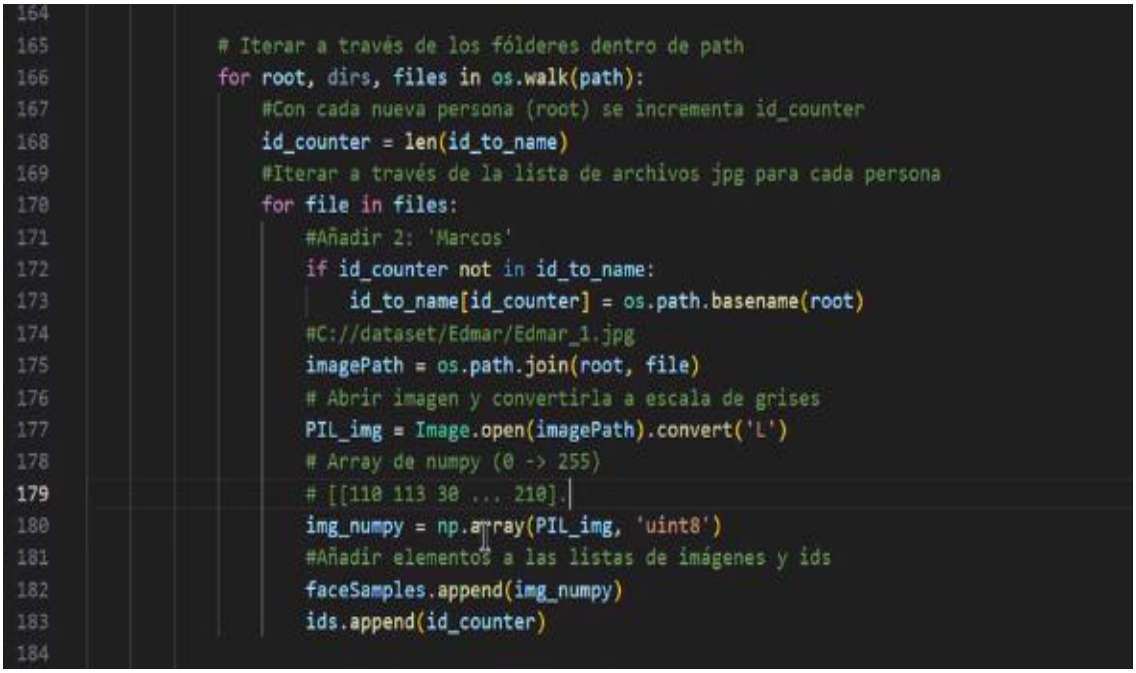


```
154
155     # Función que devuelve las imágenes con sus respectivos ids
156     # Por ejemplo
157     # faceSamples = [Edmar_1, Edmar_2, Edmar_300, Jhony_1, Jhony_2, Jhony_300]
158     # ids = [0, 0, 0, 1, 1, 1]
159     # id_to_name = {0: 'Edmar', 1: 'Jhony', 2: 'Marcos'}
160     def getImagesAndLabels(path):
161         faceSamples = []
162         ids = []
163         id_to_name = {}
164
```

Nota. Elaboración propia*,* en figura 34 se muestra las funciones para extraer un archivo de rostro para cada persona corresponde se guarda en una lista que se usa para el entrenamiento y la evaluación, la forma de la data, es una lista donde cada elemento es una imagen convertida en un arrive una imagen de pixeles de [0 255 255] sería un arrive de dimensión 2,[filasxcolumnas], cada de unos de los elementos es un arrrive, correspondiente a cada elemento o imagen se tiene el id identifica a la persona guardando en orden la id correspondiente a cada imagen, si hay trecientos Carlos debería de aver trecientos ceros, si existe trecientos Karlas debe haber trecientos ceros, guardando el orden y el ide a dichas imágenes a la persona correspondiente, si ubiera otra persona (imagen), también tendría que haber otra lista de ids correspondiente para aquella persona en un mismo orden. Esta función get entrega la lista de los ids con la persona correspondiente a dicha id, la persona 1 y la persona 2 tiene diferentes ids, persona 1 tienes id cero, persona 2 tiene id uno con sus respectivos fólderes.

Figura 35

Iterando a través de los fólderes dentro de path



```
164
165         # Iterar a través de los fólderes dentro de path
166         for root, dirs, files in os.walk(path):
167             #Con cada nueva persona (root) se incrementa id_counter
168             id_counter = len(id_to_name)
169             #Iterar a través de la lista de archivos jpg para cada persona
170             for file in files:
171                 #Añadir 2: 'Marcos'
172                 if id_counter not in id_to_name:
173                     id_to_name[id_counter] = os.path.basename(root)
174                 #C://dataset/Edmar/Edmar_1.jpg
175                 imagePath = os.path.join(root, file)
176                 # Abrir imagen y convertirla a escala de grises
177                 PIL_img = Image.open(imagePath).convert('L')
178                 # Array de numpy (0 -> 255)
179                 # [[110 113 30 ... 210].
180                 img_numpy = np.array(PIL_img, 'uint8')
181                 #Añadir elementos a las listas de imágenes y ids
182                 faceSamples.append(img_numpy)
183                 ids.append(id_counter)
184
```

Nota. Elaboración propia, en la figura 35 se muestra, iterando a través de los sub-fólderes se le asigna la variable root y con cada nueva persona se aumentará los ids, si se tiene dos fólderes el id_counter aumentara con cada persona que se agrega, con cada nueva persona se va iterar

la lista de sus imágenes (rostro), en caso la persona no se encuentra en el diccionario id_to_name, por ejemplo encontremos el nombre Marcos que no se encuentra registrado, entonces se crea una nueva entrada [2, nombre de la persona (Marcos)] esto se realiza una sola vez cada vez que se encuentra una persona nueva.

Figura 36

Extracción de las imágenes de rostro en ruta

Nota. Elaboración propia, en la figura 36 se muestra en cada uno de las imágenes se extrae nuevas rutas de imágenes mediante la función PIL_img = Image.Open(imagepath), ejemplo para la imagen edma1 jpg la ruta puede ser c://dataset/Edmar/Edmar_1.jpg, loque hacemos luego convertir ese archivo en escala de grises.

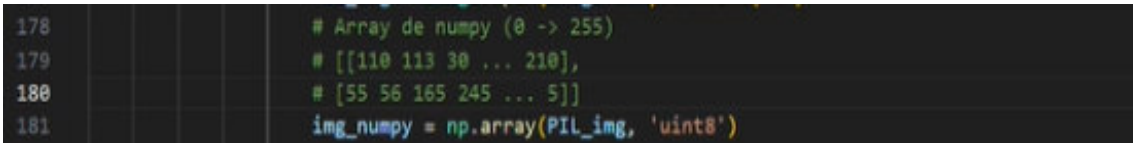
Figura 37

Función convierte en escala de grises

Nota. Elaboración propia, en la figura 37 se muestra la función que habré el archivo de imagen convirtiendo a escala de grises mediante la función (convert('L')) con la opción L.

Figura 38

El Array tiene un numero de filas y un número de columnas de la imagen, donde cada pixel corresponde a un número



Nota. Elaboración propia, en la figura 38 se muestra luego de convertir en escala de grises, para luego convertir en un array de numpy que cada pixel va a corresponder a un número (0 → 255) este array tiene un numero de filas de la imagen y un número de columnas del imagen donde cada pixel corresponde a un número.

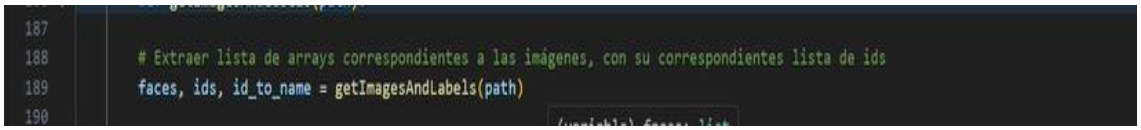
Figura 39

Añadir elementos a las listas de imágenes

Nota. Elaboración propia, en la figura 39 se muestra finalmente tanto ese array correspondiente a rostro como el id correspondiente a dicho array o dicho imagen son guardados en la data faceSamples.append(img_numpy), ids.append(id_ccunter).

Figura 40

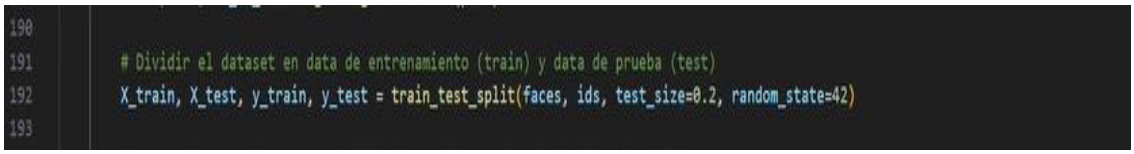
Obteniendo lista de imágenes, lista de ids y diccionario



Nota. Elaboración propia, en la figura 40 se muestra luego de ejecutar la anterior función se ha logrado obtener lista de imágenes, lista de ids y el diccionario id_to_name, que es la lista de ids y su correspondiente nombre o persona # id_to_name {0: 'Edmar', 1: 'Jhonny', 2: 'Marcos'}

Figura 41

Dividir el dataset en data de entrenamiento (train) y data de prueba tesh



Nota. Elaboración propia, en la figura 41 de muestra a continuación, se realiza con la data dividirla en data de entrenamiento que nos permite entrenar el modelo y data de prueba que nos va a permitir validarlo, para saber si es confiable y preciso, se utiliza la función train_test_split que es una utilidad de sklearn, loque se hace es pasar todo la data de faces, ids, y le asiganamos que la quinta parte de test_size=0.2 de esa data que clasifique como data de prueba y las otras cuartas quintas partes queden como parte de entrenamiento. Lo que hace esto, de la lista de faces , lista de ids aleatoreamente coje la quinta parte y la guarda en X_tast y en Y_test en las imágenes de prueba y en ids de prueba y el resto también lo mescla aleatoriamente lo guarda en X_train y en Y_train, lo que se va lograr esto es tener data separa en dos partes, una para entrenar y otra para validar.

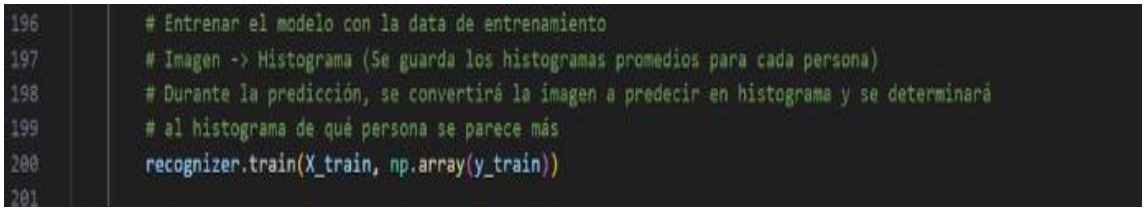
Figura 42

Reconocimiento de rostro, empleando lbph

Nota. Elaboración propia, en la figura 42 se muestra un nuevo modelo de la función del algoritmo lbph para reconocimiento de patrones y posturas y que se emplea para reconocimiento de rostro,

Figura 43

Función modelo de entrenamiento



Nota. Elaboración propia, en la figura 43 se muestra la función del entrenamiento, este modelo que se extrae previamente para ello vamos a invocar la función Xtrain y nuestras y ids de entrenamiento que es Ytrain. Lo que se logra que a cada imagen le convierte en una representación de dicha imagen histograma y de todos los histogramas de una persona se saca

un histograma promedio que es como una representación del rostro de esa persona, y luego al momento de predecir de la imagen de la persona, predecir qué persona es, convirtiéndose en un histograma y se compara con los histogramas con las personas guardadas en el modelo determinando a que persona pertenece se parece más.

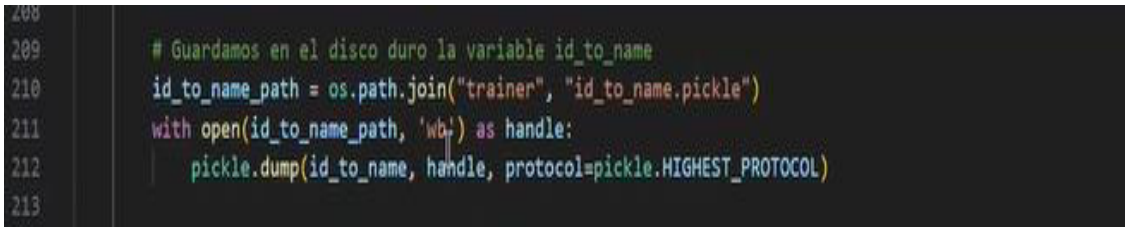
Figura 44

Crear folder trainer

Nota. Elaboración propia, en la figura 44 se muestra las imágenes de rostros son guardadas en los fólderes el modelo y se va a guardar en el disco duro mediante la función (model_path) guardando el modelo.

Figura 45

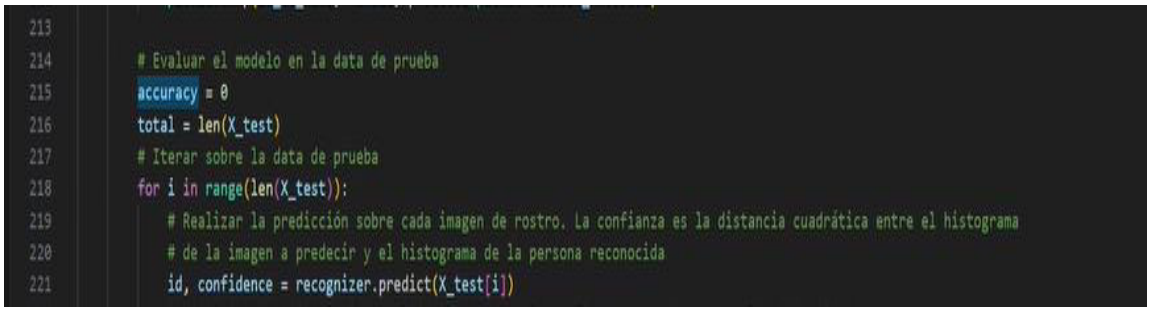
Guardar en el disco duro la variable id_to_name.pickle



Nota. Elaboración propia, en la figura 45 se muestra la función donde guardamos en el disco duro la variable ids y las personas que correspondían y el diccionario (“id_to_name.pickle”),usando el modulo pickle nos permite guardar en la variable y luego poder extraer nuevamente, en que momento de la detección vamos a poder extraer, si en la predicción, nos arroja un id 1 tenemos que saber a que persona corresponde ese id, entonces con esa variable que se va a recuperar que corresponde el id 1→ {1:’jhony’}.

Figura 46

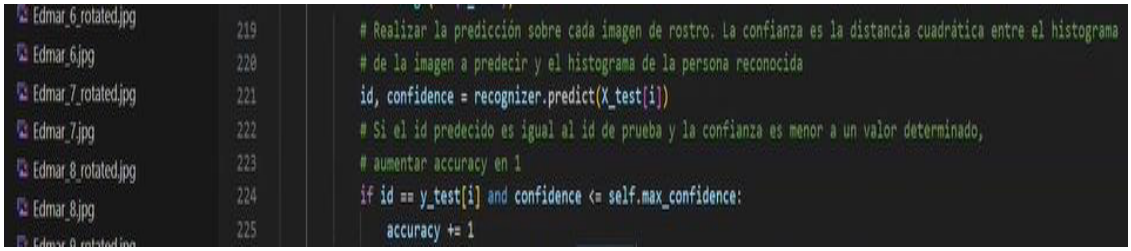
Evaluar el modelo en la data de prueba



Nota. Elaboración propia, en la figura 46 se muestra un modelo entrenado, un modelo a que evaluar, que tan preciso y confiable es ello, vamos iterar solo la data de prueba X_tesh y en cada X_tesh se realiza la predicción y vamos a determinar a qué id de persona corresponde para ello vamos a iterar en la data de prueba (x_test), y en cada uno de esos (x_test).

Figura 47

Función realiza predicción sobre cada imagen de rostro



Nota. Elaboración propia, en la figura 47 se muestra la predict con el modelo ya entrenado, y vamos a determinar, a que persona id de persona corresponde dicha imagen, también, además de la persona, nos arroja un valor de confianza, este valor es la distancia cuadrática entre el histograma de este imagen (x_test(i)) que queremos clasificar y el histograma de la persona que hemos conocido mientras no esté valor confidence, más cerca, más confiables era muestra predicción, más seguro estamos que ese rostro corresponde a la persona.

Se considera una predicción correcta si el id que se a predicho es igual al id de prueba al id almacenada (y_test) y la confianza es lo suficientemente pequeña, en caso de que aumente en 1 el número de operaciones correctos o aciertos.

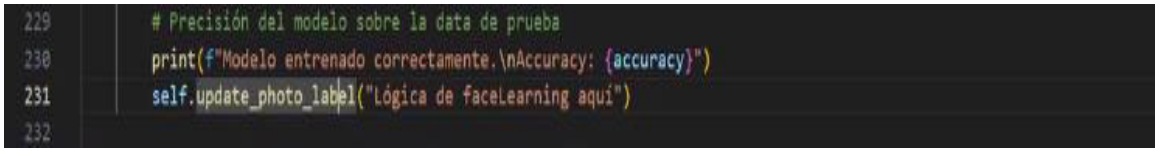
Figura 48

Dividir el número de predicciones correctas entre el número total de predicciones

Nota. Elaboración propia, en la figura 48 se muestra a continuación una vez que termine de iterar sobre toda la data de prueba, se calcula cuantas operaciones correctas hubo con respecto al número total de predicciones y eso no da la precisión o accuracy, esa predicción la devolvemos a la consola.

Figura 49

Precisión del modelo sobre la data de prueba



Nota. Elaboración propia, en la figura 49 se muestra, Finalmente tenemos un modelo entrenado guardado en el disco duro, y luego este modelo se puede utilizar o emplear en la parte de detección y reconocimiento.

3.2. DESARROLLAR UN PROTOTIPO DE SISTEMA DE SOFTWARE QUE INTEGRE EL MÓDULO DE CAPTURA, PROCESAMIENTO, ALMACENAMIENTO E INTERFAZ DE RECONOCIMIENTO FACIAL

En el contexto de la creciente necesidad de fortalecer la seguridad pública y la eficacia en la identificación de personas con órdenes de captura, se propone el desarrollo de un prototipo de sistema de software especializado. Este sistema tiene como objetivo integrar de manera eficiente el proceso de captura, procesamiento, almacenamiento e interfaz de reconocimiento facial, con el propósito de identificar delincuentes con órdenes de captura de manera rápida y precisa.

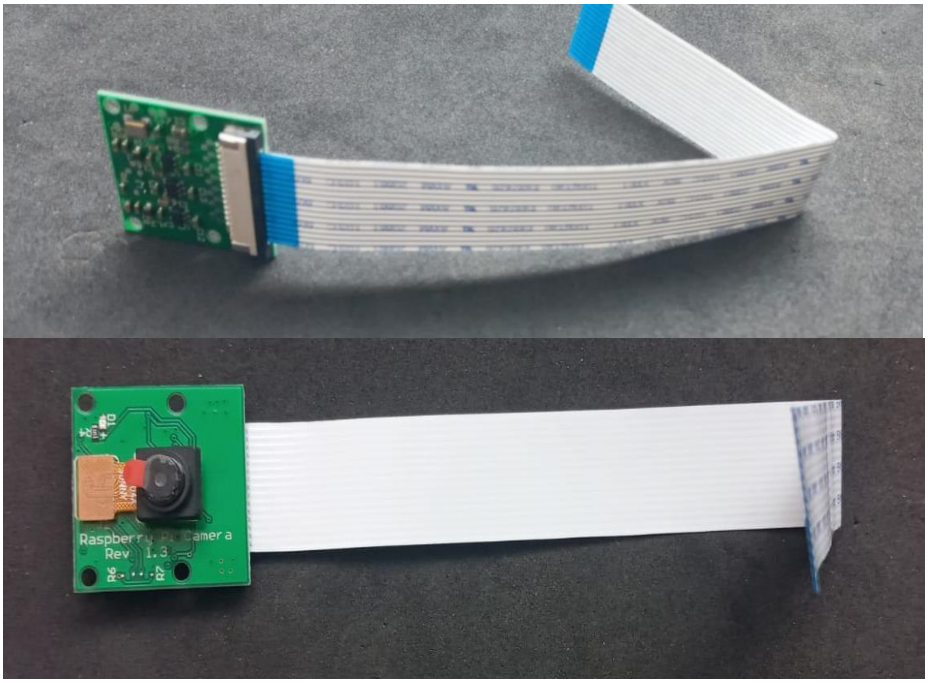
El prototipo se centrará en la aplicación de tecnologías avanzadas de visión por computadora y reconocimiento facial para mejorar los métodos tradicionales de identificación de individuos. La integración de un módulo de captura permitirá la adquisición de imágenes faciales en tiempo real, mientras que el módulo de procesamiento se encargará de analizar y extraer características distintivas de cada rostro capturado.

El sistema también contemplará un robusto módulo de almacenamiento que garantizará la seguridad y accesibilidad de la información recolectada, asegurando la confidencialidad de los datos sensibles relacionados con las identidades de los individuos buscados. Además, se implementará una interfaz de usuario intuitiva y funcional que facilitará la interacción con el sistema, permitiendo búsquedas rápidas y eficaces.

Este proyecto no solo aspira a optimizar los procesos de identificación de delincuentes con órdenes de captura, sino también a contribuir significativamente a la mejora de la seguridad pública y la capacidad de las fuerzas del orden para abordar de manera eficiente situaciones críticas. A través de la implementación de este prototipo, se busca explorar y demostrar la viabilidad de la tecnología de reconocimiento facial como una herramienta valiosa en el ámbito de la seguridad y la justicia.

Figura 50

Cámara PI 8 Mg pixeles resolución 1080p y 720p



Nota. Fuente propia, en la figura 50 se muestra la placa cámara Raspberry Pi v2 de 8 megapíxeles que se caracteriza por su gran efectividad en el desempeño de captura de alta resolución de videos, imágenes. Así mismo, generando imágenes estáticas de 3280x2464 pixeles, video de 1080p30, 720p60 y 640x480p90.

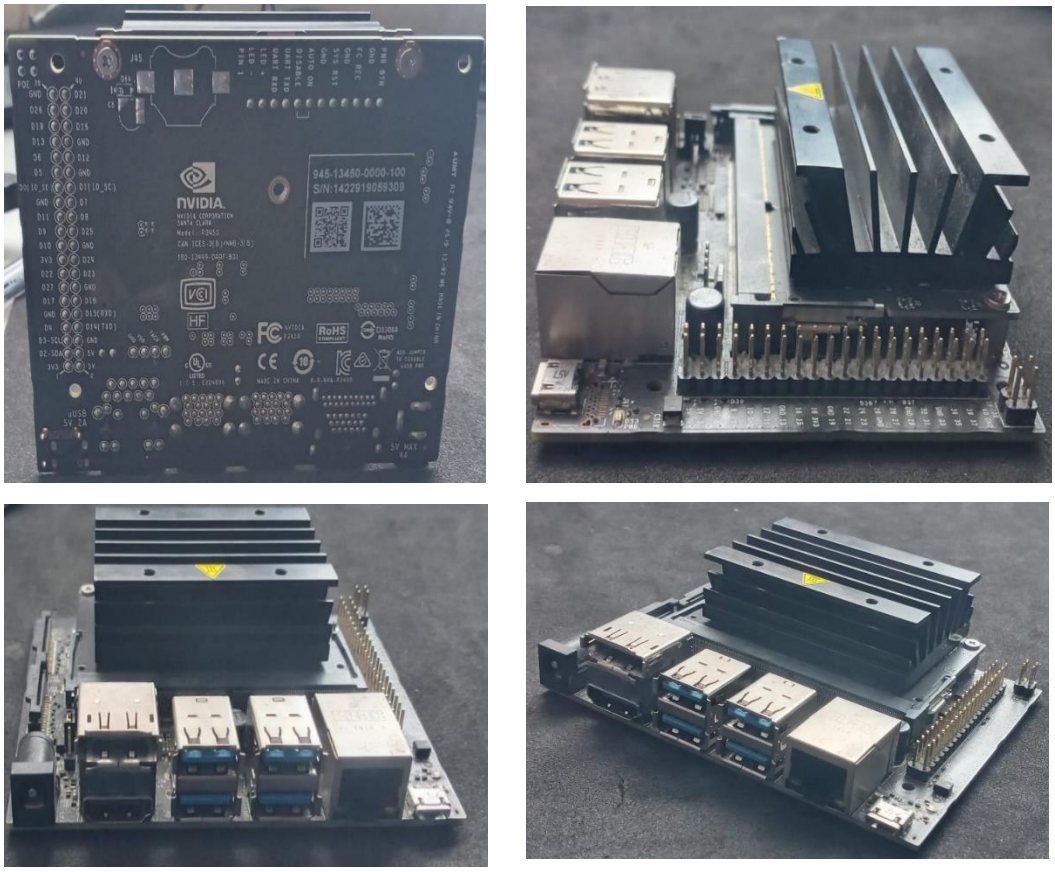
Para la interfaz que conecta la placa, Raspberry Pi Camera tiene el estándar CSI (Camera Serial Interface) que puede ser compatible con la mayoría de las placas de desarrollo como Raspberry Pi 4, NVIDIA Jetson Nano.

Características de la Cámara Raspberry Pi V2 8MP:

- Alta calidad: intercala un sensor de imagen Sony IMX219 de 8 megapíxeles.
- Alta resolución: Capaz de imágenes estáticas de 3280x2464 píxeles.
- Gran rendimiento: captura video a 1080p30, 720p60 y 640x480p90.
- Amplia compatibilidad: compatible con la mayoría de SBC o placas de desarrollo que tienen CSI estándar.

Figura 51

Placa de desarrollo Jetson Nano



Nota. Fuente propia, en la figura 51 se muestra la placa de desarrollo Jetson Nano de la marca Nvidia 4gb RAM lo cual es ideal para la aplicación de inteligencia artificial, programación, robótica. Por medio del HMDI se va conectar a una pantalla(monitor) y mediante el mouse para su configuración inicial para este proyecto de investigación de reconocimiento facial de supuestos delincuentes con orden de captura.

Placa de desarrollo Jetson nano de la marca Nvidia 4gb RAM de lo cual es ideal para la aplicación con inteligencia artificial, programación y robótica.

Por medio del HMDI se va a conectar a una pantalla (monitor) y ratón para su configuración inicial.

Especificaciones Técnicas de hardware Jetson nano

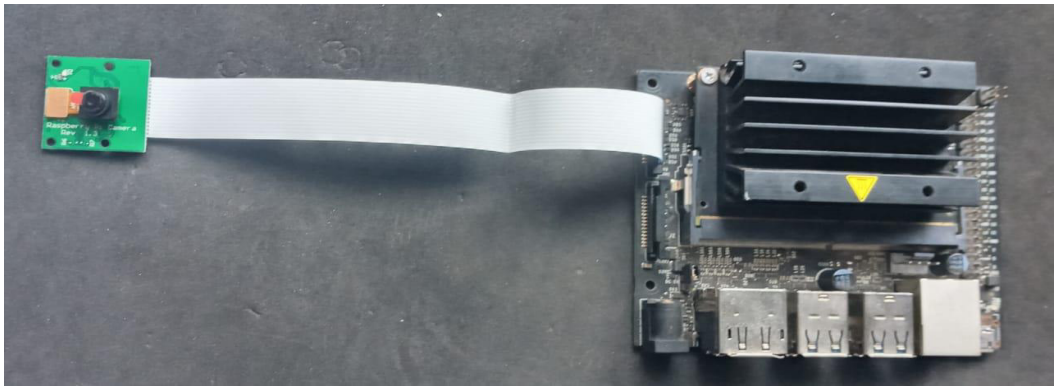
- GNU: NVIDIA Maxwell con 128 núcleos
- Procesador: ARM A57 de cuatro núcleos a 1,43Ghz
- Memoria: LPDDR4 de 64bits 4GB 25,6GB/s
- Almacenamiento: microSD (no incluido)

- Codificador de video: 4K @ 30| 2x1080p @ 60 | 4x1080p @ 30 | 9x720p @ 30 (H.264/H.265)
- Decodificador de video: 4K @ 60 | 2x 4K @ 30 | 4x 1080p @ 60 | 8x1080p @ 30 | 18x720p @ 30 (H.264/H.265)
- Cámara: x2 conectores MIPI CSI-2 de 15 patillas y 2 vías
- Pantalla: HDMI 2.0 y DP 1.2
- Conectividad: Gigabit Ethernet, dongle M.2 E
- USB: x4 USB 3.0, USB 2.0 Micro-B
- x40 GPIO, I2C, 12S, SPI, UART, toma de ventilador de 4 patillas, toma POE de 4 patillas.
- Alimentación: Micro-USB 5V 2A o DC 5V 4ª
- Botones de encendido, recuperación forzada y reinicio Mecánica: 100 mmx79 mmx31 mm (incluye placa de soporte, módulo y solución térmica)

Un ejemplo de proyecto innovador que contribuye al avance de la tecnología es el Jetson Nano, una potente pieza de hardware utilizada para robótica, programación, inteligencia artificial y otros campos. Uno de estos proyectos es un sistema de control de acceso para entrar y salir de una empresa determinada. Del mismo modo, el mismo hardware se utiliza en el proyecto de investigación de reconocimiento facial en curso, cuyo objetivo es identificar a presuntos delincuentes con órdenes de detención pendientes y causar malestar público en la sociedad actual.

Figura 52

Cámara PI de 8 Mg pixeles con resolución 1080p y 720p conectada por flat a la Jetson nano conectada a uno de los puertos de conexión flat de 16 bits



Nota. Fuente propia, en la figura 52 se muestra el vínculo de instalación de cámara pi de la Raspberry, mediante un flex de 16 bits y el hardware Jetson nano.

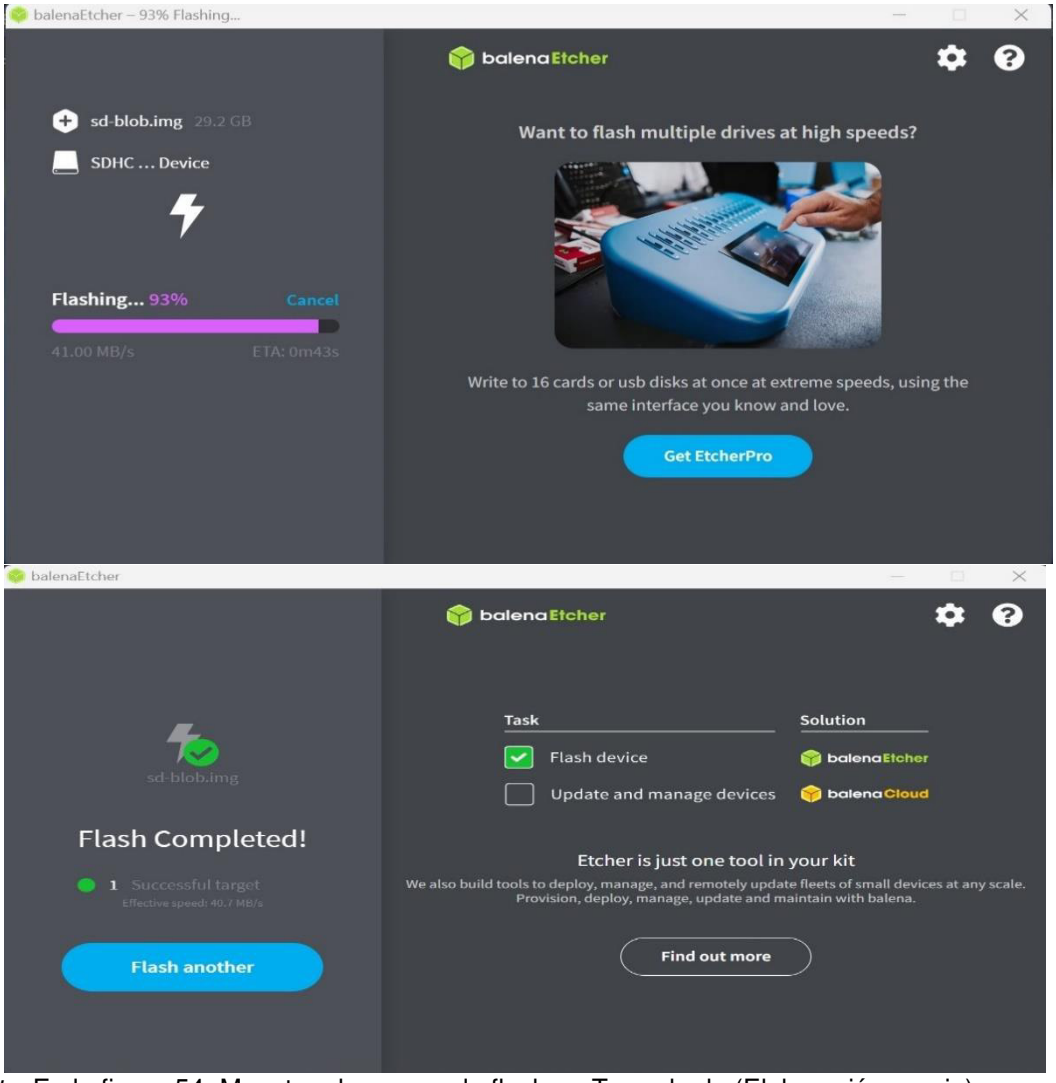
Figura 53

Tarjeta SD

Nota. En la figura 53. Tarjeta SD la imagen del sistema operativo es grabada en la tarjeta SD, ya entrando grabado se coloca.

Figura 54

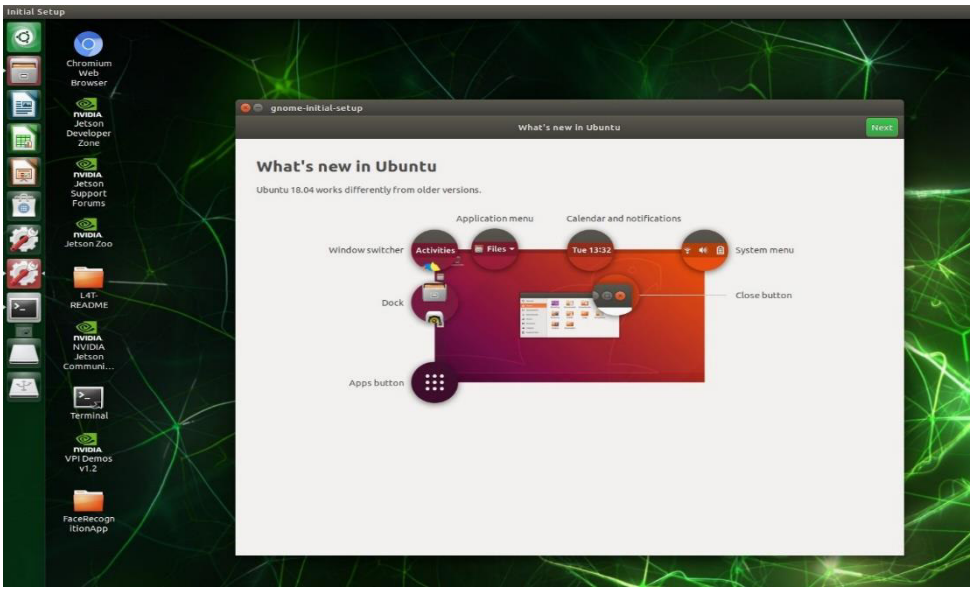
Flasheo del sistema operativo en tarjeta SD



Nota. En la figura 54. Muestra el proceso de flasheo. Tomado de (Elaboración propia)

Figura 55

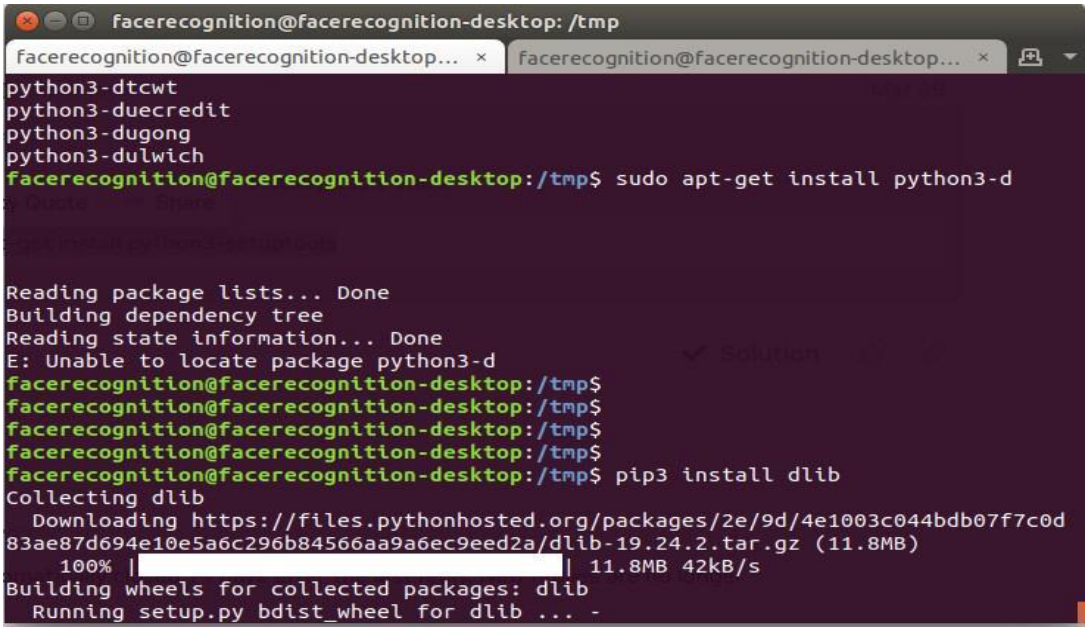
Inicio del sistema operativo y configuración



Nota. En la figura 55. Muestra la configuración, consiste en nombre del usuario, contraseña, idioma del sistema operativo (Elaboración propia)

Figura 56

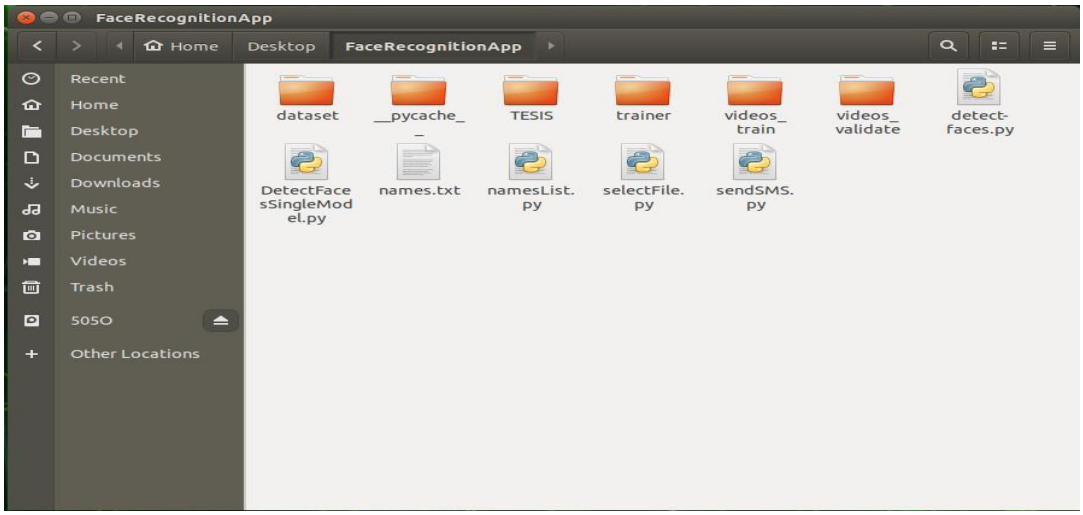
Instalación de dependencias



Nota. En la figura 56. Muestra instalación de los módulos de los que depende el programa de reconocimiento facial. Archivo de DetetFacessingleModel.py X algoritmo send sms.pyx. Tomado de (Elaboración propia)

Figura 57

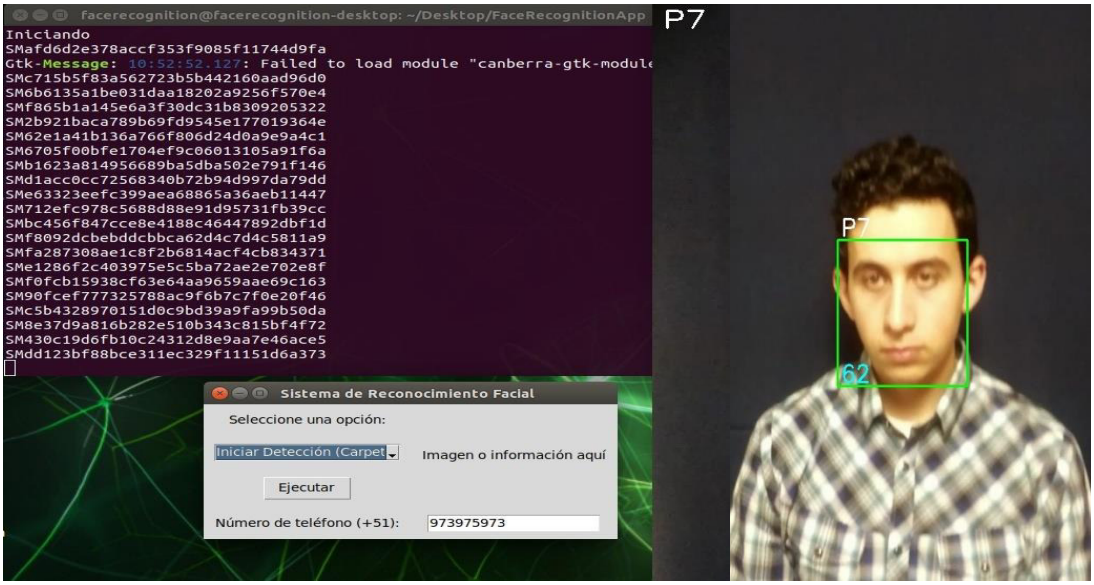
Folder correspondiente al programa de reconocimiento



Nota. En la figura 57. Muestra incorporación del programa de reconocimiento desde la PC al sistema Jetson Nano. Tomado de (Elaboración Propia)

Figura 58

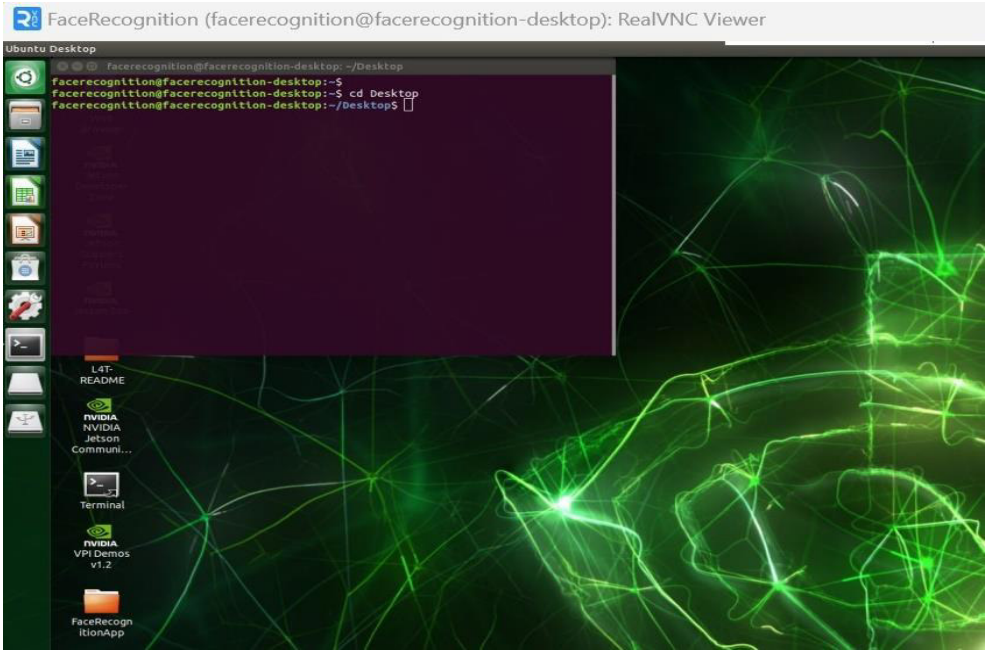
Reconocimiento facial de supuesto delincuente con orden de captura



Nota. En la figura 58. Muestra la ejecución del programa de reconocimiento facial en el sistema Jetson Nano. Tomado de (Elaboración propia)

Figura 59

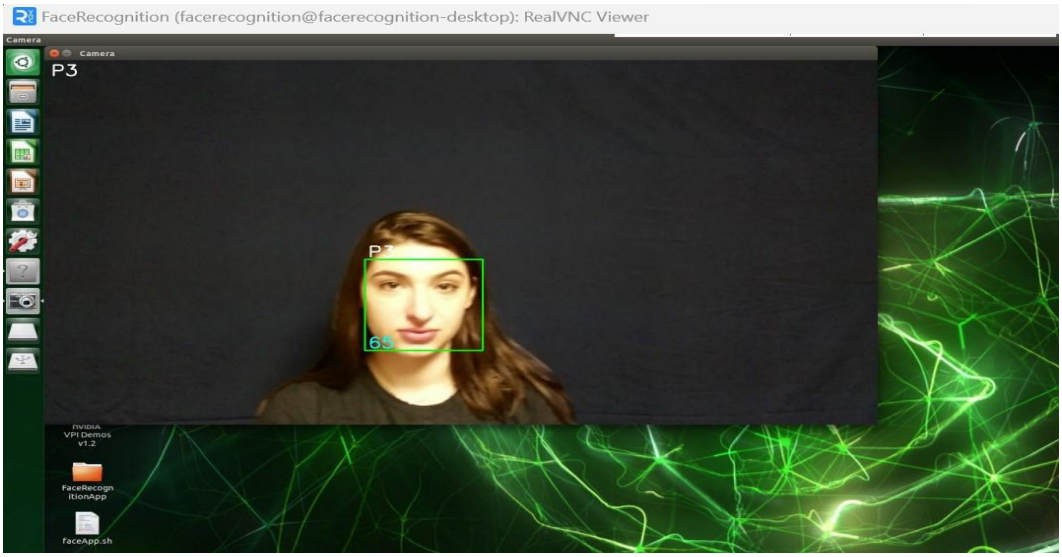
El proceso de reconocimiento en el sistema operativo jetson nano



Nota. En la figura 59. Se muestra el proceso de reconocimiento en el sistema operativo para la validación entre el software y hardware para el reconocimiento facial (elaboración propia)

Figura 60

Reconocimiento facial de sistema operativo jetson nano



Nota. En la figura 60. Se muestra el reconocimiento facial de la persona p-3, (elaboración propia)

3.3. IMPLEMENTAR TÉCNICAS DE PROCESAMIENTO DIGITAL DE IMÁGENES QUE BRINDAN UN ALTO NIVEL DE CONFIABILIDAD

En la búsqueda constante de perfeccionar las capacidades de los sistemas de reconocimiento facial, se propone la implementación de técnicas avanzadas de procesamiento digital de imágenes. Este objetivo tiene como finalidad incorporar métodos que eleven la confiabilidad y precisión del reconocimiento facial, contribuyendo así a la efectividad de la identificación de individuos, en particular, aquellos sujetos a órdenes de captura.

El procesamiento digital de imágenes desempeña un papel fundamental en la mejora de la calidad visual de las imágenes faciales capturadas, permitiendo la extracción de características distintivas y la reducción de posibles fuentes de error. En este contexto, la implementación de técnicas innovadoras busca superar desafíos comunes, como variaciones en la iluminación, cambios en la pose facial y otros factores que puedan afectar la exactitud del reconocimiento.

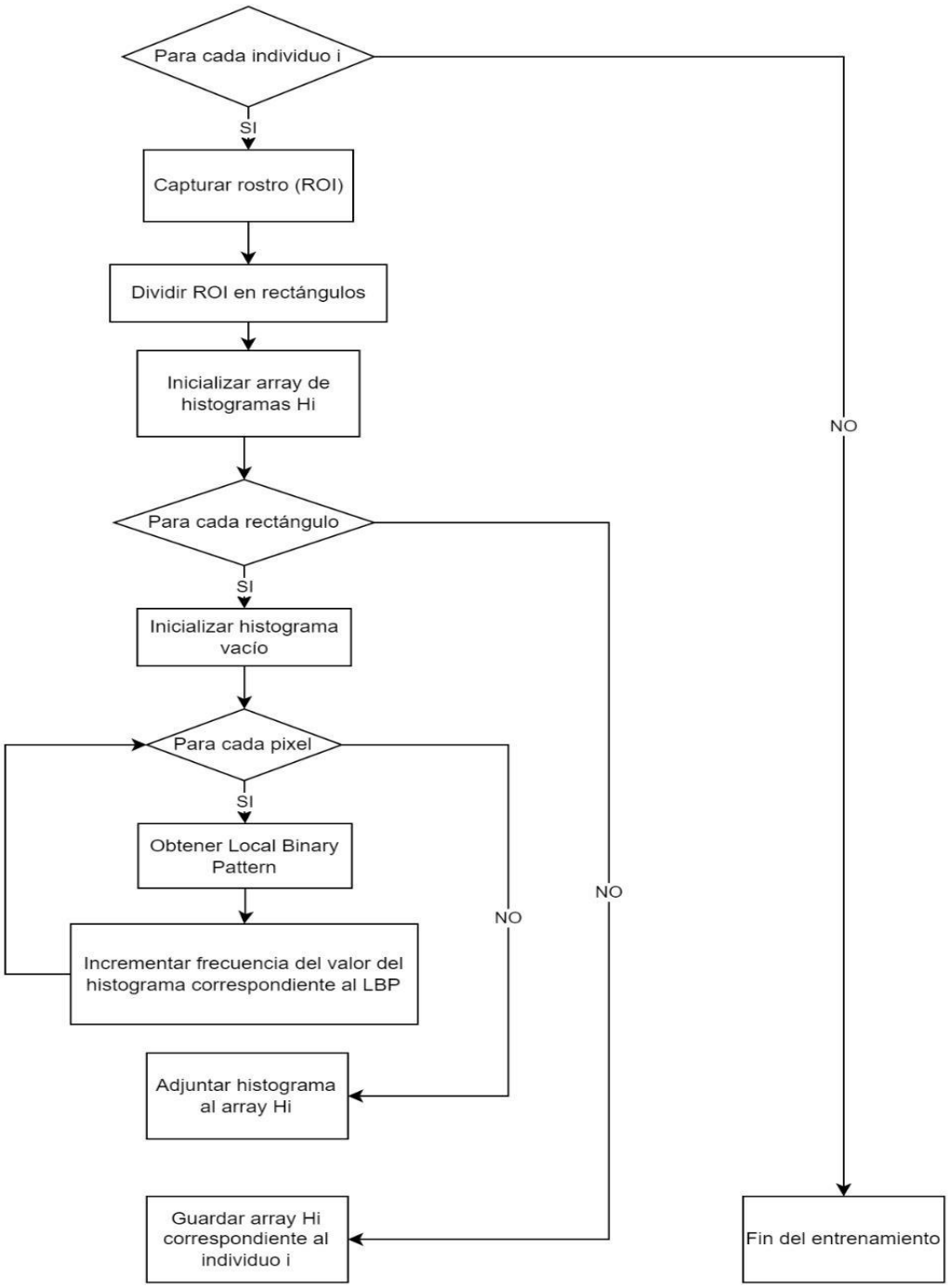
Este esfuerzo se centrará en explorar y aplicar algoritmos avanzados de procesamiento digital que incluyen, pero no se limitan a, técnicas de filtrado, normalización y mejora de contraste. La meta principal es dotar al sistema de una capacidad mejorada para analizar imágenes faciales de manera más precisa, incluso en condiciones adversas.

La implementación de técnicas de procesamiento digital de imágenes no solo busca incrementar la confiabilidad del sistema de reconocimiento facial, sino también consolidar su utilidad en escenarios del mundo real. Se pretende demostrar cómo estas técnicas pueden ser clave para superar obstáculos inherentes al entorno de captura de imágenes y, por ende, elevar la efectividad del sistema en la identificación de individuos con órdenes de captura. Con ello, se aspira a establecer un estándar más alto en la aplicación de tecnologías de reconocimiento facial en el ámbito de la seguridad y la justicia.

3.3.1 Diagrama de flujo del algoritmo del aprendizaje

Figura 61

Diagrama de flujo del aprendizaje

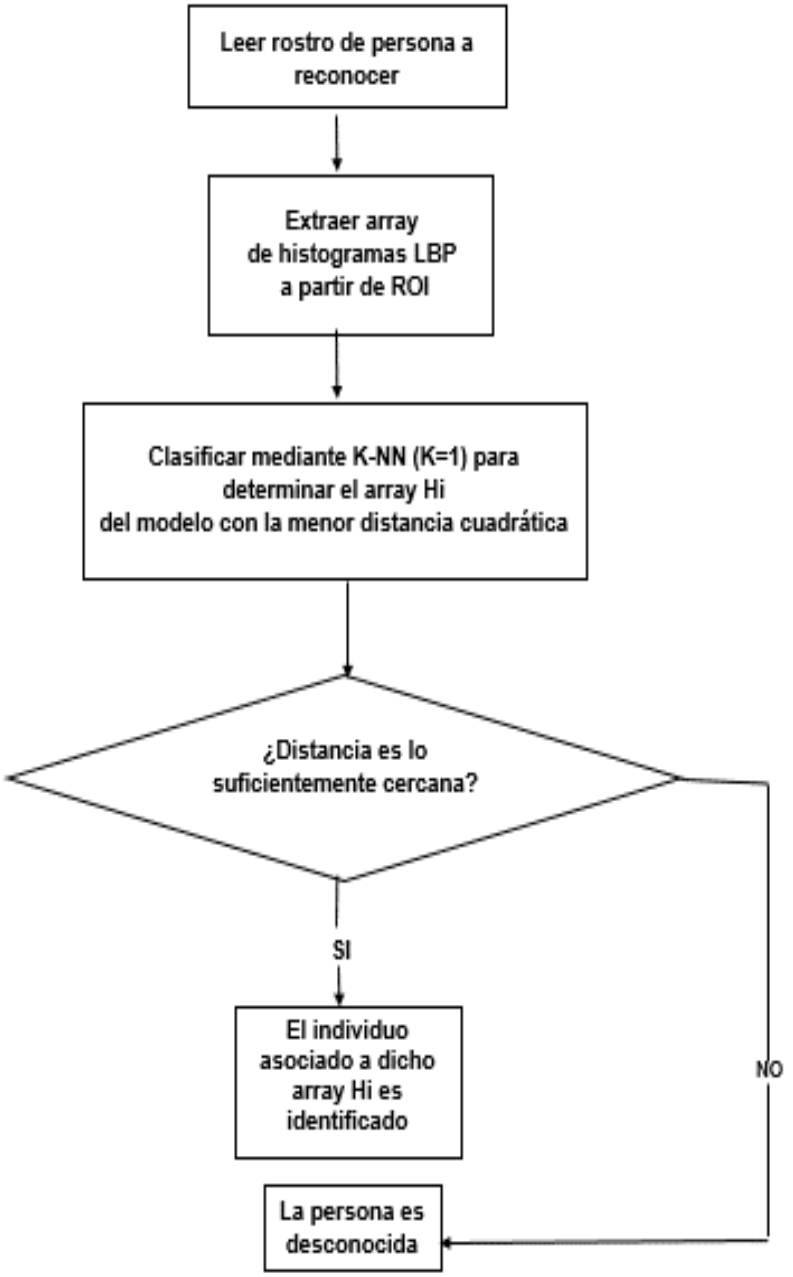


Nota. Elaboración Propia, en la figura 61 se muestra diagrama de flujo del aprendizaje (elaboración propia)

3.3.2 Diagrama de flujo del algoritmo de reconocimiento

Figura 62

Diagrama de flujo del reconocimiento



Nota. Elaboración propia, en la figura 62 se muestra el diagrama de flujo del reconocimiento (elaboración propia)

3.3.3 El algoritmo de LBPH consiste en:

3.3.3.1 Parámetros: el LBPH utiliza 4 parámetros:

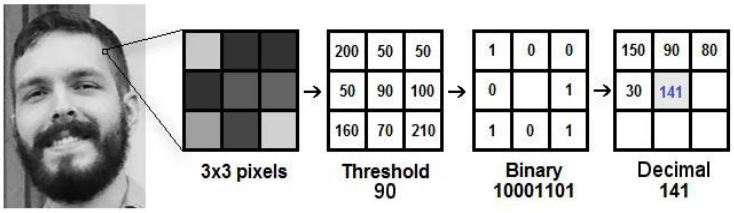
- Radio: en relación con el píxel central, radio es el que se utiliza en la construcción del patrón binario circular local. Normalmente se establece en 1.
- Vecinos: la cantidad de puntos de muestra necesarios para construir el patrón binario circular del vecindario. Tenga en cuenta que el coste computacional aumenta con el número de puntos de muestra que proporcione. Normalmente se fija en 8.
- Cuadricula X: la cantidad de celdas dispuestas horizontalmente. Cuantas más celdas y más fina sea la cuadrícula, mayor será la dimensionalidad del vector de características resultante. Normalmente, se fija en 8.
- Cuadricula Y: la cantidad de celdas en orientación vertical. Cuantas más celdas y más fina sea la cuadrícula, mayor será la dimensionalidad del vector de características resultante. Normalmente, se fija en 8.

3.3.3.2 Entrenar el algoritmo: Primero debemos entrenar el algoritmo. Para ello, debemos utilizar un conjunto de datos que contenga los rostros de las personas que queremos reconocer. Para que el algoritmo identifique una imagen de entrada y proporcione una salida, también tenemos que establecer un ID (que puede ser un número o el nombre de la persona) para cada imagen. El mismo ID debe aparecer en todas las imágenes del mismo individuo. Una vez construido el conjunto de entrenamiento, examinemos las fases de cálculo de LBPH.

3.3.3.3 Aplicar de la operación LBP: La LBPH computa una imagen intermedia, que enfatiza las características del rostro y caracteriza mejor la imagen original. Esta es la etapa computacional inicial del proceso. Basándose en los parámetros radio y vecinos, el método aplica una idea de ventana deslizante para lograrlo.

Figura 63

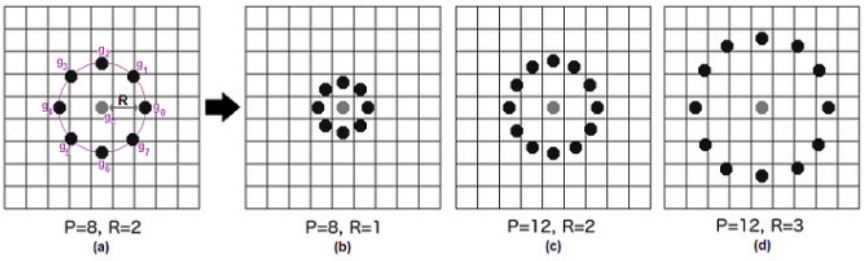
Procedimiento de imagen a histograma



Nota. El rostro del sujeto se muestra en escala de grises en la Figura 63, con una porción de la imagen representada por una caja o matriz de tres por tres píxeles, cada uno de los cuales tiene una intensidad que va de 0 a 255. Del mismo modo, el valor central de la matriz sirve como umbral, definiendo los nuevos ocho valores para el contorno circular adyacente. A cada vecino del umbral central se le asigna un nuevo valor binario: uno si el vecino es igual o mayor que el umbral central, y cero si el vecino es menor que el umbral central. A continuación, la matriz extrae valores binarios omitiendo el valor central. A continuación, concatena cada valor binario de una posición continua con un nuevo valor binario en el sentido de las agujas del reloj, convirtiendo el valor binario en un valor decimal y configurándolo en el valor del centro de la matriz, que es un píxel de la imagen auténtica original. Tenemos una nueva imagen superior a la conclusión del proceso (LBP) que captura mejor las características paramétricas del rostro de la persona de la imagen original (Saltón, 2017).

Figura 64

Variación del parámetro P

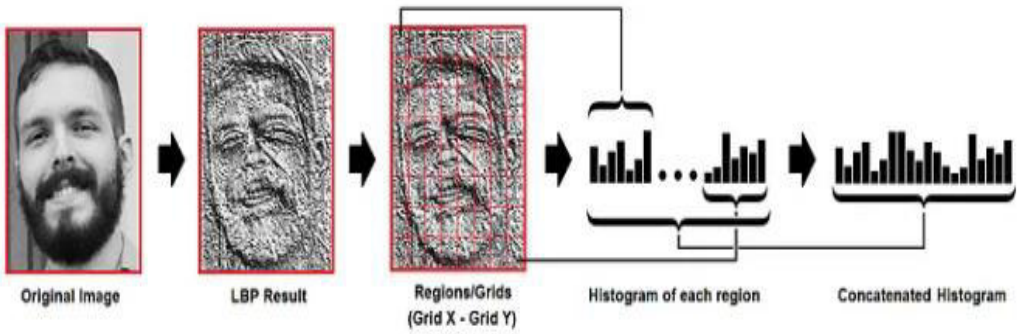


Nota. Interpolación bilineal, que varía el parámetro, como se ve en la Figura 64. En caso de que un punto de datos se encuentre entre píxeles, el valor del nuevo punto de datos P se estima dividiendo por dos los valores de los cuatro píxeles más cercanos (Saltón, 2017).

3.3.4 Extracción de los histogramas: Como se ve en la imagen 63, ahora podemos dividir la imagen en diferentes cuadrículas utilizando los parámetros Cuadrícula X y Cuadrícula Y utilizando la imagen creada en la fase anterior.

Figura 65

Concatenación de histogramas



Nota. En la figura 65 se muestra donde se extrae histograma de cada región (Saltón, 2017).

- Como la imagen es en escala de grises, sólo habrá 256 lugares (0-255) en el histograma de cada cuadrícula que correspondan a las ocurrencias de la intensidad de cada píxel.
- Para hacer un nuevo histograma más grande, debemos concatenar cada histograma. Suponiendo cuadrículas de 8x8, el histograma final contendrá 8x8x256 = 16.384 posiciones. Las cualidades de la imagen original se reflejan en el histograma final.

3.4. Desarrollar pruebas y validar el sistema de alerta remota hacia un dispositivo móvil

El objetivo de este paso es el desarrollo de pruebas y la validación de un sistema de alerta remota hacia dispositivos móviles, integrándolo con el sistema de reconocimiento facial previamente propuesto.

El propósito principal de este sistema de alerta remota es notificar a las autoridades competentes o al personal de seguridad designado cuando se identifica a un individuo con orden de captura mediante el sistema de reconocimiento facial. Esta notificación, transmitida de forma inmediata a través de dispositivos móviles, tiene como fin agilizar la respuesta y facilitar la toma de decisiones en tiempo real.

Las pruebas se enfocarán en evaluar la velocidad y confiabilidad del sistema de alerta remota en condiciones diversas, considerando posibles escenarios operativos y variaciones en la conectividad de red. La validación del sistema abordará aspectos clave como la efectividad de la notificación, la interfaz de

usuario del dispositivo móvil y la capacidad de gestionar alertas concurrentes de manera eficiente.

Este desarrollo busca no solo mejorar la capacidad de respuesta ante situaciones críticas, sino también demostrar la viabilidad y utilidad práctica de la integración de sistemas de alerta remota con tecnologías de reconocimiento facial. La implementación exitosa de este componente del sistema contribuirá significativamente a la efectividad general de la plataforma, consolidando su posición como una herramienta integral y confiable en el ámbito de la seguridad y la aplicación de la ley.

Figura 66

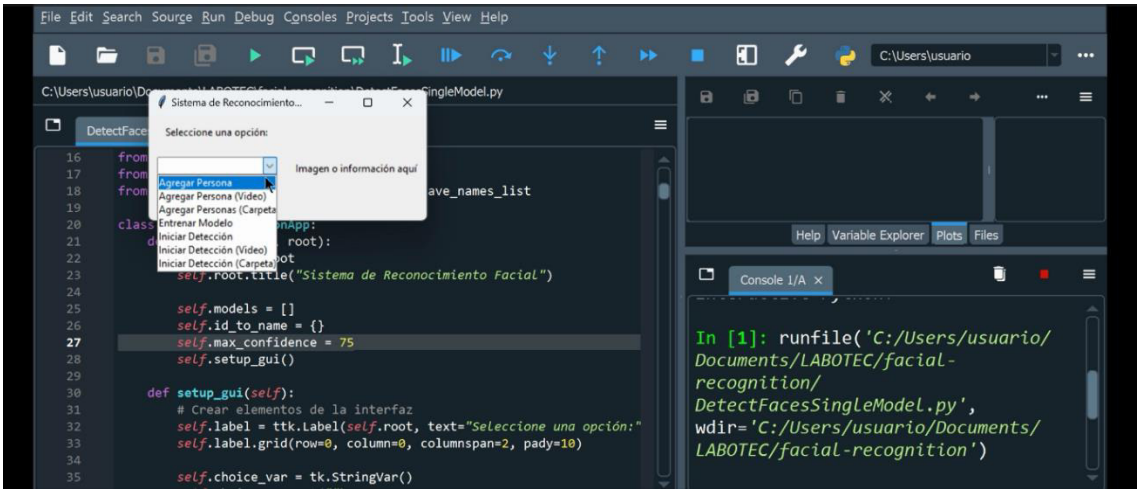
Figuras banco de rostros

Nota. En la figura 66 se muestra data de videos (Elaboración propia)

Data de entrenamiento de 15 personas con una duración de un segundo cada video, extraído de una base de datos Rufts Face Database que contiene alrededor de 112 personas, extrayendo quince de ellos, dividiendo el video en dos, entrenamiento y extracción de rostro, entrenamiento del modelo para reconocer a personas, luego se captura videos para hacer la validación, siendo videos de dos segundos en los que va a comparar que la persona efectivamente sea reconocida

Figura 67

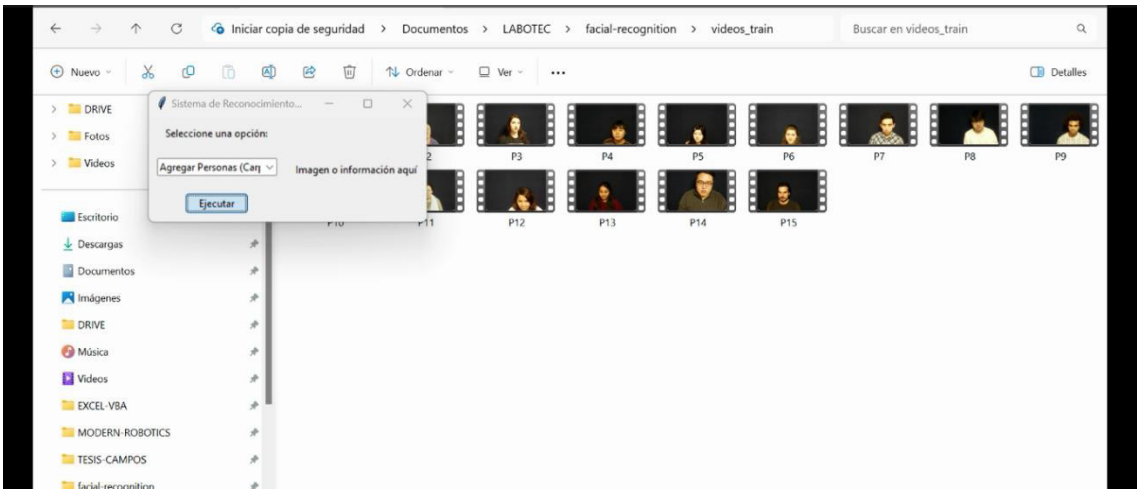
Aplicación de agregar personas



Nota. En la figura 67. (elaboración propia), se muestra al abrir la variable Self donde se habla de la aplicación, en este caso hay varias opciones, al seleccionar se agrega la imagen de rostros de personas (carpeta), supuesto delincuente con orden de captura. Busca la carpeta que contiene en la base de dato y cada carpeta video lo abre y extrae el rectángulo que extrae la persona, entonces se ejecuta y extra el rectángulo que extrae el rostro.

Figura 68

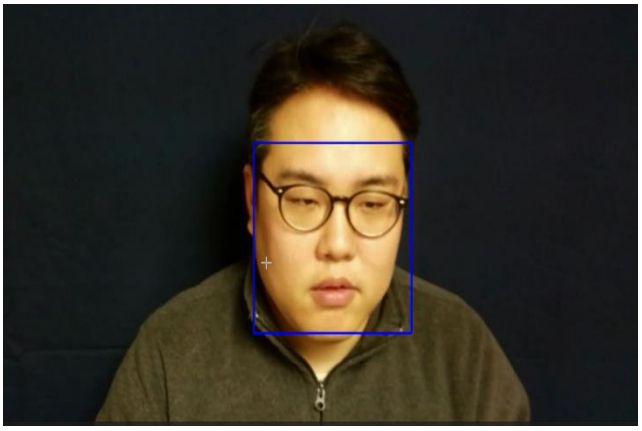
Aplicación a ejecutar carpetas de personas



Nota. En la figura 68. Muestra la búsqueda de carpeta, donde se encuentra la base de dato, extrayendo cuadro que contiene el rostro del supuesto delincuente con orden de captura, cada video se detecta el rostro y se extrae (Elaboración propia).

Figura 69

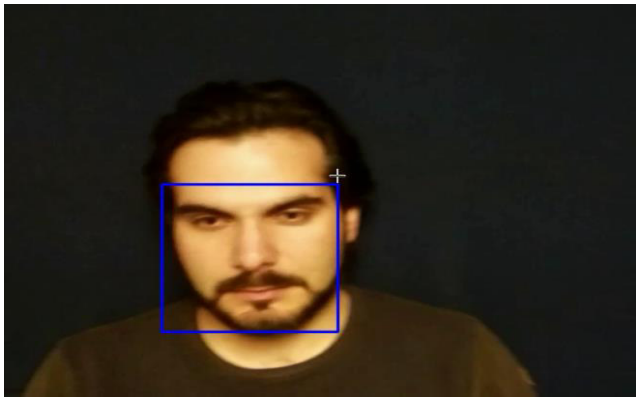
Extracción de imagen 14



*Nota***.** Figura 69. Muestra la extracción de persona (p-14). Toma de (Elaboración propia)

Figura 70

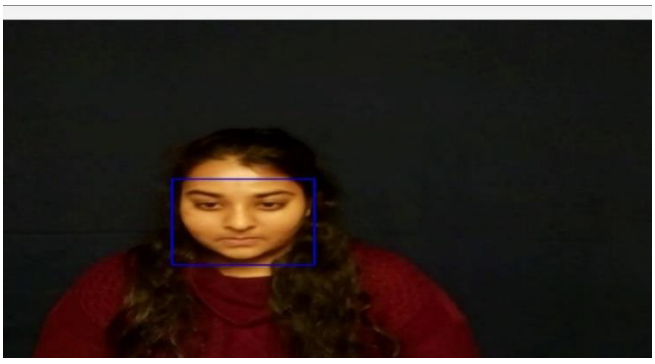
Extracción de imagen 15



*Nota***.** Figura 70. Muestra la extracción de persona (p-15). Toma de (Elaboración propia)

Figura 71

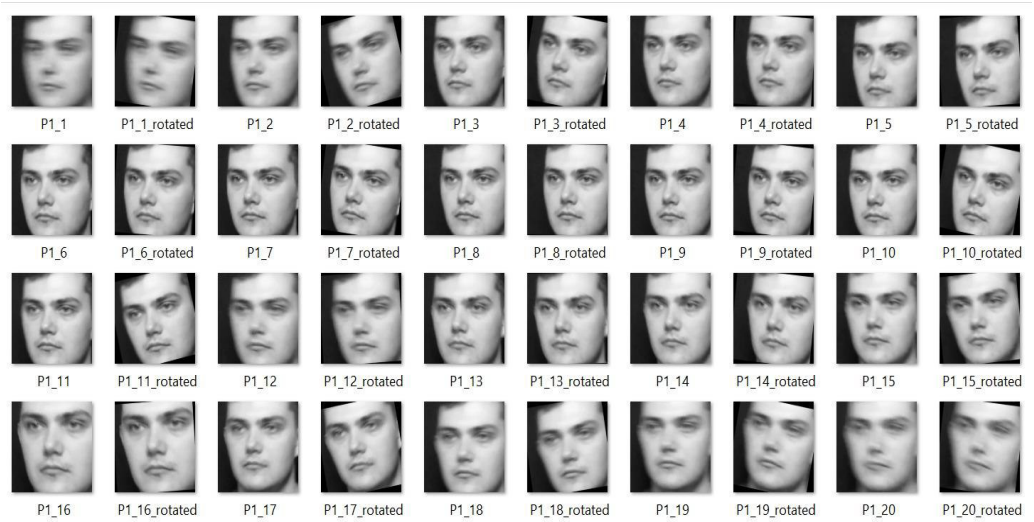
Extracción de imagen 13



Nota. Figura 71. Muestra la extracción de persona (p-13). Toma de (Elaboración propia)

Figura 72

Muestra de imágenes por persona



Nota. Figura 72. Muestra de 40 imágenes por persona. Tomado de (Elaboración propia)

Figura 73

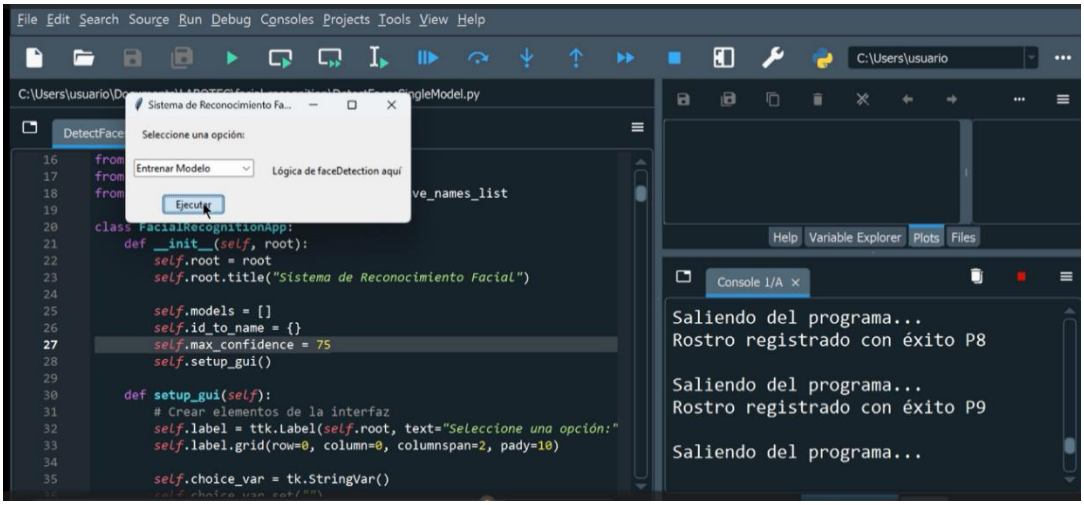
Muestra de imágenes por persona

Nota. Figura 73. Muestra de 40 imágenes por persona. Tomado de *(*Elaboración propia*)*

Por cada persona se tiene una carpeta donde contiene los rostros, así como rostros inclinadas o rotadas para poder reconocer rostros con más posiciones. Para luego realizar el entrenamiento, con el modelo.

Figura 74

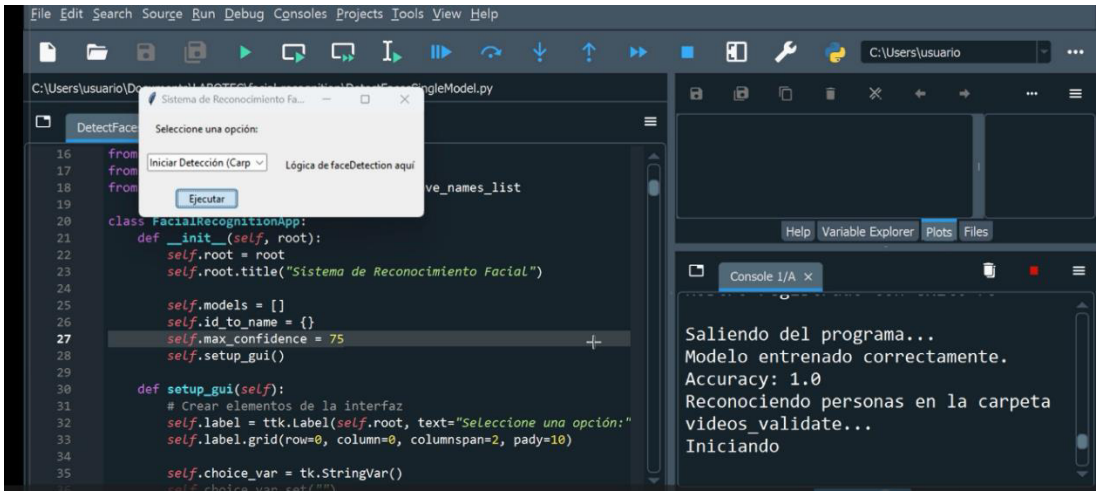
Para realizar entrenamiento



Nota. Figura 74. Muestra entrenamiento del modelo pre-entrenado Tomado de (Elaboración propia)

Figura 75

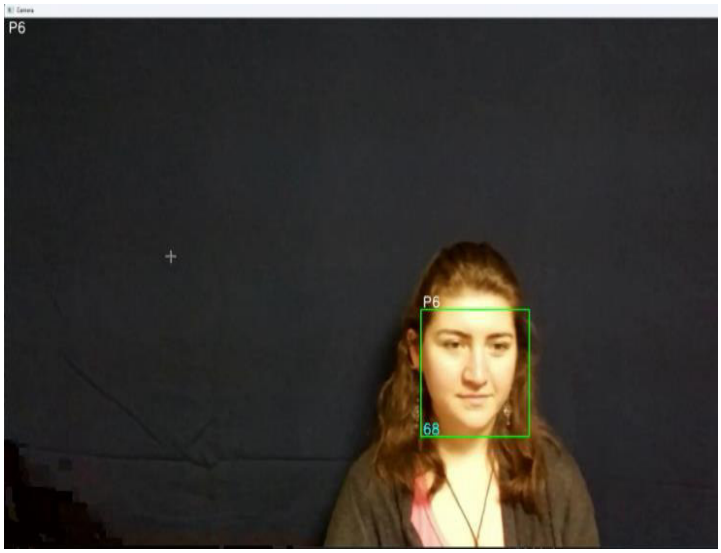
Iniciando la detección(captura)



Nota. Figura 75. Muestra la detección de rostro facial mediante la captura Tomada de (Elaboración propia)

Figura 76

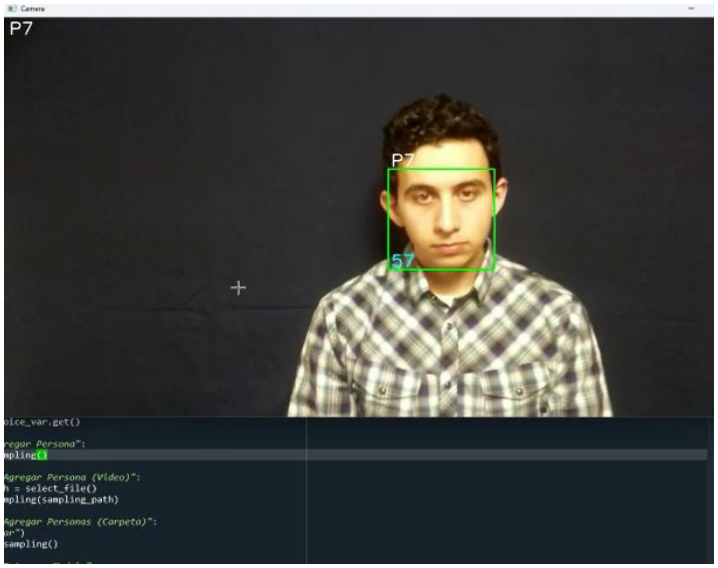
Reconocimiento facial a persona 6



Nota. Figura 76. Muestra el reconocimiento facial detectado de la persona (P 6), tomada de (Elaboración propia)

Figura 77

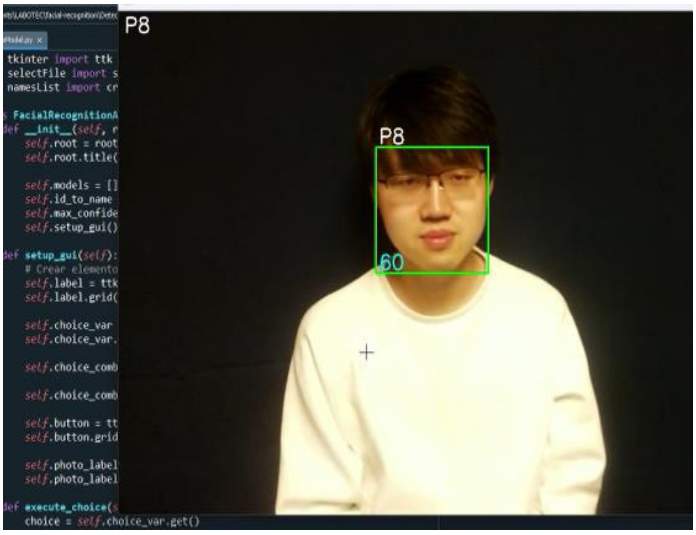
Reconocimiento facial a persona 7



Nota. Figura 77. Muestra el reconocimiento facial detectado de la persona (P 7), tomada de (Elaboración propia)

Figura 78

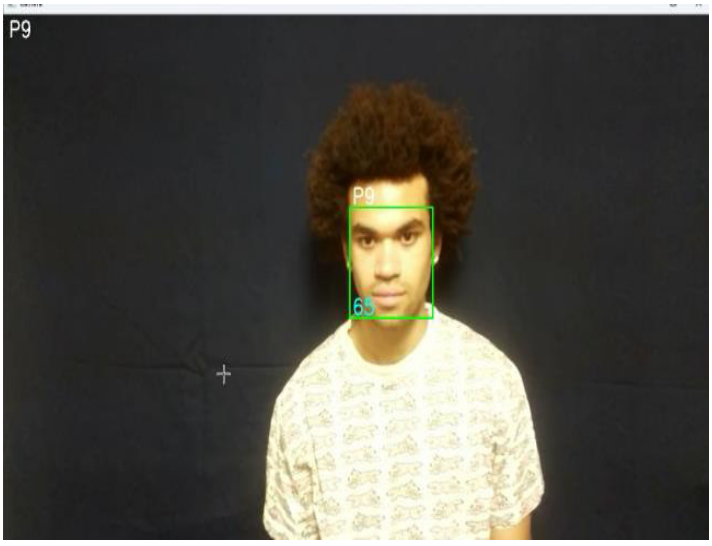
Reconocimiento facial a persona 8



Nota. Figura 78. Muestra el reconocimiento facial detectado de la persona (P 8), tomada de *(*Elaboración propia*)*

Figura 79

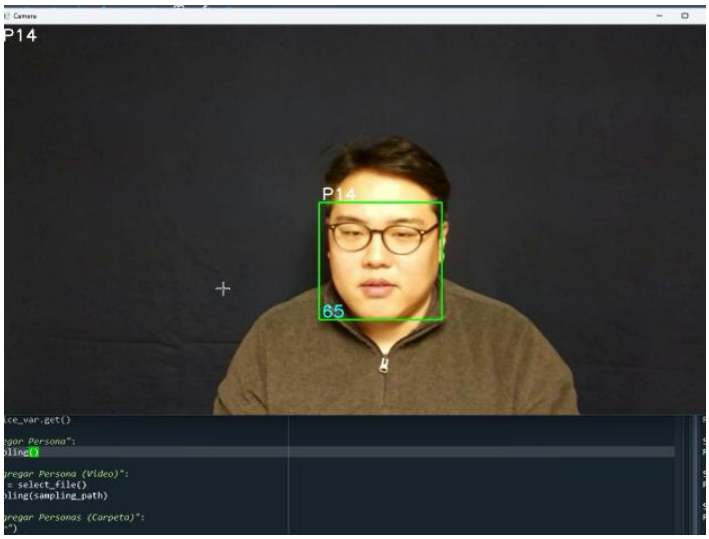
Reconocimiento facial a persona 9



Nota. Figura 79. Muestra el reconocimiento facial detectado de la persona (P 9), tomada de (Elaboración propia)

Figura 80

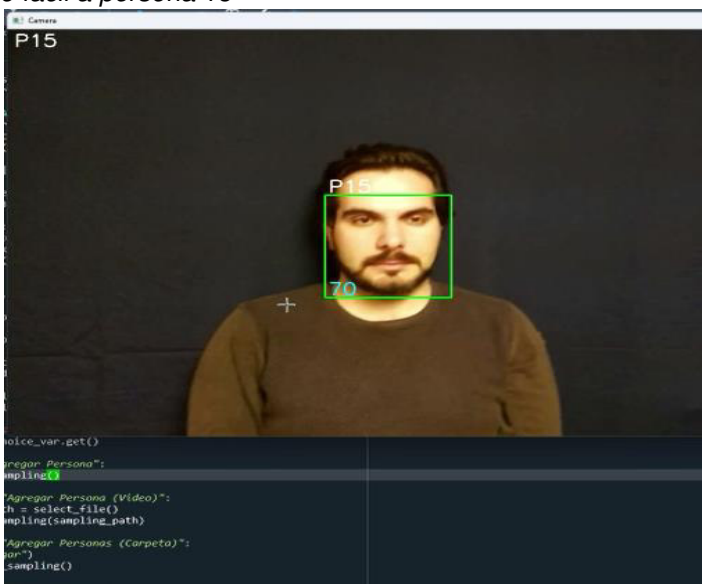
Reconocimiento facial a persona 14



Nota. Figura 80. Muestra el reconocimiento facial detectado de la persona (P 14), toma de (Elaboración propia)

Figura 81

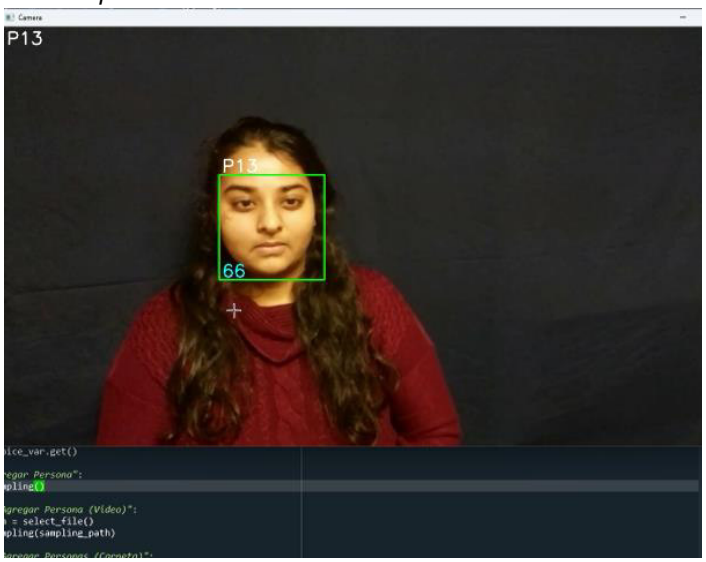
Reconocimiento fácil a persona 15



Nota. Figura 81. Muestra el reconocimiento facial detectado de la persona (P 15), tomado de (Elaboración propia)

Figura 82

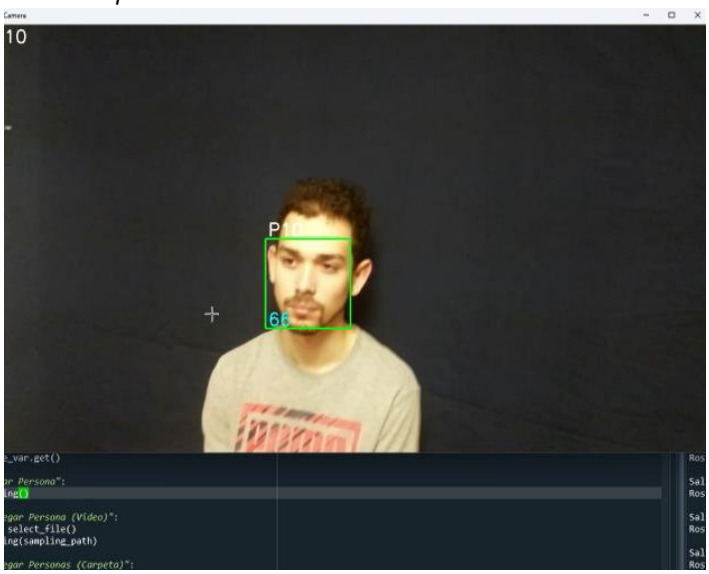
Reconocimiento fácil a persona 13



Nota. Figura 82. Muestra el reconocimiento facial detectado de la persona (P 13), tomada de (Elaboración propia)

Figura 83

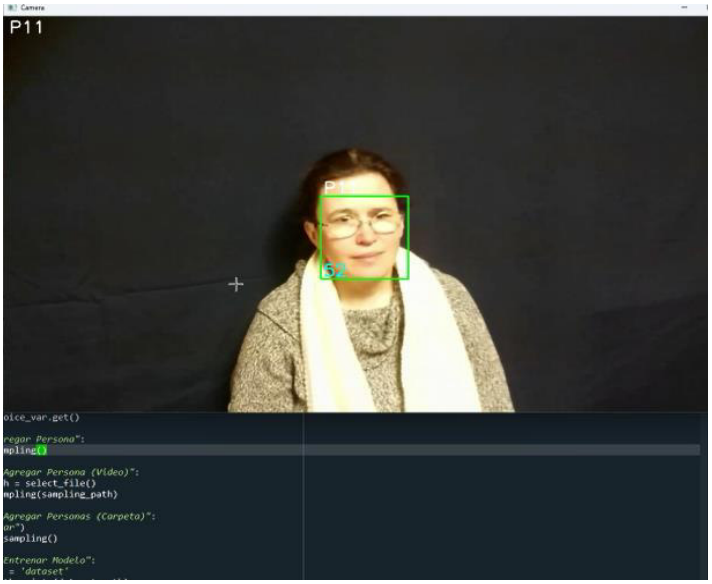
Reconocimiento facial a persona 10



Nota. Figura 83. Muestra el reconocimiento facial detectado de la persona (P 10), tomada de (Elaboración propia)

Figura 84

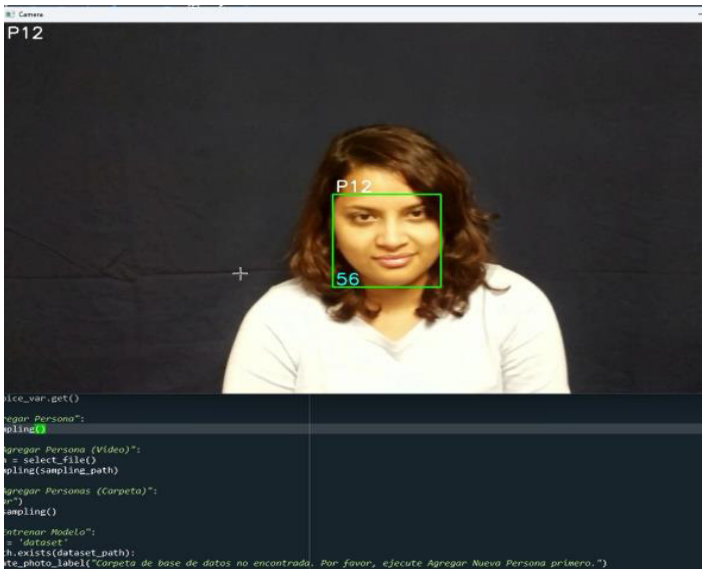
Reconocimiento facial a persona 11



Nota. Figura 84. Muestra el reconocimiento facial detectado de la persona (P 11), tomada de (Elaboración propia)

Figura 85

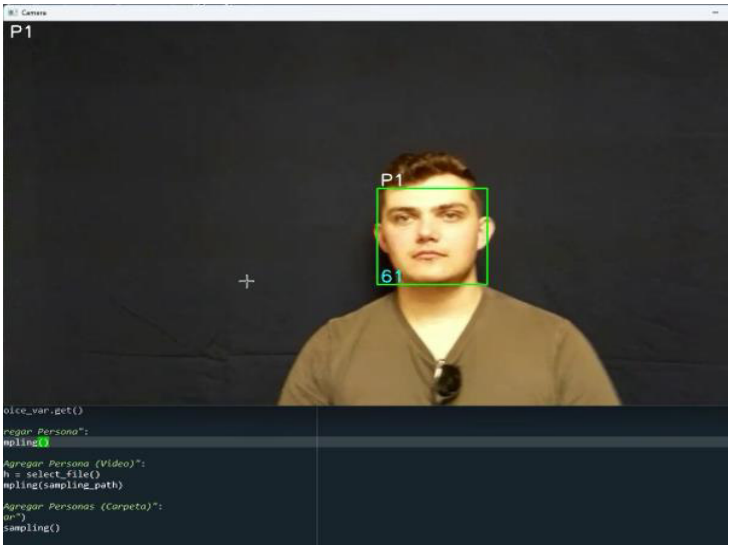
Reconocimiento facial a persona 12



Nota. Figura 85. Muestra el reconocimiento facial detectado de la persona (P 12), tomada de (Elaboración propia)

Figura 86

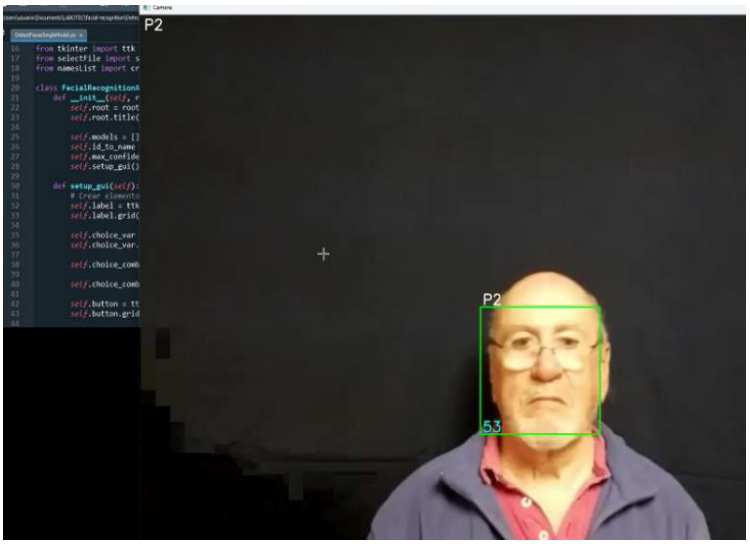
Reconocimiento facial a persona 1



Nota. Figura 86. Muestra el reconocimiento facial detectado de la persona (P 1), tomada de (Elaboración propia)

Figura 87

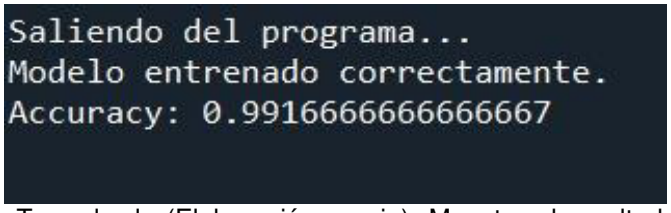
Reconocimiento facial a persona 2



Nota. Figura 87. Muestra el reconocimiento facial detectado de la persona (P 2), tomada de (Elaboración Propia)

Figura 88

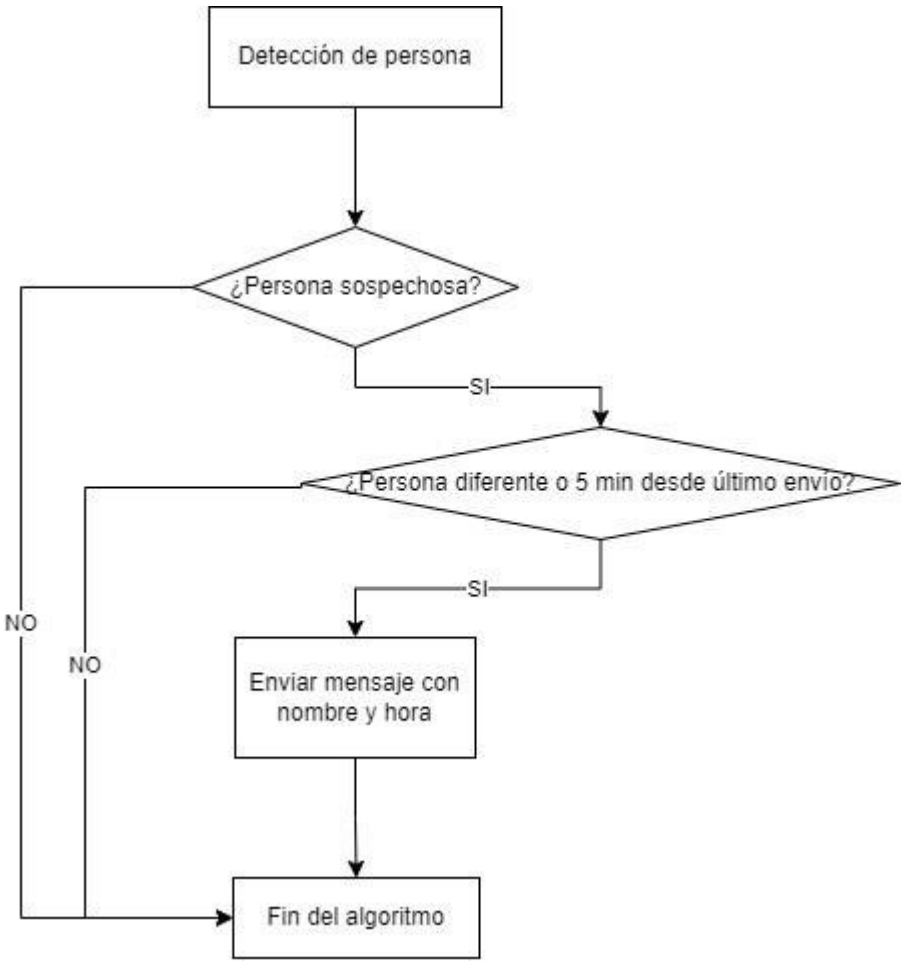
Entrenamiento de reconocimiento



Nota. Figura 88. Tomada de (Elaboración propia). Muestra el resultado de entrenamiento de reconocimiento con exactitud de 99% del porcentaje de eficacia del reconocimiento facial respecto a la detección de rostro de supuestos delincuentes con orden de captura.

Figura 89

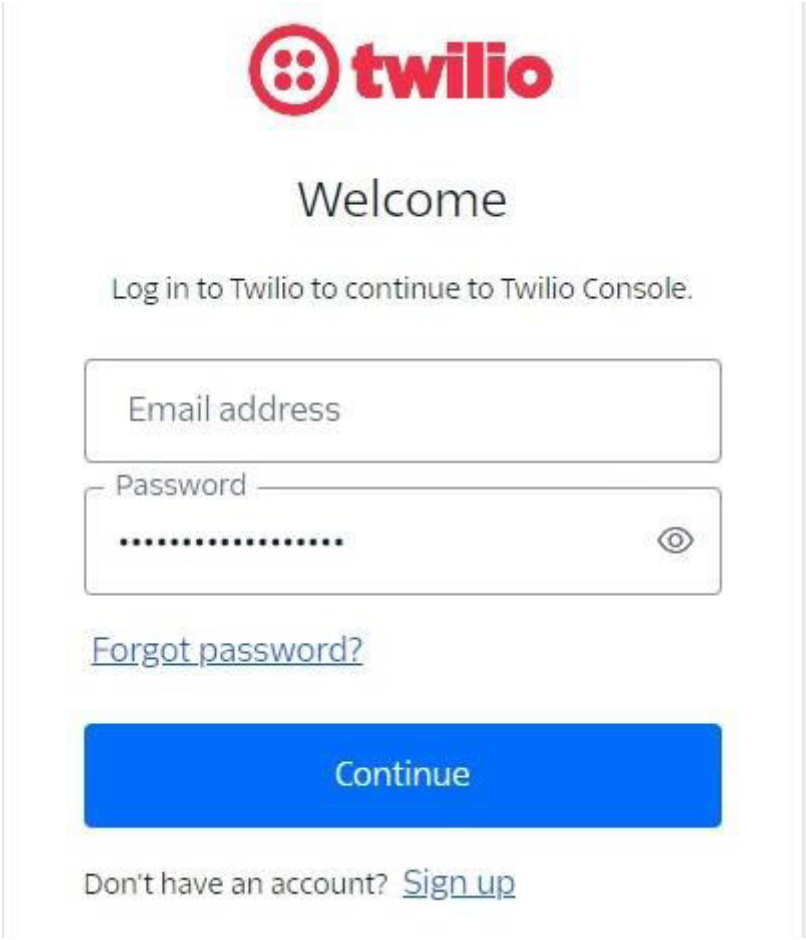
Diagrama de flujo de envió de mensaje



Nota. Figura 89. Muestra el diagrama de flujo del envió de mensajería (Elaboración propia)

Figura 90

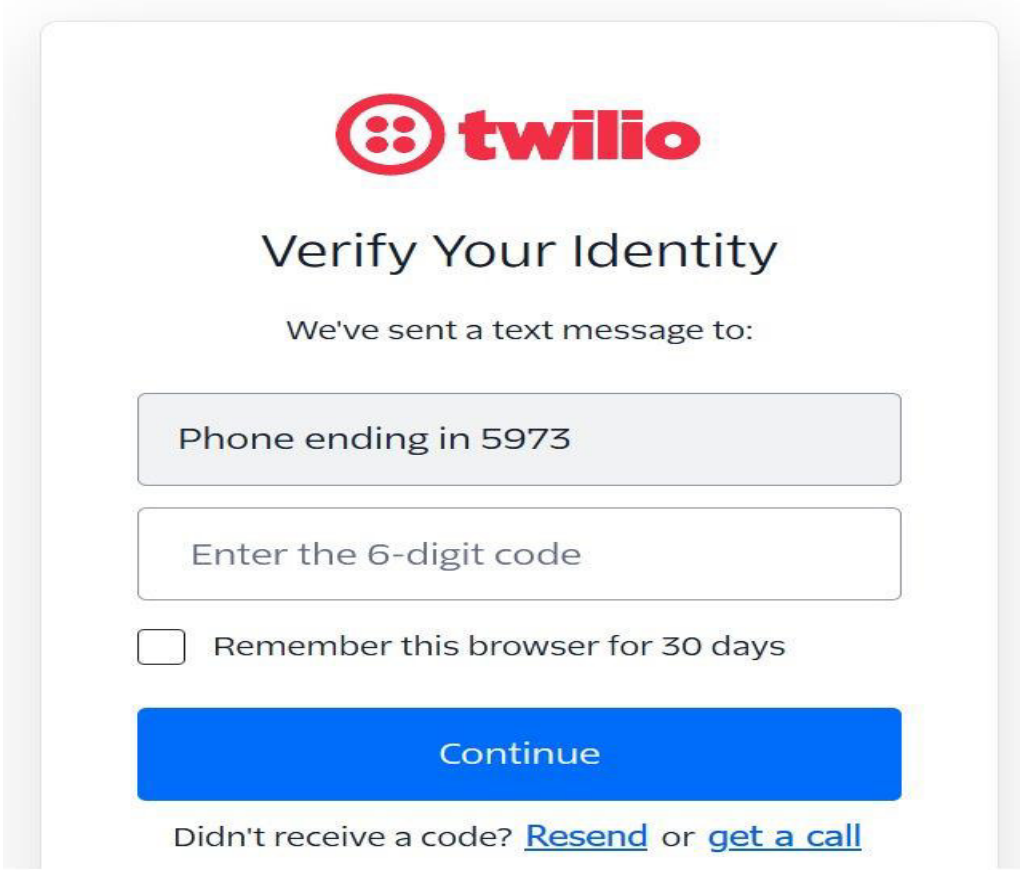
Mensajería móvil



Nota. En la figura 90. Muestra la plataforma de mensaje alerta móvil, el cual será logeado con nuestra cuenta de Gmail para tener acceso a la plataforma twilio de envió mensaje (Elaboración propia).

Figura 91

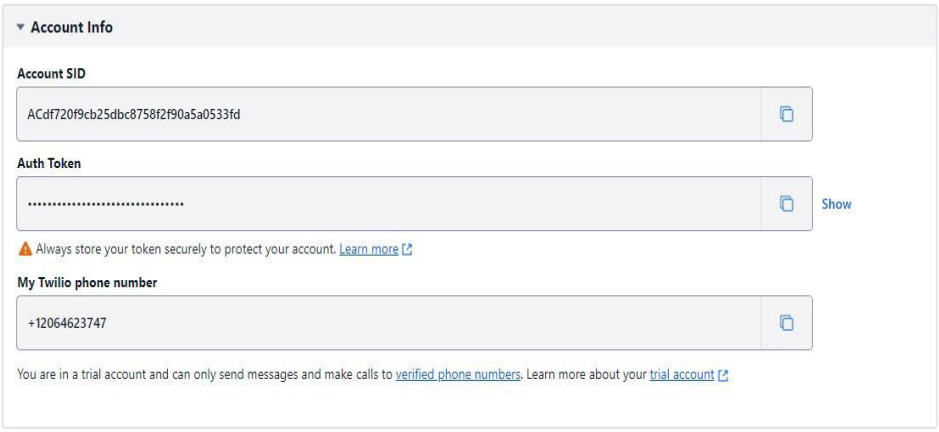
Código de verificación para twilio



Nota. En la figura 91. Muestra el vínculo de teléfono móvil y la plataforma twilio

Figura 92

Plataforma de token e índice



Nota. En la figura 92. Muestra ID de cuenta, mi token de autenticación y el número de teléfono de twilio.

Figura 93

Visualización de código Auth token y account SID

Nota. En la figura 93. Muestra códigos de parámetros de comunicación de la plataforma vinculado al teléfono móvil

Figura 94

Algoritmo de mensajería de texto

```python
from twilio.rest import Client

# Credenciales
account_sid = 'ACdf720f9cb25dbc8758f2f90a5a0533fd'
auth_token = 'cc209e0831764195f050e4d086eea836'
twilioNumber = '+12064623747'

# Clase cliente
client = Client(account_sid, auth_token)
# Función que realiza el envío del mensaje
def sendSMS(mssg, to='973975973'):
    message = client.messages.create(
        from_= twilioNumber,
        body=mssg,
        to=f"+51{to}"
        )
    print(message.sid)

if __name__ == "__main__":
    sendSMS("Mensaje desde python!")
```

Nota. En la figura 94. Se muestra los parámetros de twilio vinculados al algoritmo de reconocimiento facial mediante la detección de rostros y envió de mensajería con clave token y numero de mensaje twilio.

Figura 95

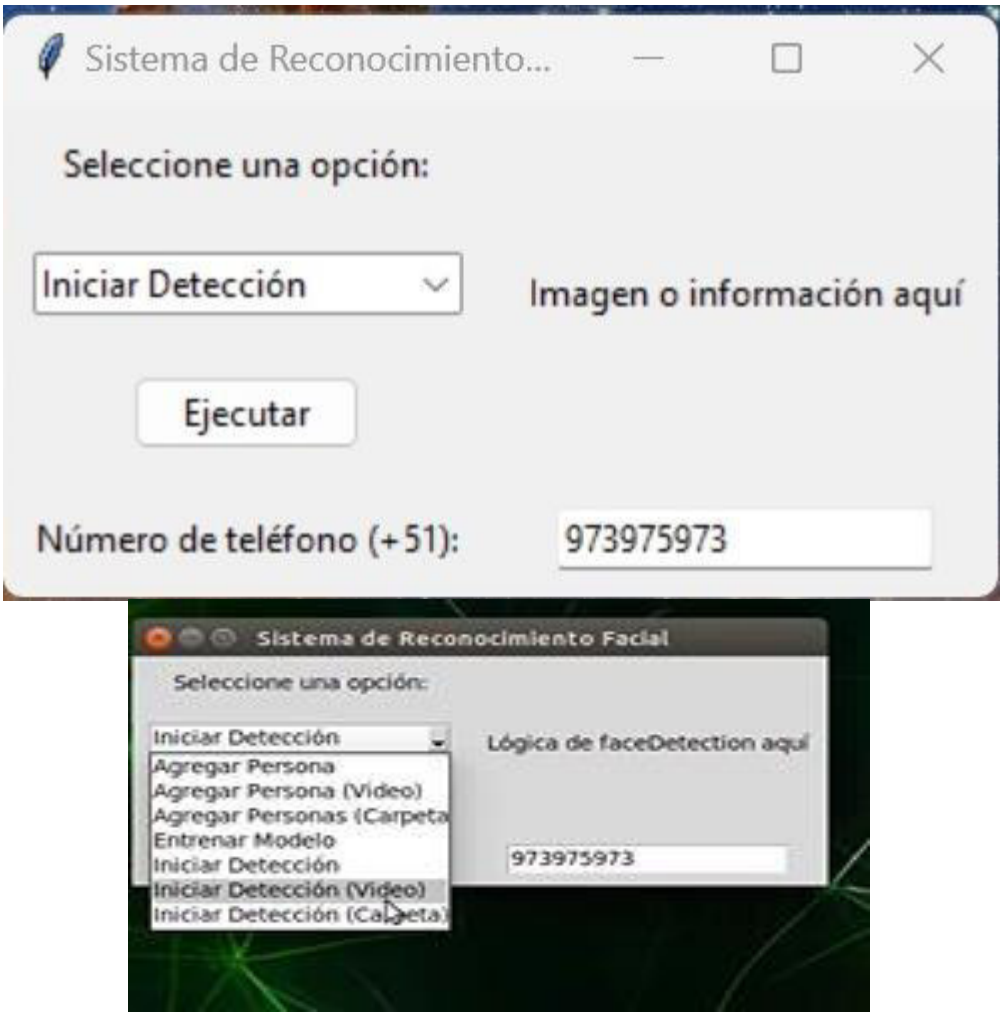
Detección y alerta de envió de mensaje

```
# Si la persona fue reconocida y es una nueva persona o han pasado 5 minutos
if name != lastName or time.time() - lastTime > 300:
    timeAndDate = datetime.datetime.fromtimestamp(time.time()).strftime('%Y-%m-%d %H:%M:%S')
    # Enviar mensaje: nombre ha sido reconocido fecha y hora
    sendSMS(f"{name} ha sido reconocido/a a las {timeAndDate}")
    lastName = name
    lastTime = time.time()
```

Nota. En la figura 95. Muestra el desempeño del algoritmo en la detección de rostro de supuestos delincuentes con orden de captura mediante envió de mensajería cada cinco minuto, nombre ha sido reconocido fecha, hora y año

Figura 96

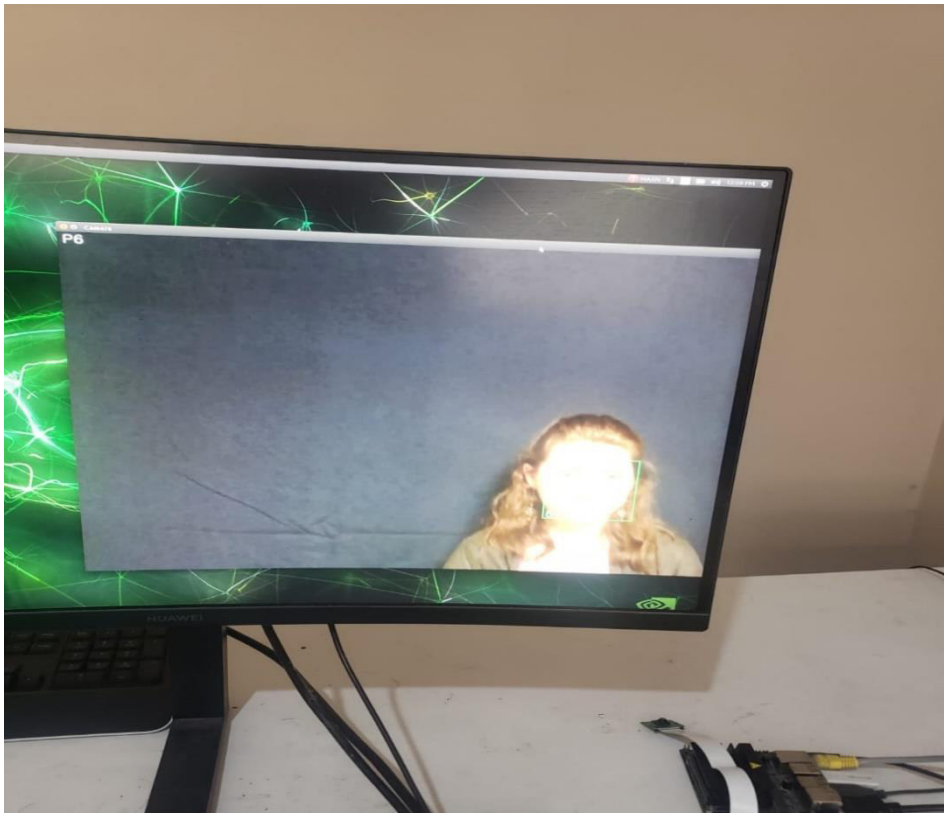
Módulo de captura de imagen facial



Nota. En la figura 96. Se muestra el proceso de captura de reconocimiento facial de supuesto delincuente con orden de captura mediante el número de alerta twilio

Figura 97

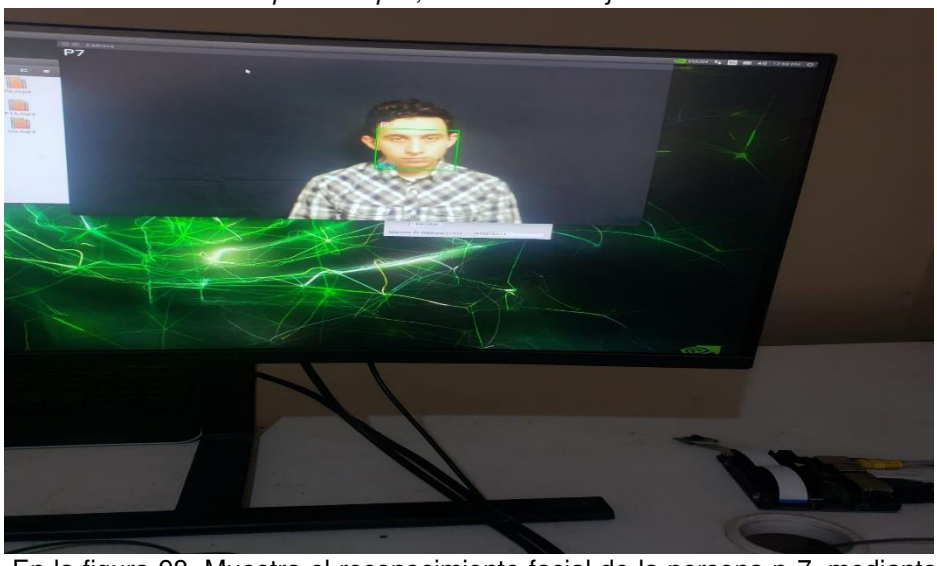
Reconocimiento facial de la persona p-6, mediante la tarjeta Jetson Nano



Nota. En la figura 97. Muestra el reconocimiento facial de persona p-6, mediante un monitor externo y la tarjeta electrónica Jetson nano para la identificación de rostro de (Elaboración propio)

Figura 98

Reconocimiento facial de la persona p-7, mediante la tarjeta Jetson Nano



Nota. En la figura 98. Muestra el reconocimiento facial de la persona p-7, mediante un monitor externo y la tarjeta electrónica Jetson nano para la identificación de rostro de (Elaboración propia).

Figura 99

Reconocimiento facial de la persona p-8, mediante la tarjeta jetson nano

Nota. En la figura 99. Muestra el reconocimiento facial de la persona p-8, mediante un monitor externo y la tarjeta electrónica Jetson nano para la identificación de rostro de (Elaboración propia).

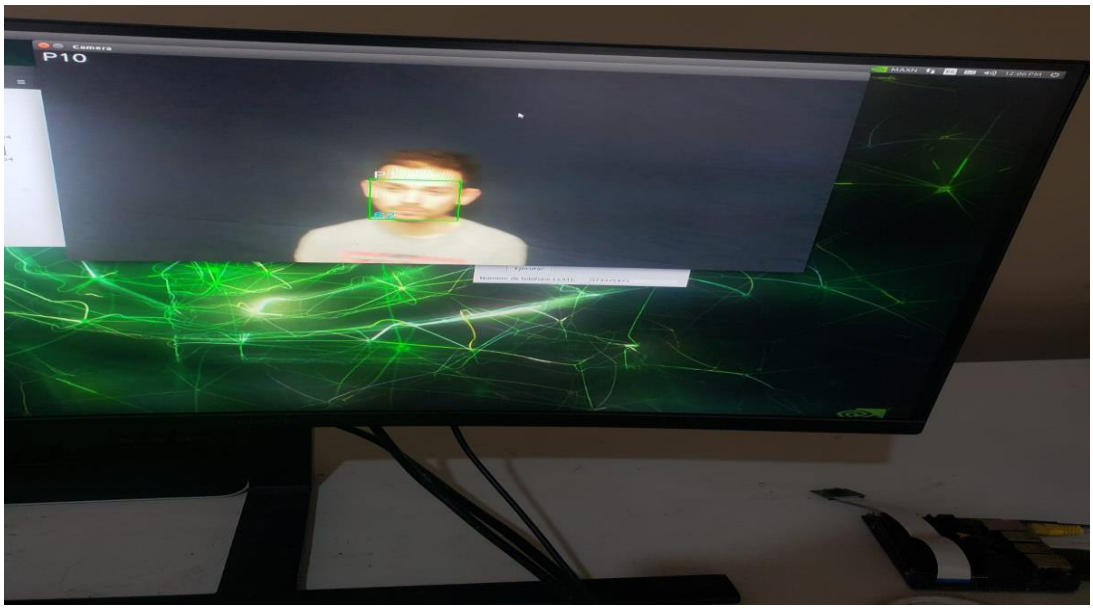
Figura 100

Reconocimiento facial de la persona p-9, mediante la tarjeta jetson nano

Nota. En la figura 100. Muestra el reconocimiento facial de la persona p-9, mediante un monitor externo y la tarjeta electrónica Jetson nano para la identificación de rostro de (Elaboración propia).

Figura 101

Reconocimiento facial de la persona p-10, mediante la tarjeta jetson nano



Nota. En la figura 101. Muestra el reconocimiento facial de la persona p-10, mediante un monitor externo y la tarjeta electrónica Jetson nano para la identificación de rostro de (Elaboración propia).

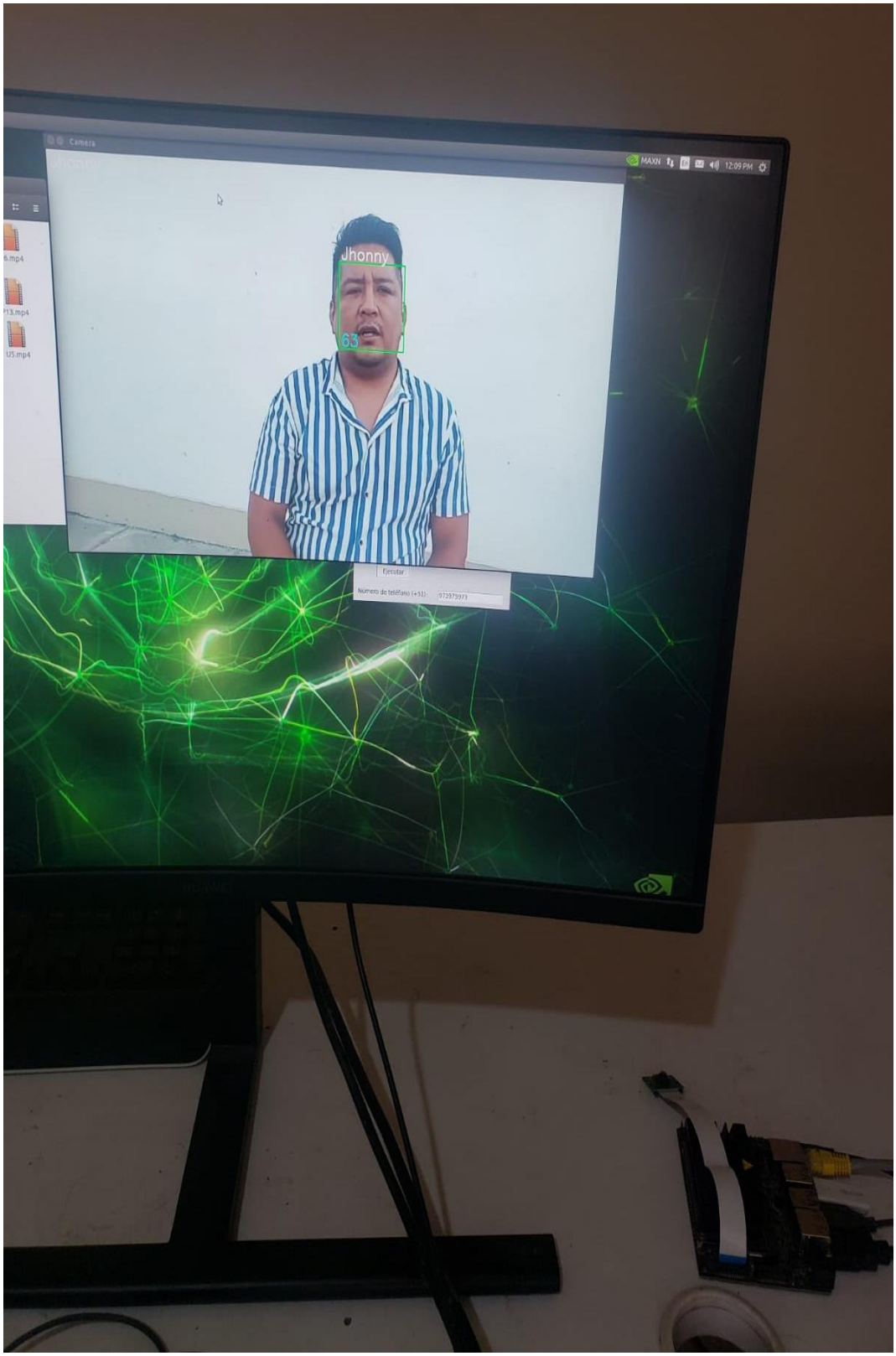
Figura 102

Reconocimiento facial de la persona p-14, mediante la tarjeta jetson nano

Nota. En la figura 102. Muestra el reconocimiento facial de la persona p-14, mediante un monitor externo y la tarjeta electrónica Jetson nano para la identificación de rostro de (Elaboración propia).

Figura 103

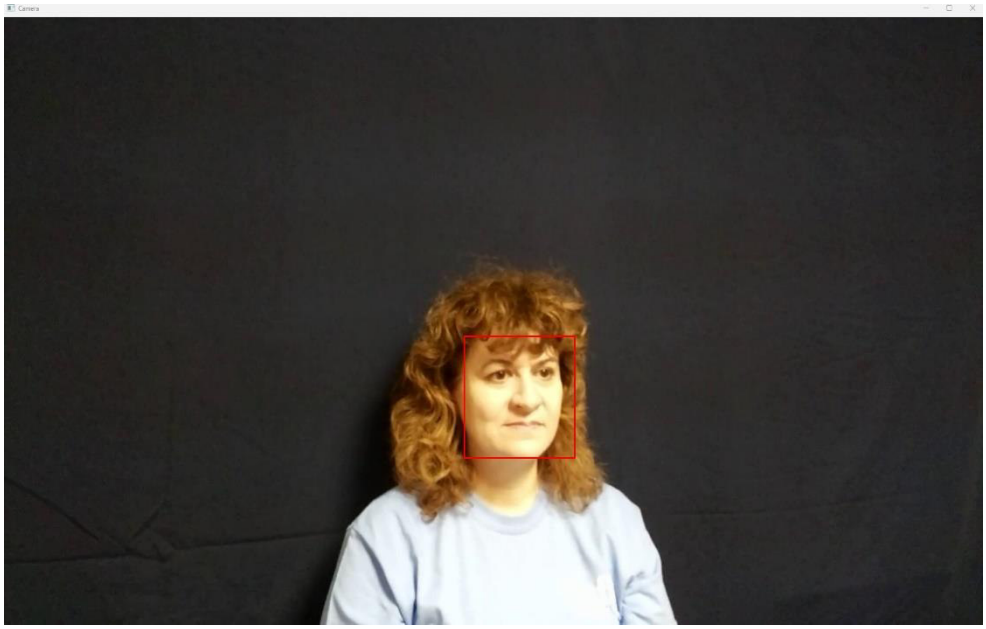
Reconocimiento facial de la persona jhonny, mediante la tarjeta jetson nano



Nota. En la figura 103. Muestra el reconocimiento facial de la persona jhonny, mediante un monitor externo y la tarjeta electrónica Jetson nano para la identificación de rostro de (Elaboración propia)

Figura 104

Persona desconocida, no detectada



Nota. En la figura 104. Muestra a una persona no detecta por el sistema de reconocimiento facial de (Elaboración propia)

Figura 105

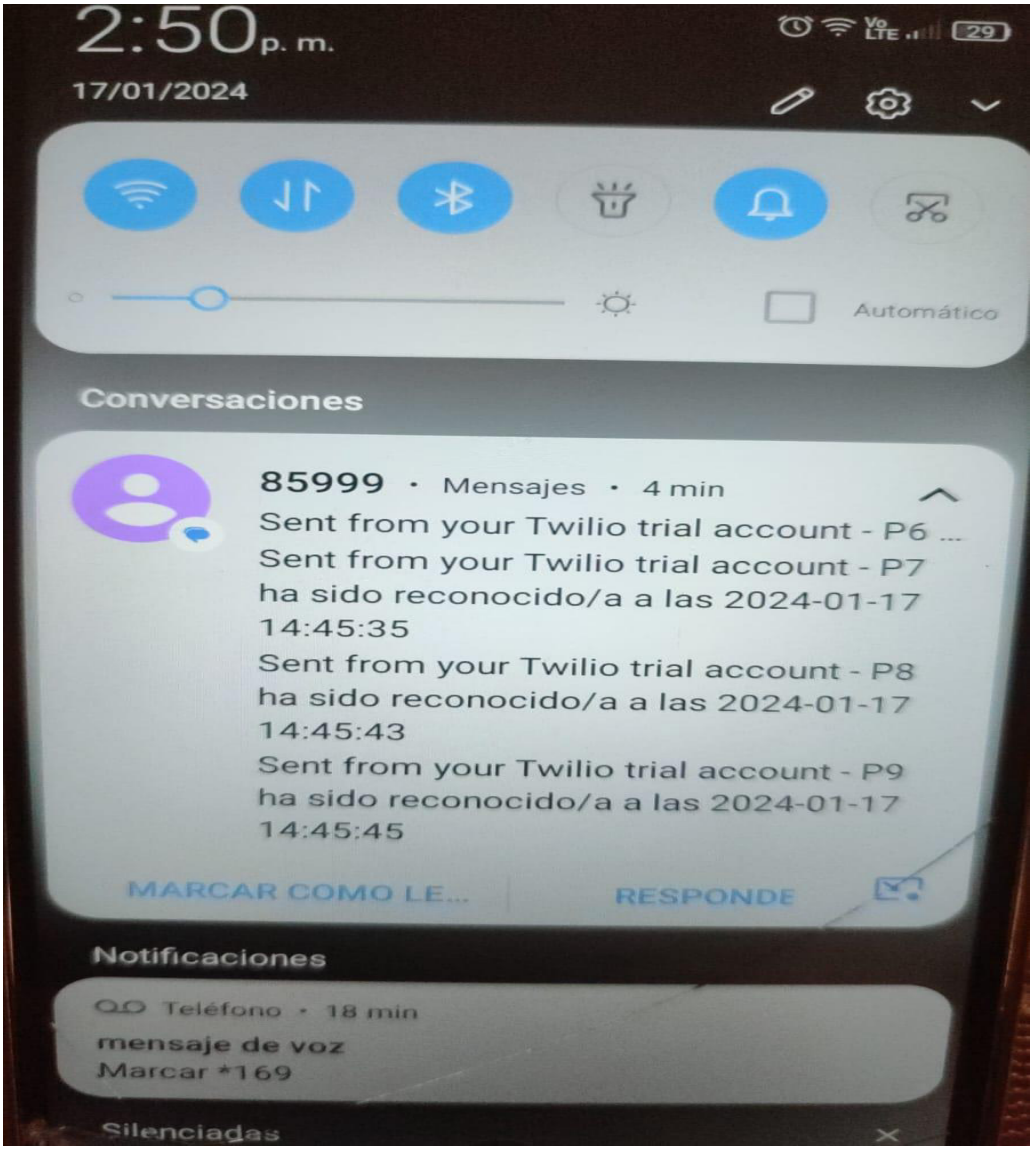
Desarrollo Pruebas y validar el sistema de reconocimiento facial

Identidad esperada	Identidad reconocida	Acierto	Error
P1	P1	✓	
P2	P2	✓	
P3	P3	✓	
P4	P4	✓	
P5	P5	✓	
P6	P6	✓	
P7	P7	✓	
P8	P8	✓	
P9	P9	✓	
P10	P10	✓	
P11	P11	✓	
P12	P12	✓	
P13	P13	✓	
P14	P14	✓	
P15	P15	✓	
Jhonny	Desconocido		✗
Desconocido 1	Desconocido	✓	
Desconocido 2	Desconocido	✓	
Desconocido 3	Desconocido	✓	
Desconocido 4	Desconocido	✓	
Desconocido 5	Desconocido	✓	

Nota. En la figura 105. Muestra el desempeño del modelo del sistema de reconocimiento facial de (Elaboración propia)

Figura 106

Validación de envió de mensaje a dispositivo móvil



Nota. En la figura 106. Se muestra la recepción de envió desde la cuenta prueba de twilio la detección de las personas (p-6), (p-7), (p-8), (p-9) son reconocidos en el año fecha y hora a tiempo real mediante dispositivo móvil.

En primer lugar, validamos el desempeño, el modelo de reconocimiento. Para ello, empleamos un data.set de rostros etiquetadas de 15 personas. Este data.set lo obtenemos de la base datos Rulfs face data base. Para lo cual, los rostros se capturen a partir de los videos de la base de datos del data.set es dividido en data de entrenamiento y data de validación.

Se ejecutan el aprendizaje sobre la data de entrenamiento y se obtiene un modelo LBPH.

Se realiza el reconocimiento de los videos de validación correspondientes a las 15 personas. Se determina el desempeño del desempeño del reconocimiento sobre esta data de validación.

SISTEMA DE ALERTA REMOTA

En primer lugar, se ha reconocido la identidad de la persona en caso que haya sido una persona sospechosa se envía un mensaje de texto con el nombre de la persona y la hora que ha sido detectada.

Para evitar el video excesivo de mensajes, solo se admite el mensaje de texto cada 5 minutos o si, se reconoce a una persona distinta.

Para realizar él envió de mensajes, empleamos el servicio de mensajería twilio, que nos proporciona un número de teléfono virtual. Empleamos nuestra credencial de nuestra cuenta en conjunto con la librería twilio de Python, por su efectividad de servicio.

CAPÍTULO IV: ANÁLISIS DE COSTOS Y BENEFICIOS

4.1. ANÁLISIS DE COSTOS

4.1.1. Recursos Humanos

En el siguiente apartado, se toma en cuenta los costos de los profesionales involucrados en el proyecto, en este caso es del mismo tesista. Por ello, el tesista es un de un especialista con experiencia tanto en el desarrollo de algoritmos de detección facial como en la programación en Python. Además, fue responsable de liderar el diseño y la implementación del algoritmo de detección de rostro, así como de utilizar sus habilidades en Python para desarrollar el software necesario para la integración del sistema.

Tabla 10

Presupuesto de Recursos Humanos

Presupuesto de Recursos Humanos	
Desarrollo del algoritmo de detección de rostro	S/. 1500
Capacitación en lenguaje de programación Python	S/. 1000
Costo Total 1	**S/. 2500**

Nota. En la Tabla 10, se dan a conocer el monto asignado por el desarrollo e implementación del algoritmo de detección de rostro, así como también el conocimiento en programación Python, ello debido a que el algoritmo está basado en este lenguaje.

4.1.2. Recursos de Hardware

En este apartado, se toma en cuenta los equipos utilizados para la implementación del sistema de detección de rostros, así como también herramientas que fueron útiles para la estructuración y desarrollo.

Tabla 11

Presupuesto de Recursos de Hardware

Equipos	P. Unitario	Cantidad	Precio Total
Tarjeta Jetson nano	S/. 450	1	S/. 450
Camara Raspberry	S/. 110	1	S/. 110
Tarjeta S.D	S/. 45	1	S/. 45
Monitor	S/. 120	1	S/. 120
Laptop Asus (ASUSTek COMPUTER INC.)	S/. 3200	1	S/. 3200
Cámara Web Full HD	S/. 275	1	S/. 275
Costo Total 2			**S/. 4200**

Nota. En la Tabla 11, se detallan los costos de los equipos utilizados para desarrollo del sistema, cabe destacar que algunos de los equipos ya se tenían en propiedad del tesista, pero de igual forma se tomó en cuenta para verificar un costo total de implementación.

4.1.3. Recursos de Software

En los recursos de hardware se va a tener que contemplar tipos de costos de hardware y también de implementación para obtener el costo total.

Tabla 12

Presupuesto de Recursos de Software

Presupuesto de Recursos de Software	
Software Python	S/. 350
Costo Total 3	**S/. 350**

Nota. En la tabla 12, se muestra el software utilizado para el desarrollo del trabajo de investigación

4.1.4. Costos del Desarrollo

Finalmente, se suma todos los costos identificados para el desarrollo total del trabajo de investigación, teniendo como conclusión la siguiente tabla.

Tabla 13

Sumatoria de Presupuestos

Sumatoria de Presupuestos	
Costo Total 1	**S/. 2500**
Costo Total 2	**S/. 4200**
Costo Total 3	**S/. 350**
Costo total	**S/. 7050**

Nota. En la tabla 13, se presenta la sumatoria total de los presupuestos indicados anteriormente.

4.2. ANÁLISIS DE COSTOS

4.2.1. Beneficios tangibles

- **Mejora de la seguridad:** El sistema de reconocimiento facial proporcionará un método avanzado de autenticación, lo que ayudará a mejorar la seguridad en entornos como edificios corporativos, instalaciones gubernamentales o sistemas de acceso a dispositivos móviles.
- **Eficiencia operativa:** Al automatizar procesos de identificación y autenticación, el sistema reducirá el tiempo y los recursos necesarios para verificar la identidad de las personas, lo que resultará en una mayor eficiencia operativa.
- **Reducción del fraude:** Al utilizar tecnología biométrica para verificar la identidad de las personas, se reducirá el riesgo de fraudes y suplantaciones de identidad.
- **Respuesta rápida a situaciones de seguridad:** Las alertas remotas permitirán a los usuarios recibir notificaciones inmediatas en sus dispositivos móviles en caso de detección de personas no autorizadas o comportamientos sospechosos, lo que permitirá una respuesta rápida a situaciones de seguridad.

- **Experiencia del usuario mejorada:** La integración de alertas remotas en dispositivos móviles proporcionará a los usuarios una mayor comodidad y accesibilidad, ya que podrán recibir notificaciones en tiempo real sin necesidad de acceder a sistemas centralizados.
- **Adaptabilidad y escalabilidad:** El sistema diseñado con software y interfaces flexibles permitirá su adaptación a diferentes entornos y su escalabilidad para cubrir las necesidades de organizaciones de diferentes tamaños y sectores.

4.2.2. Beneficios intangibles

- **Mejora de la percepción de seguridad:** La implementación de tecnología avanzada de reconocimiento facial podría mejorar la seguridad para los ciudadanos, lo que contribuiría a una sociedad mas segura.
- **Innovación tecnológica:** La adopción de sistemas de reconocimiento facial y alertas remotas demostraría el compromiso de las autoridades con la innovación y la adopción de tecnología de vanguardia para la seguridad de la población.
- **Confianza y satisfacción:** La implementación de medidas de seguridad avanzadas, como el reconocimiento facial, podría aumentar la confianza de los ciudadanos y mejorar su satisfacción al sentirse protegidos y seguros.
- **Contribución a la comunidad:** La implementación de sistemas de reconocimiento facial podría contribuir a la seguridad y tranquilidad de la comunidad en general, lo que podría mejorar las diversas actividades económicas y sociales con un menor riesgo de inseguridad.

4.3. ANÁLISIS DE SENSIBILIDAD

4.3.1. Desarrollo de Flujo de Caja

Empezamos por calcular el presupuesto global antes de estimar el coste anual de poner en marcha el sistema que utiliza la alerta remota de un dispositivo móvil para identificar los rostros de presuntos delincuentes que tienen órdenes de detención. Para el cálculo se usó los montos totales identificados anteriormente, así como también el beneficio por el sistema. Ya por el tercer año se identifica una ganancia neta.

Tabla 14

Flujo de Caja del Proyecto

AÑOS	**1**	**2**	**3**	**4**
	BENEFICIO			
BENEFICIO CON EL SISTEMA DE DETECCIÓN DE ROSTRO	**S/. 2750**	**S/. 2750**	**S/. 2750**	**S/. 2750**
	EGRESOS			
INVERSIÓN INICIAL	**S/. 7050**			
RECURSOS HUMANOS	**S/. 2500**			
RECURSOS DE HARDWARE	**S/. 4200**			
RECURSOS DE SOFTWARE	**S/. 350**			**S/. 350**
TOTAL DE EGRESO	**S/. 7050**	**S/. 0**	**S/. 0**	**S/. 350**
SUMATORIA (BENEFICIO-EGRESO)	**- S/. 4350**	**S/. 2750**	**S/. 2750**	**S/. 2400**
GANANCIA	**- S/. 4350**	**- S/. 1650**	**S/. 1100**	**S/. 3600**

Nota. En la Tabla 14, se muestra el flujo de caja del proyecto, que se identifica que a partir del tercer año ya se está obteniendo ganancias.

4.3.2. Análisis de VAN

Para el cálculo del Valor Actual Neto (VAN) de la tesis, utilizaremos los valores identificados en el flujo de caja y en los análisis de costos.

Datos:

- Tiempo (t) = 4 años
- Inversión Inicial I_0= S/. 7050
- Tasa de Interés (i) = 8%

La fórmula para calcular el VAN es el siguiente:

$$VAN = -i_0 + \sum_{t=1}^{n} \frac{V_t}{(1+k)^n}$$

$$VAN = 1911.71$$

4.3.3. Análisis de TIR

Para analizar la Tasa de Interés de Retorno (TIR), se calculará en base al VAN obtenido siguiendo la siguiente formula.

$$F1\ (1+TIR)1 + F2\ (1+TIR)2 \ldots \ldots \ldots Fn\ (1+TIR)\ n - Io = 0$$

Obteniendo un TIR de 20.769%

4.3.4. Análisis de ROI

El análisis del Retorno de Inversión se realiza en base a las ganancias obtenidas restado con la inversión divido con la misma, el resultado debería dar positivo para identificar que si tendremos un retorno de inversión efectivo.

$$ROI = \frac{(Ganancias - Inversión)}{Inversión}$$

ROI: 20.4%

CONCLUSIONES

- Con el presente trabajo ayudara a paliar la inseguridad que se presenta en el país y que cada día se observa un sostenido incremento, para identificar personas con orden de captura, el sistema embebido desarrollado en base a un algoritmo LBPH para realizar las etapas de detección y aprendizaje de rostros, proporciona un aliado en la lucha contra la delincuencia. El modelo se almacena en la base de datos y luego se utiliza comparación de imágenes extraídas con modelos variables pre-entrenados utilizando un algoritmo de reconocimiento facial Python de código libre.
- El sistema desarrollado tiene un menor costo, utilizando para este fin hardware de costos accesibles y de buen desempeño, como son la tarjeta Jetson Nano y la cámara Rasberry. Se realiza el flasheo del sistema operativo en tarjeta SD donde el proceso es completado y la imagen del sistema operativo es grabado en tarjeta SD, luego grabado se coloca a la Jetson Nano. Para una correcta instalación de dependencias, se necesita de los módulos de los que dependen el programa de reconocimiento facial. Archivo y algoritmo send SMS.py x. Folder correspondiente al programa de re-cocimiento facial para incorporación del programa desde la PC al sistema Jetson Nano. Reconocimiento facial del supuesto delincuente con orden de captura ejecutado en sistema Jetson Nano.
- Se concluye que el algoritmo de LBPH consta de dos etapas; entrenamiento y reconocimiento. El entrenamiento consiste en una data de entrenamiento de imágenes de rostros, Etiquetadas con su propia identidad. Cada rostro se divide en 64 celdas. Sobre cada celda se extraen histogramas de valores, mediante el valor central del histograma que surge como umbral, definiendo los ocho valores para el entorno circular adyacente. A cada vecino del umbral central se le asigna un nuevo valor binario: uno si el vecino es igual o mayor que el umbral central, y cero si el vecino es menor que el umbral central. Así mismo, la matriz extrae valores binarios omitiendo el valor central. A continuación de concatena los histogramas correspondientes a cada celda y esto se guarda en el

modelo. Finalmente, el reconocimiento facial se determina el array histograma LBP correspondiente al rostro que se requiere reconocer.

- El desarrollo de este proyecto de investigación usando la tarjeta embebida Jetson Nano y una tarjeta SD mediante un monitor externo permitieron obtener resultados a un 99,9% de aciertos. Se Validó el desempeño, modelo de reconocimiento facial mediante el empleo de una data.set de rostros etiquetados de 15 personas. Este data.set se obtiene de la base de datos Rulf face data.base. Para lo cual los rostros se capturan a partir de los videos de la base de datos el data.set, el cual es dividido en data de entrenamiento y data de validación. A continuación, se ejecuta el aprendizaje sobre la data de entrenamiento y se obtiene un modelo LBPH. Se consideran imágenes de 15 rostros de supuestos delincuentes con orden de captura para realizar el pre-entrenamiento y determinar el desempeño del reconocimiento con la base de dato como modelo. Reactivando un mensaje de texto alerta al dispositivo móvil cuando se ha reconocido la identidad de la persona, en caso que haya sido una persona sospechosa, con respectivos parámetros facial almacenada en la base de dato, se envía un mensaje de texto con el nombre de la persona, año, fecha y hora a tiempo real.
- Para realizar él envió de mensaje se emplea la plataforma de mensajería twilio que proporciona un número de teléfono virtual. Empleando una credencial para la cuenta en conjunto con la librería twilio de Python, dicho servicio es efectivo. Para evitar el envió excesivo de mensajes, solo se admite el mensaje de texto cada 5 minutos o si, se reconoce a una persona distinta.

RECOMENDACIONES

- Este proyecto se desarrolla mediante dos etapas de reconocimiento de supuestos delincuentes con orden de captura, pero se podría acoplar otro tipo de algoritmo que sea más robusto con respecto a variaciones en las condiciones de iluminación, posición y entorno de la persona a la cual se desea identificar, siendo de multiplataforma de código abierto o patentado.
- Componente de procesamiento podría implementarse sobre un servicio web en la nube. De esta manera se reduciría la carga de procesamiento de Jetson nano y se podría abarcar base de datos de personas con un tamaño mucho mayor.
- En Futuras mejoras al proyecto se podrían emplear distintas técnicas de reconocimiento de rostro, por ejemplo; Haar cascade, modelos deep learning. Esto se haría con el fin de comparar el desempeño de las distintas técnicas y seleccionar la más adecuada según consideraciones de eficiencia computacional, precisión, entre otras.
- A si mismo se valida, el reconocimiento facial con un muestreo de 15 personas en calidad de supuestos delincuentes con orden de captura, pero si a futuro se tiene 2 videos u fotografías distintos del delincuente con orden de captura, entonces se podría realizar el reconocimiento facial a tiempo real, mediante la comparativa entre la prueba y el modelo de la base de dato a entrenar, esto debido a que el (MININTER), solo brinda una foto del individuo como base de dato adquirido mediante la Reniec para el programa recompensa.
- En el presente proyecto se emplea la plataforma twilio para enviar alertas de alarma a los dispositivos móviles lo cual requiere una conexión de internet continua. En lugar de ello se podría emplear el módulo SIM800L para enviar las alertas, puesto que solo se requeriría un chip de móvil con saldo para enviar mensajes de texto.

REFERENCIAS

Altamirano, S. y Untuña Toalombo, V. (2022). *Sistema de control de acceso por medio de reconocimiento facial con uso de mascarilla y monitoreo de temperatura* [Tesis de licenciatura, Universidad Técnica de Ambato]. Repositorio Institucional Digital, Ambato, Ecuador. https://repositorio.uta.edu.ec/jspui/handle/123456789/34899

Barreto , R. y Lizarraga, D. (2019). *Modelo de sitema de reconocimiento facial para el control de la trata de personas.* [Tesis de Licenciatura, Universidad Tecnológica del Perú] Repositorio Institucional de la UTP. https://hdl.handle.net/20.500.12867/2063

Briceño, A. (11 de junio de 2019). *Recompensas fallidas: programa del Mininter y miles de soles bajo la lupa*. El Comercio. https://elcomercio.pe/lima/seguridad/recompensas-fallidas-programa-mininter-miles-soles-lupa-informe-video-noticia-ecpm-642908-noticia/?ref=ecr

Briones, E. y Sotomayor, M. (2020). Sistema de reconocimiento de personas y géneros aplicando técnicas machine learning en establecimientos comerciales [Tesis de licenciatura, Universidad de Especialidades Espíritu Santo]. Repositorio Institucional UEES. http://repositorio.uees.edu.ec/123456789/3194

Carmilema, L. M. y Pérez Malte, J. D. (2019). *Implementación de un sistema de rastreo y encendido automático Vehicular, mediante reconocimiento facial* [Tesis de licenciatura, Escuela Politécnica Nacional]. Repositorio Institucional EPN. https://bibdigital.epn.edu.ec/handle/15000/20347

Centurión, D. y Almeida Delgado, C. (2022). Reconocimiento facial utilizando redes neuronales artificiales en Raspberry Pi [Tesis de Licenciatura, Universidad Nacional del Este]. Respositorio Institucional de la Universidad Nacional del Este. http://servicios.fpune.edu.py:83/fpunescientific/index.php/fpunescientific/article/view/234

Domínguez, J. (2019). *Fundamentos de Programación en Lenguaje C*. IEASS Editores. https://shorturl.at/bdeoS

Flores, R. (2022). *Asistente virtual de acompañamiento mediante redes neuronales aplicados en un sistema embebido que verifica la presencia de niños en busetas escolares durante el trayecto* [Tesis de licenciatura, Universidad Técnica del Norte]. Repositorio Institucional Digital UTN, Ibarra, Ecuador. http://repositorio.utn.edu.ec/handle/123456789/12152

González, J. y Zhindón Mora, M. (2020). Plataforma de Servicios de Reconocimiento Facial para Detección de Prófugos de la Justicia en Ecuador. *Revista Ciencia e Investigación, 5*(3), 31-41. https://doi.org/10.5281/zenodo.3926901

Gutiérrez, D. (2021). *Diseño y desarrollo de un sistema de video vigilancia basado en dispositivos embebidos, técnicas de visión artificial y algoritmos inteligentes* [Tesis de licenciatura, Universidad Politécnica Salesiana sede Cuenca]. Repositorio Institucional de la Universidad Politécnica Salesiana. https://dspace.ups.edu.ec/handle/123456789/19956

Ibarra, W. (2020). *Sistema de control de acceso mediante identificación y verificación facial fundamentado en algoritmos de aprendizaje automático y redes neuronales* [Tesis de licenciatura, Universidad de las Fuerzas Armadas ESPE]. Repositorio Institucional ESPE. http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/22530/T-ESPE-043856.pdf?sequence=1&isAllowed=y

Kelvin, S. (10 de noviembre de 2017). *Reconocimiento facia l: comprensión del algoritmo LBPH.* Medium. https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b

López, O. (2020). *Planteamiento de un algoritmo para la identificación de posibles delincuentes mediante* [Tesis de grado, Universidad Distrital Francisco José de Caldas]. Repositorio Institucional de Universidad Distrital Francisco José de Caldas. http://hdl.handle.net/11349/28210

Manga, R. (2019). *Sistema de biometría facial para la búsqueda e identificación de personas desaparecidas en Colombia* [Tesis de titulo, Universidad de Cartagena]. Repositorio institucional Universidad de Cartagena. https://repositorio.unicartagena.edu.co/bitstream/handle/11227/8512/TESIS%20PEDROZA%20MANGA.pdf?sequence=1

Muñoz, E. (2021). *Desarrollo de un sistema de control de acceso de personal empleando reconocimiento facial respaldado con técnicas de aprendizaje profundo* [Tesis de licenciatura, Universidad de las Fuerzas Armadas]. Repositorio Institucional ESPE. http://repositorio.espe.edu.ec/bitstream/21000/25302/1/T-ESPE-044623.pdf

Rodríguez, S., Giménez, M. y Molina, M. (2022). *Marco Metodológico para el Desarrollo de un Sistema de Reconocimiento Biométrico Mediante Técnicas de Machine Learning.* Editorial de la Universidad Nacional de la Rioja (EUDELAR). http://sedici.unlp.edu.ar/handle/10915/149102

Toalombo, S. y Toalombo, V. (2022). *Sistema de control de acceso por medio de reconocimiento facial con uso de mascarilla y monitoreo de temperatura* [Tesis de licenciatura, Universidad Técnica de Ambato]. Repositorio institucional UTA, Ambato, Ecuador. https://repositorio.uta.edu.ec/jspui/handle/123456789/34899

Velarde, O. y Yabarrena Tamayo, J. (2018). *Sistema de reconocimiento facial para el control de la trata de personas en Perú* [Tesis de titulo, Universidad Andina del Cuzco]. Repositorio institucional Digital Universidad del Cusco. https://hdl.handle.net/20.500.12557/2047

Zuluaga, J. y Hernandez Reyes, S. (2022). *Reconocimiento de expresiones faciales usando un sistema embebido* [Tesis de licenciatura, Universidad Pontificia Universidad Javeriana]. Repositorio Institucional Universidad Pontificia Universidad Javeriana. https://repository.javeriana.edu.co/handle/10554/63654

GLOSARIO

Reconocimiento Facial: Proceso automatizado de identificación o verificación de la identidad de una persona mediante el análisis de características faciales únicas.

Sistema de Reconocimiento Facial: Conjunto de hardware y software diseñado para llevar a cabo la identificación o verificación de individuos mediante el análisis de sus rasgos faciales.

Software de Reconocimiento Facial: Aplicación informática que utiliza algoritmos para analizar y comparar características faciales con el fin de identificar o verificar a una persona.

Interfaz de Usuario (UI): Medio a través del cual los usuarios interactúan con el sistema, incluyendo elementos visuales y controles que facilitan la comunicación.

Alerta Remota: Sistema de notificación o aviso que se envía a una ubicación o dispositivo distinto al que originó la alerta.

Dispositivos Móviles: Dispositivos electrónicos portátiles, como smartphones o tablets, que permiten la movilidad del usuario y tienen capacidades de procesamiento y conectividad.

API (Interfaz de Programación de Aplicaciones): Conjunto de reglas y herramientas que permite la creación de software y la comunicación entre diferentes componentes de un sistema.

Algoritmo de Reconocimiento Facial: Conjunto de instrucciones lógicas diseñadas para realizar el análisis de una imagen facial y generar una representación única para su comparación.

Base de Datos Facial: Almacén de datos que contiene información sobre las características faciales de individuos registrados, utilizado para comparación durante el proceso de reconocimiento facial.

Seguridad Biométrica: Enfoque de seguridad que utiliza características biológicas únicas, como el rostro, huellas dactilares o la voz, para autenticar la identidad de una persona.

Encriptación: Proceso de codificación de la información para protegerla contra accesos no autorizados.

Notificación Push: Mensaje que se envía de forma instantánea a un dispositivo móvil, incluso cuando la aplicación relacionada no está activa.

Interconexión de Dispositivos: Conexión y comunicación entre diferentes dispositivos, permitiendo la transferencia de datos o información.

Privacidad y Cumplimiento Normativo: Consideraciones y medidas relacionadas con la protección de la privacidad de los individuos y el cumplimiento de las leyes y regulaciones aplicables.

Machine Learning (Aprendizaje Automático): Subcampo de la inteligencia artificial que se ocupa de desarrollar algoritmos y modelos que permiten a las máquinas aprender patrones y realizar tareas sin una programación explícita.

ANEXOS

ANEXO A: MATRIZ DE CONSISTENCIA

PROBLEMÁTICA	PREGUNTA DE INVESTIGACIÓN	OBJETIVO GENERAL	PREGUNTAS DE INVESTIGACIÓN	OBJETIVOS ESPECÍFICOS	ACCIONES	MARCO TEÓRICO	DESARROLLO	EVALUACIÓN	CONCLUSIONES	RECOMENDACIONES
Ante la falta de una herramienta de identificación de delincuentes, estos se fugan con facilidad por los terminales terrestres, fronteras, aeropuertos etc. De esta forma, el estado para reducir el impacto de los delincuentes prófugos ofrece recompensa que oscila entre S/ 10.000 a S/150.000 soles al brindar información de delincuentes con orden de captura por	¿Cómo realizar el reconocimiento facial con alerta remota para dispositivos móviles?	Diseñar e implementar un sistema de reconocimiento facial utilizando software e interfaces con alerta remota para dispositivos móviles	• ¿Cómo reconocer los rostros de los prófugos de la justicia? • ¿Cómo integrar el módulo de captura, procesamiento, almacenamiento e interfaz de reconocimiento facial? • ¿Cómo reducir el error en el reconocimiento de rostros? • ¿Cómo verificar el funcionamiento del sistema mediante alerta remota hacia un dispositivo móvil?	• Desarrollar un sistema de reconocimiento de rostros en función de un algoritmo. • Desarrollar un prototipo de sistema software que integre el módulo de captura, procesamiento, almacenamiento e interfaz de reconocimiento facial • Implementar técnicas de procesamiento digital de imágenes que brinden un alto	Desarrollar el algoritmo de reconocimiento de rostros Esquematizar el prototipo del sistema Aplicar técnicas de procesamiento de imágenes Prueba del sistema de reconocimiento de rostros.	El reconocimiento facial, inicio en los años sesenta mediante el sistema semiautomático que consistía en realizar marcas en la fotografía de esa forma delimitar la característica principal, ojos, nariz, boca, oído. El sistema de reconocimiento facial necesito un AMINISTRADOR para ubicar aquellos parámetros de	Desarrollo un sistema de identificación de rostros Desarrollo del Prototipo del sistema Implementar técnicas de procesamiento de imágenes Validar el funcionamiento del sistema	Análisis de Costos, Análisis de Beneficios, Desarrollo de flujo de caja, Análisis del VAN, Análisis del TIR y Análisis del ROI	Con el presente trabajo ayudara a paliar la inseguridad que se presenta en el país y que cada día se observa un sostenido incremento, para identificar personas con orden de captura, el sistema embebido desarrollado en base a un algoritmo LBPH para realizar las etapas de detección y aprendizaje de rostros, proporciona un	Este proyecto se desarrolla mediante dos etapas de reconocimiento de supuestos delincuentes con orden de captura, pero se podría acoplar otro tipo de algoritmo que sea más robusto con respecto a variaciones en las condiciones de iluminación, posición y entorno de la persona a la cual se desea identificar, siendo de multiplataforma de código abierto o patentado.

diversos delitos cometidos.				**nivel de confiabilidad.** • **Desarrollar pruebas y validar el sistema de alerta remota hacia un dispositivo móvil**		**rostro. Así mismo, "se calcularon las distancias medias y relaciones de estas marcas a un punto de referencia común, finalmente se comparaban con los datos de referencia". Ya en los años setenta en una de sus obras toca expone, menciona color de cabello, el grosor de labios, narices entre otras características (Toca, 2011 citado por Gutiérrez, 2021, p.16).** **"Los sistemas de**			**aliado en la lucha contra la delincuencia. El modelo se almacena en la base de datos y luego se utiliza comparación de imágenes extraídas con modelos variables pre-entrenados utilizando un algoritmo de reconocimiento facial Python de código libre.**	

						reconocimiento biométrico utilizan características fisiológicas o de comportamiento propias de cada individuo para identificarlo, es decir, se reconoce al usuario por lo que es en lugar de por lo que posee o sabe" (Rodríguez, Giménez, & Molina, 2022).				

ANEXO B: DIAGRAMA DE GANTT

	M de ta	Nombre de tarea	Duración	Comienzo	Fin
1		**Diseño e implementación de un sistema de reconocimiento facial utilizando software e interfaces con alerta remota para dispositivos móviles**	**240 días**	**mié 1/02/23**	**mar 2/01/24**
2		**Diseñar e implementar un sistema de reconocimiento facial utilizando software e interfaces con alerta remota para dispositivos móviles**	**80 días**	**mié 1/02/23**	**mar 23/05/23**
3		Investigación y estudio del estado del arte	1 ms	mié 1/02/23	mar 28/02/2
4		Definición y Requisitos del sistema	1 ms	mié 1/03/23	mar 28/03/2
5		Diseño y eimplementación del sistema	2 mss	mié 29/03/2	mar 23/05/2
6		**Desarrollar un prototipo de sistema software que integre el módulo de captura, procesamiento, almacenamiento e interfaz de reconocimiento facial**	**40 días**	**mié 24/05/23**	**mar 18/07/23**
7		Desarrollo del módulo de captura	10 días	mié 24/05/2	mar 6/06/23
8		Desarrollo del módulo de procesamiento	10 días	mié 7/06/23	mar 20/06/2
9		Desarrollo del módulo de almacenamiento	10 días	mié 21/06/2	mar 4/07/23
10		Desarrollo de la interfaz de reconocimiento facial	10 días	mié 5/07/23	mar 18/07/2
11		**Implementar técnicas de procesamiento digital de imágenes que brinden un alto nivel de confiabilidad.**	**20 días**	**mié 19/07/23**	**mar 15/08/23**
12		Selección de técnicas de procesamiento de imágenes	10 días	mié 19/07/23	mar 1/08/23
13		Optimización y mejora de algoritmos	10 días	mié 2/08/23	mar 15/08/2
14		**Desarrollar pruebas y validar el sistema de alerta remota hacia un dispositivo móvil**	**100 días**	**mié 16/08/23**	**mar 2/01/24**
15		Diseño de casos de prueba	20 días	mié 16/08/2	mar 12/09/2
16		Ejecución de pruebas	20 días	mié 13/09/2	mar 10/10/2
17		Ajustes y correciones	30 días	mié 11/10/2	mar 21/11/2
18		Validación del funcionamiento del sistema	30 días	mié 22/11/2	mar 2/01/24