UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

Tema:

ALGORITMOS DE PROCESAMIENTO DE SEÑALES PARA EL RECONOCIMIENTO FACIAL Y DE VOZ EMPLEANDO REDES NEURONALES

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones.

ÁREA: Electrónica

LINEA DE INVESTIGACIÓN: Tecnología de la información y sistemas de control

AUTOR: Carlos Alexander Orozco Analuiza

TUTOR: Ing. Juan Pablo Pallo Noroña, Mg

AMBATO – ECUADOR

septiembre - 2022

APROBACIÓN DEL TUTOR

En calidad de tutor de Trabajo de Titulación con el tema: ALGORITMOS DE PROCESAMIENTO DE SEÑALES PARA EL RECONOCIMIENTO FACIAL Y DE VOZ EMPLEANDO REDES NEURONALES, desarrollado bajo la modalidad Proyecto de Investigación por el señor Carlos Alexander Orozco Analuiza, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2022

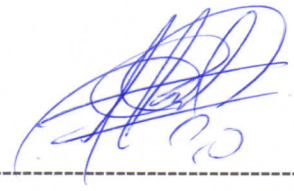
Ing. Juan Pablo Pallo Noroña, Mg

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: ALGORITMOS DE PROCESAMIENTO DE SEÑALES PARA EL RECONOCIMIENTO FACIAL Y DE VOZ EMPLEANDO REDES NEURONALES es absolutamente original, autentico y personal. En tal virtud el contenido, efectos legales y académicos que se desprendan del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022



Carlos Alexander Orozco Analuiza

C.I 1804946489

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Carlos Alexander Orozco Analuiza, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado ALGORITMOS DE PROCESAMIENTO DE SEÑALES PARA EL RECONOCIMIENTO FACIAL Y DE VOZ EMPLEANDO REDES NEURONALES, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora presidenta del Tribunal.

Ambato, septiembre 2022.

Firmado electrónicamente por:
ELSA PILAR URRUTIA

Ing. Pilar Urrutia, Mg.
PRESIDENTA DEL TRIBUNAL

Firmado electrónicamente por:
MARCO ANTONIO JURADO LOZADA

Ing. Marco Jurado
PROFESOR CALIFICADOR

Firmado electrónicamente por:
FABIAN RODRIGO SALAZAR ESCOBAR

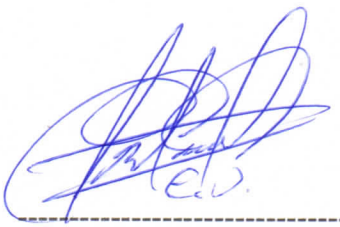
Ing. Fabián Salazar
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación a favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizó su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2022



Carlos Alexander Orozco Analuiza

C.I 1804946489

AUTOR

DEDICATORIA

A Dios todo poderoso que con su santísimo manto me brindo salud, trabajo y sabiduría para llegar a esta etapa final de mi carrera.

A mi padre Luis Orozco que con su sacrificio, trabajo y dedicación nunca me falto su apoyo durante el transcurso de mi carrera.

A mi madre Blanca Analuiza que las buenas y las malas con su amor infinito supo guiarme por el buen camino para ser un hombre de bien.

A mis Hermanos William, Javier, Diego y Jessenia que supieron apoyarme durante mi vida universitaria.

A mis primos, amigos y familiares que me dieron un aliento de superación para cumplir mi objetivo

Carlos Alexander Orozco Analuiza

AGRADECIMIENTOS

Con un emotivo y un gran agradecimiento quiero primeramente agradecer a mi Dios poder haber brindado la sabiduría e inteligencia para cumplir esta meta en mi vida.

Gracias Padres por su apoyo, compresión y sacrificio; por brindarme la oportunidad de ser alguien en la vida. El esfuerzo y sacrificio que usted lo pusieron para que nunca me falte nada, se ve reflejado hoy en la obtención de una meta más en vida.

Gracias hermanos y hermana por estar en las buenas y en las malas y ser partícipe de este gran sueño.

Gracias a la Universidad Técnica de Ambato en especial a la facultad de Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

Gracias a los docentes de la carrera de Ingeniería Electrónica y Comunicaciones por bridarme sus conocimientos, experiencias y palabras de aliento durante mi vida universitaria, en especial al Ing. Juan Pablo Pallo Noroña por haber brindado su apoyo técnico en la realización de este proyecto.

Gracias a mis amigos y compañeros que compartimos el salón de clases y brindarme una amistad desinteresada y sincera. Y a todas aquellas personas que estuvieron a lo largo de este caminar y formaron de mí una persona de bien.

A todos un Dios le pague……

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN EJECUTIVO

El presente trabajo de titulación trata del desarrollo de un sistema de control de acceso por medio de reconocimiento facial y voz, para la autentificación de personas de una vivienda. En la actualidad los niveles de inseguridad han aumentado y esto ha llevado a que cada vez más se incremente los robos y daños hacia los inmuebles por el bajo nivel de seguridad en una vivienda.

El dispositivo desarrollado en este proyecto se basa en un biométrico de acceso bimodal, a través del aprendizaje automático y las redes neuronales, un subcampo de la Inteligencia Artificial. El sistema cuenta de dos métodos de autentificación: facial y voz, para lo cual se empleó modelos de redes neuronales diseñadas por el investigador con las etapas de: formación de la base de datos, procesamiento de imágenes y audios, y diseño la red neuronal.

Se implementa dos métodos de autentificación facial y voz para evitar suplantaciones de identidad, una cámara se encarga de capturar el rostro de la persona y realizar el reconocimiento facial, si el usuario está registrado se activa el micrófono para capturar la clave de acceso y permitir el acceso, para registrar los datos se emplea un servidor LAMP donde se guarda la información del sistema y notificaciones al usuario mediante la aplicación Telegram. Este proyecto se orienta al control de acceso de personas hacia una vivienda evitando utilizar métodos de autentificación tradicionales.

Palabras clave: Redes neuronales, reconocimiento facial, reconocimiento de voz, control de acceso, RaspberryPi.

ABSTRACT

The present titling work deals with the development of an access control system through facial and voice recognition, for the authentication of people in a home. Currently, the levels of insecurity have increased and this has led to an increase in theft and damage to real estate due to the low level of security in a home.

The device developed in this project is based on a bimodal access biometric, through machine learning and neural networks, a subfield of Artificial Intelligence. The system has two authentication methods: facial and voice, for which neural network models designed by the researcher were used with the stages of: database formation, image and audio processing, and neural network design.

He implements two methods of facial and voice authentication to avoid identity theft, a camera is responsible for capturing the person's face and performing recognition through the neural network, if the user is registered, the microphone is activated to capture the key of access and process it through the neural network, to record the data a LAMP server is used where the system information and user notifications are stored through the Telegram application. This project is aimed at controlling the access of people to a home, avoiding the use of traditional authentication methods.

Keywords: Neural networks, facial recognition, voice recognition, access control, RaspberryPi.

CAPÍTULO I

MARCO TEÓRICO

1.1 Tema de investigación

ALGORITMOS DE PROCESAMIENTO DE SEÑALES PARA EL RECONOCIMIENTO FACIAL Y DE VOZ EMPLEANDO REDES NEURONALES

1.2 Antecedentes Investigativos

Los avances más esenciales para la creación del prototipo se tuvieron en cuenta mediante el análisis de diversos trabajos sobre proyectos de investigación y artículos relacionados con algoritmos de reconocimiento facial y de voz basados en redes neuronales artificiales, que se encuentran documentados en los siguientes casos:

En la Universidad Técnica de Ambato, Jonny Bastidas en el trabajo con el tema "Registro de asistencia de alumnos por medio de reconocimiento facial utilizando visión artificial" en el año 2019. Donde hace mención sobre el uso de las redes neuronales como Histograma de Gradientes orientados (HOG) y la red neuronal Convoluciónal de Redes Neuronales (CNN) para el registro de asistencia de los estudiantes de la Universidad de las Fuerzas Armadas, solucionando así el problema de tomar lista a mano. Como tecnologías hace uso de visión artificial por computador simplemente a través de un Software Python en la cual elaborara un script con la capacidad de obtener el registro de los estudiantes, a través de una captura de con los rostros de los estudiantes. Al final del trabajo concluye que la red neuronal CNN es mucha más eficiente a la hora de realizar métodos de reconocimiento facial, a diferencia de la red neuronal HOG, esto debido al tiempo de procesamiento y precisión siendo la red neuronal CNN más eficiente, teniendo como resultado de un 92% de efectividad y la red HOG con un nivel de efectividad del 50%. Estos resultados dependen mucho de la velocidad de procesamiento de cada red y también de la iluminación del lugar. [1]

En el año 2019, Obando Darío en su trabajo realizado en la Universidad de las Fuerzas Armadas ESPE en la ciudad de Sangolquí con el tema "Implementación de un control de acceso biométrico mediante reconocimiento facial" en la cual menciona el uso de un sistema robusto de verificación para el control de acceso de personas, el cual detalla la aplicación del método utilizando tarjetas RFID y otro método llamado sistema biométrico en tiempo real. Dentro de las tecnologías utilizadas hace uso de la inteligencia artificial utilizando un Arduino para el control RFID con una cerradura y para la detección facial utiliza la red neuronal CNN implementada en un servidor de Linux en la cual procesa la información a través de un script realizado en Python, comprendida de librerías tales como Open CV y Tkinter. Se concluye que se utilizaron redes neuronales artificiales para la detección de rostros a una distancia de 50cm obteniendo como resultado del 88.94 % de confiabilidad y de 92.88% a una distancia de 80 cm. [2]

Otra trabajo realizada por Christian Arroyo en año 2019 con el tema "Desarrollo de un sistema prototipo de acceso a los laboratorios de redes de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional basado en reconocimiento facial" para la Escuela Politécnica Nacional en la ciudad de Quito , en la cual menciona la utilización de librerías como OpenCV , EmguCv y Ozeki para poder realizar el reconocimiento facial que permite el acceso al personal indicado a los laboratorios de dicha institución . El diseño propuesto en este tema es capaz de registrar al personal indicado y guardar en una base de datos con todos los registros necesarios de cada usuario y poder controlar el acceso mediante reconocimiento facial a través de las librerías OpenCV a través de la tecnología de reconocimiento de rostros, esta tecnología permite acceder a diferentes lugares e incluso les permite obtener información personal. Como tecnología hardware implementada en este tema hace referencia a la utilización de una computadora como servidor y una cámara IP para la detección de las personas. Como conclusiones de esta investigación se menciona que la efectividad del reconocimiento facial fue de un 68% en comparación a otros métodos de control de acceso debido a factores externos como la resolución de la cámara y la iluminación del lugar, además de los complejos métodos de reconocimiento facial para los sistemas biométricos que existen en la actualidad. [3]

El trabajo realizado en Vellore Instituta of Technology por Navya Saxena y Devina Varshney con el tema “Smart Home Security Solutions using Facial Authentication and Speaker Recognition through Artificial Neural Networks” en año 2021, en este proyecto de titulación se presenta un enfoque integral de la seguridad del hogar inteligente. Utiliza las dos tecnologías independientes de vanguardia de reconocimiento de voz y autenticación facial para ayudar a aumentar la privacidad y la seguridad. Este método implica el reconocimiento facial al tomar una señal en tiempo real de la persona en la puerta y luego realizar un análisis de transmisión en vivo donde la cara reconocida se autentica con los datos del propietario en la base de datos que coincide con la cara con un nombre. El reconocimiento de voz se ha utilizado para comprobar doblemente el resultado de la autenticación facial. Todo el proceso se lleva a cabo con la ayuda de redes neuronales. La precisión general del modelo propuesto es del 82,71% con una precisión del 87,5% para la autenticación facial y del 84,62% para la autenticación de voz. [4]

Y por último la investigación por Shanthakumar H.C, Nagaraja G.S, Mustafa Basthikodi con el tema “Performance Evolution of Face and Speech Recognition system using DTCWT and MFCC Features”, realizado en el 2021, En el trabajo propuesto hace uso de inteligencia artificial como la principal tecnología , el reconocimiento facial se realiza mediante DTCWT (Dual Tree Complex Wavelet Transform) integrado con QFT (Quick Fourier Transform) predominante y el reconocimiento de voz se realiza mediante el algoritmo MFCC (Mel Frequency Cepstral Coefficients). Las variables de rendimiento como EER, FRR, FAR y TSR se evalúan para el reconocimiento de personas. Como conclusión se observa que la Tasa de Éxito Total es del 98,83% para la base de datos de rostros y del 97,50% para la base de datos de voz. [5]

Todas las investigaciones analizadas previamente tienen en común la aplicación de redes pres entrenados basados en redes neuronales, ya que es un método más confiable y utilizado para proyectos de reconocimiento de rostros y voz. Por lo cual para este

proyecto no se pretende utilizar redes neuronales entrenadas, si no que se propone crear una red neuronal desde que permita realizar el reconocimiento facial y de voz para un sistema de control y acceso, teniendo en cuenta el previo análisis del procesamiento de señales , por lo cual en este proyecto se realizar un sistema biométrico bimodal con dos maneras de autenticación , mediante el reconocimiento de rostros y voz en un mismo sistema electrónico y además se guardara la detección de la persona en la base de datos a través de SQL y notificaciones de acceso a través de la aplicación de mensajería Telegram.

1.2.1 Contextualización del problema

Según la fiscalía general del Estado del Ecuador los robos a domicilios entre los meses de enero y noviembre del año 2021 fueron alrededor de 6128 robos en todo el país, teniendo un incremento de un 48% con respecto al año 2020, según el horario de los robos el 23.1% se lo realiza en las mañanas, un 27.5% por las tardes y 49.4 % de robos en las noches, con promedio de 677 robos a domicilios por mes. Pichincha y Guayas fueron las provincias que registraron más robos a domicilios, con el 21% y 20.3% respectivamente. Además, recalca que la mayoría de los robos son llevados a cabo los sábados y domingos, cuando las personas no se encuentran dentro del mismo y toman como ventaja que no cuentan con sistemas de seguridad seguros que protejan una vivienda. Ya que la mayoría de las personas simplemente utilizan un candado para cerrar sus hogares al momento de salir del mismo, siendo muy fáciles de vulnerar y romper. [6]Todos los días se registran actos de delincuencias en el país, según el Banco Interamericano de Desarrollo (BID) el país pierde alrededor de $3000 millones anuales por actos de la delincuencia organizada y además establece que los niveles crecientes de delincuencia impiden que la economía del país crezca un 3.5% anualmente. [7]

En la actualidad con el avance de la tecnología se ha logrado implementar sistemas de seguridad inteligentes más seguros y fiables gracias a la domótica, logrando así tener un resultado mucha más satisfactorio en la seguridad del hogar. Sin embargo muchas personas no conocen sobre los beneficios de los sistemas de seguridad inteligentes tienen en la protección de una vivienda y esto es debido a la falta de información e

implementación de estos sistemas de seguridad, haciendo que las personas prefieren seguir utilizando los sistemas de seguridad convencionales.

No obstante, una de las características más relevantes que pueden otorgar los sistemas de seguridad inteligentes es la protección y seguridad en el hogar, dado que la inseguridad es uno de los problemas más recurrentes en la ciudad, y más aún los delitos contra el patrimonio en los cuales se incluye hurto a personas, a establecimientos, vehículos, celulares y residencia. Este último es uno de los más preocupantes ya que se vulnera la seguridad e integridad de una persona dentro de su hogar donde se supone que es más seguro que en las calles. Según un balance de seguridad en Ambato por parte de la Policía Nacional, ya se contabilizo más de 134 ilícitos de este tipo en el 2021, siendo los domicilios pertenecientes a estratos sociales medio o alto, sobre todo aquellos ubicados en sectores donde no existe buen alumbrado público o están circundadas por calles sin mantenimiento los preferidos por los delincuentes. Según el coronel Marco Antonio Muñoz, jefe de la Sub zona de Tungurahua los robos a domicilios han aumentado debido a la pandemia, ya que muchas personas no cuentan con los recursos económicos, ni trabajo y se dedican a este tipo de delitos, además afirma que en el mes de noviembre del 2021 se registró el mayor hurto a una vivienda en la ciudad de Ambato, en la cual los delincuentes lograron sustraer alrededor de 450000 dólares bajo amenazas a los propios ocupantes. [8]

Con una ciudad menos segura y el aumento de este tipo de delitos, se ha visto en la necesidad de implementar maneras más eficientes que ayuden a reducir o prevenir los robos en las viviendas. En los últimos años con los avances tecnológicos, la inteligencia artificial ha sido una de las nuevas tendencias en la transformación digital, teniendo un amplio crecimiento en ámbitos de seguridad, salud, agricultura, pesca, entre otras. Ecuador ha sido una de los países pioneros en implementar el uso de la inteligencia artificial al contar con un sistema auxiliar de diagnóstico con inteligencia artificial, basado en la Cloud de la empresa china Huawei, en la cual analiza miles de imágenes almacenadas, de lesiones sospechosas de los pulmones de pacientes afectados por el COVID-19, el cual ayuda a contar con un diagnóstico más certero y

rápido de personas contagiadas con COVID-19. Además, en el país existen muchas más aplicaciones implementadas en la industria, como la creación de Chat Bots para mensajería automática para la atención al cliente por parte de las Entidades Bancarias, utilización de drones inteligentes para la el transporte y fumigación de cualquier tipo de plantación, utilización de gafas inteligente para la visión artificial y realidad virtual. [9]

El desarrollo de este proyecto está enfocado en buscar solventar necesidades relacionadas con la seguridad y automatización de una vivienda mediante el control de personas que puedan ingresar y salir de la misma. En la actualidad existen muy pocos sistemas de seguridad basadas en autentificación facial y reconocimiento de voz que ayuden la seguridad de los hogares, con esto se propone desarrollar un sistema biométrico multimodal empleando algoritmos de reconocimiento facial y voz que utilicen redes neuronales. El prototipo se desarrollara en una tarjeta de bajo coste junto con una cámara y un micrófono que permita que el prototipo sea portable para que pueda ser implementado en cualquier ambiente, teniendo como beneficiarios a los integrantes de una vivienda en la cual podrán tener acceso de manera más segura hacia el interior de la vivienda, sin la necesidad de utilizar una llave o clave para poder ingresar al domicilio.

1.2.2 Fundamentación teórica

Sistema de Control de acceso

El sistema de control de acceso tiene la capacidad de controlar y permitir el acceso de un individuo o varios, a una zona específica utilizando varias maneras de identificación. Además, sirve como un sistema de seguridad ya que permite saber quién ha hecho que, donde y cuando, y permite controlar quien pude hacer que, donde y cuando. [10]

Para el funcionamiento de un sistema de control se integran varias tecnologías como tarjetas inteligentes, sensores, lectores biométricos, métodos de autentificación e inclusive puertas automáticas como se observa en la figura 1. Todos estos elementos

son controlados mediante un software especializado ejecutado en una computadora o tarjeta de desarrollo.

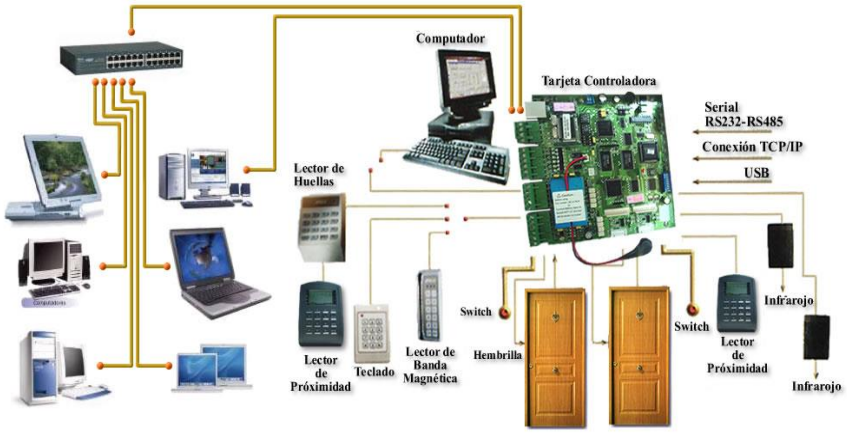


Figura 1 Sistema de control de acceso. [11]

Un sistema de control de acceso tiene como finalidad:

- Controlar quién tiene acceso a departamentos o partes específicos de una inmueble.
- Controlar el acceso a bases de datos, sistemas informáticos y otros servicios de información.
- Evitar el acceso no autorizado a los activos físicos, maquinaria o datos de las empresas por parte de terceros.
- Detectar el acceso no aprobado y establezca medidas de seguridad para detenerlo.
- Realizar un seguimiento de las acciones importantes realizadas por los usuarios del sistema y revíselas.
- Facilitar la gestión de empleados y la organización del negocio.

Un sistema de control de acceso trabaja mediante tres etapas: identificación, autentificación y autorización.

Identificación

Identificación es el proceso en el cual se realiza la identificación del usuario, para la cual existen varias maneras para la identificación como: huellas dactilares, tarjetas de identidad, el reconocimiento facial o voz, entre otros. [12]

Autentificación

Autentificación es el proceso que realiza la verificación de la identidad del usuario (mediante una credencial o una contraseña) que se encarga de verificar si los datos del usuario se encuentran en la base de datos y si posee los permisos de acceso. [12]

Autorización

Autorización es el último paso que permite autorizar el acceso de un usuario a un lugar o a cierta información, siempre y cuando ha cumplido con la identificación y autentificación de manera correcta. [12]

Tipos de control de acceso

Hay diferentes maneras de clasificar los controles de acceso esto depende del lugar a controlar o por el método de recopilar información para su acceso, entre los cuales se tiene:

Sistema de control de acceso autónomo

Un sistema de control de acceso autónomo es cuando no se requiere de un dispositivo para permitir el control de una o más elementos, estos sistemas carecen de limitar el acceso por horarios o grupos. Un ejemplo de estos sistemas son las cerraduras electrónicas como se observa en la figura 2, son muy sencillos ya que solo identifica a la persona y le permite ingresar o salir de un lugar. [13]

Figura 2 Control de acceso autónomo. [14]

Sistema de control de acceso en red

Un sistema de control de acceso en red es un sistema que está conectado a una red inalámbrica a través de un dispositivo local o remoto como se observa en la figura 3, y hace uso de un software de control que permite tener un registro de todas las operaciones realizadas sobre el sistema con fecha, horario, autorización, etc. Algunos ejemplos son los sistemas biométricos. [13]

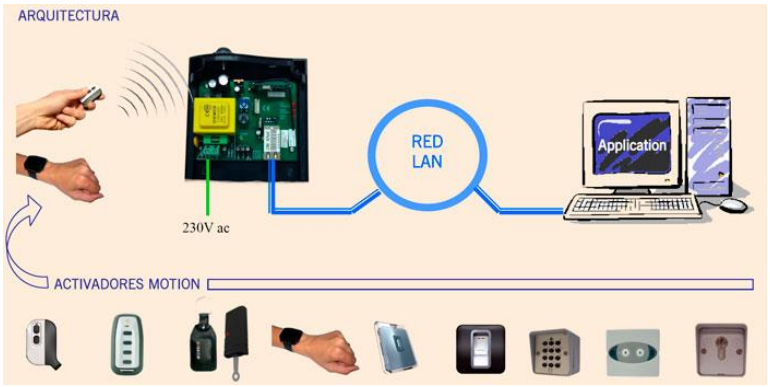


Figura 3 Control de acceso en red. [13]

Métodos de verificación para el control de acceso

El control de acceso es sumamente importante para que todos los usuarios tengan el acceso correspondiente a datos y recursos de sistema. Para lo cual existen diferentes dispositivos entre los cuales se tiene:

Dispositivo autónomo por teclado

El dispositivo autónomo por teclado es un método de seguridad basado en un teclado anclado a un dispositivo que permite el control de acceso mediante un código de números, generalmente estos dispositivos solo cuentan con una fuente de alimentación (pilas recargables) y son muy fáciles de instalar en cualquier puerta. Sin embargo, a pesar de que este sistema es muy seguro cuenta con algunas desventajas entre las cuales son: pocos utilizados para exteriores, ya que su uso es más comercial en departamentos de lujos, y además se cuenta con el riesgo de olvidar el código de acceso. [13]

Dispositivo autónomo por tarjeta

Un dispositivo autónomo por tarjeta es un sistema basado en una fuente de alimentación y un lector de tarjetas, son muy similares a los sistemas basados en teclado ya que utilizan una tarjeta en vez de un código. Estos sistemas tienen la capacidad de soportar varias tarjetas de identificación para el control de acceso. [13]

Dispositivo autónomo por huella dactilar

Un dispositivo autónomo por huella dactilar es una tecnología que permite identificar a una persona a través del análisis de características que pueden ser de fisiológicas como: las huellas dactilares o la retina. Sin ninguna duda estos sistemas son los más seguros y empleados en la actualidad en muchos ámbitos como: empresas, casas, autobuses, aeropuertos, etc. [15]

Problemas de control de acceso

El control de acceso es el proceso de controlar quién hace qué y va desde la administración del acceso físico a los equipos hasta determinar quién tiene acceso a un recurso. Muchas vulnerabilidades de seguridad se generan por el uso incorrecto de los controles de acceso. Casi todos los controles de acceso y las prácticas de seguridad pueden superarse si el atacante tiene acceso físico alas equipos objetivo. Por ejemplo, no importa que haya configurado los permisos de un archivo, el sistema operativo no puede evitar que alguien eluda el sistema operativo y lea los datos directamente del disco. Para proteger los equipos y los datos contenidos, el acceso físico debe restringirse y deben usarse técnicas de encriptación para proteger los datos contra robo o daño, para lo cual se hace un resumen de los problemas más comunes que existen en los sistemas de control de acceso como se observa en la tabla 1.

Tabla 1 Problemas de los sistemas de control de acceso. [16]

Problemas de los sistemas de control de acceso		
Método	Problema	Ilustración
Candado	Perdida de la llave, Deterioro de la llave y candado, Rotura o Deformación de la llave, Clonación de la llave, Fácil método de alteración	
Dispositivos autónomos por teclado	Olvidar el código de acceso, Desgaste de fuente de alimentación, fallos en lectura del código	
Dispositivos autónomos por tarjeta	Perdida de la tarjeta de identificación, Desgaste de fuente de alimentación, fallos en lectura de la tarjeta, Clonación de la tarjeta	
Dispositivos autónomos por huella dactilar	Deformación o heridas de la huella, Sudoración de las manos.	

Elaborado por: Investigador

Reconocimiento Facial

El reconocimiento facial es una forma de identificar mediante algoritmos de procesamiento de imágenes o video la identidad de una persona, todo esto es posible mediante su rostro de manera automática con la utilización de cámaras.

El reconocimiento facial se puede realizar de varias maneras, pero todos los procesos siguen una serie de etapas como: detección del rostro, acondicionamiento, normalización, extracción de características y reconocimiento. Estas etapas permitan

extraer y analizar características faciales de un individuo a partir de una imagen o un video como se observa en la figura 4, para después esta información convertirla en un modelo y compararla con varias imágenes de una base datos verificando la identificación de la persona. [17]

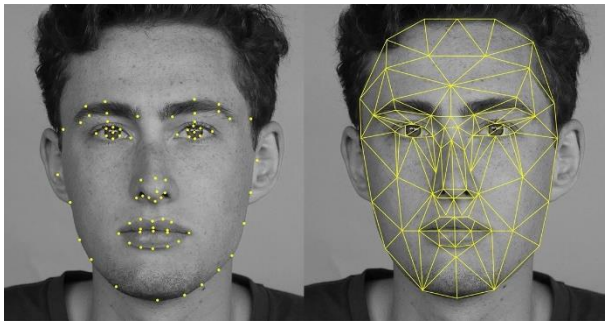


Figura 4 Método de control de acceso por reconocimiento facial. [18]

Fases del Sistema de Reconocimiento Facial

Los sistemas de tecnología facial pueden variar, pero en general tienden a funcionar de la siguiente manera:

La primera fase de detección facial se encarga de localizar el rostro en la imagen o video. Después se realiza el procesado a la imagen para alinear y normalizar los rostros, para lograr obtener características geométricas en común con todas las imágenes que se procesan. Luego, se lleva a cabo una extracción de características faciales para poder obtener información útil para la distinción de rostros. Y finalmente, las características obtenidas son comparadas con la base de datos para reconocer el rostro, como se observa en la figura 5. [19]

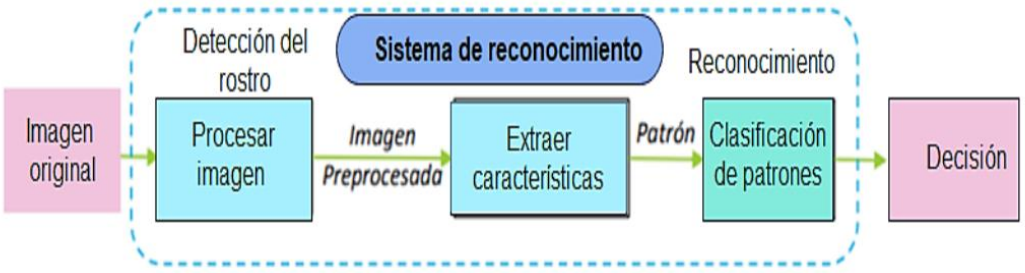


Figura 5 Etapas del reconocimiento facial. [19]

Cada fase es crítica ya que el resultado final está determinado por la precisión con la que se recopilaron las características, que depende tanto de la ubicación del rostro en la imagen como de la estandarización de la imagen.

Detección

Detección es la técnica mediante la cual el sistema localiza la posición de los rostros humanos en una imagen o cuadro, esto se conoce como detección de rostros como se observa en la figura 6. En el pasado para realizar el reconocimiento facial, era necesario realizar un proceso de detección manual del rostro en una imagen, en la actualidad existen una gran variedad de enfoques de detección automática, cada uno con su propio conjunto de bases.

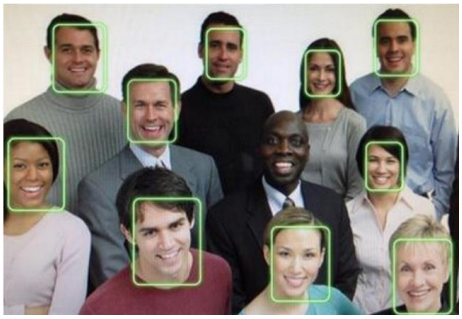


Figura 6 Ejemplo reconocimiento facial con detector de sesgo facial. [20]

Debido a la gran cantidad de algoritmos (muchos de los cuales tienen variantes), es imposible nombrar y describir todos los métodos de detección disponibles. No obstante, se mencionarán el algoritmo que se han considerado relevante en este proyecto.

Algoritmo de Viola-Jones

Un algoritmo es una secuencia de pasos finitos bien definidos que resuelven un problema. Desde el punto de vista informático un algoritmo es cualquier procedimiento computacional bien definido que parte de un estado inicial y un valor o un conjunto de valores de entrada, a los cuales se les aplica una secuencia de pasos computacionales finitos, produciendo una salida o solución. [21]

El método ideado por Paul Viola y Michael Jones fue el primer sistema de detección facial que podía utilizarse en tiempo real por su rapidez y precisión.

Como se observa en la figura 7, este método se basa en un conjunto de características conocidas como Haar-like features, que se derivan del producto escalar de una imagen y un patrón simple de igual tamaño que la imagen original, este escalar tiene un signo positivo o negativo dependiendo de cómo se establezca el patrón, y hay muchas formas diferentes de describir los patrones y además es el resultado que se comparará para ver si se puede encontrar una cara en un área determinada de la imagen.

Figura 7 Aplicación de Haar-like features a una imagen. [22]

El trabajo de Viola y Jones incluye tres contribuciones principales: el desarrollo de una nueva representación de imágenes (imagen integral) que reduce los tiempos de extracción de características, el diseño de un clasificador basado en AdaBoost y el uso de clasificadores en cascada.

Características

El método Viola-Jones detecta características en una cara usando características llamadas clasificadoras de Haar. En la visión por computadora, las características de Haar se utilizan para identificar la intensidad de los píxeles en una región de manera rastreable. Las características de Haar son regiones de imágenes rectangulares, y los clasificadores se componen de dos o tres características rectangulares que escanean constantemente la ventana en busca de características de rostros humanos, como se observa en la figura 8.

Figura 8 Haar feature. [22]

En el método de Viola - Jones se incluyen tres estrategias para la detección de componentes faciales:

- Las características de tipo Haar utilizadas en la extracción de características tienen forma rectangular y están determinadas por una imagen integral.
- AdaBoost es un enfoque de detección de rostros basado en el aprendizaje automático.
- Clasificador en cascada para fusionar de manera eficiente muchas de las características. El término 'cascada' en un clasificador se refiere a los diversos filtros que componen el clasificador final.

Imagen Integral

Una imagen integral es el nombre de una estructura de datos y un algoritmo utilizado para obtener esta estructura de datos. Se utiliza como una forma rápida y eficiente de calcular la suma de valores de píxeles en una imagen o parte rectangular de una imagen. La contribución de una imagen integral, es un nuevo formato de imagen que permite la extracción rápida de funciones al reducir la cantidad de operaciones en los píxeles. Los píxeles de la imagen integral incluyen la información de brillo acumulada entre el punto de coordenadas (0, 0) y el píxel en cuestión, y tiene el mismo tamaño que la imagen original, como se muestra en la figura 9.

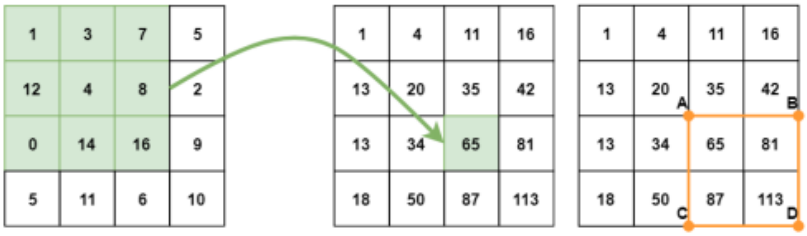


Figura 9 Integral image. [22]

Refuerzo Adaptativo (AdaBoost)

AdaBoost (AD) es un método para combinar un grupo de clasificadores débiles. El método AD es un poderoso clasificador. Una imagen integral es un pobre aprendiz; sin embargo, cuando se combinan numerosas imágenes integrales mediante AD, se crean clasificadores robustos para determinar las características faciales en la región de la ventana. Como se muestra en la figura 10 el proceso AD, se mueve de izquierda a derecha. La relevancia de las muestras azules que faltaban se indica por su tamaño. Los círculos azules más grandes son capturados por el segundo clasificado, y los círculos naranjas mal clasificados reciben mayor peso, mientras que los demás se reducen. Los círculos naranjas restantes son capturados por el tercer clasificado, y el clasificador fuerte final integra los tres clasificadores débiles.

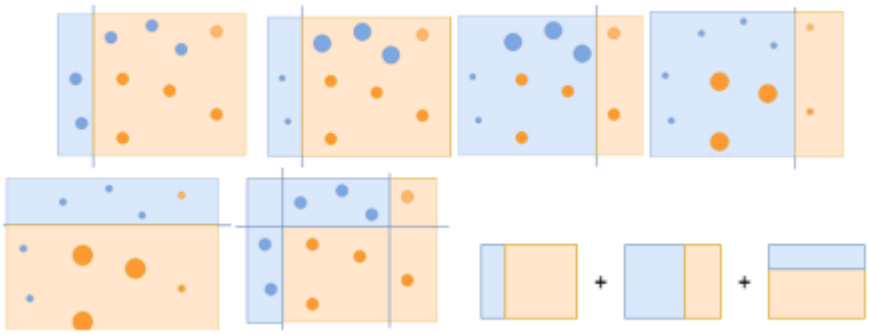


Figura 10 Adaptive Boosting. [22]

Clasificador en cascada

El clasificador en cascada es el proceso de organizar un grupo de características en una forma de categorización jerárquica se conoce como clasificación en cascada. Para establecer si hay o no características faciales en el área de características dada, hay al menos tres clases. Cada sub imagen se clasificará usando una característica en el filtro de clasificación de la primera etapa; si el valor de la característica en el filtro no se

ajusta a los requisitos esperados, será rechazado. Luego, el algoritmo pasa a la siguiente sub ventana y calcula el valor de la característica; si los resultados alcanzan el umbral necesario, pasa a la segunda etapa del filtro, donde el número de sub ventanas que pasan la clasificación disminuye hasta llegar a la imagen del rostro reconocido, como se observa en la figura 11.

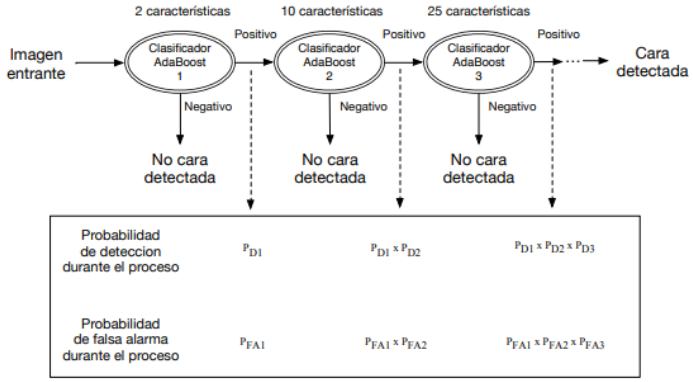


Figura 11 Clasificadores en cascada. [22]

Histograma de Gradientes Orientados (HOG)

El HOG es un descriptor de caracteres que se ha aplicado con éxito a la detección de objetos y patrones, representando un objeto como un solo vector de valor en lugar de una colección de vectores de caracteres, cada uno de los cuales representa una región de la imagen calculada por una ventana deslizante. El descriptor HOG se calcula para cada posición, mientras que la escala de la imagen se ajusta para proporcionar una función HOG como se observa en la figura 12.

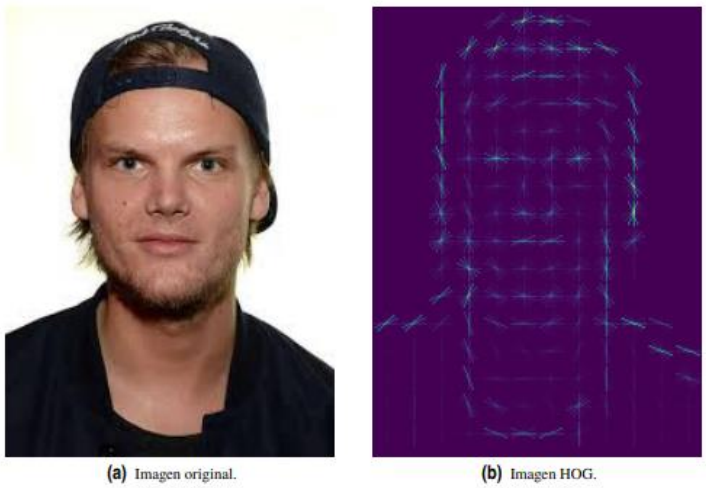


Figura 12 Ejemplo del cálculo de una imagen HOG. [23]

DSP

El término DSP se aplica a cualquier chip que trabaje con señales representadas de forma digital. En la práctica, el término se refiere a microprocesadores específicamente diseñados para realizar procesado digital de señal. Los DSP utilizan arquitecturas especiales para acelerar los cálculos matemáticos intensos implicados en la mayoría de sistemas de procesado de señal en tiempo real. [24]

Procesamiento digital de imágenes

El procesamiento de imágenes digitales es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

Para realizar el reconocimiento facial es necesario realizar un procesamiento de las imágenes para su respectivo análisis. El procesamiento digital de imágenes hace referencia al uso del ordenador con un conjunto de técnicas que se aplican a una imagen digital para mejorar calidad de la imagen e identificar detalles dentro de la misma y obtener un mejor resultado de la imagen obtenida, para lograr esto hay algoritmos que permiten eliminar ruido, mejorar la intensidad, recortar, retocar el contraste, etc. [25]

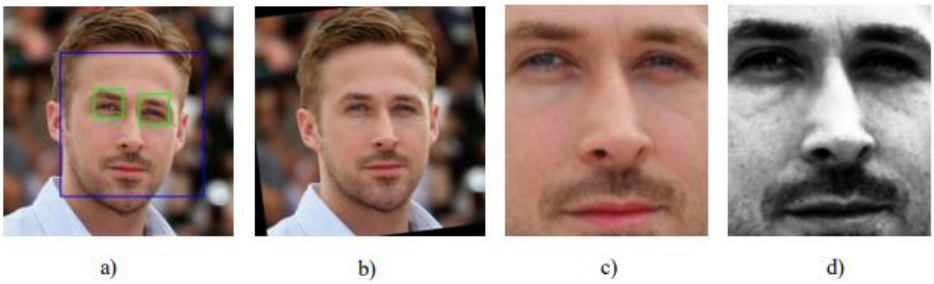


Figura 13 Transformaciones Afines aplicado a una imagen. *[25]*

Generalmente para el procesamiento de imágenes aplicado en el reconocimiento de imágenes se emplean algoritmos basados en transformaciones geométricas como la rotación, traslación, escalado y recorte, un método que es muy empleado son las transformaciones afines como se observa en la figura 13.

Transformación Afín

Una transformación afín es una transformación lineal de una coordenada que abarca las transformaciones básicas de traslación, rotación, escala y deformación o sesgo. La rectitud y el paralelismo se conservan con transformaciones afines, al igual que las proporciones a lo largo de las líneas, como se observa en la figura 14. [26]

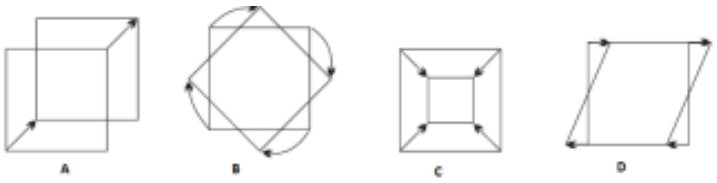


Figura 14 A.Traslación. B: Rotación, C: Escalado y D: Deformación. [26]

Rotación

Una rotación es un tipo de transformación que toma cada punto de una imagen y lo hace girar un cierto número de grados alrededor de un punto dado. Debido a que las caras no siempre se descubren con el mismo grado de inclinación, es típico que las imágenes giren en el reconocimiento facial. Como resultado, los algoritmos de rotación son necesarios para normalizar la posición del rostro en la imagen. [27]

Escalado

Escalado es el proceso de agrandar o achicar una imagen. Hay un sin número de algoritmos que permiten reducir o aumentar el tamaño de una imagen, así como lo haría un zoom.

Deformación e inclinación

La deformación e inclinación es el proceso de transformar una región rectangular en un rombo. La deformación e inclinación es empleada cuando un rostro no está enfocado o no está de frente a la cámara, y esto hace que se tenga un cambio de perspectiva.[27]

Recorte

Recortar es una técnica para extraer solo la parte de una imagen que contiene una cara. Dado que una imagen es una matriz numérica, la técnica es sencilla.

En general, una transformación afín está compuesta de transformaciones lineales compuestas con una traslación o desplazamiento que permiten realizan el procesamiento a una imagen original en una imagen procesada como se observa en la figura 15.

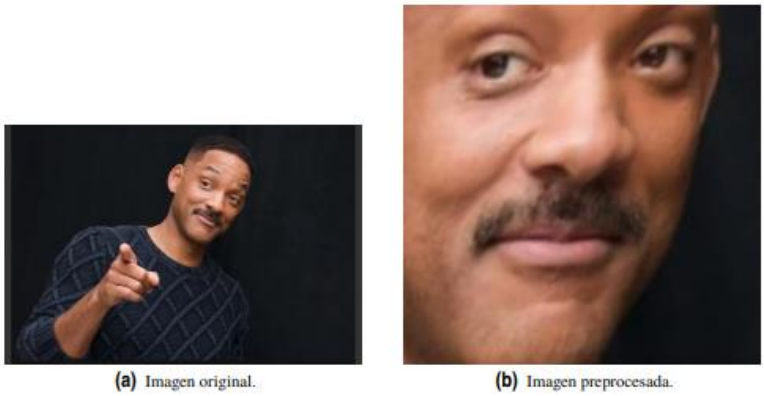


Figura 15 Ejemplo del pre procesamiento de una imagen. [28]

Extracción de características

La extracción de características es el proceso de aplicar algoritmos a fotografías digitales para eliminar la repetición y la información irrelevante. Para reconocer e identificar rostros en fotografías de manera efectiva, se debe obtener un conjunto de

cualidades que los describan y representen. Por lo tanto, es fundamental que esta etapa se lleve a cabo con precisión y siguiendo criterios bien definidos. [29]

Las características geométricas y analíticas o basadas en la apariencia son dos tipos de características que se pueden usar para caracterizar rostros en fotografías.

- Las características geométricas son aquellas que miden la longitud y la ubicación de rasgos particulares de la cara, como los ojos, la nariz y la boca. Se crea un vector de características extrayendo estos componentes o puntos de características faciales. [30]

- Características de la apariencia: representan cambios en la textura del rostro en función de las arrugas, las regiones alrededor de la boca, los ojos y otras cualidades globales del rostro humano. Por lo general, estos enfoques holísticos codifican la matriz de intensidad de píxeles sin depender de rasgos faciales específicos.

Comparación y Clasificación

Hasta ahora se han descrito el proceso de los algoritmos que permiten obtener una representación fiel del rostro que aparece en una fotografía, pero falta la última fase se encarga de realizar le evaluación de la representación mediante una comparación con otra base de datos y poder lograr la clasificación de rostros. A continuación, se muestran algunas de las técnicas más aplicadas para los algoritmos explicados anteriormente. [31]

Distancia euclídea

La distancia euclidiana es la distancia directa entre dos puntos en un plano se define como esta distancia euclidiana. Está basado en el Teorema de Pitágoras como se observa en la ecuación 1, puede generalizarse a un espacio N-dimensional $X = [x_1, x_2, \dots\dots, x_N]$e $Y = [y_1, y_2, \dots\dots, y_N]$.

$$d(X,Y) = \sqrt{\sum_{i=1}^{N}(y_i - x_i)^2} = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + \cdots . + (y_N - x_N)^2} \qquad \textit{Ecuación 1}$$

Máquinas de soporte vectorial

Las máquinas de soporte vectorial (Support Vector Machine) son un método de aprendizaje supervisado que genera funciones de clasificación a partir de un conjunto de datos de entrenamiento etiquetados. [32]

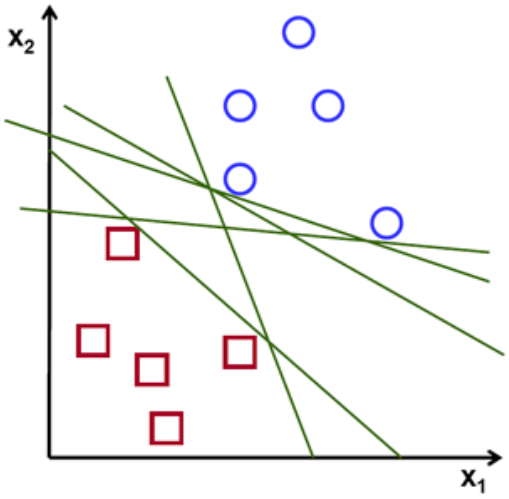


Figura 16 Máquinas de soporte vectorial. [32]

Existen numerosos hiperplanos potenciales que se pueden utilizar para dividir los dos grupos de puntos de datos. El margen máximo, o la mayor distancia entre puntos de datos para ambas clases, es lo que buscamos en un plano. Maximizar la distancia del margen agrega algo de soporte, lo que aumenta la confianza con la que se pueden clasificar los puntos de datos futuros, como se observa en la figura 16. [33]

Reconocimiento de voz

El reconocimiento de voz es la capacidad de una máquina o programa para identificar palabras y frases en lenguaje hablado y convertirlas a un formato legible por máquina. El objetivo es crear un archivo de datos a partir de una señal auditiva generada por un altavoz como se observa en la figura 17. Los usos son numerosos, incluida la capacidad de escribir documentos sin usar el teclado. [34]

Figura 17 Reconocimiento de voz a texto. [34]

La señal que se adquirió y transformó en una señal eléctrica se puede presentar usando un micrófono conectado al equipo de visualización. Cabe señalar que la gráfica de esta señal contiene mucha información inútil que crearía ineficiencias en el sistema a la hora de realizar el reconocimiento de voz si no se hubiera tratado con segmentación o un filtro de eliminación de ruido. [35]

Voz

La voz es un sonido rígido y voluntario producido por una persona cuando el aire que se encuentra en los pulmones, esta pasa por las cuerdas vocales de la garganta generando así vibraciones. Es uno de los métodos de comunicación que utilizan los seres humanos a través de sonidos, y lo emplean para hablar, reír cantar o gritar. La voz humana tiene diferentes tonos, esto es debido a que cada persona tiene diferentes tipos de aparato fonador. [36]

La voz puede ser generado por un hombre o por una mujer, el hombre posee un tono de voz más grave ya que posee cuerdas vocales más gruesas con una longitud de 17 y 25mm, a diferencia de la mujer que suele tener un tono más fino ya que su longitud va de 12.5 y 17mm.

Frecuencia

La frecuencia es la cantidad de oscilaciones por unidad de tiempo, generalmente es medido en segundo, que realiza una onda en movimiento como se observa en la figura 18. Su unidad de medida se realiza en hercios (Hz) y nos permite identificar si los sonidos son graves o agudos, a mayor frecuencia el tono del sonido es más agudo, y a menor frecuencia el tono grave. El oído humano es capaz de oír sonido que se encuentre entre los 20Hz y los 20000Hz. [37]

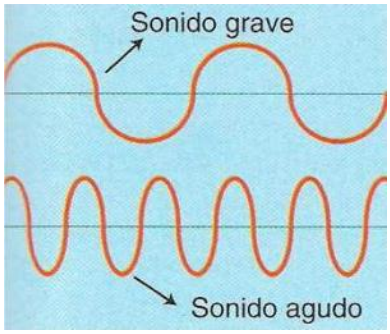


Figura 18 Frecuencia del sonido. [37]

Amplitud

La amplitud es el encargado de indicar la magnitud de las variaciones de la presión y permite identificar entre los sonidos fuertes y débiles. A mayor amplitud el sonido es más fuerte y a menor amplitud el sonido es más débil como se observa en la figura 19. La amplitud se mide en decibeles (dB), por ejemplo, el oído humano puede escuchar entre 10 y 120 Db denominada umbral de audibilidad, si se supera los 120 Db pasa a la zona de umbral de dolor causando una sensación dolorosa en el oído. [37]

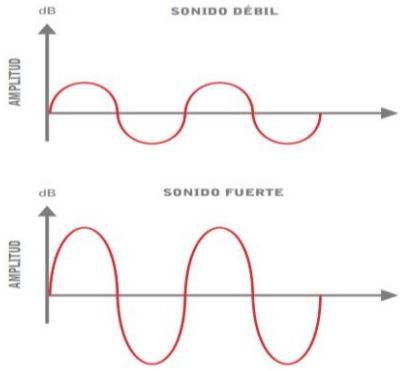


Figura 19 Amplitud. [37]

Longitud de Onda

La longitud de onda es la distancia que recorre una perturbación periódica que se propaga por un medio en un ciclo. La longitud de onda es unos de los parámetros que se emplea para definir una onda. Se la define con la letra griega λ (lambda). La longitud de onda que el oído humano puede escuchar va desde menos de 2cm hasta aproximadamente 17 metros como se observa en la figura 20. [37]

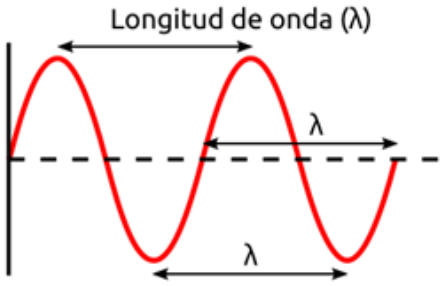


Figura 20 Longitud de onda. [37]

Timbre

El timbre es la característica que permite distinguir entre dos o más sonidos con la misma frecuencia y amplitud, pero producidos por distintas fuentes sonoras, como la voz o los instrumentos musicales. Un sonido se compone de varias frecuencias que son múltiplos de su frecuencia principal, que se denominan armónicos. La cantidad de intensidad de armónicos en diferentes sonidos que permite distinguir entre diferentes fuentes de sonido. [37]

Extracción de características

La extracción de características implica calcular una serie de vectores de características que ofrecen una representación compacta de la señal de voz proporcionada, lo que permite un procesamiento futuro.

Es fundamental extraer las principales propiedades de la señal de voz para realizar una identificación correcta. [38]

MFCC Coeficientes Cepstrales de Escala de Mel

Coeficientes Cepstrales de Escala de Mel (MFCC) es la técnica de extracción de características más ampliamente utilizada en el reconocimiento de voz porque describen de forma compacta la amplitud del espectro del habla. El procedimiento para

desarrollar un vector característico de MFCC se observa en la figura 21. La señal se enfatiza previamente antes de separarla en fotogramas y se utiliza una función de ventaneo. El ventaneo es una técnica para eliminar los bordes de la señal y enfatizar la parte central del marco para su examen. [39]

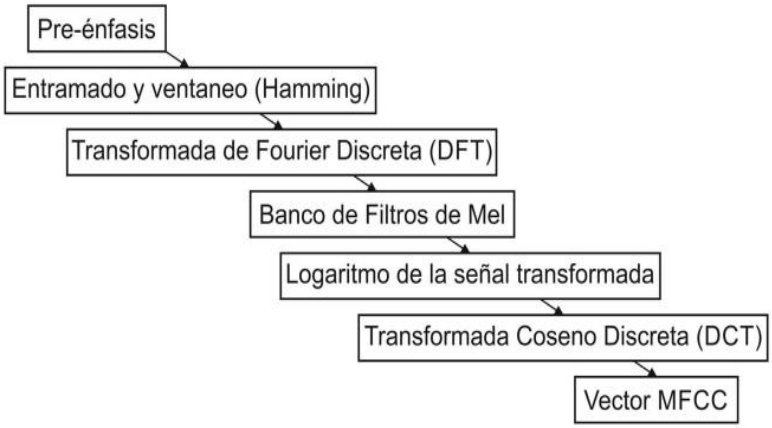


Figura 21 Proceso de obtención de los coeficientes MFCC. [39]

La amplitud del espectro se utiliza para obtener la Transformada Discreta de Fourier (DFT) de cada cuadro, y esta información se transmite al dominio Mel a través del Banco de Filtros. La escala Mel se basa en un mapeo entre la frecuencia actual y el tono percibido por un oyente humano simulado; es lineal por debajo de 1 kHz y logarítmica por encima de este umbral como se observa en la figura 22. A continuación, se calcula el logaritmo de la señal y, por último, se utiliza la transformada discreta del coseno (DCT) para extraer el número deseado de coeficientes por fotograma del vector resultante.

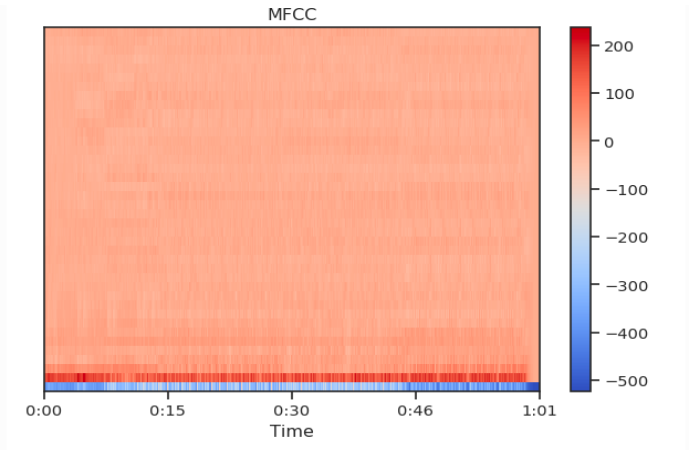


Figura 22 Coeficientes cepstrales de escala de Mel. [39]

A continuación, se describe cada etapa del procesamiento MFCC como se observa en la figura 23:

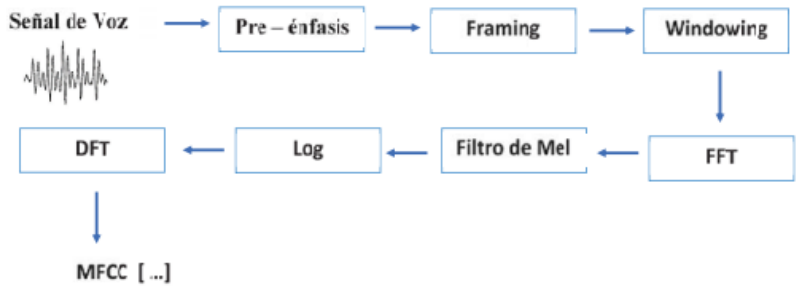


Figura 23 Procesamiento de audio mediante MFCC. *[39]*

- **Preénfasis (pre-énfasis):** La fase de preénfasis consiste en pasar la señal a través de un filtro que enfatiza las frecuencias altas, lo que permite extraer más información de la señal (las frecuencias altas poseen más información dependiendo del hablante) y equilibrar el espectro de frecuencias. [39]

  Los componentes de alta frecuencia en las señales de voz tienen un bajo nivel de energía. El preénfasis es una técnica para aumentar la energía de los elementos de alta frecuencia. En la siguiente ecuación 2 se puede utilizar para aplicar el filtro de preénfasis en una señal X:

  $$Y(t) = X(t) - \alpha X(t-1)$$ ***Ecuación 2***

  Donde:

  - Y(t): Representa la salida del filtro en el dominio del tiempo.
  - X(t): Representa la forma de onda del habla en función de tiempo.
  - α: Toma valores en el intervalo [0.95, 0.97]

- **Framing:** El Framing es el proceso de dividir una señal en muchos marcos de pequeños intervalos de tiempo. Los anchos de fotograma oscilan entre 20 y 30 milisegundos, con una superposición del 40 al 60 por ciento. Como se observa en la figura 24 la señal de sonido se ha separado en muchos cuadros, con la superposición (en rojo). [39]

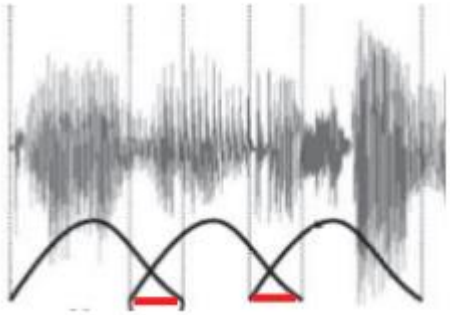


Figura 24 Señal de sonido que se ha separado en muchos cuadros. *[39]*

- **Windowing:** El Windowing es el proceso de "enventanar" la señal, segmentándola en tramas consecutivas de 20ms de duración, con un solapamiento de 10 ms por cada segmento, para asegurar la continuidad de la información de la señal, de tal manera de obtener información distintiva entre los diferentes tipos de sonidos producidos por la voz, mediante el posterior análisis espectral de la señal. Esencialmente, la función reduce la parte de la señal de voz a cero al principio y al final de cada cuadro como se observa en la figura 25.

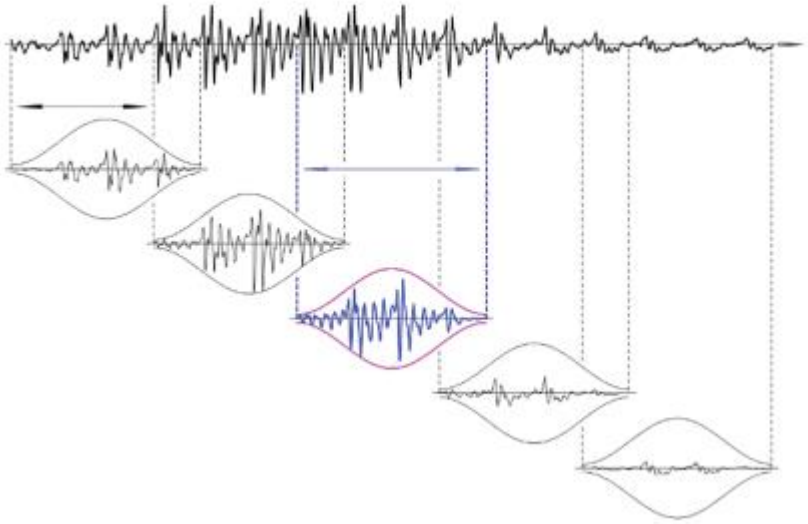


Figura 25 Proceso Windowing del MFCC. [39]

La función enventanar está representada por la ecuación 3:

$$W[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) & \text{donde } 0 \leq n \leq N-1 \\ 0 & \text{otro caso} \end{cases}$$

Ecuación 3

Donde:

- W (n) = enventanar
- N = número de muestras en cada frame

La señal de salida Y es idéntica a la señal de entrada X multiplicada por la función Windowing para cada fotograma como se observa en la ecuación 4.

$$Y(n) = X(n) * W(n)$$

Ecuación 4

- **FFT:** La transformada rápida de Fourier es un algoritmo que calcula la transformada discreta de Fourier de una secuencia, o su inversa. Este método se encarga de quitar la magnitud de la frecuencia de cada cuadro como se observa en la ecuación 5.

$$X[K] = \sum_{n=0}^{N-1} x[n] e^{-\frac{2\pi}{N}kn} \qquad k = 0,1,\ldots . N-1$$

Ecuación 5

Donde:

- X[k]: Es un número complejo que indica la magnitud y la fase de ese componente de frecuencia en la señal original.
- x [n] = Señal Hamming Windowing

- **Procesamiento Mel - banco de filtros:** Un banco de filtros (o filterbank) es una matriz de filtros de paso de banda que separa la señal de entrada en múltiples componentes, cada uno de los cuales transporta una única subbanda de frecuencia de la señal original.El objetivo de esta fase es duplicar la magnitud de la frecuencia obtenida en la fase anterior utilizando un conjunto

de filtros de 20 - 40 Mel, con cada filtro de salida igual al total de los componentes filtrados utilizando la ecuación 6.

$$m = 2595 * \log 10(1 + \frac{f}{700}) \qquad \textit{Ecuación 6}$$

Donde:

- f: representa la frecuencia en Hz.
- m: Banco de filtros en mel.

- **Log:** El logaritmo es una función monótona estrictamente cóncava (creciente) comprendida en el conjunto de los números reales positivos y es la inversa de la función exponencial**.** El propósito de este punto es calcular el logaritmo de la magnitud de la frecuencia. Esta fase reduce la sensibilidad de las estimaciones de frecuencia a fluctuaciones menores de la señal. Por ejemplo, la diferencia en la potencia de audio provocada por la distancia entre el altavoz y el micrófono.
- **DCT :** La transformada de coseno discreta (DCT del inglés Discrete Cosine Transform) es una transformada basada en la Transformada de Fourier discreta, pero utilizando únicamente números reales .Esta fase permite convertir la señal del dominio de la frecuencia al dominio del tiempo utilizando la DCT (Transformada de coseno discreta). MFCC es el nombre que se le da al resultado de esta conversión, y el conjunto de coeficientes MFCC se conoce como vector acústico. Para realizar este procese se utiliza la ecuación 7 :

$$y_t[k] = \sum_{m=1}^{M} \log(|Y_t(m)|^2) \cos[k * (m - 0.5) * \frac{\pi}{M}] \qquad \textit{Ecuación 7}$$

$$K = 1,2, \dots, M$$

Donde:

- $y_t[k]$ = Conjunto de coeficientes de MFCC
- M = número de filtros de Mel o dimensión del vector $y_t[k]$
- m: Banco de filtros en mel.

Inteligencia artificial

La inteligencia artificial es la ciencia que tiene como objetivo el diseño y construcción de máquinas con la capacidad de imitar el comportamiento inteligente de las personas cómo se observa en la figura 26. Estas máquinas utilizan algoritmos para aprender mediante datos y usar lo aprendido en la toma de decisiones tal y como lo haría un ser humano, a diferencia de los humanos estas máquinas basadas en Inteligencia artificial no necesitan descansar y pueden procesar grandes volúmenes de información. [40]

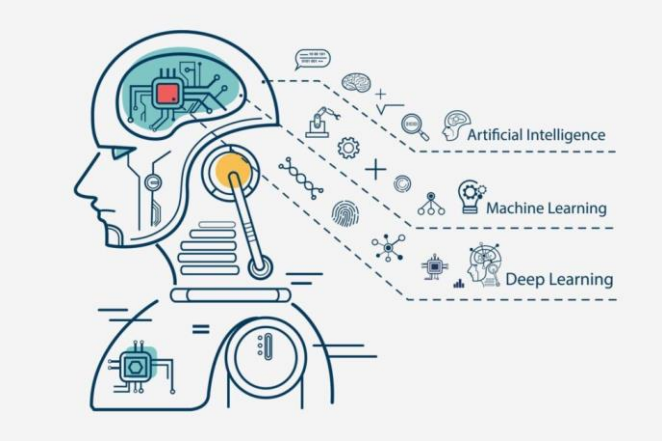


Figura 26 Aplicaciones de la inteligencia artificial. [40]

La inteligencia artificial es muy utilizada para facilitar y ayudar a las personas en todos los ámbitos de la vida, algunas de las aplicaciones que están creciendo en los últimos años son: [41]

- Reconocimiento, clasificación y etiquetado de imágenes
- Mejoras en el rendimiento de la estrategia comercial algorítmica
- Procesamiento de datos de pacientes escalable y efectivo
- Mantenimiento predictivo
- Detección y clasificación de objetos
- Distribución de contenido en redes sociales

- Defensa contra amenazas de seguridad cibernética

Aprendizaje automático

Aprendizaje automático es una rama de la inteligencia a artificial que significa aprendizaje automático, se encarga de generar algoritmos con la capacidad de aprender y no tener que ser programados de manera explícita. Los algoritmos de aprendizaje automático utilizan método de cálculo para procesar datos sin la necesidad de una ecuación específica, ya que la característica principal de sus algoritmos es extraer de manera autónoma información precisa y relevante de los datos que se están procesando.

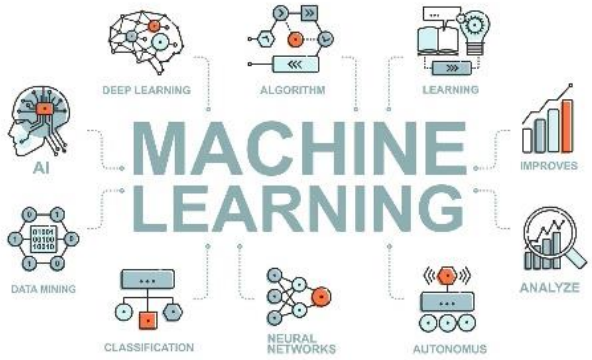


Figura 27 Aprendizaje automático. [42]

Un programa no tiene la necesidad de programar algoritmo por horas tomando en cuenta los posibles escenarios ni todas las excepciones cómo se observa en la figura 27, lo único que realiza es alimentar al algoritmo con volúmenes inmensos de datos para que el algoritmo aprenda solo y sepa tomar decisiones como un ser humano, mientras más datos procese un algoritmo esta tiende a mejorar su rendimiento y su precisión. [42]

Los tipos de Aprendizaje automático son: Aprendizaje supervisado y Aprendizaje no supervisado.

Aprendizaje supervisado

El aprendizaje supervisado es cuando un algoritmo cuenta con las preguntas (características) y las respuestas (etiquetas) para poder identificar patrones de datos, aprender de las observaciones y realiza predicciones, las predicciones realizadas son corregidas por el operador y este proceso continuo hasta que el algoritmo alcanza un alto nivel de precisión y rendimiento. [43]

Aprendizaje no supervisado

El aprendizaje no supervisado es cuando el algoritmo recibe solo las características más no las etiquetas, y el algoritmo interpreta estos datos para poderlo organizarlo en grupos con las características iniciales. Su capacidad para tomar decisiones mejora gradualmente y se vuelve más refinada mientras más datos evalúan. [43]

Redes neuronales artificiales

Una red neuronal artificial (ANN) es una rama de la inteligencia artificial basada en la estructura del sistema nervioso del ser humano. La característica de una red artificial es la capacidad de aprender a realizar tareas humanas, a partir de un conjunto de patrones de entrenamiento de aprendizajes supervisados o no supervisados. [44]

Una red neuronal está conformada por un conjunto de neuronas artificiales, es decir, está formada por un conjunto de funciones $\{f^{(1)}, \ldots \ldots, f^{(k)}\}$, que están conectados entre sí con cada una de sus salidas que conducen a las entradas del otro. Las redes neuronales artificiales no son más que un conjunto funciones, representadas mediante la composición de varias funciones $f(x) = f^{(k)}(\ldots(f^{(1)}(x)))$ cómo se observa en la figura 28. [45]

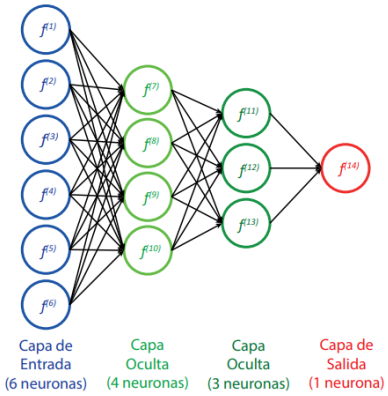


Figura 28 Esquema de una red neuronal artificial. [46]

La capa de la izquierda se la conoce como capa de entrada, y las neuronas dentro de ella se denominan neuronas de entrada. La o las capas ocultas son las capas intermedias y tienen interconexiones que se encargan de procesar la información hasta enviarlas a la capa de salidas. Finalmente, el resultado del proceso de la información se ve en la capa de salida, el cual se encarga de representar las probabilidades de cada clase. [46]

Perceptrón

El perceptrón es una red neuronal antigua desarrolla entre 1950 y 1960, esta red neuronal tiene la capacidad de aprender a reconocer patrones muy sencillos y poder realizar una clasificación binaria. [47]

El perceptrón es capaz de tomar uno o más valores binarios de entrada y obtener como resultado una sola salida binaria cómo se observa en la figura 29. Matemáticamente se define con la siguiente ecuación 8 para poder comprobar su resultado:

$$z(x) = \sum_{j=1}^{n} (w_j x_j) + b \qquad \textit{Ecuación 8}$$

$$a(z) = \begin{cases} 0 & z \leq 0 \\ 1 & \text{c. c.} \end{cases}$$

Donde:

- a(z): Representa la función de activación.
- x_j: Son componentes vectoriales de entradas {x1, … … . , xn}.
- w_j : Son los pesos relativos de las entradas.

- b: Valor del sesgo.
- z: Representa la función escalonada que, si su entrada es mayor que 0, devuelve 1; de lo contrario, devuelve 0.

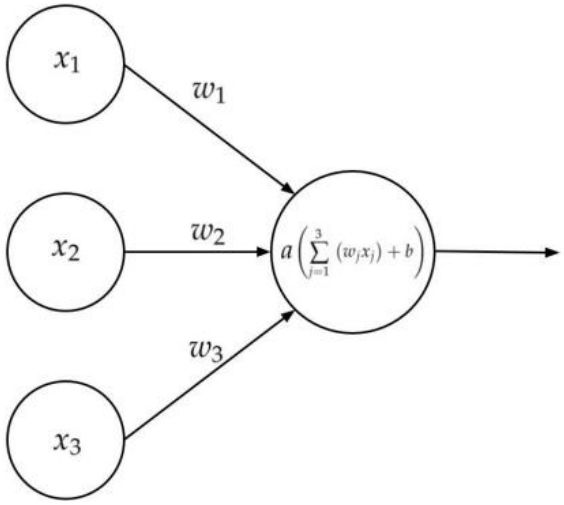


Figura 29 Diagrama de perceptrón con tres entradas (x1, x2, x3) y una única salida. [47]

El valor de variable del sesgo, se encarga de regular que tan fácil es obtener un valor de 1 del perceptrón. Será fácil devolver el valor 1 para un perceptrón con un valor de sesgo muy grande, y siempre obtendremos el número 0 como resultado de salida si el sesgo es una cantidad muy negativa.

Redes Feedforward

Las redes neuronales feedforward (FANN) es una red neuronal artificial donde las conexiones entre las unidades no forman un ciclo. Son redes de clase ANN más investigadas por el campo científico para la aplicación en muchos campos. Debido a que se construye ensamblando funciones, esta forma de algoritmo se denomina "red" (o perceptrones). La frase "feedforward" deriva del idioma inglés y alude al hecho de que la información siempre fluye hacia adelante, de una neurona a la siguiente, sin conexiones entre la salida y la entrada de la misma neurona. El modelo final se representa como un gráfico acíclico, que muestra cómo se vinculan las funciones y cómo fluye la información. [48]

Arquitectura de una red FANN

La arquitectura de este tipo de modelo se puede separar en tres elementos principales, conocidos como capas. Debido a que cada neurona en una capa está conectada a todas las neuronas en la siguiente capa, pero nunca a las neuronas en la misma capa, esta forma de red se conoce como completamente conectada. [48]

Las capas se componen en:

- Capa de entrada
- Capa oculta
- Capa de salida

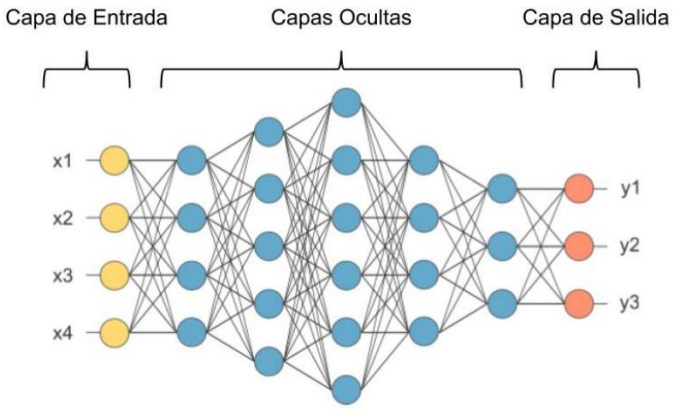


Figura 30 Modelo de una red neuronal artificial feedforward. [48]

Las diferentes capas que componen una red neuronal son tres como se puede observar en la figura 30. En primer lugar, está la capa de entrada, que, como su nombre indica, especifica cuántos valores de entrada aceptará el modelo antes de transferirse a la primera capa oculta. Las neuronas de la primera capa oculta reciben los mismos cuatro valores de entrada, los procesan y luego pasan el resultado a las neuronas de la siguiente capa, quienes, a su vez, se encargan de procesar los datos recibidos y son capaces de hacer más complejos y decisiones abstractas que las neuronas de la primera capa. Finalmente, la capa de salida recibe el resultado de la última capa oculta; las neuronas de esta capa a menudo no tienen una función de activación. Debido a que el resultado de esta capa se usa para representar las probabilidades de cada clase (en un problema de clasificación) o cualquier forma de valor real. [48]

Funciones de activación

Una función de activación es, por tanto, una función que transmite la información generada por la combinación lineal de los pesos y las entradas, es decir son la manera de transmitir la información por las conexiones de salida.

La función de activación determina, como su propio nombre indica, el nivel de activación que alcanza cada neurona una vez que ha recibido los impulsos aferentes. Estas funciones ostentan un rol muy importante en la determinación del poder computacional de la red neuronal. Como resultado, existen numerosos tipos de funciones de activación, sin embargo, algunas funcionan mejor que otras en la práctica. Para ello existen un conjunto de funciones de activación que aportan una serie de no linealidades a la suma de salida de la neurona como se observa en la figura 31. [49]

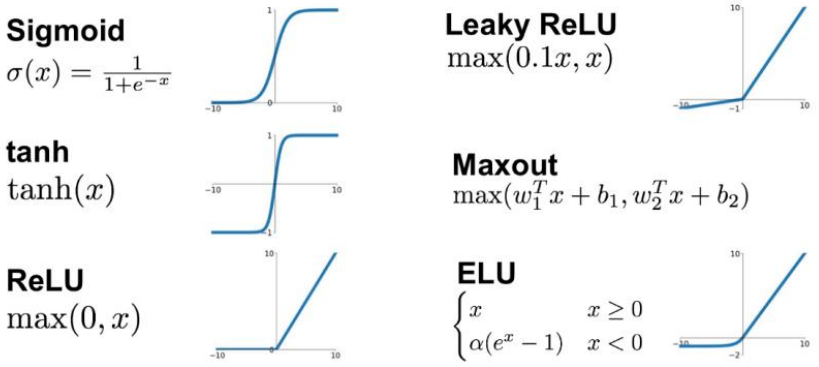


Figura 31 Funciones de activación. [49]

A continuación, se resumen las diferentes funciones de transferencia cómo se observa en la tabla 2 que se pueden emplear para diversas aplicaciones.

Tabla 2 Resumen de funciones de transferencia. [49]

Función de activación	Ecuación
Función escalón	$\emptyset(x) = \begin{cases} 0 \text{ si } x < 0 \\ 1 \text{ si } x \leq 0 \end{cases}$
Función Sigmoidal	$\emptyset(x) = \frac{1}{1+e^{-x}}$
Función Rectificadora	$\emptyset(x) = \max\{0, x\}, \text{siendo } x \geq 0$

Función Tangente Hiperbólica	$\emptyset(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$
Función de Base Radial	Gausianas, multicuadráticas, multicuadráticas inversas

Regla de aprendizaje

La regla de aprendizaje establece el método a través del cual la red neuronal ajusta los pesos de sus conexiones en respuesta a los datos entrantes. Según su regla de aprendizaje, las redes neuronales se pueden clasificar de dos maneras. Primero, dependiendo de si se requiere o no un agente externo para regular el proceso, una red neuronal puede exhibir un aprendizaje supervisado o no supervisado. Por otro lado, el aprendizaje será Offline u Online según se distinga una fase de aprendizaje y una fase de funcionamiento. [48]

El Perceptrón Multicapa

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

El Perceptrón simple es incapaz de crear categorías que no estén linealmente separadas, ya que solo puede distinguir patrones que pueden estar separados por un hiperplano (una línea recta en el caso de dos neuronas de entrada). La incorporación de capas ocultas, que da como resultado una red neuronal es conocida como perceptrón multicapa (MLP) como se observa en la figura 32, es una técnica para superar las limitaciones del perceptrón simple.

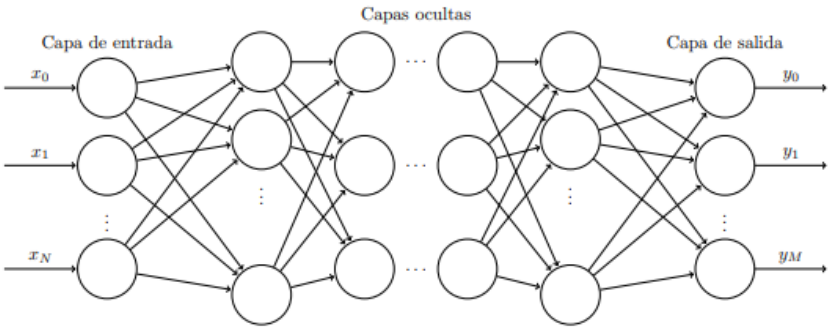


Figura 32 Estructura de un Perceptrón Multicapa. [30]

El MLP tiene tres tipos de capas: una capa de entrada que recibe parámetros en la red, capas ocultas que ejecutan el procesamiento de información y capas de salida que transmiten el resultado del sistema. [30]

Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) son una especie de red neuronal artificial que puede aprender a distinguir entre características en un conjunto de datos mediante el cálculo de convoluciones. Como resultado, esta forma de red se emplea comúnmente para reconocer objetos en fotografías. [28]

Las imágenes no son percibidas de la misma manera por una computadora y un ser humano, pero podemos afirmar que este tipo de red neuronal funciona de manera similar, porque la computadora ve las imágenes como conjuntos de matrices bidimensionales con valores relacionados con la imagen en cada punto, es decir, píxeles. [50]

Las redes neuronales convolucionales (CNN) son un tipo de red con una arquitectura específica que sobresale en el reconocimiento de patrones de imagen, lo logran mediante el empleo de dos nuevos tipos de capas conocidas como convolución y pooling, que realizan procedimientos para la extracción de propiedades geométricas de imágenes. Así, la red detecta formas simples como curvas y aristas en las primeras capas ocultas, que se propagan a lo largo de las siguientes capas de convolución, donde se jerarquiza el aprendizaje desde los elementos más simples a los más complejos, hasta llegar finalmente a las capas densas, donde se hará la clasificación. [30]

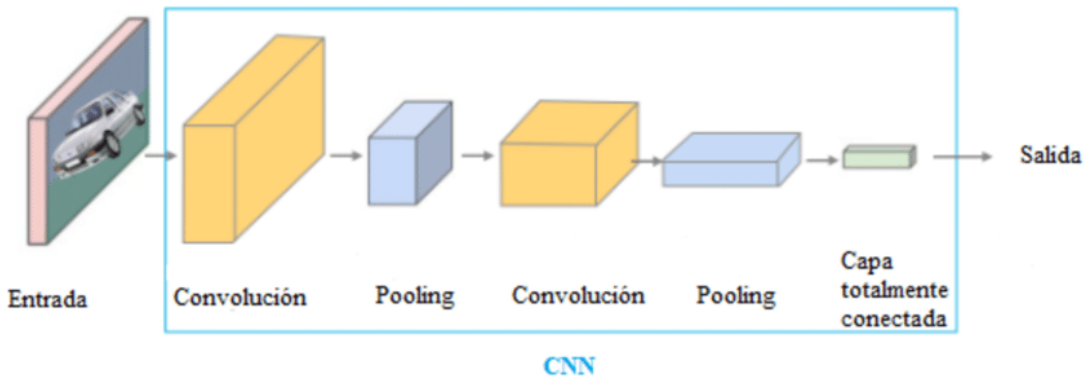


Figura 33 Diagrama de una red neuronal convolucional. [30]

Los datos recopilados de la capa de entrada se transforman en un conjunto de valores determinados por capas ocultas totalmente conectadas cuando se utiliza una CNN. En este enfoque, la estructura más básica de una CNN se divide en tres capas: una capa de entrada, una capa de extracción de características y una capa de clasificación como se observa en la figura 33.

Capa de entrada

La capa de entrada es responsable de recibir datos desde fuera de la CNN. Si los datos contienen imágenes, las entradas serán tridimensionales, con canales de ancho, alto y color, como los tres valores RGB de cada píxel.

Capa convolucional

La capa convolucional es alterar los datos entrantes utilizando una colección de neuronas que están conectadas localmente desde la capa anterior. Esta capa calculará el producto escalar de una región específica de la capa de entrada y los pesos que la conectan con la capa de salida. En la mayoría de los casos, este proceso mantiene las mismas dimensiones espaciales.

La convolución, un procedimiento realizado en este tipo de capa, se utiliza para detectar las propiedades de CNN. Esto puede tomar los datos de entrada sin procesar o una salida de una convolución anterior como entrada. Con frecuencia, esta operación se malinterpreta como filtrado de datos de entrada, y el kernel a cargo del filtrado generalmente corresponde al conjunto de pesos de la capa convolucional. [30]

Como se observa en la figura 34 el kernel se mueve durante el curso de ciertos datos de entrada. En cada paso, el filtro se multiplica por los valores de los datos de entrada, lo que da como resultado una nueva característica de salida.

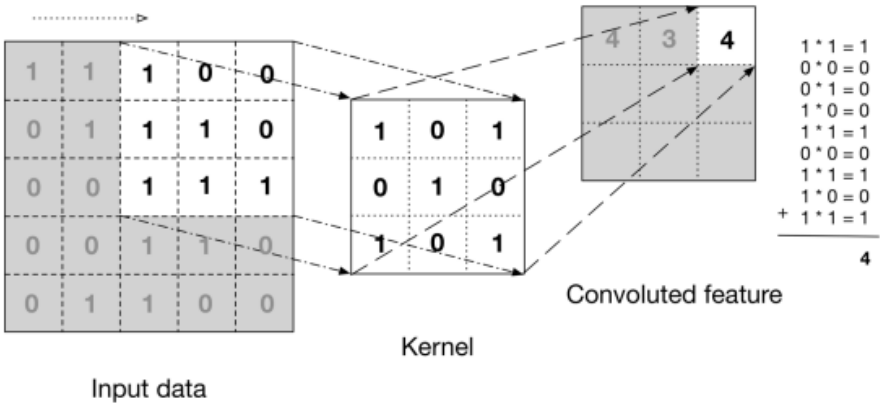


Figura 34 Representación de la operación de convolución con un kernel. [30]

Debido a que el proceso de convolución reduce el tamaño de la matriz resultante, puede seleccionar si desea que tenga el mismo tamaño que la matriz de entrada o no. Se agregan filas y columnas auxiliares de valor cero para conservar el mismo tamaño. Esto no siempre es aceptable porque evita reducir la cantidad de parámetros y, al mismo tiempo, hay más datos para buscar patrones. [30]

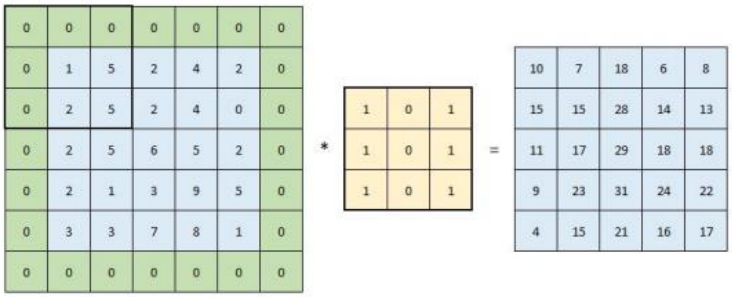


Figura 35 : Proceso de convolución con padding. [30]

El tamaño del mapa de características se reduce como resultado de este proceso en comparación con la matriz de datos original, como se observa en la figura 35. El padding, que consiste en agregar ceros alrededor de los bordes de la matriz de entrada, se usa para evitar esto. Además del relleno, stride es un parámetro que especifica cuántas celdas desplaza el kernel para cada paso de operación en la matriz de entrada.

La siguiente ecuación 10 se puede utilizar para calcular el tamaño del mapa de características de salida:

$$F = [\frac{N + 2P - F}{S} + 1] \qquad \textit{Ecuación 10}$$

Donde:

- F = Tamaño de la matriz de salida.
- N = Tamaño de la matriz de entrada.
- P = Parámetro de padding, por lo general equivale a 0.
- S = Parámetro de stride, por lo general equivale a 1.

Capa de pooling

La capa pooling se coloca generalmente después de la capa convolucional. Para evitar dificultades durante la fase de entrenamiento, como el overfitting, la capa de pooling se utiliza para simplificar aquellos valores dentro del mapa de características que son semánticamente cercanos. Debido a que la reducción de tamaño implica la pérdida de información, esta técnica a menudo se conoce como reducción de muestreo.

La función más utilizada es Max-pooling, que devuelve una matriz que contiene los valores más altos del mapa de características y define una submatriz de tamaño MxM con stride M sobre el mapa de características. Otra función utilizada es la Average Poooling que se encarga de calcular el valor medio de cada subconjunto de la matriz, como se observa en la figura 36. [51]

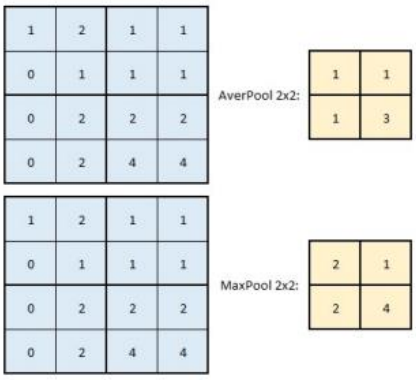


Figura 36 AverPooling y MaxPooling 2x2. [51]

Las capas de pooling se colocan entre los límites de convolución para reducir la dimensionalidad de los datos que pasan a través de ellos. Para hacer esto, las capas de pooling comúnmente emplean un filtro de 2x2. Este filtro se aplica a los datos, lo que da como resultado el reemplazo de los cuatro píxeles filtrados con un valor único que corresponde al valor máximo de los píxeles filtrados, como se observa en la figura 37.

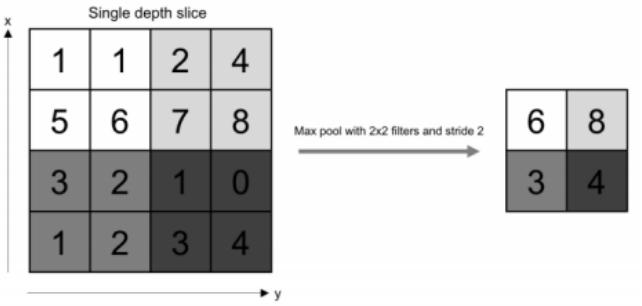


Figura 37 Ejemplo de una operación de Max-pooling con una matriz de tamaño 2x2. [51]

Capa de clasificación

La capa de clasificación es la capa más reciente de la CNN, aunque puede haber más de una de este tipo en algunos casos. Este tipo de capa está completamente conectado a su capa asistente, y su objetivo es tomar las características finales calculadas por el resto de la red y producir estimaciones o probabilidades que correspondan a las clases. Los datos de estas capas se representarán como un vector con tantos elementos como clases se hayan analizado, con dimensiones mucho más pequeñas. [51]

Asistente de Mensajería Telegram

Telegram es un servicio de chat extremadamente rápido, fácil de usar y gratuito, Telegram se centra en la velocidad y la seguridad. Telegram se puede usar simultáneamente en todos los dispositivos. Cualquier teléfono, tabletas o computadoras de escritorio pueden acceder a sus mensajes sin ningún problema. Tiene la capacidad transmitir mensajes, imágenes, videos y archivos de cualquier tipo con Telegram (doc, zip, mp3, etc.).

La aplicación ofrece una serie de funciones, incluido un bot estándar que se comunica con los usuarios a través de comandos en mensajes privados y puede ser controlado mediante Python.

1.3 Objetivos

1.3.1 Objetivo general

- Implementar un control de acceso por medio de algoritmos de procesamiento de señales para el reconocimiento facial y de voz empleando redes neuronales.

1.3.2 Objetivos específicos

- Analizar los sistemas de control de acceso tradicionales más implementados en una vivienda.
- Analizar un algoritmo de procesamiento de señales empleando reconocimiento facial y voz que utilicen redes neuronales.
- Diseñar un sistema electrónico portable de para el control de acceso de personas.

CAPÍTULO II

METODOLOGÍA

2.1 Materiales

Para la realización del presente proyecto de investigación, se utilizaron materiales como artículos, revistas, tesis, cursos, internet, fichas técnicas y además se implementó utilizando el lenguaje de programación adecuado para el proyecto.

2.2 Métodos

2.2.1 Modalidad de investigación

Investigación Aplicada

Para el presente proyecto se empleó la investigación aplicada, porque el objetivo principal es poner en práctica conocimientos de, DSP, Comunicación Avanzadas, Electrónica Digital y Programación, adquiridos durante la formación académica.

Investigación Bibliográfica

La revisión bibliográfica se realizó en libros, artículos científicos y proyectos de investigación previos de los diferentes repositorios de las Universidades del país y del extranjero, relacionados con el tema de investigación.

Investigación Experimental

Se realizó diversas pruebas de funcionamiento del algoritmo de reconocimiento facial y voz para la implementación, estos algoritmos deben ser medidos y probados de manera correcta para poder obtener una efectividad alta en el reconocimiento facial y voz.

Investigación de campo

Para el presente proyecto de investigación se utilizó la investigación de campo puesto que se pretende implementar el prototipo en una vivienda que permitan recolectar la mayor información posible para obtener los mejores resultados.

2.2.3 Recolección de información

Para la recolección de la información en el presente trabajo de titulación se revisó artículos científicos de revistas indexadas, libros, artículos académicos y proyectos de investigación de las bases de datos de repositorios de las Universidades del país relacionados al estudio e implementación de algoritmos de reconocimiento facial y de voz para un control de acceso.

2.2.4 Procesamiento y Análisis de Datos

Para el procesamiento y análisis de datos se realizó los siguientes pasos:

- Análisis y revisión de la información recolectada.
- Tomar decisiones sobre cómo usar la información.
- Interpretación de la información.
- Mejoras y anexos en los datos e información
- Pruebas piloto.
- Verificación y control de errores
- Interpretación de los resultados

2.2.4 Desarrollo del Proyecto

Para cumplir con los objetivos planteados en el proyecto de investigación se llevó a cabo los siguientes pasos:

a) Identificación de sistemas de seguridad implementados en los inmuebles.
b) Identificación de los problemas de los sistemas de seguridad implementados en los inmuebles.
c) Identificación de los métodos y algoritmos de reconocimiento facial y de voz que empleen redes neuronales.
d) Análisis las aplicaciones y sistemas de los métodos de reconocimiento facial y de voz que empleen redes neuronales.
e) Elaboración del script que permita detención de rostros y de voz utilizando software libre.
f) Implementación código final en una tarjeta de desarrollo.
g) Diseño de Placas PCB del prototipo.

h) Ensamblaje del prototipo.

i) Pruebas del prototipo.

j) Identificación y corrección de errores.

k) Elaboración del informe final

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

La ejecución del proyecto e implementación del sistema de control de acceso basado en reconocimiento facial y voz con algoritmos de Aprendizaje Automático fue implementado en la casa de la familia Orozco ubicado en parroquia Ambatillo de la ciudad de Ambato, el cual permitió tener un mejor control de las personas que ingresan a una vivienda , mejorando así el nivel de protección y seguridad hacia el interior de la vivienda , si una persona no registrada o desconocida con malas intenciones intenta ingresar al domicilio no podrá hacerlo de manera fácil y será reportada de manera inmediata como alguien sospechoso , logrando así evitar daños a la propiedad. Se establece de esta manera un sistema de seguridad seguro y confiable, que fomentara con el tiempo la implementación en más viviendas de la localidad. Tanto el software y el hardware empleados en este proyecto son de uso libre, además se utilizó elementos y componentes existente dentro del mercado ecuatoriano.

3.1.1 Análisis de Factibilidad

Factibilidad Técnica

Este trabajo se considera técnicamente factible ya que los múltiples componentes eléctricos y electrónicos utilizados son comercializables y pueden adquirirse dentro del país, además de que se ha investigado el funcionamiento de cada elemento.

Factibilidad Económica

El desarrollo del proyecto es financieramente factible, porque el investigador cubrió todos los costos del proyecto.

Factibilidad Bibliográfica

La documentación recopilada hizo posible este proyecto porque está disponible públicamente, incluidas revistas indexadas, videos, tutoriales, libros y tesis, entre otros recursos que fueron fundamentales para el éxito del proyecto.

3.2 Desarrollo de la propuesta

Requerimiento del prototipo

Para el desarrollo del sistema de control de acceso basado en redes neuronales en la vivienda de la familia Orozco, se plantearon diversos requerimientos por parte del responsable de las instalaciones entre las cuales se tiene que cumplir:

- Control de acceso mediante el uso del reconocimiento facial y de voz.
- Control cerradura solenoide mediante el sistema de control de acceso.
- Sistema de notificación al momento de que una persona usa el sistema mediante la aplicación Telegram.
- Comunicación entre en sistema y servidor la cual guardad la información del sistema.

Etapas del sistema

En este capítulo se describe el diseño del sistema de control de acceso en tiempo real a través de reconocimiento facial y autenticación mediante comando de voz empleando inteligencia artificial. En el diseño del sistema se hace uso de un algoritmo de aprendizaje profundo basado en una red neuronal para evaluar el desempeño e implementar en una tarjeta de desarrollo inteligente.

Se describe las etapas del sistema los cuales son: adquisición de datos, procesamiento de los datos y visualización de datos, como se observa en la figura 38.

La primera etapa, abarca la recolección de información o la generación de la base de datos para para los rostros y clips de audio empleando técnicas de aprendizaje automático y de procesamiento de señales mediante una cámara de alta resolución y un micrófono en el entorno de programación Python. La segunda etapa tercera etapa es la parte más importante del sistema ya que recibe los datos para ser procesados a través de un algoritmo basado en una red neuronal, donde realiza el tratamiento de imágenes, la fase de entrenamiento y la fase de prueba para el control de acceso en tiempo real. En esta etapa se utiliza un algoritmo de reconocimiento facial empleando Keras con Tensorflow en el entorno Jupyter Lab de Python, para programar y definir las características que va a tener la red neuronal. En la tercera etapa se desarrollaron la visualización de datos, aquí se realiza el acceso y el registro de las personas

reconocidas en un motor de base de datos MySQL y notificación mediante mensajería Telegram.Todas estas etapas se ejecutaron de manera secuencial y están interconectas ya que funcionan dentro de la tarjeta de desarrollo.

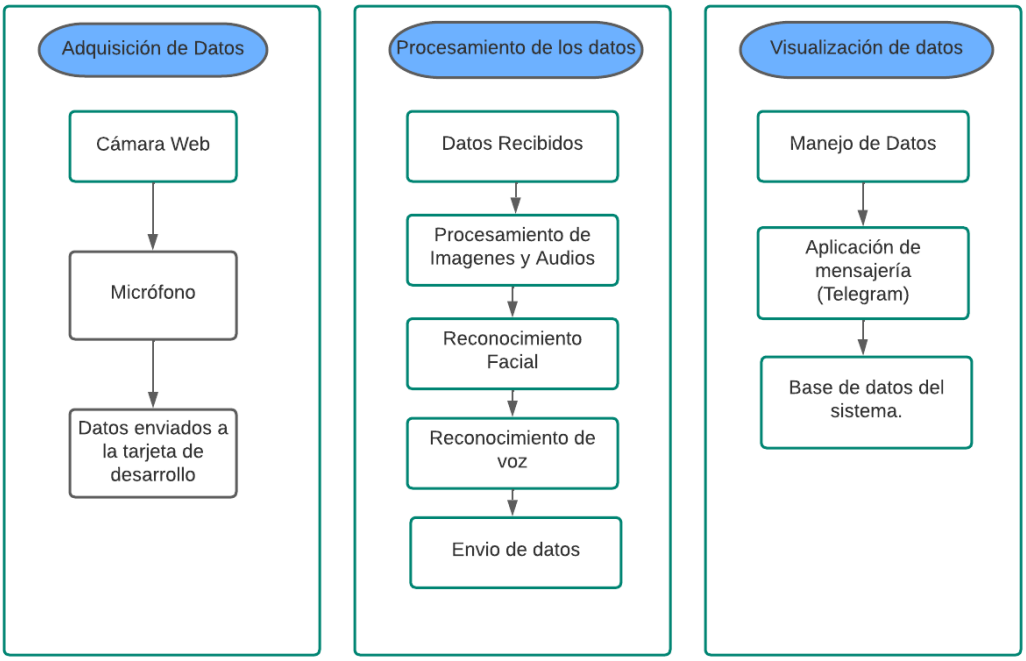


Figura 38 Etapas del sistema de control de acceso.

Elaborado por: Investigador

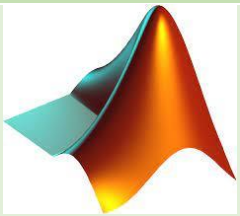
Sin embargo, antes de revisar los algoritmos empleados en la red neuronal CNN, se realizará una revisión de las herramientas utilizadas para ello, como el lenguaje de programación y módulos.

Herramientas de desarrollo

Lenguaje de programación

Conjunto de instrucciones que permiten realizar algoritmos y procesos lógicos que luego serán ejecutados por una computadora o sistema informático, brindando control tanto del comportamiento físico y lógico de la máquina como de conexión con el usuario. [52]

Tabla 3 Tabla comparativa de los distintos lenguajes de programación. *[52]*

Características	**Lenguajes de programación**		
	Python	LabVIEW	Matlab
			
Sistema Operativo	Linux, Mac y Windows	Multiplataforma	Multiplataforma
Licencia	Libre	Propietaria	Software privativo
Aplicaciones	*Desarrollo Web *Procesamiento de imágenes *Interfaz gráfica *Juegos de desarrollo	*Diseño de sistemas industriales *Procesamiento de imágenes *Sistemas de pruebas automatizadas	*Manipulación de matrices *Interfaz de usuario *Procesamiento de imágenes

Ventajas	*Lenguaje de código abierto orientado a objetos *Facilita la programación concurrente *Portabilidad	*Simplificación de tareas *Gran flexibilidad al Sistema *Dotado de un compilador gráfico	*Extenso soporte de funciones ya desarrolladas *Visualización de gráficos de alta calidad *Proporciona IDE más rápido para el cálculo matemático
Desventajas	*Consumo de memoria *Lentitud	*Poca capacidad de procesamiento *Lentitud	*Problemas de velocidad *La computadora necesitan MCR (Matlab Component Runtime) para que funcionen adecuadamente.

Elaborado por: Investigador

Se compararon los distintos lenguajes de programación mencionados en la tabla 3, por lo cual, el más óptimo es Python debido a que es un lenguaje de código abierto orientado a objetos que soporta procesamiento de imágenes y es de licencia libre, además, es compatible con Linux lo que facilita su implementación en la Raspberry Pi porque se va a instalar Raspberry Pi OS llamado también Raspbian para la realización del proyecto de investigación.

Se empleó Python para realizar el proyecto ya que proporciona una gran cantidad de bibliotecas que facilitan el desarrollo de aplicaciones de inteligencia artificial [53]. Se ha utilizado la versión 3.10 ya que es compatible con las bibliotecas utilizadas y es la versión estable más reciente de Python en el momento en que se desarrolló el sistema.

Librerías utilizadas

Se utilizó varias bibliotecas para crear el sistema, lo que facilitó el desarrollo gracias a un conjunto de funciones o clases descritas en este documento. Las siguientes bibliotecas que se han utilizado ampliamente son:

- NumPy: Es un conjunto de herramientas de Python para computación numérica y análisis de datos, con un enfoque en grandes conjuntos de datos. [54]
- OpenCV: Es una biblioteca de programación informática de código abierto que admite aplicaciones de visión artificial. [55]
- Tensor Flow: Es un marco de aprendizaje automático de extremo a extremo de código abierto. [56]
- Keras: Es una biblioteca de Python para crear redes neuronales. Se puede usar con Tensor Flow, Microsoft Cognitive Toolkit o Theano como base. [57]
- SpeechRecognition: Es una función que permite la captura de audios y sonidos para la aplicación en la domótica y la inteligencia artificial, etc. [58]
- Librosa: Es un paquete de utilidades de Python para analizar y procesar audio y música. [59]

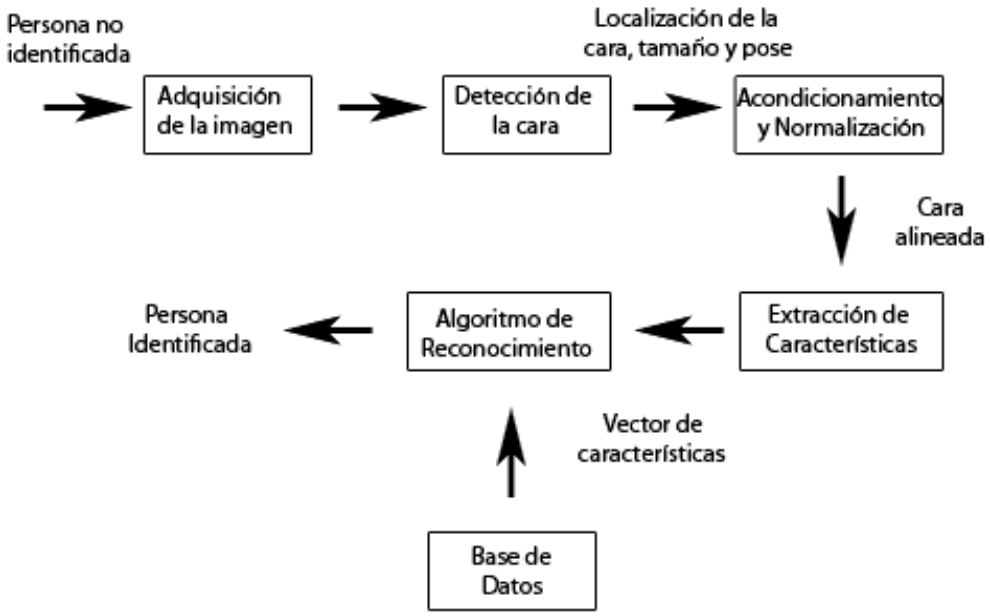
Adquisición de datos

En el presente modulo se describe la adquisición de los datos de imagen y clips de audio, la base de datos del sistema se forma por una pequeña muestra de 4 personas con un mínimo de 250 fotografías y 15 audios por persona. La obtención de los datos se lleva a cabo en el lugar donde se implementará el prototipo teniendo en cuenta la iluminación y el ruido del lugar.

Sistema de Reconocimiento facial

El primer método aplicado en el sistema de control de acceso es el reconocimiento facial como se muestra en la figura 39 el cual consta de 3 etapas. La primera etapa se

encarga de la adquisición de los datos que son: captura y detección del rostro de una persona desconocida, los cuales serán adquiridos mediante la cámara web, estos datos serán imágenes digitales que se guardaran para ser procesados. La segunda etapa se encarga del procesamiento de los datos, en la cual se realiza el acondicionamiento y normalización de las imágenes que servirán para el algoritmo final. La última etapa se encarga de aplicar el algoritmo de reconocimiento facial que será el encargado de identificar y permitir el paso a la persona correcta.



Figuran 39 Etapas de reconocimiento facial.

Elaborado por: Investigador

Captura y almacenamiento de imágenes

Para poder realizar el reconocimiento de rostros, fue necesario forma una base de datos propia de las personas a reconocer, para lo cual se formó una base de datos llamada Dataset de imágenes. Este dataset se formó por 1000 imágenes de 4 de personas. Solo se utilizó cuatro personas debido al requerimiento computacional en el momento del entrenamiento de la red neuronal.

Para llevar a cabo el proceso se utilizó uno de los lenguajes de programación que se mencionó anteriormente: Python. Se representa el algoritmo utilizado para obtener las imágenes. En pocas palabras, el algoritmo desarrollado tiene como objetivo recopilar

fotos únicas de la cara que tienen un tamaño de 250x250 píxeles, como se observa en la figura 40, además el código que se utilizó se verán en el anexo A.

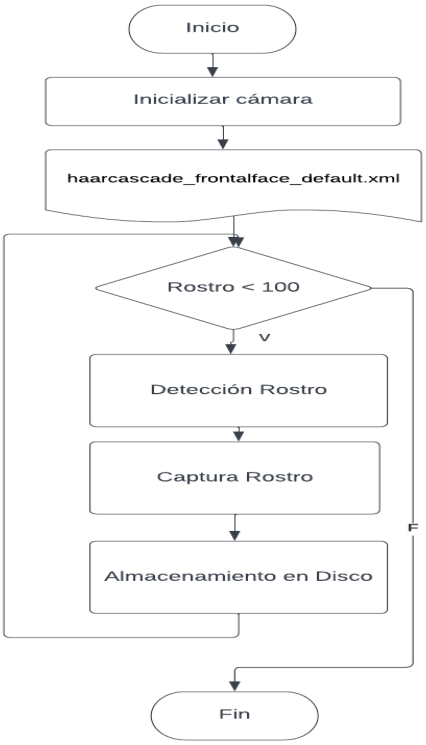


Figura 40 Diagrama de flujo para la captura de rostros.

Elaborado por: Investigador

Después de capturar las imágenes de los 4 individuos, se dividió la cantidad de imágenes en tres grupos: Entrenamiento, Validación y test. Para dividir el total de las imágenes en tres grupos tomo referencias del trabajo realizado por Artola [60] , en la cual utiliza el criterio de equilibrio de aprendizaje. Se dividió el total de las imágenes en porcentajes: 70% para el conjunto de entrenamiento, 20% para el conjunto de validación y 10% para el conjunto de pruebas, se utilizó este método ya que evita problemas con el funcionamiento y rendimiento de la CNN por el ajuste insuficiente y el ajuste excesivo. El primero es generado por tener datos insuficientes, lo que impide que CNN generalice lo que ha aprendido. El segundo ocurre cuando la red no es lo suficientemente capaz de distinguir entre clases después del entrenamiento utilizando imágenes que son demasiado particulares sobre las clases que se van a evaluar. La distribución de las imágenes fue en tres categorías cómo se observa en la tabla 4.

Tabla 4 Número de fotos agrupadas en tres grupos de datos separados para las 4 personas.

Nombre	Cantidad de imágenes			Total
	Entrenar	Validar	Probar	
Carlos	175	50	25	250
Javier	175	50	25	250
Diego	175	50	25	250
Jessenia	175	50	25	250
Total	**700**	**200**	**100**	**1000**

Elaborado por: Investigador

Pre procesamiento de las imágenes

Es necesario realizar un pre procesamiento a las imágenes obtenidas previamente, ya que esto permite entrenar y comprobar el funcionamiento de la red. Para esta etapa, las imágenes obtenidas pasaron por procesos que simplificaron y minimizaran su tamaño, estas etapas son: conversión a escala grises, escalado, rotación, traslación y recorte, se empleó una seria de transformación geométricas a través de un método llamado transformaciones afines. [61]

Para permitir la comparación, todas las caras ecualizadas deben tener el mismo tamaño y las mismas coordenadas oculares. En este proyecto, se ha optado por utilizar el formato de imagen ISO/IEC 19794-5 cómo se observa en la figura 41, basado en los formatos de intercambios de datos geométricos, que define un área similar a una foto de pasaporte, con unas dimensiones de 168x192 píxeles (ancho x alto), además el código que se utilizó se verán en el anexo B. [62]

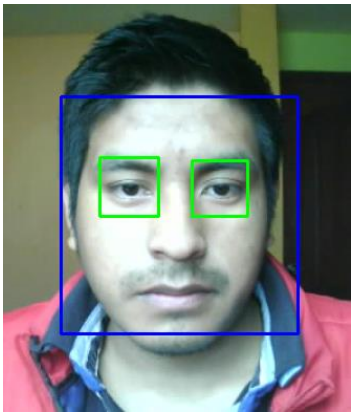


Figura 41 Ejemplo de fotografía de un rostro para el preprocesado.

Elaborado por: Investigador

- **Conversión a escala de grises:** Las imágenes están obtenidas mediante una webcam, por lo tanto las imágenes están en formato digital, específicamente es imagen en RGB (rojo, verde, azul) el cual para procesarla con los colores originales sería un trabajo muy duro ya que la imagen estará representada por una matriz tridimensional. Para evitar este problema se realizó conversión a escala grises como se observa en la figura 42, se aplicó el método llamado "promedio equivalente" el cual consiste en derivar de los tres planos de la imagen en color y obtener una imagen uniforme, mitigando así los cambios de luminosidad que podrían afectar en el análisis. [63]

Figura 42 Resultado conversión imagen a escala grises.

Elaborado por: Investigador

- **Rotación:** La rotación es el segundo cambio que se realiza para poder centrar el rostro de la imagen, como se observa en la figura 43.

Figura 43 Resultado rotación de la imagen.

Elaborado por: Investigador

- **Escalado:** En este proceso se ha dado un cierto tamaño a la cara que ya ha sido rotada y nivelada. Se decidió adherirse al estándar ISO/IEC 19794-5 , como se observa en la figura 44.

Figura 44 Resultado escalado de la imagen.

Elaborado por: Investigador

- **Recorte**: Es necesario recortar la imagen para eliminar los fondos y bordes que rodean al rostros. Las proporciones finales de todas las fotos son las mismas. El estándar que marca la norma mencionada anteriormente se utilizó para el escalado de la imagen. Como resultado, las imágenes finales tienen un tamaño de 168 x 192 píxeles (ancho x alto) como se observa en la figura 45.

Figura 45 Resultado recorte de la imagen.

Elaborado por: Investigador

Reconocimiento facial

El proceso de reconocimiento facial se realiza mediante una serie de pasos aplicando algoritmos para obtener un resultado esperado. A continuación, se presenta un análisis de los algoritmos empleados.

Detección

Debido a su fiabilidad y eficiencia, el algoritmo Haar-like features de Viola & Jones es una de las técnicas de identificación de rostros más utilizadas e implementadas por la librería OpenCV, es por esto que se usara esta librería de Machine Learning que permite identificar el rostro y otras partes de la cara como: ojos, nariz, boca, de una manera más fácil y sencilla como se observa en la figura 46.

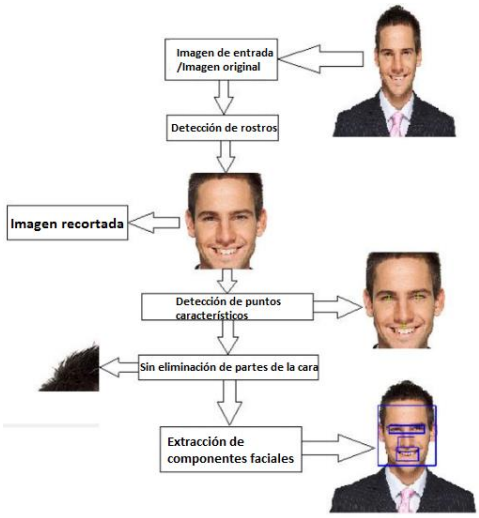


Figura 46 Detección de partes faciales usando el algoritmo de Viola Jones.

Detección de rostros

Se usó un detector de rostros robusto basado en un clasificador en cascada empleando el algoritmo Viola-Jones y la biblioteca OpenCV, cómo se observa en la figura 47. La técnica comienza cargando el clasificador Haar para que el elemento sea detectado desde la biblioteca, en este caso el clasificador de detección frontal de rostros llamado 'haarcascade_frontalface'. A continuación, utilizando la función de detección de la biblioteca se cambió el tamaño de la ventana de detección mínima y finalmente se aplicó una escala a la imagen para reducir los tiempos de procesamiento. El código utilizado para la detección de rostros se detalla en el anexo C.

El programa sigue una seria de pasos que permiten la detección de rostros:

- La imagen de la cara del resultado de la captura cambia de RGB a escala de grises.
- Detección de cascada Haar con estandarización de iluminación
- Para el reconocimiento facial se utiliza un extracto facial que comprende cálculos de rostros propios.
- PCA (construcción de una matriz de imagen de vector plano e identificación del vector de imagen) [64]

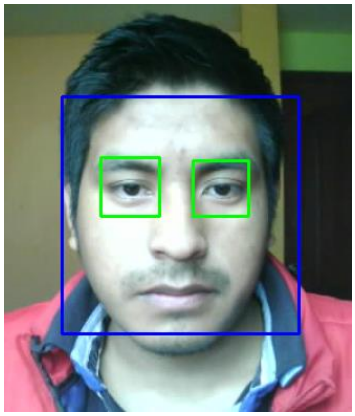


Figura 47 Resultado de reconocimiento de rostro.

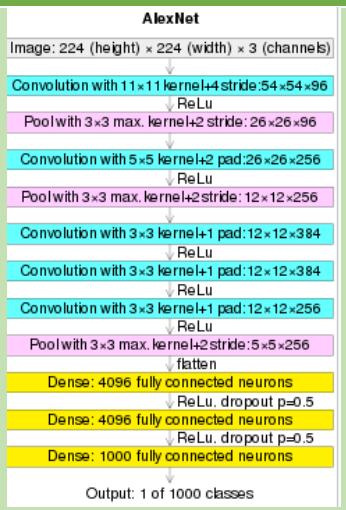
Elaborado por: Investigador

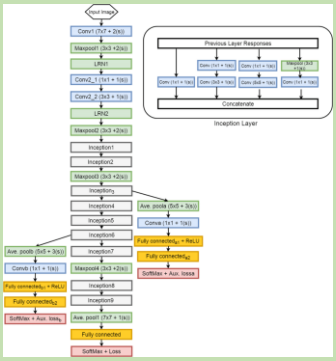
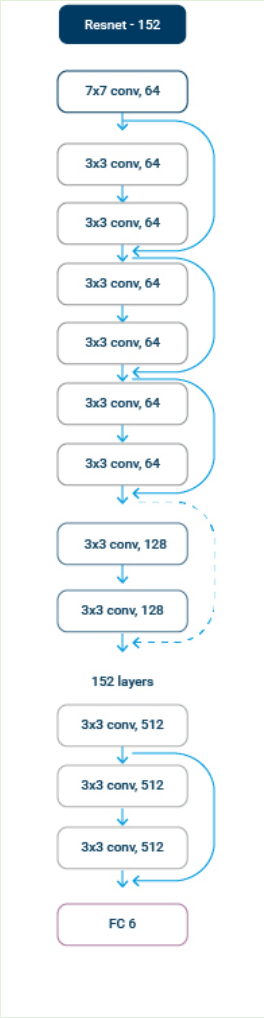
Modelo de Red Neuronal

Una red neuronal convolucional es un tipo de red neuronal artificial donde las neuronas artificiales, corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria de un cerebro biológico.

Después de obtener los conjuntos de datos y analizar las técnicas de detección, entrenamiento, se revisaron múltiples redes neuronales convolucionales para descubrir la estructura óptima para esta situación, como se muestra en la tabla 5.

Tabla 5 Tabla comparativa de los redes neuronales.

Nombre	Concepto	Características	ventajas	Estructura
AlexNet	es el nombre de una arquitectura de red neuronal convolucional (CNN), diseñada por Alex Krizhevsky en colaboración con Ilya Sutskever y Geoffrey Hinton	La arquitectura consta de ocho capas: cinco capas convolucionales y tres capas totalmente conectadas.	Uso de las GPUs para reducir el tiempo de entrenamiento.	
VGG	es una arquitectura de red neuronal convolucional (CNN) profunda estándar con varias capas	Existen dos tipos: VGG-16 o VGG-19 que consta de 16 y 19 capas convolucionales.	La arquitectura VGG es la base de modelos innovadores de reconocimiento de objetos.	

GoogLeNet e Inception	GoogLeNet es un tipo de red neuronal convolucional basada en la arquitectura Inception.	La arquitectura de GoogLeNet consta de 22 capas (27 capas, incluidas las capas de agrupación), y parte de estas capas son un total de 9 módulos de inicio	GoogLeNet ahora es una arquitectura básica dentro de las bibliotecas ML más comunes, como TensorFlow, Keras, PyTorch, etc	
ResNet	ResNets es una arquitectura de red neuronal común utilizada para aplicaciones de visión artificial de aprendizaje profundo	ResNet puede contener una gran cantidad de capas convolucionales, comúnmente entre 18 y 152, pero admite hasta miles de capas	ResNet permite entrenar cientos, si no miles de capas, mientras se logra un rendimiento fascinante.	

Elaborado por: Investigador

Existen dos posibilidades en este punto: emplear una red ya entrenada y simplemente entrenar las capas finales de las redes neuronales ya existentes como se observa en la figura 48 y tabla 5, o entrenar una red desde cero. Como no es necesario calcular todos

los parámetros de la red, la primera alternativa es más rápida y fácil de implementar. La desventaja de estas estructuras es que su precisión puede ser inferior a la de una red formada desde cero si la aplicación para la que se desarrollaron originalmente no es idéntica.

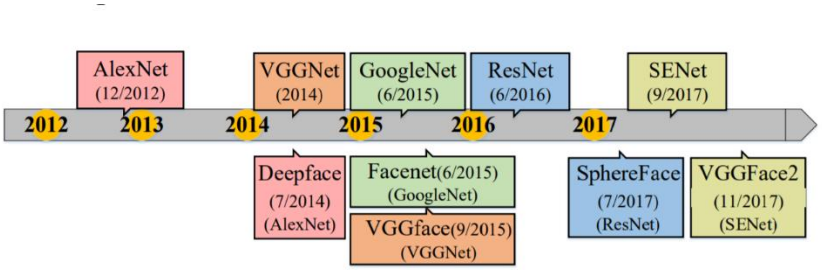


Figura 48 Arquitecturas de red típicas en la clasificación de objetos.

En este proyecto no se empleó un modelo entrenado, ya que se creará una red CNN particular desde cero empleando bibliotecas de inteligencia artificial. La implementación y entrenamiento del modelo de red neuronal convolucional se realizó en esta sección utilizando la API de Keras y la biblioteca Tensorflow en el lenguaje de programación Python.

Modelo de Red Neuronal Convolucional

Se utilizó el método de prueba y error para el diseño de la red neuronal. Para esto, se realizaron revisiones de trabajos y proyectos, sobre cómo implementar una red neuronal convolucional usando Keras y TensorFlow. El modelo de la red neuronal CNN consta de 10 capas las cuales se observa en la figura 49:

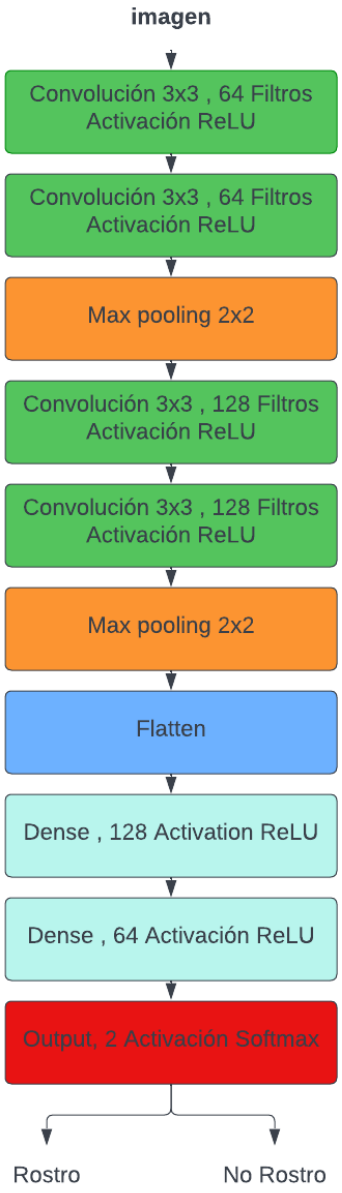


Figura 49 Modelo red neuronal convolucional para el reconocimiento facial.

Elaborado por: Investigador

Este modelo es una red neuronal convolucional consta de diez capas, cada una de las cuales se describe en profundidad a continuación:

- La primera capa de convolución, que consta de 64 filtros con kernels 3x3, se implementa y emplea la activación de ReLU para transformar los valores negativos a cero. El funcionamiento de esta capa implica extraer pequeños grupos de píxeles de la imagen de entrada y realizar un producto punto con el kernel.

- De la misma forma que la capa anterior, se creó una segunda capa de convolución con 64 filtros y kernels de tamaño 3x3. Esta capa también contiene una activación de ReLU, pero esta vez las entradas son 64 matrices de 168x192x1. Esta capa sigue los mismos pasos que la anterior, dando como resultado 64 matrices de salida con dimensiones de 168x192x64 cada una.

- Se utilizó una capa de Max Pooling de tamaño 2x2, lo que significa que en lugar de tomar píxeles uno por uno, se tomó 2 por 2 tanto en altura como en ancho, lo que resultó en una reducción de dimensiones comparable a 84x96 y una reducción en el número de neuronas.

- Con Kernels de 3x3 y activación ReLU, se creó de dos capas de convolución de 128 filtros. En esta ocasión se generan 128 matrices de 84x96x64.

- Se utilizó una segunda capa de Max Pooling, con un tamaño de 2x2, lo que resultó en una reducción de las dimensiones a 42x48 píxeles.

- La capa Flatten se agregó para "aplanar" las dimensiones, lo que significa que pasa de tres a una dimensión. Posee la misma cantidad de neuronas que la anterior capa.

- Se agregó unas dos capas típicas con activación ReLU, que consta de 128 y 64 neuronas. Estas capas son las encargadas de conectar todas las neuronas de la capa anterior con las neuronas de su capa. Además, estas capas son capaces de minimizar la cantidad de neuronas que se acoplan entre sí.

- La capa Salida se agregó en último lugar y presenta una activación Softmax, que asigna un valor probabilístico a cada una de las dos clases. El pronóstico del modelo será la salida con el valor de probabilidad más alto. Esta predicción indicará si la imagen evaluada es un escenario de "Rostro detectado" o "Rostro desconocido".

La red se compiló usando la función model.compile() después de implementar la arquitectura del modelo. La función de optimización, el optimizador a usar y la métrica de evaluación son entradas que se pueden pasar a esta función. La función de optimización en este ejemplo será "Loss " de tipo "entropía cruzada categórica", con

un optimizador llamado "Adam" y "accuracy " como medida de evaluación. El código utilizado se verá el anexo D.

Entrenamiento red neuronal

El sistema se configuro en este paso para que luego pueda aprender a reconocer cualquier cara con la que se le haya enseñado. Para lograrlo, se necesitó un conjunto de fotografías para cada individuo. Es fundamental que estas fotografías sean únicas, apareciendo únicamente el individuo.

El paso de entrenamiento se completó después del establecimiento del modelo de red neuronal convolucional. Los siguientes procedimientos se llevaron a cabo para lograr esto:

- Establecer el valor del hiperparámetro " steps_per_epoch ", comúnmente conocido como "pasos". Esto se calculó dividiendo el número de muestras en el conjunto de entrenamiento (1000 imágenes) por el parámetro de tamaño de lote (batch_size=32, 64).
- El hiperparámetro “validation_step”, a veces conocido como "pasos de validación", recibió un valor. Este parámetro indica que se llevarán a cabo una serie de "pasos de validación" al final de cada "época", utilizando el conjunto de validación, para ver si el aprendizaje del algoritmo implementado es correcto. El número de "pasos de validación" se calculó multiplicando el número de muestras en el conjunto de validación por el hiperparámetro " batch_size".
- Se empleó la función "fit", que ejecuta el proceso de entrenamiento. Para lograr esto, los hiperparámetros que se definieron previamente y además se declararon argumentos de la función.
- Finalmente, se realizó el entrenamiento específico del modelo. Se guardaron los resultados de la prueba de entrenamiento en archivo denominado "model-cnn-facerecognition.h5". La ejecución del programa se verá en el anexo E.

Sistema de Reconocimiento de voz

El sistema propuesto que se realizó para el reconocimiento de voz consta de tres etapas: captura de la señal de voz, pre procesamiento de la señal y entrenamiento del hablante utilizando las funciones adquiridas en las dos primeras etapas. Como se observa en la figura 50.

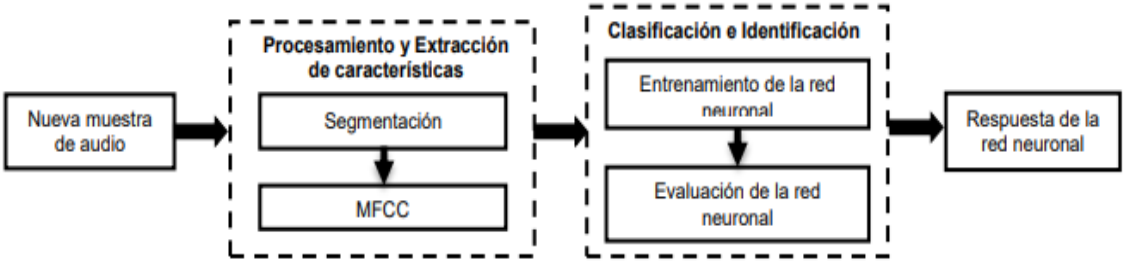


Figura 50 Sistema propuesto reconocimiento de voz.

Captura de la señal de voz

Para la captura de los clips de audio se tomó una muestra de 15 audios por persona con una frase que son sus cuatro números finales de la cédula, los audios son grabados con diferentes expresiones de emociones, tonos y con una duración de 4 segundos cada uno, como se puede observar en la figura 51.

Figuran 51 Expresiones de emociones de la voz. [65]

Para que sea más fácil reconocer y entrenar a los participantes a lo largo de la investigación, se etiqueto sus rangos de audio con sus nombres como se observa en la tabla 6.

Tabla 6 Rangos de audios por persona.

Persona	Código-Frase	Rango	Edad	Sexo
Carlos	6489	audio1 - audio 15	25	Masculino
Diego	1052	audio16 - audio 30	23	Masculino
Javier	1279	audio31 - audio 45	27	Masculino
Jessenia	4291	audio46 - audio 60	21	Femenino

Elaborado por: Investigador

Se utilizó los números finales de cedula como frase, ya que se ocupó el api de Google Cloud Speech para una mejor identificación de la voz y autenticación. Este método de grabación ayuda a tener un audio procesado complementamente libre de ruido y directamente para ser procesado como se observa en la figura 52, además el código utilizado se verá en el anexo F.

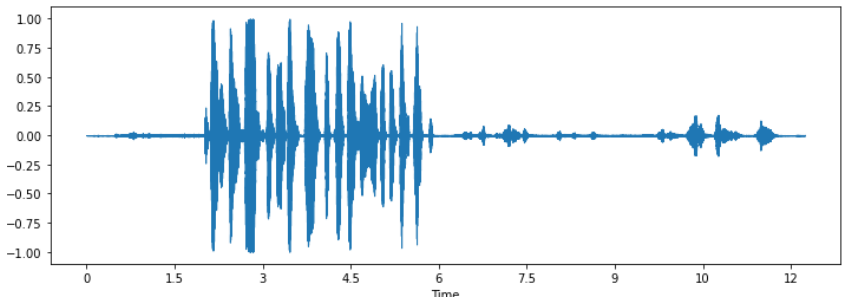


Figura 52 Señal de audio grabada mediante Google Speech Cloud.

Elaborado por: Investigador

El número de muestras se obtuvo mediante la tarjeta de audio de computadora, se logró obtener 60 audios, de los cuales 44 audios son para el entrenamiento de la red, 8 audios para la validación, y 8 audios para las pruebas (test) de la red neuronal. Todos los audios están en formato *. WAV.

Después de tener el total de audios a capturar, se comenzó a grabar los audios configurando los siguientes parámetros, como se observa en la tabla 7:

Tabla 7 Parámetros de los clips de audio.

Velocidad de Transmisión.	**128Kbps**
Tamaño de muestra de sonido.	16 bits
Tipo de canal.	Monofónico
Velocidad de muestreo de sonido.	22Khz
Formato de audio.	*. Wav

Elaborado por: Investigador

Después de terminar con la captura de los audios grabados, era necesario procesar y analizar las señales. Para ello se utilizó LIBROSA, un paquete de Python que permite graficar y el analizar fragmentos de audio como se puede ver en la figura 53.

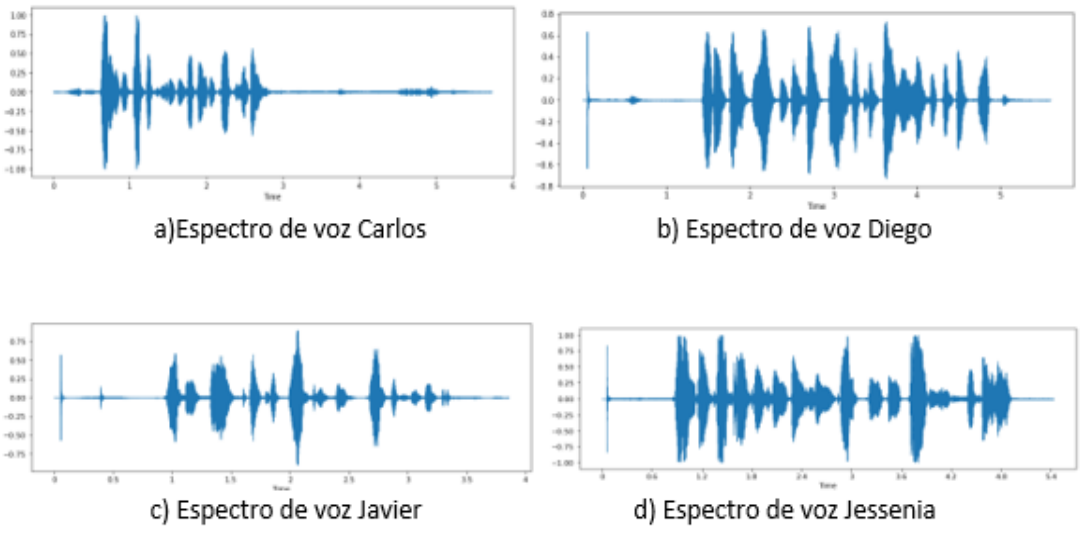


Figura 53 Representación de los audios capturados en función del tiempo.

Elaborado por: Investigador

Procesamiento de audio

Se realizó el procesamiento para calcular algunas características de las muestras de audio, así como también se definió una función que permitió carga los archivos WAV

de una muestra antes de calcular las características. Es necesario normalizar los datos de entrada antes de procesar las características, ya que las amplitudes de las señales de audio grabadas pueden fluctuar significativamente dependiendo del dispositivo de grabación y otros factores.

El propósito de esta etapa es eliminar cualquier información innecesaria de los diversos audios del conjunto de datos y acentuar la pronunciación. Para ello, se aplicó el método de los coeficientes cepstrales de las frecuencias de Mel, que fue la técnica de extracción de características utilizada.

Extractor de características MFCC

Primero se ingresó el audio a procesar donde se aplicó el método llamado Framing, que consiste en segmentar el audio en intervalos de ms. Para reducir el impacto de las fugas al ejecutar la transformada de Fourier en el paso siguiente, conviene "recortar" las muestras de la señal de audio utilizando una ventana como la ventana de Hamming.

Se determinó el espectro de potencia, o la magnitud al cuadrado del espectro para cada muestra, calculando primero la FFT de la señal. Se empleó la transformada de Fourier el cual permitió cambiar nuestro audio en el dominio del tiempo a un audio en el dominio de la frecuencia.

Es necesario aplicar un banco de filtros Mel en el espectro de potencia de cada fragmento, para lo cual se utilizó las frecuencias de Mel que explica el hecho de que las frecuencias más altas dan como resultado menos variaciones de tono (o frecuencia) perceptibles en la percepción auditiva humana. La suma de las "energías" en cada filtro se obtiene multiplicando estos filtros por el espectro de potencia.

También se tomó el registro de estas energías agrupando los segmentos de audio para aplicar el log y obtener las energías logarítmicas. El logaritmo de energía promedio en cada muestra está representado por el coeficiente cero, que puede eliminarse o no.

Las energías del banco de filtros de registro de cada fragmento se transforman mediante una transformada de coseno discreta, dando como resultado una matriz de audio con de 193 indicadores que representa el audio de entrada como un todo. Además se presenta los códigos y resultados obtenidos en el anexo G.

Modelo de Red Neuronal Convolucional

Al igual que la primera red convolucional de reconocimiento de rostros, esta red será creada desde cero con la misma metodología y tecnologías. Para la red neuronal primero se dividió todos los datos de audios en tres grupos como se observa en la figura 54, **entrenamiento** con 44 audios, **validación** con 8 audios y **test** con 8 audios.

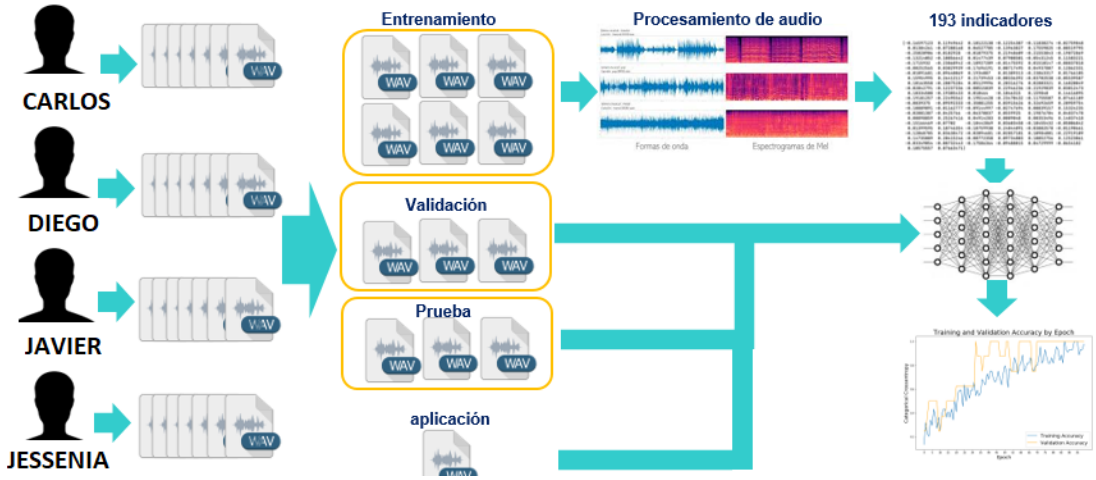


Figura 54 Estructura del modelo convolucional de voz.

Elaborado por: Investigador

Como se explicó en la etapa de pre procesamiento de señales se obtuvo 193 indicadores de audio como datos de entrada en una matriz o matriz de datos para cada grabación. Finalmente, la matriz se introduciría en un modelo CNN particular, que aprendería de la muestra de entrenamiento después de varias repeticiones.

Implementación modelo de Red Neuronal Convolucional particular

Antes de desarrollar la red neuronal, se organizó los datos creando un archivo de Excel que incluía todos los audios que se capturo, poniendo el nombre como identificación y completando cierta información. El archivo de distribución de los audios se puede ver en el anexo H.

La red neuronal CNN particular consta de 7 capas como se observa en la figura 55:

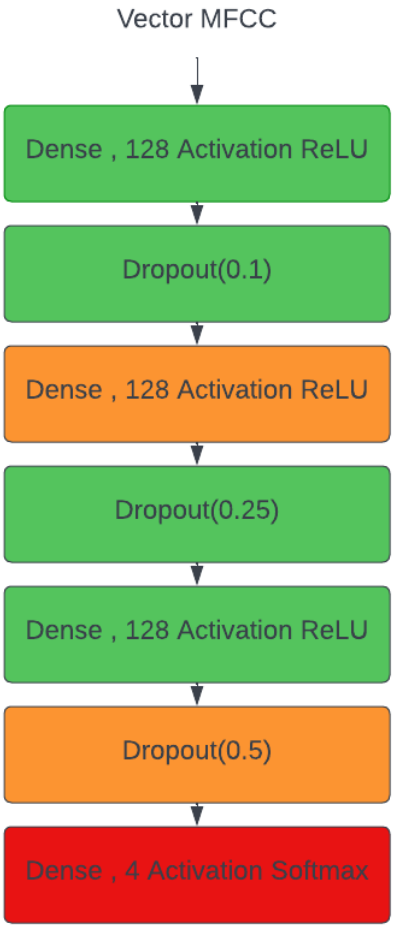


Figura 55 Modelo red neuronal convolucional para el reconocimiento de voz.

Elaborado por: Investigador

En esta red neuronal, agregamos tres capas con activación RELU, la primera de las cuales tendrá una densidad de 193 y una caída de 0,1, que es el número total de indicadores recopilados en la extracción de datos anterior. La segunda y tercera RELU tendrán una densidad de 128 con un dropout de 0,25 y 0,5, respectivamente; y la cuarta capa tendrá una función de activación softmax con una densidad igual al número de indicadores únicos en el experimento, que en este caso es 4. Usaremos la optimización de aprendizaje de Adam para su optimización.

Entrenamiento de la red neuronal

El paso de entrenamiento se completó después del establecimiento del modelo de red neuronal convolucional. Los siguientes procedimientos se llevaron a cabo para lograr esto:

- Establecer el valor del hiperparámetro "steps_per_epoch ", comúnmente conocido como "pasos".
- Finalmente, se empleó el modelo model.fit () para el entrenamiento. Se especifican los conjuntos de datos de entrenamiento y validación, así como la cantidad máxima de épocas para entrenar, las devoluciones de llamada y el tamaño del lote. El tamaño del lote tiene un impacto en el entrenamiento, especialmente cuando se emplea SGD. El promedio de los gradientes por lotes se calcula y se utiliza para actualizar los pesos en cada iteración. Un lote más pequeño produce un gradiente "ruidoso", que puede ser beneficioso para explorar más el espacio de pesas (y evitar quedarse atascado demasiado pronto en un mínimo local), pero es perjudicial para la convergencia cerca del final del entrenamiento.
- Finalmente, se realizó el entrenamiento específico del modelo. Se guardaron los resultados de la prueba de entrenamiento en archivo denominado "model-cnn-speechrecognition.h5".

Debido a que un lote más grande produce menos gradientes ruidosos, se pueden ejecutar pasos más grandes (es decir, una tasa de aprendizaje más alta), lo que da como resultado un entrenamiento más rápido. Además, los lotes más grandes tienen una sobrecarga computacional más baja.

Después de entrenar el modelo, se utilizaron la muestra de validación y la muestra de prueba para determinar la precisión real del modelo. El código utilizado para el diseño y entrenamiento se puede ver en el anexo I.

Selección de los elementos para la implementación del sistema

Se ejecutó un análisis técnico de cada uno de los elementos tomando en cuenta las características, el precio de cada uno de ellos y sobre todo si se encuentran disponibles en el país. La selección de los diferentes componentes para el sistema de control de acceso y monitoreo se indica a continuación:

Tarjeta de desarrollo

Son circuitos con un micro controlador primario que ejecuta una serie de instrucciones de un programa en particular, beneficiando varios procesos de diseño, ya sea para sistemas digitales o analógicos. [66]

Tabla 8 Tabla comparativa de tarjetas de desarrollo. [67] [68] [69]

Características	**Tarjetas de Desarrollo**		
	Raspberry pi 4	Banana PI BPIM64	Arduino Uno
			
Tipo	Microordenador	Microordenador	Microordenador
RAM	4GB	2GB	2KB
GPU	VideoCore IV 400 MHz	Dual core Mail 400 MP2	ATmega 328
Procesador	Broadcom BCM2837, CortexA53(ARMv8) 64 bits SoC	ARM A53 QuadCore	ATmega 328
Almacenamiento	Micro SD	Micro SD	EEPROM 1KB
Wifi	2.4 GHz 802.11n	802.11 b/g/n	No

Bluetooth	BT 4.1	BT 4.0	No
Frecuencia de reloj	1,5 GHz	1,2 GHz	16 MHz
Conectividad de red	Fast Ethernet 10/100 Gbps	Ethernet 100 Mbps	No
Cámara	Puerto CSI	MIPI-CSI	No
Precio	$200	$75	$15
Disponibilidad en el país	Si	Si	Si

Elaborado por: Investigador

Una vez realizado el análisis en la tabla 8, la tarjeta de desarrollo seleccionada es la Raspberry pi 4 debido a que es un ordenador de tamaño reducido, el cual contiene una conectividad de red, se puede conectar también a través de wifi, y es de fácil acceso en el mercado.

Cámara Web

Dispositivo para capturar imágenes, va conectado a la Raspberry Pi por medio del puerto USB donde las fotografías se transmiten a través de internet.

Tabla 9 Tabla comparativa de las cámaras. *[70] [71]*

Características	**Cámaras**		
	Youlisn	Adesso CyberTrack H4	Conexis
			
Resolución	1080 pixeles	720P 1280×720 pixeles	720P 1280×720 pixeles
Velocidad de Fotogramas	30 FPS	30 FPS	30 FPS

Tipo de sensor	CMOS	CMOS	CMOS
Interfaz	USB2.0 + micrófono	USB2.0	USB2.0 + Cable de Audio 3.5mm
Precio	20	36	15
Disponibilidad en el país	Si	Si	Si

Elaborado por: Investigador

En la tabla 9, se realizó la comparación de las posibles cámaras a utilizar para un funcionamiento adecuado del prototipo, se eligió la Cámara Web FULL-HD 1080P Youlisn, debido a que tiene una resolución adecuada para aplicar al sistema y el costo es económico, además como la Raspberry pi tiene puertos USB es factible utilizar este tipo de cámara.

Diagrama de bloques del dispositivo

El esquema general del sistema se observa en la figura 56, el cual consta de una Raspberry Pi 4 de 8GB a la cual se le conecta la cámara USB y una tarjeta de sonido para el micrófono, una fuente de alimentación, una pantalla para la visualización del sistema, un módulo relé, parlante, un ventilador para evitar que la Raspberry Pi se sobrecaliente y garantizar el correcto funcionamiento del sistema. Se presenta las características de la tarjeta utilizada en el anexo J.

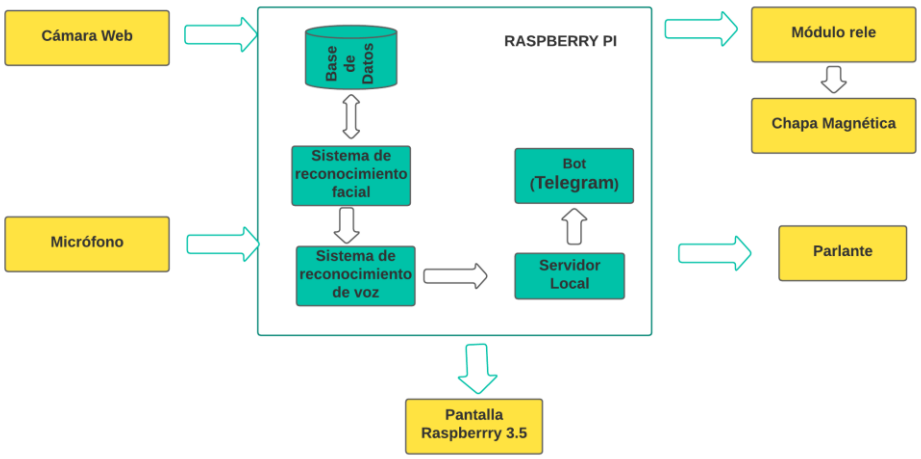


Figura 56 Esquema del sistema biométrico

Elaborado por: Investigador

Además, se presenta el diagrama de conexión físico que se utilizó para el prototipo, analizado los componentes del diagrama anterior, cómo se observa en la figura 57.

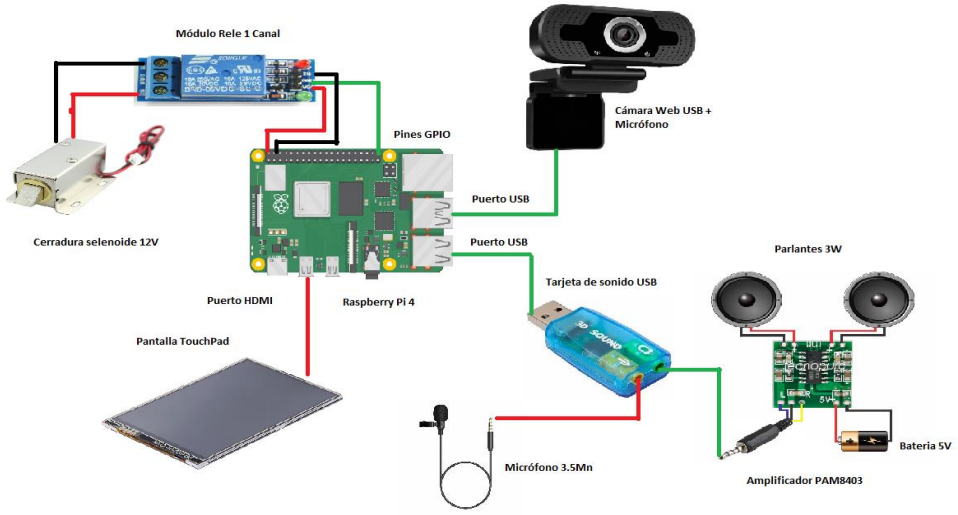


Figura 57 Esquema General del prototipo.

Elaborado por: Investigador

Se muestra el esquema pictográfico del sistema con las conexiones correspondientes, el diagrama se lo realizó en el software Fritzing, como se observa en la figura 58.

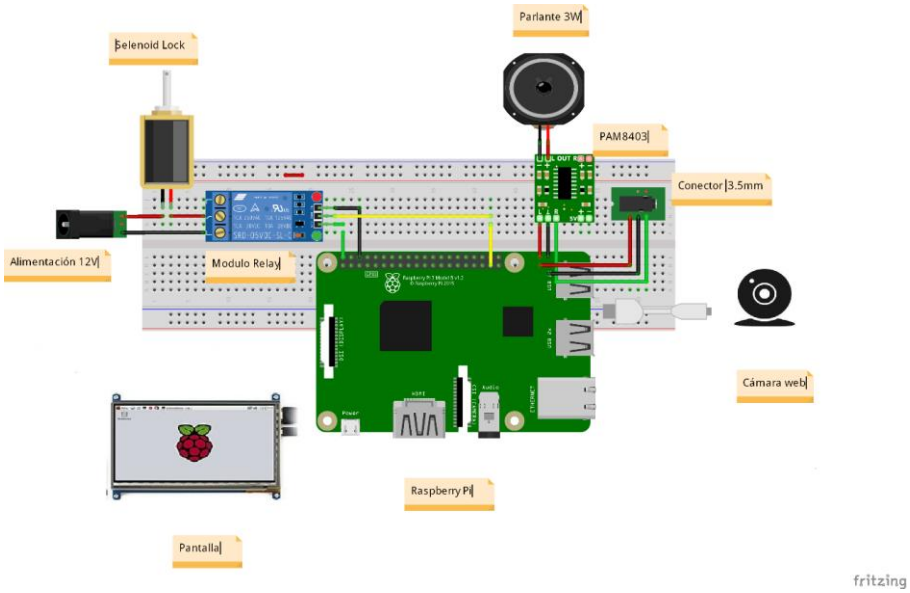


Figura 58 Diagrama pictográfico del sistema.

Elaborado por: Investigador

Montaje del circuito

Las conexiones entre los componentes del circuito se implementaron en una protoboard como se observa en la figura 59.

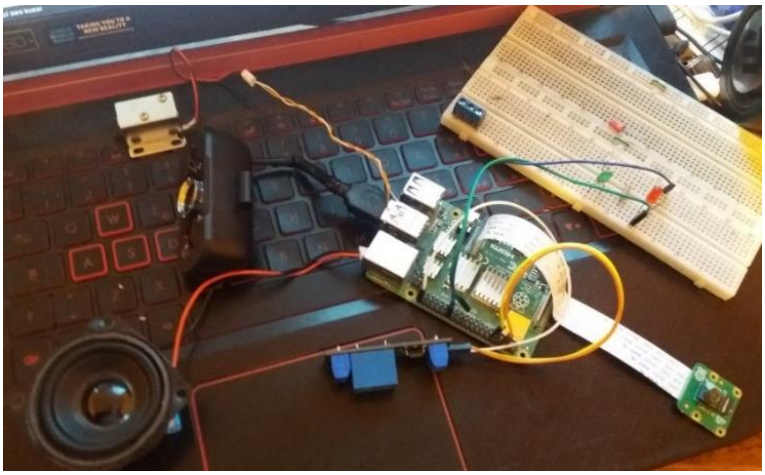


Figura 59 Montaje del circuito antes de la implementación.

Elaborado por: Investigador

Una vez realizado el diagrama de conexión se colocó todos los materiales dentro de una caja hermética para proteger de daños producidos por el clima como se observa en la figura 60.

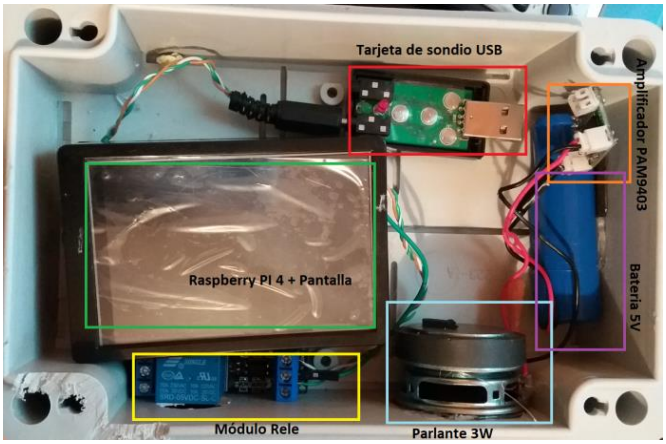


Figura 60 Implementación dentro del prototipo.

Elaborado por: Investigador

Diseño de software del proyecto

Se muestra el funcionamiento del sistema de manera secuencial, en donde se inicia con la activación de la cámara para capturar el rostro, la imagen capturada es procesada

para pasar a la detección de rostros, si el sistema registra un rostro pasa comparar con nuestra base de datos, si el rostro pertenece al usuario registrado pasa a la activación del micrófono para decir la frase a comparar o si no el sistema vuelve a comenzar la detección. Si él se cumple la detección de rostro y la autentificación de voz se cumplen de manera correcta el sistema permite el acceso y realiza el registro del usuario en la base de datos con la notificación vía mensajería Telegram, como se observa en la figura 61.

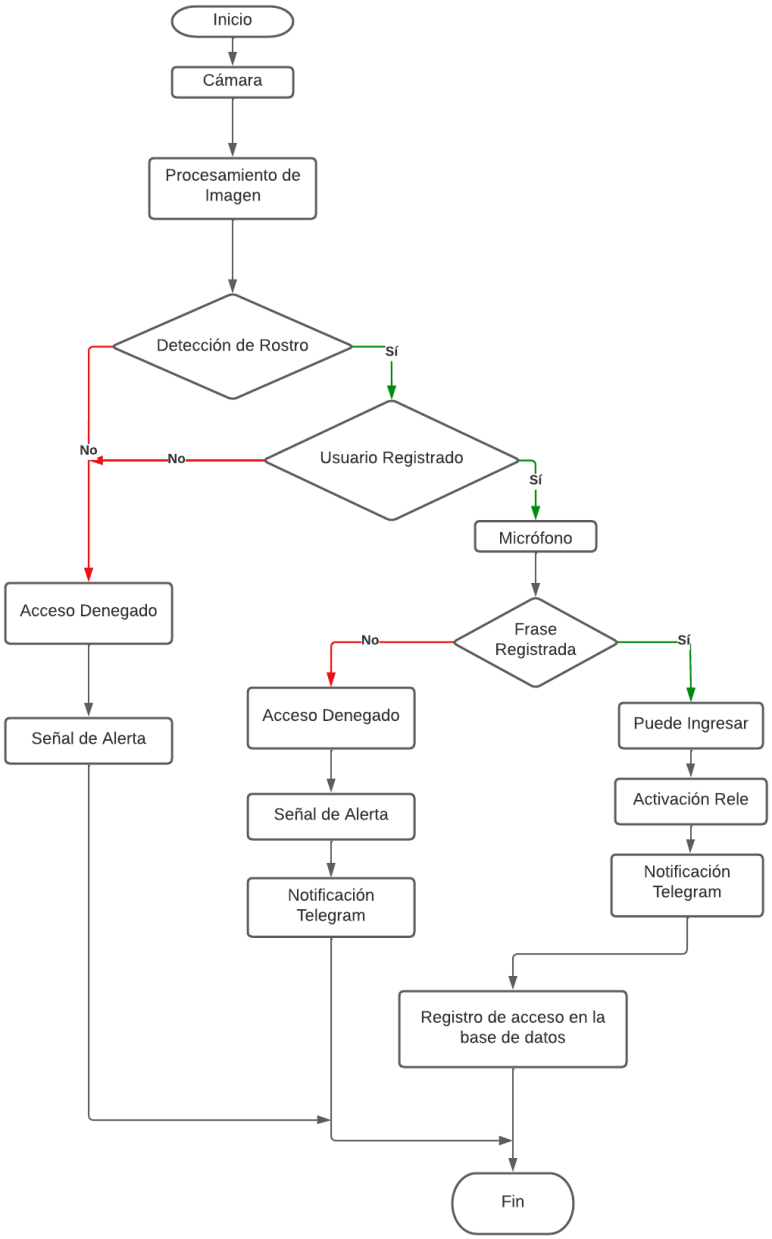


Figura 61 Diagrama de flujo del sistema.

Elaborado por: Investigador

A continuación, se detallan cada uno de los procedimientos para el reconocimiento de voz y facial en la placa de desarrollo Raspberry Pi.

Instalación del Sistema Operativo en la Raspberry pi

Para la instalación del software en la Raspberry Pi se usó una tarjeta MicroSD, en la cual permitió la instalación del sistema operativo Raspberry Pi OS. Para la instalación se usó el sistema operativo más actual disponible en la página oficial de Raspbian como se observa en la figura 62.

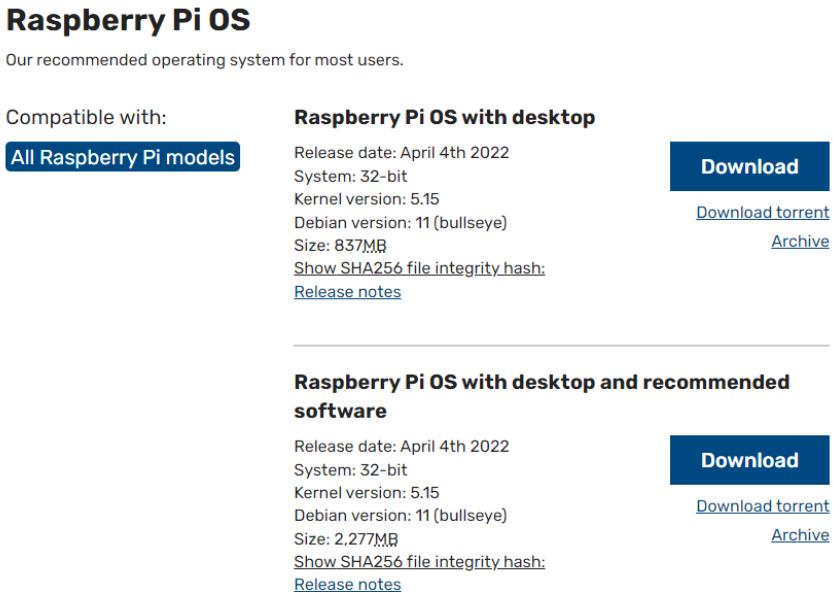


Figura 62 Sistemas Operativos para Raspberry.

Elaborado por: El Investigador

Una vez escogido el sistema operativo adecuado deberemos quemar en una memoria SD con el programa Raspberry Pi Imager como se observa en la figura 63.

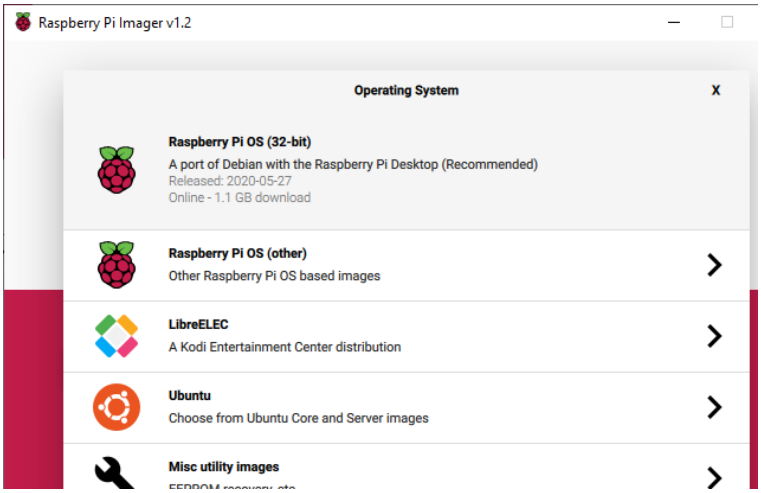


Figura 63 Instalación del sistema operativo en la microSD.

Elaborado por: El Investigador

Una vez instalado el sistema operativo se empezó a configurar el sistema operativo de la manera más conveniente y adaptable a las necesidades del sistema.

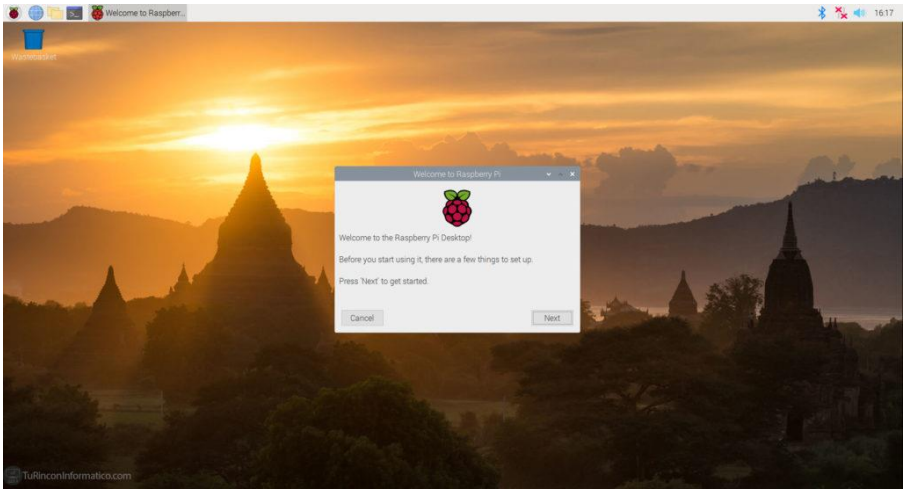


Figura 64 Pantalla de inicio del sistema operativo.

Elaborado por: El Investigador

Cuando se inicia por primera vez como se observa en la figura 64 se necesitó instalar y actualizar todas las dependencias necesarias del sistema para lo cual se ejecutó el siguiente comando:

apt-get update y apt-get upgrade

Instalación de dependencias para el sistema

Se instaló Tensor Flow y Keras, entre otras bibliotecas, para reconocimiento facial y de voz; sin embargo, se debe construir un ambiente virtual para instalar los paquetes exactos que se utilizarán en el desarrollo del proyecto y evitar problemas con la actualización de las librerías.

Instalar TensorFlow en Raspberry Pi

Hay que asegurarse de que el dispositivo Raspberry Pi tenga instalada la versión más reciente de los paquetes antes de iniciar cualquier instalación. Se ejecutó la siguiente línea de comando en una ventana de terminal.

sudo apt update

Para instalar algunos paquetes en el dispositivo Raspberry Pi que son necesarios para la instalación de Tensor Flow, así que fue necesario regir una serie de pasos como se observa en la figura 65:

```
# install gdown to download from Google drive
$ sudo -H pip3 install gdown
# download the wheel
$ gdown https://drive.google.com/uc?id=1YpxNubmEL_4EgTrVMu-
kYyzAbtyLis29
# install TensorFlow 2.8.0
$ sudo -H pip3 install tensorflow-2.8.0-cp39-cp39-linux_aarch64.whl
```

Figura 65 Instalación de Tensor Flow 2.8.0

El siguiente mensaje de pantalla debería aparecer después de que se complete la instalación como se observa en la figura 66.

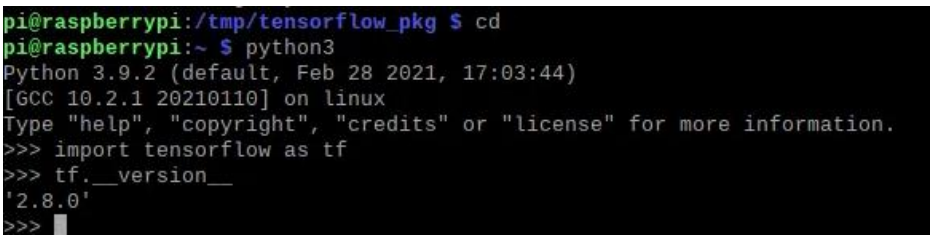


Figura 66 Mensaje de instalación satisfactoria.

Elaborado por: El Investigador

Inicialización y ejecución

Para la inicialización no es necesario realizar todo el proceso desde cero ya que se guardó el modelo entrenado con el nombre "model-cnn-facerecognition.h5" y "model-cnn-speechrecognition.h5", simplemente se creó un archivo de Python llamado Prinicipal.py en la cual se inició los modelos entrenados y se ejecutó en tiempo real como se observa en la figura 67.

Se creó un directorio principal /Tesis donde estará todo nuestro subdirectorios y programas ejecutables:

- Principal.py: Es el archivo principal del programa en formato Py. Este archivo se encarga de realizar el reconocimiento facial y voz al mismo tiempo de manera secuencial.
- Conexión.py: es el archivo que permite la conexión a la base de datos.
- Data set: aquí se almacenan los archivos de entrenamiento y validación. Este directorio se divide a su vez en subdirectorios que contendrán las imágenes y los audios de las personas a guardar en nuestra base de datos.
    - Imágenes: En este directorio se ubican las imágenes en formato .jpg que probará el sistema.
    - Audios: En este directorio se ubican los audios en formato .WAV que probará el sistema.
- model-cnn-facerecognition.h5: Es el modelo de la red neuronal convolucional.
- model-cnn-speechrecognition.h5: Es el modelo de la red neuronal convolucional.
- haarcascade_frontalface_default.xml: Es el modelo que permite realizar el reconocimiento facial mediante viola jones.
- dataset.xlsx: Archivo de Excel con la distribución de audios del sistema para el entrenamiento.

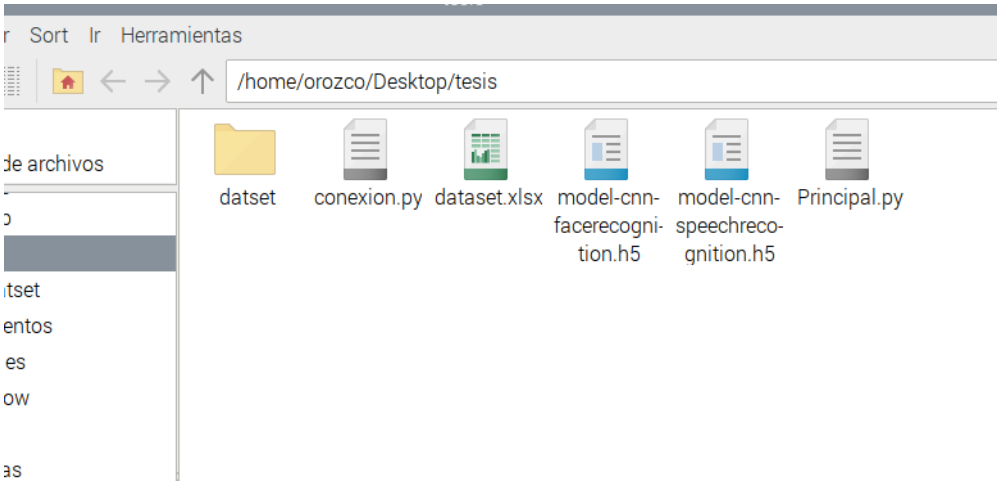


Figura 67 Directorio principal del programa.

Elaborado por: El Investigador

Instalación servidor LAMP

Para poder guardar los registros de las personas después del control de acceso fue necesario montar un servidor LAMP, que permitió crear una base de datos para el registro.

Se ejecutó los siguientes comandos en la Rasberry Pi para actualizarlo antes de comenzar el proceso de instalación:

sudo apt update & apt upgrade

Para instalar apache se ejecutó el siguiente comando:

sudo apt install apache2 -y

Una vez instalado Apache, se comprobó su funcionalidad. Para ello, se accedió a la dirección IP de Raspberry en un navegador y busque la página oficial de Apache, como se observa en la figura 68 que el servidor ya está funcionando. En mi situación, la dirección IP de Raspberry Pi es 192.168.1.107.

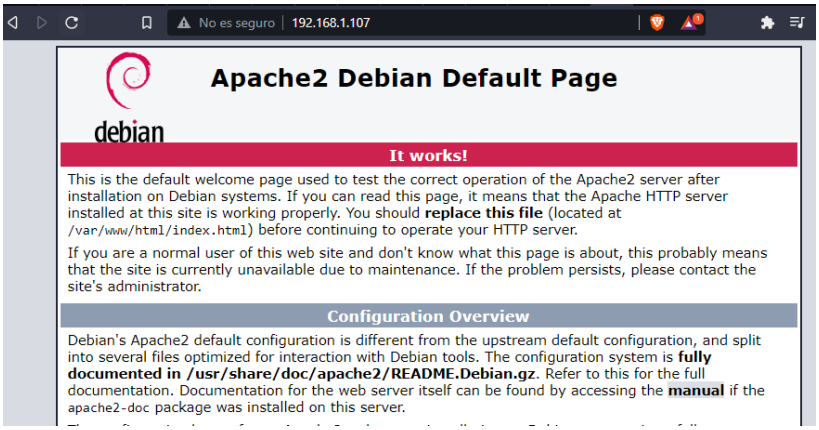


Figura 68 Instalación de Apache.

Elaborado por: El Investigador

También fue necesario instalar php, para lo cual se ejecutó el siguiente comando:

sudo apt install php -y

Para ver la correcta instalación de php se creó un archivo info.php que permitió ver la versión instalada en el sistema como se observa en la figura 69.

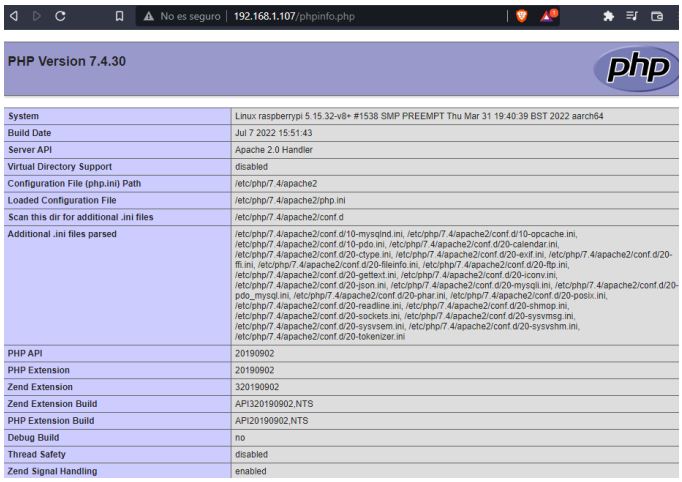


Figura 69 Instalación de php 7.4.30.

Elaborado por: El Investigador

Para el manejo de la base de datos se debe instalar después de los servidores apache y php, por lo que se decidió instalar MySQL mediante el siguiente comando:

sudo apt install mariadb-server

Con esto ya se instaló MYSQL, pero fue necesario realizar su configuración de gráfica, para eso se utiliza el código:

sudo mysql_secure_installation

Se verifico la instalación de la base de datos como se observa en la figura 70.

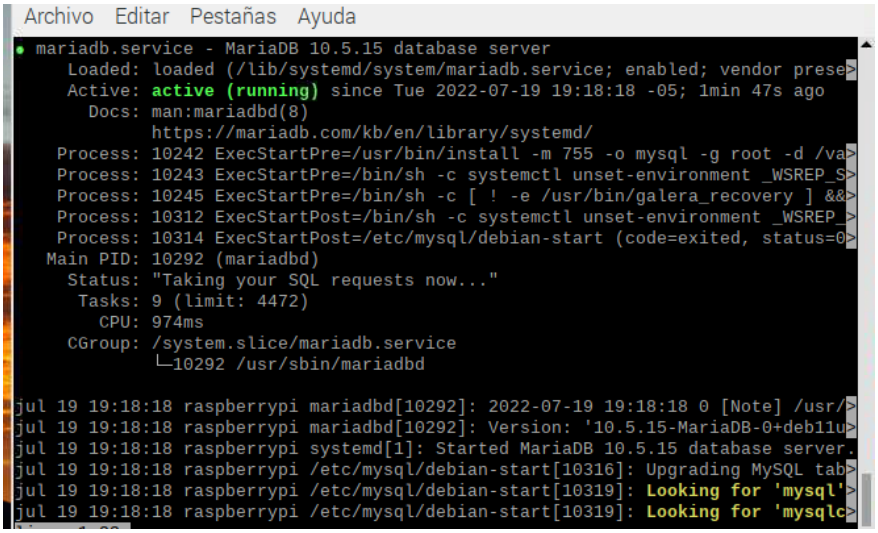


Figura 70 MariaDB funcionando correctamente.

Elaborado por: El Investigador

Creación de la base datos

Se creó una base datos con las características que se observa en la figura 71, donde se registró el nombre, fecha, la hora, código de acceso y el porcentaje de similitud detectado.

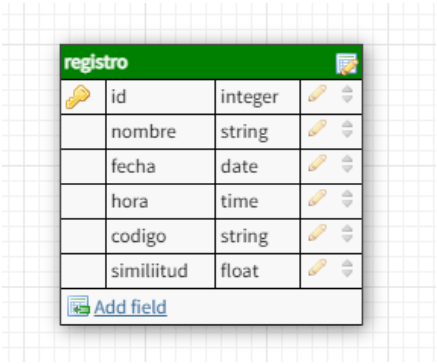


Figura 71 Base de datos del sistema.

Elaborado por: Investigador

Se creó la base de datos manualmente empleando comandos SQL, se puede emplear un gestor de base de datos, pero en esta ocasión solo se creó mediante el código SQL como se observa en la figura 72.

```
CREATE TABLE `registro2` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) NOT NULL,
  `fecha` date NOT NULL,
  `hora` time NOT NULL,
  `codigo` varchar(50) NOT NULL,
  `similitud` varchar(50) NOT NULL,
   PRIMARY KEY (Personid)
);
```

Figura 72 Código SQL para la base de datos.

Elaborado por: Investigador

Aplicación de notificaciones por Telegram

Se descargó la aplicación de Telegram desde la Play Store como se observa en la figura 73.

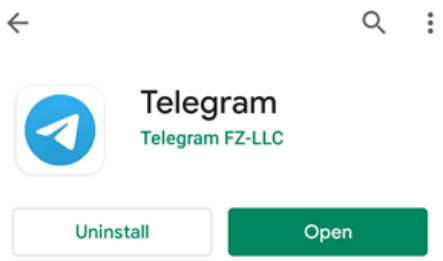


Figura 73 Instalación de aplicación Telegram.

Elaborado por: Investigador

Se abrió la aplicación y se buscó la palabra “botfather” para crear el bot como se observa en la figura 74.

Figura 74 Búsqueda del bot de telegram.

Elaborado por: Investigador

Una vez abierto el mensaje, se escribió el comando /start para iniciar la conversación como se observa en la figura 75.

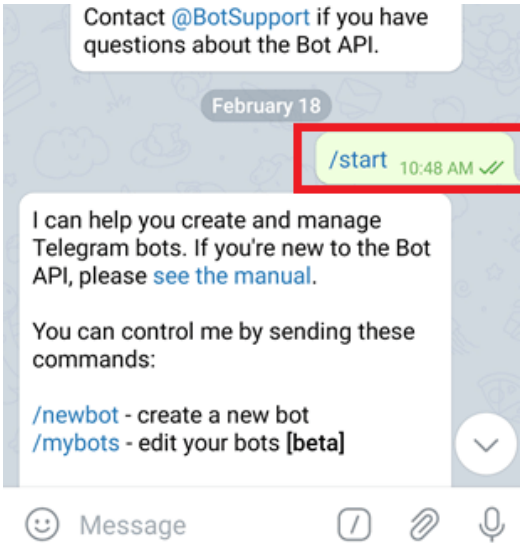


Figura 75 Creación del Bot de telegram.

Elaborado por: Investigador

Luego aparecerá una lista de comandos que se pueden usar para personalizar el bot. Escriba "/ newbot" para crear un nuevo bot, asígnele un nombre, luego ingrese el nombre de usuario (en este caso, FaceIDBot), y luego se creará el bot , como se observa en la figura 76.

Figura 76 Nombre del bot de telegram.

Elaborado por: Investigador

Recibirá un mensaje con un enlace para ver su Bot y **token** si su Bot se creó correctamente como se observa en la figura 77.

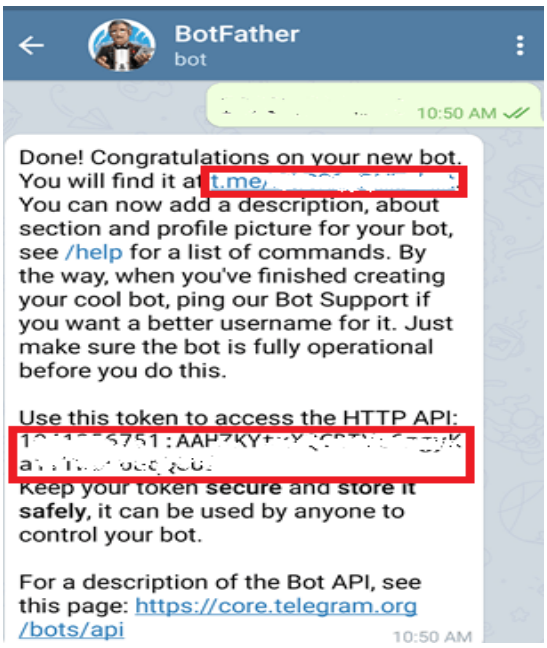


Figura 77 Token del bot a utilizar.

Elaborado por: Investigador

Después de esto se necesitó obtener el id de cada usuario para lo cual se debe buscar la opción IDbot y escribir /getid como se observa en la figura 78.

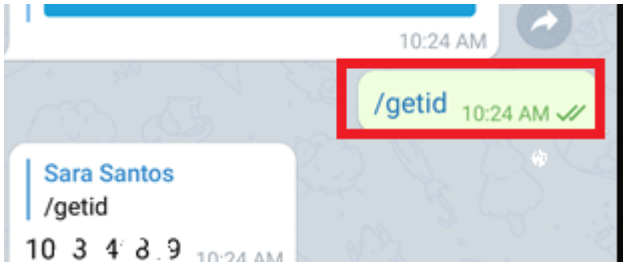


Figura 78 Obtención del id personal de Telegram.

Elaborado por: Investigador

La obtención del id se hizo en cada dispositivo de las personas registradas ya que el id es único para cada persona.

Implementación del prototipo

La implementación del prototipo permite controlar el acceso de las personas mediante dos métodos de autentificación, además se tiene el control y monitoreo del sistema desde una computadora personal, a través de una conexión ssh.

Como se observa en la figura 79, el dispositivo fue implementado en una caja hermética en la cual el usuario puede visualizarse a sí mismo durante el acceso, el sistema se encarga de detectar si la persona está registrada en la base de datos, si la persona es desconocida el sistema activa un mensaje de alerta. En la misma figura 79 se puede ver la colocación de la cámara delimitado por el círculo rojo y el micrófono delimitado por el círculo azul.

Figura 79 Implementación del sistema.

Elaborado por: Investigador

Se precedió a ensamblar el prototipo dentro de una caja hermética que sea capaz de proteger de los cambios del clima, como se observa en la figura 80.

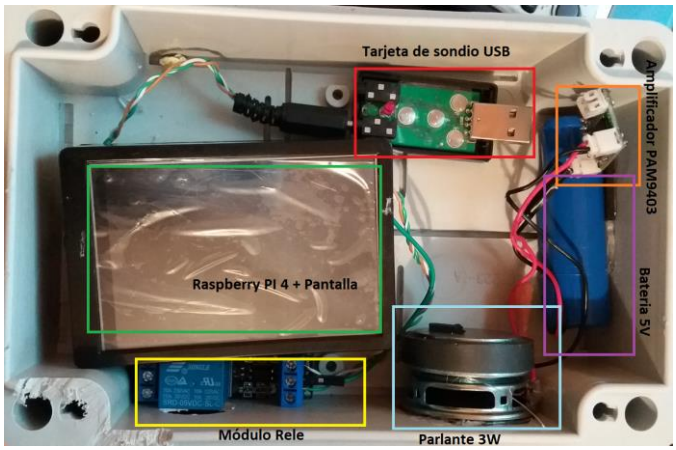


Figura 80 Conexiones internas del prototipo.

Elaborado por: Investigador

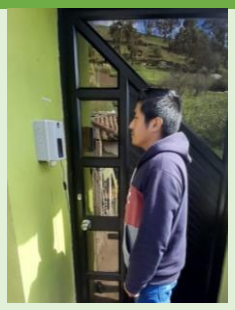
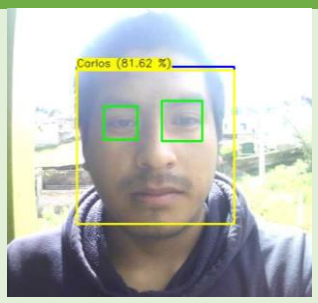
El escenario en el cual se implementó sistema fue en el acceso hacia la vivienda de la familia Orozco ubicado en la parroquia Ambatillo de la ciudad de Ambato, cómo se puede observar en la figura 81, la zona de implementación es una zona semicubierta en la cual el lugar cuenta con luz natural durante el día y luz artificial para el reconocimiento durante la noche.

Figura 81 Entorno de implementación del prototipo.

Elaborado por: Investigador

Se muestra el funcionamiento del sistema implementado, donde la persona debe colocar su rostro frente a la cámara y se visualiza a sí mismo en la pantalla, el sistema se encarga de capturar la imagen y realizar la predicción de si la persona está registrada en la base de datos para permitir el control. Después de reconocer el rostro el sistema pasa al segundo método de autentificación de voz donde el usuario deberá decir su clave personal, como se observa en la tabla 10.

Tabla 10 Implementación del Sistema de Control.

Autentificación para el usuario 1	
	
Autentificación para el usuario 2	

Autentificación para el usuario 3

Elaborado por: Investigador

Verificación de la hipótesis

En esta etapa se representa las pruebas de entrenamiento, predicción y resultados obtenidos de cada modelo de redes neuronales desarrollado en el anterior capítulo 3. Para ello se utilizó imágenes de las personas registradas en varios escenarios. Las pruebas de predicción sirvieron para comprobar el funcionamiento de los modelos creados desde cero para el reconocimiento de rostros y de voz. El objetivo de esta fase fue evaluar el desempeño del sistema después de la implementación. Se utilizó dos categorías, experimentos relacionados con la detección de rostros y experimentos relacionados con el reconocimiento de voz, para evaluar varios aspectos del rendimiento del sistema.

Este apartado representa la evaluación del rendimiento de cada red neuronal en el proceso de identificación y verificación de rostros y voz a través de dos pruebas básicas.

Evaluación del algoritmo de reconocimiento fácil

Una de las etapas más cruciales del sistema, es la detección de rostros ya que una mala detección puede dar cuásar errores en las siguientes etapas. Como ya se ha indicado, se utilizó la propia base datos generada en la sección anterior, con la que mejor se ha probado el funcionamiento de la red.

En el sistema se utilizó el método Viola-Jones para detectar rostros, lo que requiere que las personas aparezcan en la cámara de frente y tengan el rostro limpio y sin adornos para que la detección sea lo más precisa posible. El sistema funciona en tiempo real como se observa en la figura 82.

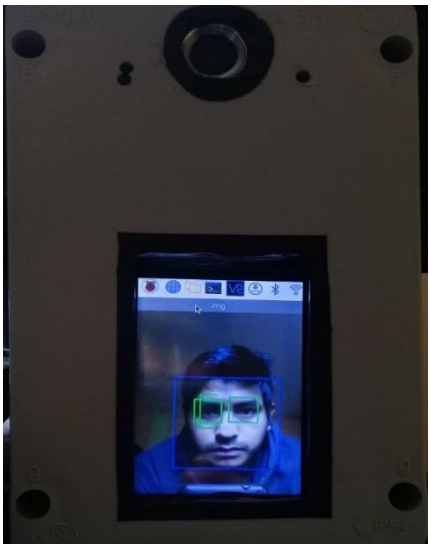


Figura 82 Detección de rostros mediante viola-jones.

Elaborado por: Investigador

Las pruebas que se llevó a cabo están relacionadas con el nivel de umbral y la distancia del rostro hacia la cámara con el fin de identificar la distancia correcta, no solo para el reconocimiento facial sino también para el reconocimiento de voz.

La primera muestra los resultados del entrenamiento de la red, en el cual se analizó las gráficas con los parámetros de medición en función de la optimización “Perdida” y la medición de “precisión”. También se muestra la matriz de confusión lo cual muestra el número de aciertos y fallos que tuvo la red.

La segunda prueba está asociada con la variable "determinación del umbral de confianza óptimo", lo que indica que la siguiente etapa es elegir el nivel de confianza más bajo para hacer predicciones.

La tercera prueba implica la variable "evaluación de la distancia de reconocimiento del rostro ", lo que indica que los datos sobre el reconocimiento se recopilan colocando al individuo de prueba a varias distancias de la cámara.

Experimento 1: Entrenamiento de la red

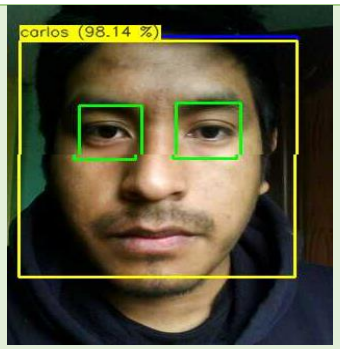
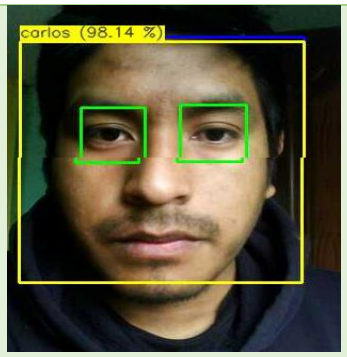
En esta prueba se realizó el entrenamiento de la red con la imagen procesada, con el objetivo de verificar el funcionamiento de la técnica de aprendizaje implementada en el proyecto de tesis.

Prueba 1 Pruebas de predicción con diferentes entrenamientos

Se realizó las pruebas de predicción para comprobar el funcionamiento del modelo por medio del algoritmo de viola jones. Este algoritmo se encargó de utilizar imágenes destinadas para la validación y predicción de resultados.

Se puede observar una prueba de un usuario registrado en la base de datos, además se incluye una comparativa entre una imagen sin procesamiento y con procesamiento de imágenes en el proceso del entrenamiento, como se observa en la tabla 11.

Tabla 11 Prueba de predicción de un usuario registrado.

Imagen sin procesamiento	**Imagen con procesamiento**
Entrenamiento: sin procesamiento	Entrenamiento: con procesamiento
Prediccion : (Desconocido)	Prediccion: Carlos
	

Elaborado por: Investigador

En la primera imagen de la tabla 11 se puede ver que el sistema realizo el reconocimiento facial cuando la imagen no es procesada pero con un porcentaje aceptable de predicción, a diferencia de la segunda imagen que la detección fue de manera correcta inclusive con poca iluminación con un valor del 98%.

Para el entrenamiento se realizó aplicando diferentes valores de "batch" y "epochs" con el objetivo de encontrar la mejor opción que permita conseguir el mejor valor de "accuracy" que pueda reconocer de manera confiable.

En la tabla 12 se observa las combinaciones que se emplearon durante el entrenamiento:

Tabla 12 Valores de Batch y Epochs a entrenar.

Batch	Epochs
32	30
64	15

Elaborado por: Investigador

Se escogió estos valores ya que los parámetros batch_size de 32 y 64 son valores constantes, y para poder encontrar el valore de los epochs se dividió el total de muestras a entrenar (1000 imágenes) dando un valor de 30 y 15 respectivamente.

Análisis de experimento 1

Los resultados que se mencionan a continuación se refieren a los valores del proceso de entrenamiento y desempeño de la red.

Proceso de entrenamiento

30 epochs y Batch Size de 32

El proceso del entrenamiento se realizó con procesamiento el cual consto de 30 "epocas" o "epochs" con el parámetro batch_size 32. Los parámetros de evaluación empleados para medir el resultado del entrenamiento fueron el "accuracy" y la función "Loss".

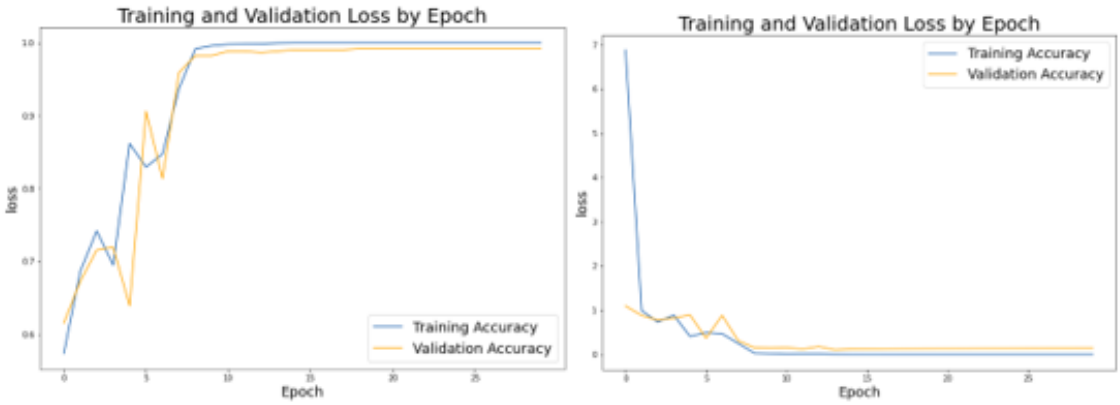


Figura 83 Gráfica de Accuracy con procesamiento.

Elaborado por: Investigador

A partir de los resultados de la gráfica 83, se pueden ver tasas de precisión son superiores al 97 por ciento, lo que indica un alto nivel de diagnóstico del modelo. Como se observa en la figura 83(a) del Accuracy, en los primeros tiempos (epoch=9) se alcanza una precisión de más del 90%, momento en el cual comienza a estabilizarse. En cuanto a la figura 83(b) de Loss (perdida), se puede observar que el modelo reduce rápidamente su pérdida hasta la época 7, momento en el que comienza a estabilizarse.

Para conocer el desempeño de la red se elaboró una comparativa de las métricas de precisión (accuracy), recall y F1-score para cada usuario registrado, como se observa en la tabla 13.

Tabla 13 Tabla de métricas de medición de la red.

	precision	recall	f1-score	support
Carlos	0.98	0.98	0.98	161
Diego	1.00	0.99	1.00	160
Javier	0.99	0.98	0.98	138
Jessenia	0.98	0.99	0.99	141
accuracy			0.99	600
macro avg	0.99	0.99	0.99	600
weighted avg	0.99	0.99	0.99	600

Elaborado por: Investigador

De la tabla 13 se pudo extraer que el valor de precisión cuando se aplicó el entrenamiento con datos procesados fue de 0.99, además se puede extraer la métrica "specificity" de 0.98 para el modelo entrenado.

Matriz de confusión

Se puede utilizar la matriz de confusión, que muestra el número de casos, incluidos los verdaderos positivos, los verdaderos negativos, los falsos positivos y los falsos negativos, para examinar el rendimiento de un modelo en particular.

Para esta prueba se analizó el algoritmo que permite determinar si un usuario registrado que está realizando su identificación facial. Para esta prueba del grupo de 4 usuarios registrados se tiene las siguientes características, como se observa en la figura 84.

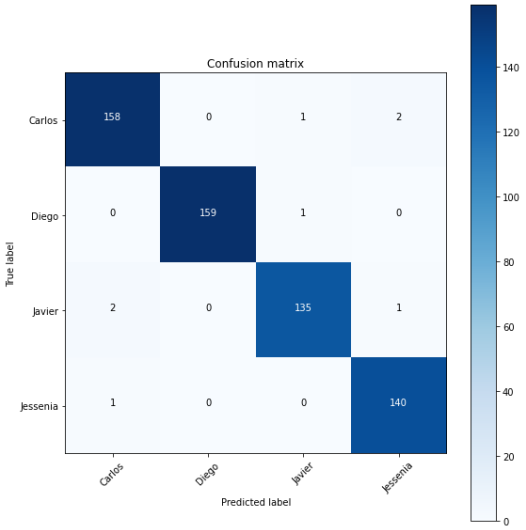


Figura 84 Matriz de confusión del modelo.

Elaborado por: Investigador

15 epochs y Batch Size de 64

El proceso del entrenamiento consto de 15 "epocas" o "epochs" con el parámetro batch_size 64. Los parámetros de evaluación empleados para medir el resultado del entrenamiento fueron el "accuracy" y la función "Loss".

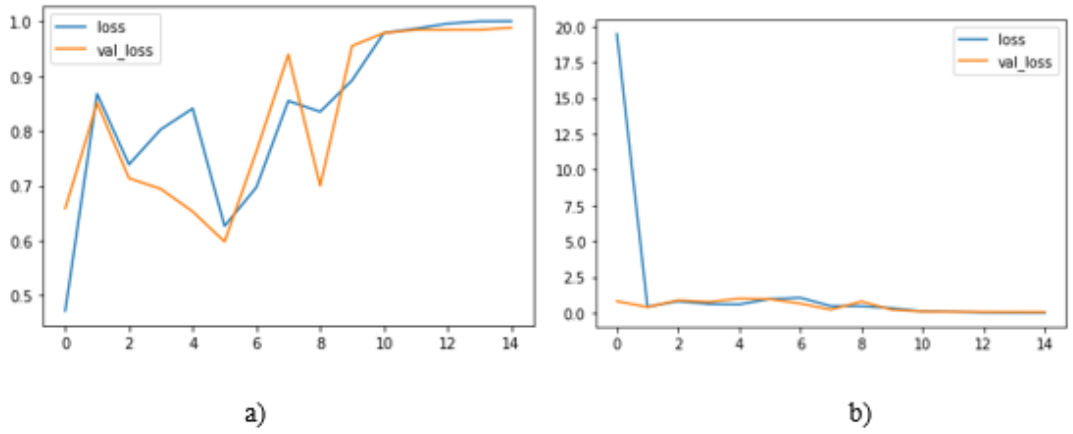


Figura 85 Imagen de accuracy y los batch.

Elaborado por: Investigador

De acuerdo a los resultados obtenidos de la figura 85, se pueden ver tasas de precisión superiores al 99 por ciento, lo que indica un alto nivel de diagnóstico del modelo. Como se observa en la figura 85(a) del Accuracy, en los últimos tiempos (epoch=10) se alcanza una precisión de más del 90%, momento en el cual comienza a estabilizarse. En cuanto a la figura 85(b) del Loss (perdida), se puede observar que el modelo reduce rápidamente su pérdida hasta la época 1, momento en el que comienza a estabilizarse.

De la figura 86 se pudo extraer que el valor de precisión cuando se aplicó el entrenamiento fue de 0.99, además se puede extraer la métrica "specificity" de 0.99 para el modelo entrenado.

	precision	recall	f1-score	support
Carlos	0.99	0.99	0.99	161
Diego	0.99	1.00	1.00	160
Javier	1.00	0.99	0.99	138
Jessenia	0.99	1.00	0.99	141
accuracy			0.99	600
macro avg	0.99	0.99	0.99	600
weighted avg	0.99	0.99	0.99	600

Figura 86 Tabla de métricas de precisión de la red.

Elaborado por: Investigador

Matriz de confusión

De igual manera para esta prueba se analizará el algoritmo que permite determinar si un usuario registrado que está realizando su identificación facial. Para esta prueba del grupo de 4 usuarios registrados se obtiene la matriz de confusión como se observa en la figura 87.

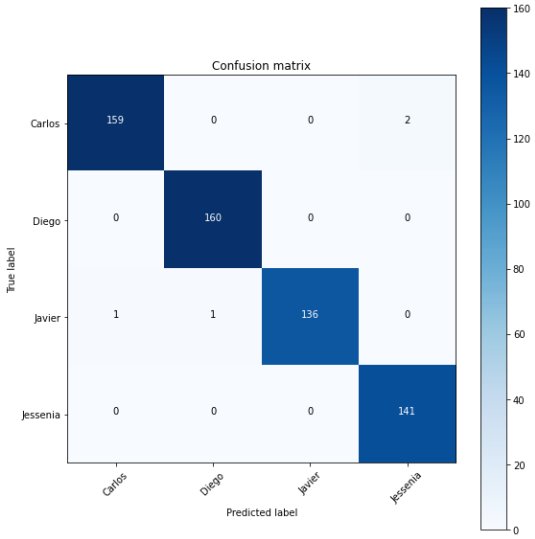


Figura 87 Matriz de confusión del segundo modelo.

Elaborado por: Investigador

Análisis de experimento 1:

Se puede ver la comparación de ambos modelos entrenados, esta prueba nos permitió ver qué modelo tiene un mejor desempeño y cual es más apto. Se realiza una comparativa de los valores de Accuracy, recall, F1-score y el tiempo de procesamiento, como se observa en la tabla 14.

Tabla 14 Comparación de los dos modelos entrenados.

Modelo		**Métricas de evaluación**				
epochs	Batch	Accuracy	precisión	recall	F1-score	tiempo
32	30	0,99	0,9875	0,985	0,99	3 horas
64	15	0,98	0,9925	0,995	0,9925	2 horas

Elaborado por: Investigador

Como se puede evidenciar la primera combinación es un modelo más confiable por el tiempo de entrenamiento y el nivel de precisión, aunque el segundo modelo no tiene muchas diferencias y también puede ser considerado para la implementación**.** Por lo tanto, se determinó utilizar la primera red entrenada para las próximas pruebas de evaluación.

Experimento 2: Fijación Umbral de confianza óptimo

Una vez realizado el entrenamiento de la red a emplear en el reconocimiento facial, el próximo paso es determinar el umbral de confianza más bajo para que realice las predicciones.

Para ello se tomó los de datos asignados a la validación, teniendo en cuanta que los datos no sean lo que se utilizaron el entrenamiento. Además, se hizo uso de otras imágenes de personas no registradas en la base de datos, con el propósito de determinar el umbral óptimo de sensibilidad.

Esta prueba consistió determinar el valor de umbral de predicción para el correcto funcionamiento de la red. Para lo cual se utilizó imágenes de las personas registradas y no registradas en diferentes ambientes.

Para validar la detección correcta se planteó tres escenarios:

- Asignación correcta: El clasificador dio la respuesta correcta a su predicción, que era sobre una persona con su nombre o le dio el valor "Desconocido" a una cara no identificada.
- Asignación incorrecta (identificación no reconocida): En este escenario, la identidad prevista de la persona registrada es "Desconocida" ya que se encuentra por debajo del nivel de confianza.
- Asignación incorrecta (identidad incorrecta): Esta situación surge cuando una predicción para una persona desconocida resulta ser uno de los individuos en la base de datos de entrenamiento.

El conjunto de datos de validación este compuesto por 5 personas, de las cuales 4 personas están registras y el 1 restante es desconocido como se observa en la siguiente tabla 15.

Tabla 15 Distribución de la cantidad de datos para validación.

Nombre	Cantidad de imágenes	Estado
Carlos	15	registrado
Luis	15	registrado
Jessenia	15	registrado
Diego	15	registrado
William	15	no registrado
Total	75	

Elaborado por: Investigador

Una vez realizada la prueba con diferentes imágenes se obtuvieron los siguientes resultados, que se observa en la tabla 16:

Tabla 16 Resultados de precisión del umbral óptimo.

Umbral de Confianza			
Precisión	Asignaciones correctas	Identidades no reconocidas	Identidades equivocadas
0,9	70	5	0
0,8	70	5	0
0,7	72	3	0
0,6	60	5	5

Elaborado por: Investigador

De la tabla 16 se puede extraer el análisis del umbral de precisión óptimo para 4 valores diferentes que permita obtener valores confiables. Se determinó que el valor umbral óptimo es 0.7 como se observa en la figura 88 ya que en este valor el número de aciertos fue el mayor con 72 correctas de un total de 75 imágenes, a diferencias de los valores de 0.8 y 0.9 que fueron 72 imágenes correctas.

Figura 88 Umbral de predicción a usuario 1.

Elaborado por: Investigador

Análisis de experimento 2

Como se observa en la figura 89 el resultado obtenido del umbral de confianza es de 0.7, en la cual este valor tiene el mayor número de aciertos correctas, el menor número de identidades equivocas y no reconocidas.

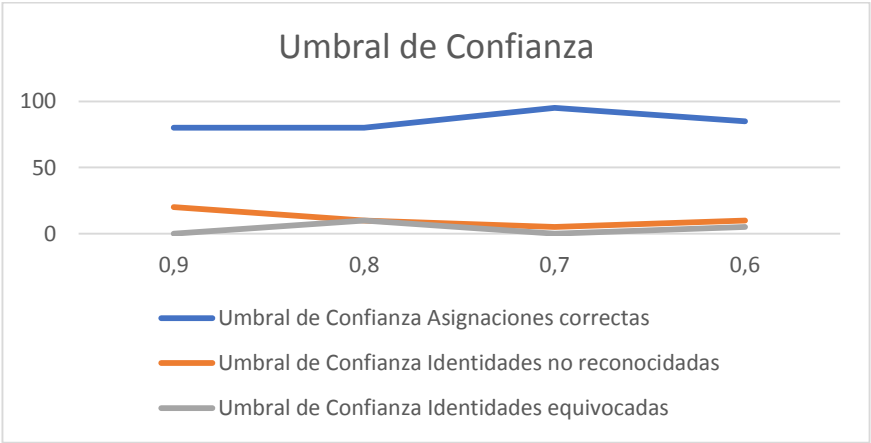


Figura 89 Resultado promedio de umbral de confianza óptimo.

Elaborado por: Investigador

En esta prueba se determinó usar el nivel de umbral de 0.7 como nivel mínimo óptico para el reconocimiento el cual se usó en los posteriores experimentos.

Figura 90 Nivel de precisión aplicado al usuario 1 y 3.

Elaborado por: Investigador

El resultado del nivel de precisión aplicado a los usuarios registrados 1 y 3, en la imagen se puede visualizar que el nivel mínimo de 0.7 se cumple para estos usuarios, como se observa en la figura 90.

Experimento 3: Evaluación de la distancia de reconocimiento del rostro

La tercera prueba corresponde a evaluar a que distancia el sistema el reconocimiento de rostros trabaja de manera adecuada. Para esta prueba se tomar como referencia el nivel de umbral 0.7 obteniendo en el anterior experimento.

Como el prototipo cuanta con un sistema de reconocimiento de voz, solo se realizaron pruebas con dos distancias, ya que el usuario debe estar lo más cerca posible del micrófono del sistema. La Tabla 17, muestra las especificaciones de las pruebas a llevarse a cabo.

Tabla 17 Valores para las pruebas de distancia.

Prueba	Detalle
distancia 1	50cm - luz normal
distancia 2	80cm -luz normal
distancia 3	100cm -luz normal

Elaborado por: Investigador

Como parámetros de evaluación se tomó en cuenta la precisión y el tiempo, como se observa en la tabla 18.

Tabla 18 Métricas de evaluación.

Métricas	Detalle
Precisión (%)	Un indicador probabilístico de qué tan exitosamente una prueba de clasificación binaria incluye o excluye una instancia dada.
Tiempo de Clasificación (seg)	La cantidad de tiempo que tarda cada método de aprendizaje en completar las tareas de clasificación y proporcionar los resultados

Elaborado por: Investigador

Prueba 1 Distancia

El sujeto debe mantener su rostro alineado frente a la cámara, el sistema se encarga de capturar el rostro y realizar la predicción, si supera el valor de umbral de 0.7 (umbral predeterminado) será un rostro conocido como se observa la figura 91.

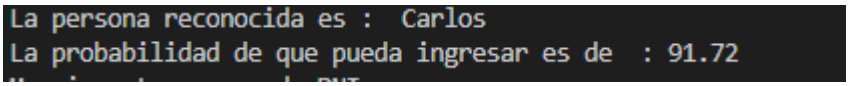


Figura 91 Porcentaje de Predicción de la persona reconocida.

Elaborado por: Investigador

La prueba a una distancia de 50cm del usuario 1 debe mantener el rostro frente a la cámara hasta que se realice el reconocimiento como se observa en la figura 92.

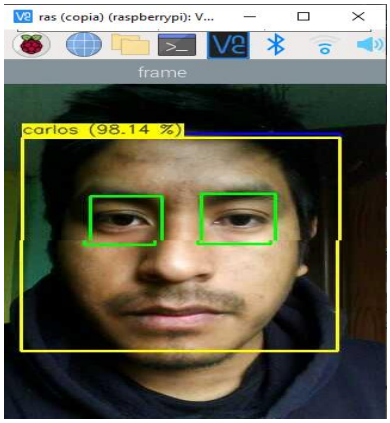


Figura 92 Individuo de prueba a la distancia 1.

Elaborado por: Investigador

Este procedimiento se realiza para todas las personas registradas a distintas distancias de prueba. Además, por la iluminación y enfoque del rostro el sistema genera un tiempo de espera para la predicción correcta. Los resultados obtenidos se presentan en porcentajes como se observa en la tabla 19.

Tabla 19 Resultado de precisión diferentes distancias.

	Distancia 50cm		**Distancia 80cm**		**Distancia 100cm**	
Usuario	precision (%)	tiempo(s)	precision (%)	tiempo(s)	precision (%)	tiempo(s)
Carlos	95,26	3,5	81,62	4,5	75,23	3
Javier	98,22	2,5	75,56	3	74,23	2
Diego	95,23	1,8	79,23	2,5	72,58	3,6
Jessenia	94,96	3,5	78,29	2,8	76,52	2,5

Elaborado por: Investigador

El resultado del contenido de la tabla 19, el cual permitió determinar la distancia para la colocación del rostro hacia la cámara. Como se observar en la figura 93 la distancia con mejores resultados de precisión fue la distancia de 50cm, el cual sirvió para realizar las pruebas de iluminación.

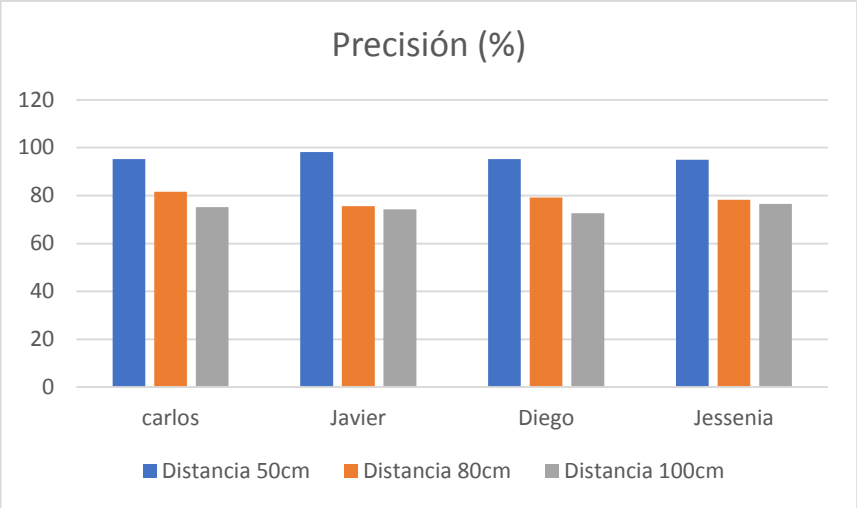


Figura 93 Resultados de precisión (%) en la prueba de Distancias

Elaborado por: Investigador

Se muestra el tiempo que se demora en reconocer los rostros, de igual manera como se puede evidenciar a la distancia de 50 cm el reconocimiento es más rápido que en las demás distancias, como se observa en la figura 94.

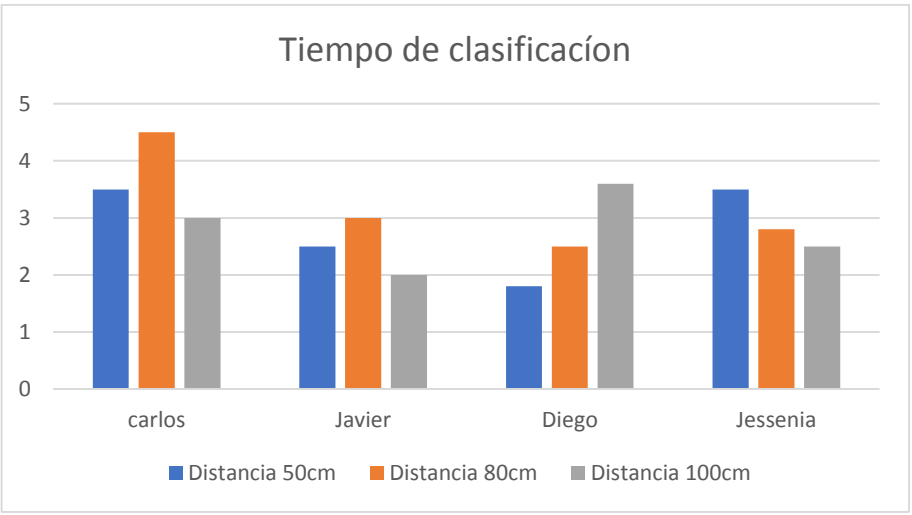


Figura 94 Resultados de tiempo(s) de clasificación en la prueba de Distancias.

Elaborado por: Investigador

Prueba 2 Iluminación

Además de la distancia se tomó en cuenta la iluminación del lugar donde se colocó el prototipo para el reconocimiento facial.

La dirección de la luz se estableció de acuerdo con los intervalos del día, que se establecen según las circunstancias de iluminación como se observa en la tabla 20. La distancia de medición permanece constante.

Tabla 20 Pruebas de iluminación a diferentes distancias.

Prueba	Detalle
Iluminación 1	8 am – 12 pm (Mañana)
Iluminación 2	14 pm 18 pm (Tarde)
Iluminación 3	7 pm - 4am (Noche y Madrugada)

Elaborado por: Investigador

Se presenta el resultado de la prueba para la iluminación 1, en un horario de 8 am a 11 am 2. Cabe recalcar que la distancia empleando con la primera iluminación fue de 50 cm obtenida en el anterior experimento como distancia confiable, como se observa en la figura 95.

Figura 95 Prueba de iluminación a usuarios.

Elaborado por: Investigador

Este procedimiento se realizó a los demás usuarios registrados. Los resultados obtenidos se observan en la tabla 21.

Tabla 21 Resultado de precisión diferentes horarios.

Iluminación						
usuario	Mañana		Tarde		Noche	
	Precisión (%)	tiempo(s)	Precisión (%)	tiempo(s)	Precisión (%)	tiempo(s)
Carlos	81,62	2,5	98,62	1,5	72,24	5
Javier	75,82	2,4	98,22	1,8	75,25	4,8
Diego	78,53	1,7	95,23	2,3	71,99	4,5
Jessenia	72,29	3	94,96	2,1	74,52	3,6

Elaborado por: Investigador

Como se observa en la figura 96, el resultado de precisión para detección en los diferentes horarios es diferentes, el horario cuando el reconocimiento es más preciso fue en la tarde con un promedio de 96.75% ya que en ese tiempo no existe mucha luz solar en el lugar del prototipo En los otros horarios se obtuvieron una precisión de 77.06% mañana y 73.5% noche, sim embargo estos valores son aceptables ya que cumplen adecuadamente con el valor del umbral de precisión.

Figura 96 Reconocimiento facial en horario nocturno.

Elaborado por: Investigador

Las Figuras 97 y 98, muestran gráficos de columnas agrupadas, lo que facilita la comprensión de la información de la Tabla 21.

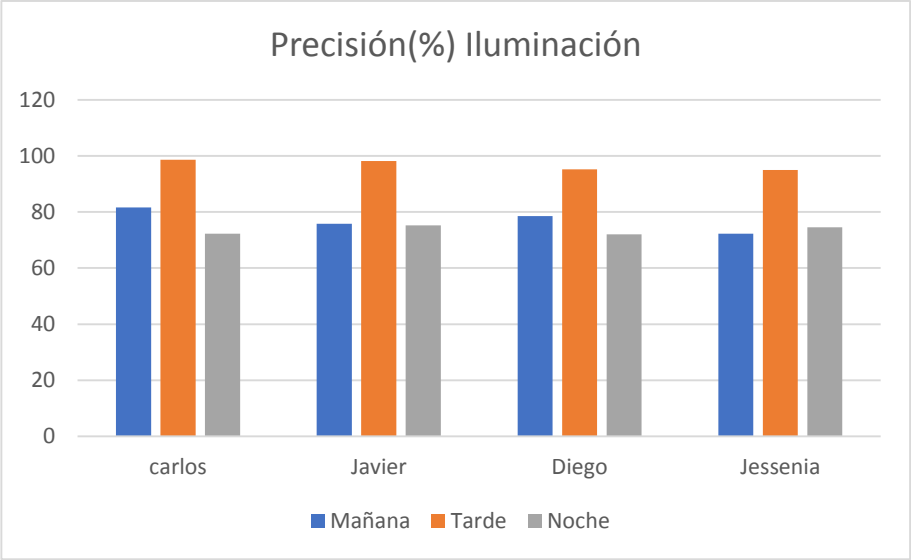


Figura 97 Precisión del prototipo (%) del reconocimiento facial con iluminación.

Elaborado por: Investigador

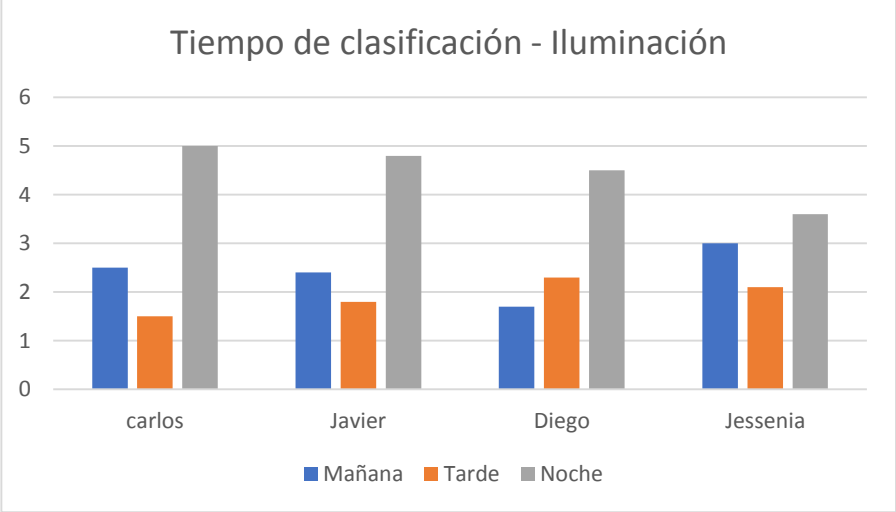


Figura 98 Tiempo de respuesta del sistema (segundos) del reconocimiento facial.

Elaborado por: Investigador

Análisis del experimento 3

Una vez realizado las pruebas, se procede a analizar los resultados obtenidos en las tres distancias analizadas.

La tabla 22 muestra los promedios de precisión de reconocimiento en las tres distancias analizadas.

Tabla 22 Resultados de precisión a diferentes distancias.

	Distancia 50cm		**Distancia 80cm**		**Distancia 100cm**	
	precisión(%)	tiempo(s)	precisión(%)	tiempo(s)	precisión(%)	tiempo(s)
Promedio	95,9175	2,825	78,675	3,2	74,64	2,775

Elaborado por: Investigador

- A una distancia de 50cm el reconocimiento facial es más preciso ya que no existe muchos obstáculos y movimientos. Esta distancia es la mejor distancia para realizar el reconocimiento facial.
- A una distancia de 80cm la precisión de reconocimiento es menor que la primera distancia con un valor de 78.67%. Esta fue una distancia óptima para utilizar durante el reconocimiento, ya que supera el nivel de umbral óptimo para el reconocimiento facial.
- A una distancia de 100cm la precisión es aceptable sin embargo el nivel de umbral de precisión es similar al aceptable lo cual podría generar las detecciones como personas desconocidas, no se utiliza detecciones desde esa distancia para evitar falsas detecciones.
- Al igual que las detecciones a diferentes distancias la más corta es la más precisa, el tiempo es más rápido a la mínima distancia.

Como el promedio más exacto es a una distancia de 50cm se estableció que la distancia a utilizar para las pruebas de iluminación es de 50cm.

Evaluación reconocimiento de voz

En este experimento se interpretarán los resultados del reconocimiento de voz, así como las pruebas que se realizaron a lo largo del desarrollo, que implicaron la verificación del sistema con la ayuda de los usuarios registrados. Se tabularán los resultados y se calculará el porcentaje de error del sistema.

Experimento 1: Entrenamiento de la Red Neuronal

Hay 4 usuarios registrados en la base de datos creada para este sistema, y se utilizaron 15 archivos de audio para cada usuario. Las palabras utilizadas son los cuatro números finales de su número de identificación para que la neurona tenga una gran capacidad de adaptabilidad a la hora del reconocimiento.

Tabla 23 Distribución de la base de datos de audios

id	genero	persona	test	frase
audio60.wav	1	carlos	test	6489
audio59.wav	1	carlos	test	6489
audio45.wav	0	diego	test	1052
audio44.wav	0	diego	test	1052
audio30.wav	0	jessenia	test	4291
audio29.wav	0	jessenia	test	4291
audio15.wav	0	javier	test	1279
audio14.wav	0	javier	test	1279
audio9.wav	0	javier	train	1279
audio8.wav	0	javier	train	1279
audio7.wav	0	javier	train	1279
audio6.wav	0	javier	train	1279
audio56.wav	1	carlos	train	6489
audio55.wav	1	carlos	train	6489
audio54.wav	1	carlos	train	6489
audio53.wav	1	carlos	train	6489
audio52.wav	1	carlos	train	6489
audio51.wav	1	carlos	train	6489
audio50.wav	1	carlos	train	6489
audio5.wav	0	javier	train	1279
audio49.wav	1	carlos	train	6489
audio48.wav	1	carlos	train	6489
audio47.wav	1	carlos	train	6489
audio46.wav	1	carlos	train	6489

audio41.wav	0	diego	train	1052
audio40.wav	0	diego	train	1052
audio4.wav	0	javier	train	1279
audio39.wav	0	diego	train	1052
audio38.wav	0	diego	train	1052
audio37.wav	0	diego	train	1052
audio36.wav	0	diego	train	1052
audio35.wav	0	diego	train	1052
audio34.wav	0	diego	train	1052
audio33.wav	0	diego	train	1052
audio32.wav	0	diego	train	1052
audio31.wav	0	diego	train	1052
audio3.wav	0	javier	train	1279
audio26.wav	0	jessenia	train	4291
audio25.wav	0	jessenia	train	4291
audio24.wav	0	jessenia	train	4291
audio23.wav	0	jessenia	train	4291
audio22.wav	0	jessenia	train	4291
audio21.wav	0	jessenia	train	4291
audio20.wav	0	jessenia	train	4291
audio2.wav	0	javier	train	1279
audio19.wav	0	jessenia	train	4291
audio18.wav	0	jessenia	train	4291
audio17.wav	0	jessenia	train	4291
audio16.wav	0	jessenia	train	4291
audio11.wav	0	javier	train	1279
audio10.wav	0	javier	train	1279
audio1.wav	0	javier	train	1279
audio58.wav	1	carlos	validacion	6489
audio57.wav	1	carlos	validacion	6489
audio43.wav	0	diego	validacion	1052
audio42.wav	0	diego	validacion	1052

audio28.wav	0	jessenia	validacion	4291
audio27.wav	0	jessenia	validacion	4291
audio13.wav	0	javier	validacion	1279
audio12.wav	0	javier	validacion	1279

Elaborado por: Investigador

En la tabla 23 se puede visualizar los archivos de audios empleados distribuidos en tres categorías: train, validation y test. Las pruebas que se llevaron a cabo se encargaron de verificar el proceso de entrenamiento y desempeño de la red.

Proceso de entrenamiento

El proceso del entrenamiento se realizó con un previo procesamiento de señales de audio explicado en el capítulo, para medir el funcionamiento de la red se entrenó la red con diferentes métricas de los valores de media de "epochs" y "batch_size", el cual nos permitió ver el resultado del "accuracy" y la función "Loss" de la red.

En la tabla 24 se muestra los valores utilizados durante el entrenamiento, como el entrenamiento para el reconocimiento de voz no se requiere de mucho consumo computacional se utilizó estas medias propuesto en el trabajo de Jassiel Rivera. [72]

Tabla 24 Valores de Batch y Epochs a entrenar.

Batch	Epochs
157	100
175	100
157	200
175	200

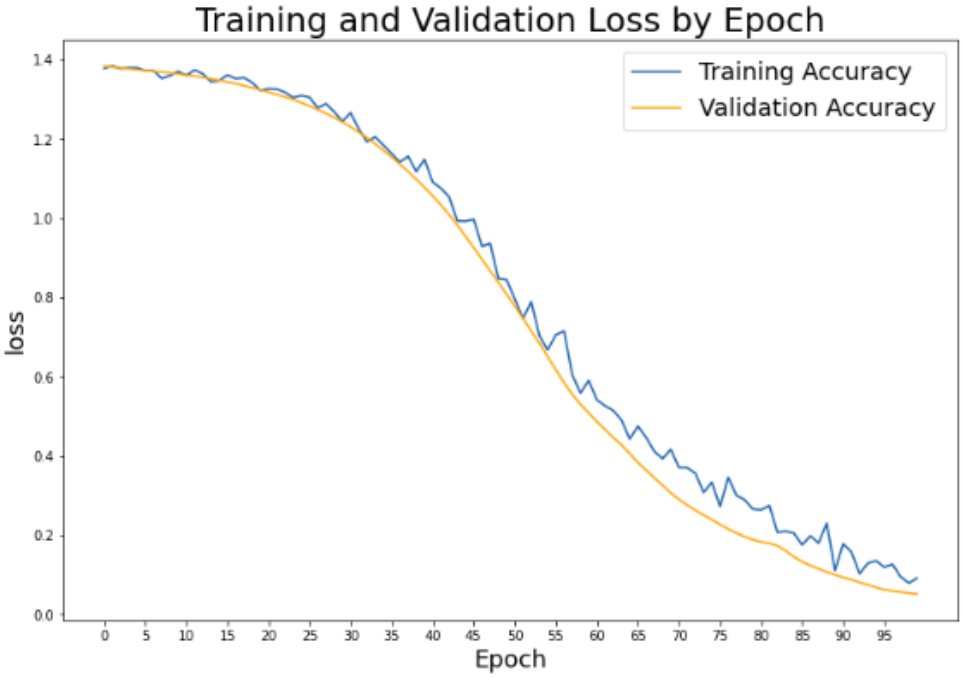
Elaborado por: Investigador

100 epochs y Batch Size de 157

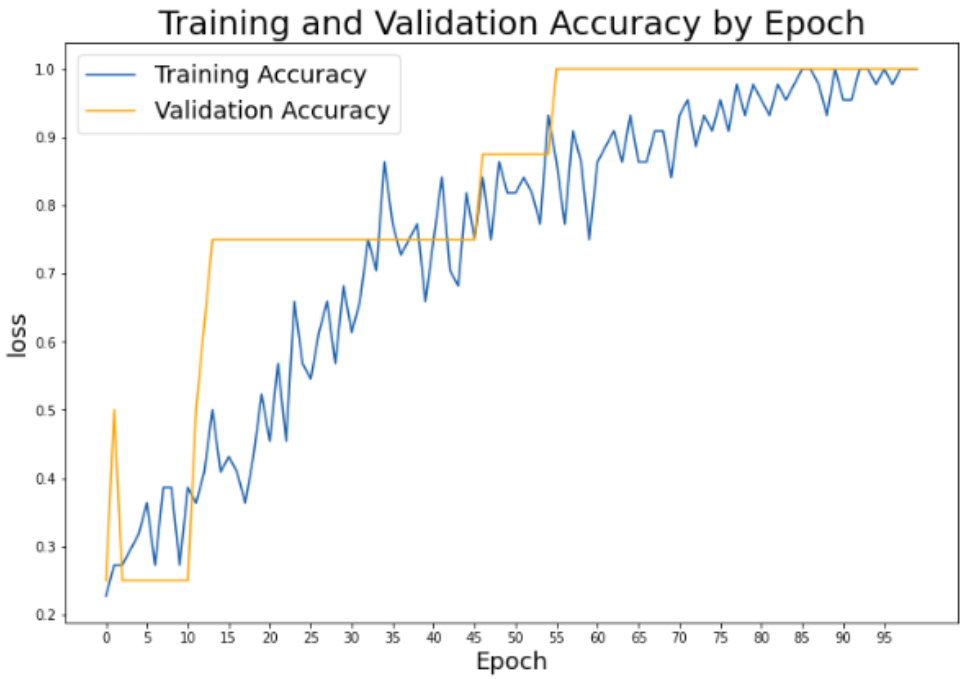
Tabla 25 Valores de accuracy y loss con batch 157.

	precision	recall	f1-score	support
carlos	1.00	1.00	1.00	16
diego	1.00	1.00	1.00	15
javier	1.00	0.94	0.97	16
jessenia	0.93	1.00	0.97	14
accuracy			0.98	61
macro avg	0.98	0.98	0.98	61
weighted avg	0.98	0.98	0.98	61

Elaborado por: Investigador



a)



b)

Figura 99 Imagen de accuracy y los batch 157.

Elaborado por: Investigador

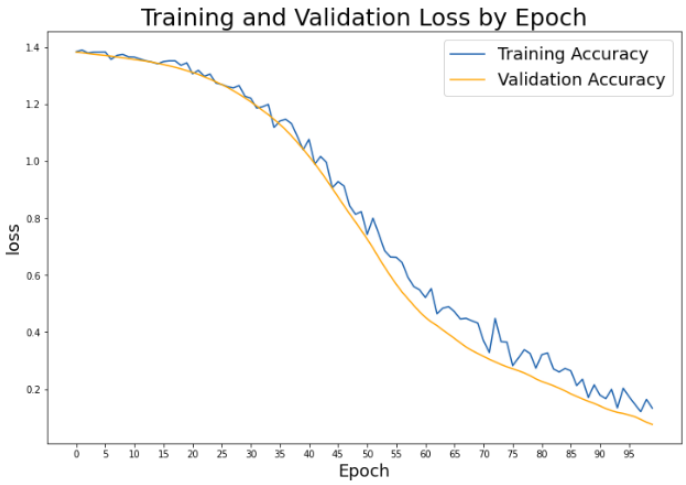
A partir de los resultados de la figura 99, se pueden ver tasas de precisión superiores al 98 por ciento, lo que indica un alto nivel de diagnóstico del modelo. La Tabla 25 y la Figura 107 dejan en claro que este modelo es una opción potencial para su uso. En cuanto al gráfico 99(a), se puede observar que el modelo reduce rápidamente su pérdida hasta la época 50, momento en el que comienza a estabilizarse. En cuanto a la figura 99(b), se puede observar que en los últimos épocas se alcanza una precisión de alrededor del 90%, momento en el cual comienza a estabilizarse.

100 epochs y Batch Size de 175

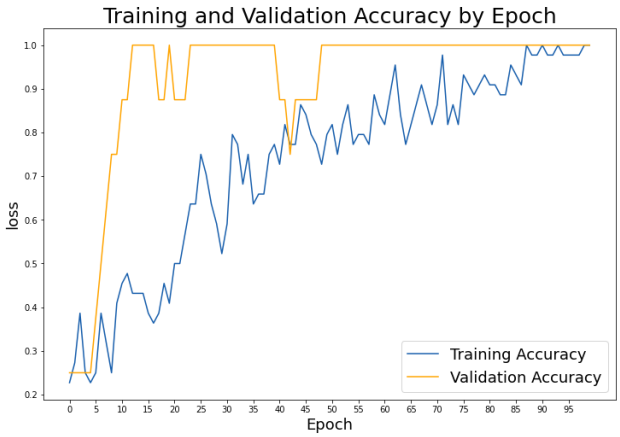
Tabla 26 Valores de accuracy y loss con batch 175.

	precision	recall	f1-score	support
carlos	0.44	0.88	0.58	8
diego	0.87	1.00	0.93	13
javier	0.93	0.67	0.78	21
jessenia	0.67	0.53	0.59	19
accuracy			0.72	61
macro avg	0.73	0.77	0.72	61
weighted avg	0.77	0.72	0.73	61

Elaborado por: Investigador



a)



b)

Figura 100 Imagen de accuracy y loss batch 175.

Elaborado por: Investigador

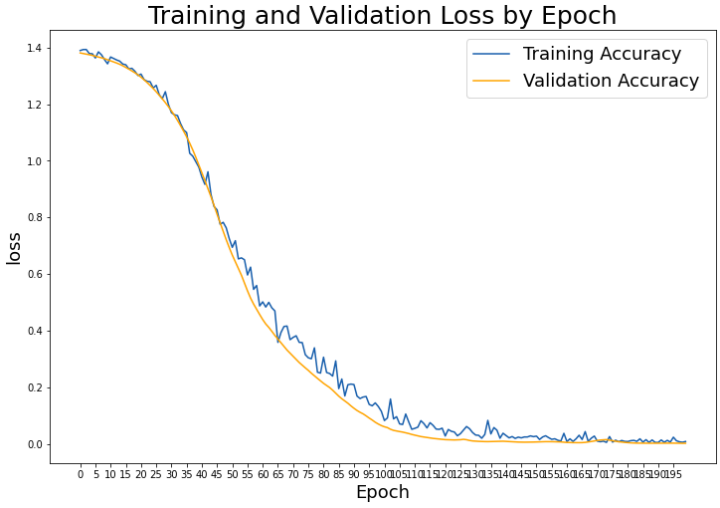
De los resultados obtenidos de la figura 100, se pueden ver tasas de precisión superiores al 72 por ciento, lo que indica un alto nivel de diagnóstico del modelo. La Tabla 26 y la Figura 100 dejan en claro que este modelo presenta problemas de aprendizaje y usos. En cuanto al gráfico 100(a), se puede observar que el modelo reduce rápidamente su pérdida hasta la época 75, momento en el que comienza a estabilizarse. En cuanto a la figura 100(b), se puede observar que en época 75 alcanza una precisión de alrededor del 90%, momento en el cual comienza a estabilizarse.

200 epochs y Batch Size de 157

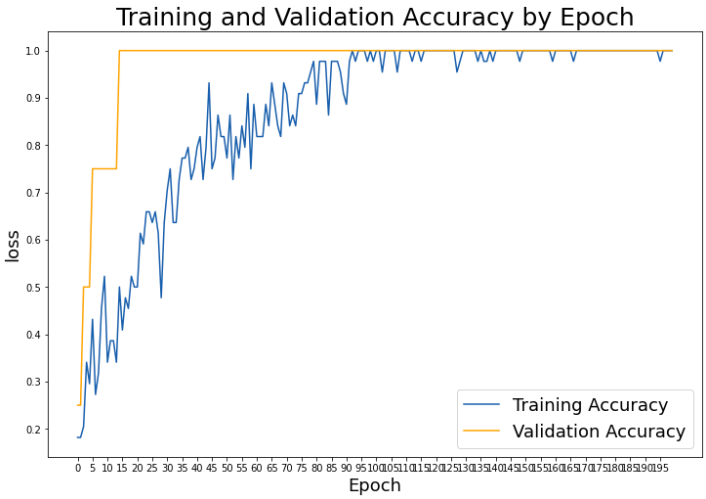
Tabla 27 Valores de accuracy y loss con batch 157.

	precision	recall	f1-score	support
carlos	0.94	0.83	0.88	18
diego	0.93	1.00	0.97	14
javier	0.93	0.82	0.87	17
jessenia	0.67	0.83	0.74	12
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.88	0.87	0.87	61

Elaborado por: Investigador



a)



b)

Figura 101 Imagen de accuracy y loss batch 157.

Elaborado por: Investigador

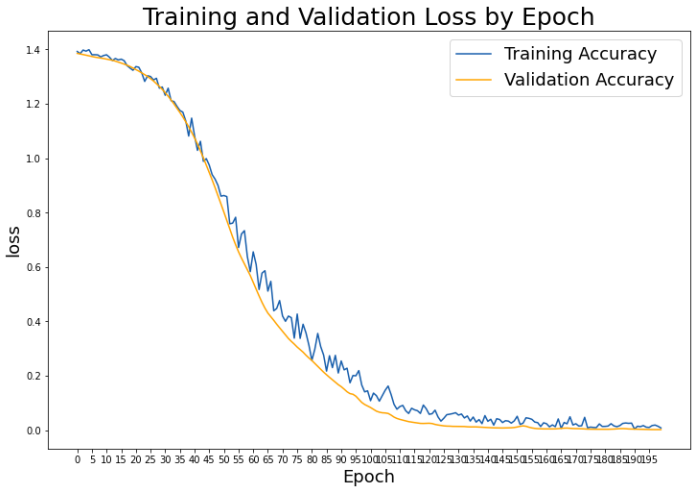
Se observa una precisión de más del 87 por ciento. Es posible ver un pequeño aumento en las pérdidas en la tabla 27 y la figura 101. La figura 101(a) muestra cómo la pérdida cae en el transcurso de 200 épocas y eventualmente se acerca a estabilizar desde la época 100. La figura 101(b) muestra cómo la precisión crece rápidamente en las primeras épocas y se mantiene casi constante a lo largo de las ultimas 100 épocas.

200 epochs y Batch Size de 175

Tabla 28 Valores de accuracy y loss con batch 175.

	precision	recall	f1-score	support
carlos	0.94	1.00	0.97	15
diego	0.93	1.00	0.97	14
javier	0.93	0.78	0.85	18
jessenia	0.80	0.86	0.83	14
accuracy			0.90	61
macro avg	0.90	0.91	0.90	61
weighted avg	0.90	0.90	0.90	61

Elaborado por: Investigador



a)

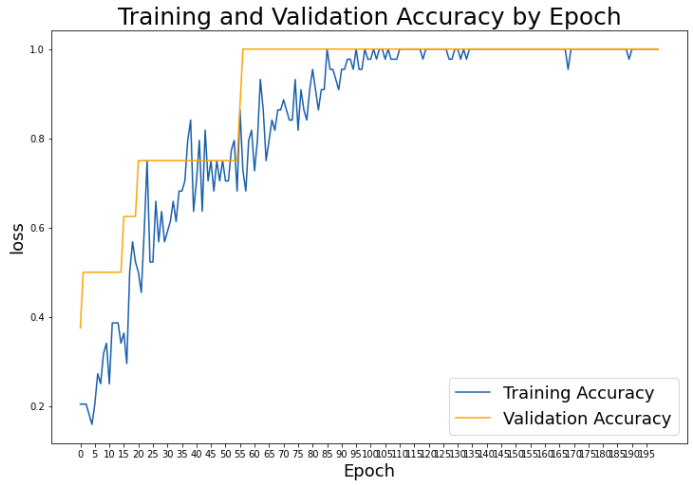


Figura 102 Imagen de accuracy y los batch 175.

Elaborado por: Investigador

Se observa una precisión de más del 90 por ciento. Es posible ver un pequeño aumento en las pérdidas en la tabla 28 y la figura 102. La figura 102(a) muestra cómo la pérdida cae en el transcurso de 200 épocas y eventualmente se acerca a estabilizar desde la época 100. La ffigura 102(b) muestra cómo la precisión crece rápidamente en las primeras épocas y se mantiene casi constante desde de las 100 épocas.

Comparación de los modelos de entrenamiento

En la tabla 25 se presenta la comparativa entre los 4 modelos entrenados a diferentes valores de epochs y batch basados en los promedios de los datos recolectados durante

el entrenamiento. Se evaluó el desempeño de los modelos para conocer la mejor opción de entrenamiento. En la tabla 29 se presenta los resultados obtenidos de los 4 modelos puestos a prueba.

Tabla 29 Comparación de los tres modelos entrenados.

Modelo		**Métricas de evaluación**			
epochs	Batch	Accuracy	precision	recall	F1-score
100	157	0,98	0,9825	0,9925	0,985
100	175	0,72	0,7275	0,77	0,72
200	157	0,87	0,8675	0,87	0,865
200	175	0,9	0,8	0,86	0,905

Elaborado por: Investigador

Como se puede evidenciar en la tabla 29 el modelo más apto y preciso fue el modelo número 1, por lo cual se usó este modelo para el sistema.

Experimento 2: Pruebas de predicción

Una vez analizado los diferentes valores de accuracy con las combinacion de los valores de bacth y epochs, se procedio a realizar las prebas de prediccion a la red escogida. A contuación se muestra las diferentes pruebas realizadas a la red.

La prueba consistió en cargar los archivos de audios destinado a los test (prueba), cada persona consta de 2 audios siendo un total de 8 muestras a analizar en la red neuronal. Se analizó de manera independiente ya que se necesitó ver el nivel de confiabilidad solo de esta red neuronal.

Para esta prueba se utilizó los audios destinados a validación y test de la red obteniendo los siguientes resultados como se observa en la figura 103.

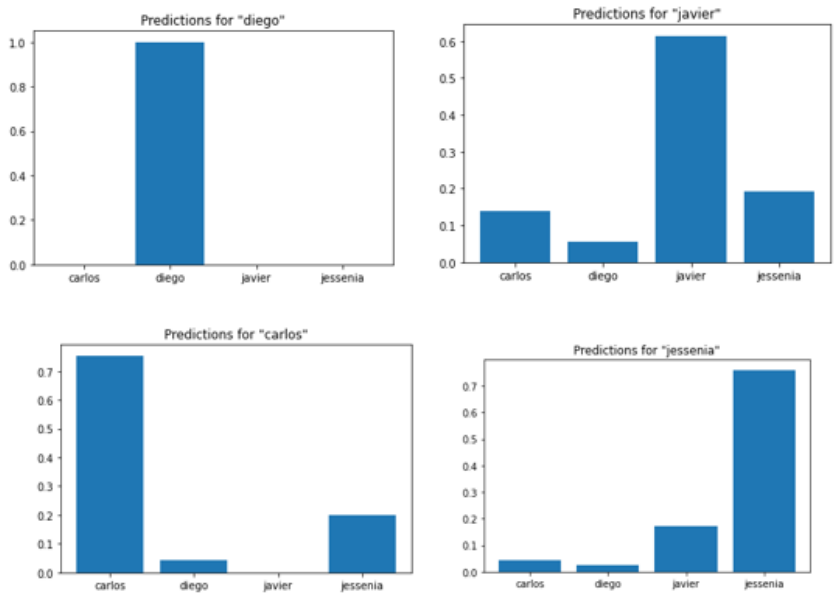


Figura 103 Resultados de predicción del reconocimiento de voz.

Elaborado por: Investigador

En la tabla 30 se muestra las pruebas realizadas a los usuarios con los diferentes audios de pruebas generador por el sistema.

Tabla 30 Resultados de precisión de reconocimiento voz.

Nombre	Audio	Precisión	Acceso	Tiempo(s)
Carlos	audio60.wav	0,85	si	2
Diego	audio44.wav	0,98	si	2
Javier	audio14.wav	0,25	no	2
Jessenia	audio29.wav	0,80	si	2
Carlos	audio59.wav	0,85	si	2
Diego	audio45.wav	0,80	si	2
Javier	audio15.wav	0,85	si	2
Jessenia	audio30.wav	0,45	no	2

Elaborado por: Investigador

Como se puede observar en la tabla 30 de las 8 pruebas realizadas a la red, 6 muestras muestran un nivel de confiabilidad del más del 90% y solo 2 muestras marcaron como desconocidas por debajo de 0.5. De estos resultados se tiene que la red tiene un 80 % de confiabilidad y un 20% de deficiencia como se observa en la figura 104. Para solucionar este problema se procedió a grabar nuevamente el audio con la frase, pero más despacio y entendible.

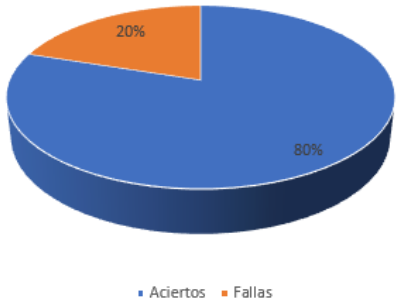


Figura 104 Resultados de predicciones

Elaborado por: Investigador

Confiabilidad

El poder verificar el correcto funcionamiento del prototipo fue necesario realizar pruebas que permitan obtener la confiabilidad del dispositivo frente a otros dispositivos del mercado, teniendo en cuanta que el dispositivo cuenta con una cámara encargada de capturar la imagen que está enfrente de la cámara para procesarla y compararla con la base del sistema , si la persona capturada registrada pasa a la autentificación de voz el cual deberá mencionar su clave registrada , si todo el proceso cumple de manera adecuada el sistema permitirá el acceso mandando una señal que permita activar la chapa magnética , si por el contrario el usuario no está identificado se el sistema enviara una señal de alerta avisando de la detección de una persona desconocida.

Para determinar el nivel de eficacia del sistema se realizan pruebas durante 3 días para los cual se obtiene la siguiente información.

Usuario Registrada - Identificada: Si una persona registrada en la base de datos se coloca frente a una cámara y es detectada como usuario registrado y dice su clave

personal correctamente permitiendo el paso, es un caso exitoso o también conocido como verdadero positivo**.**

Usuario Registrada – No Identificada: Si un usuario registrado pasa se coloca frente a la cámara y la detecta como desconocida o si la detecta, pero no reconoce su clave personal, a este caso se le conoce como fallido o falso negativo.

Usuario Desconocido – No Identificada: Si una persona que no está registrada en la base de datos se coloca frente a la cámara y se la detecta como desconocidas el caso se le conoce como verdadero negativo.

Como no se cuenta con un sistema biométrico que se venden en el marcado las pruebas de funcionamiento se realizan basadas en la norma ISO / IEC 19989-1:2020 el cual se encarga de evaluar y gestiona la seguridad de la información en las empresas. Esta norma fue desarrollada para poder de evaluar la seguridad integrada en el reconocimiento biométrico y la detección temprana de allanamientos o ataques que puedan afectar el sistema en la identificación biométrica. [73]

Según esta norma un sistema biométrico se deben considerarse varios aspectos como:

- Control de acceso único basado en la identidad.
- Control de acceso y seguridad de la información a alto nivel [61]
- Bajos costos de mantenimiento.
- Este tipo de tecnología debe estar avanzando ahora mismo.
- Crea la identidad corporativa y garantiza los más altos estándares de excelencia.
- Flexible y capaz de adaptarse a diferentes situaciones de trabajo.

Según estos parámetros se evaluó considerando cada uno de los ítems mencionados.

Se realizaron pruebas durante 3 días en los cuales para obtener un nivel de confiabilidad y para eso se aplica la siguiente ecuación 11:

$$x = \frac{\#\text{ de casos efectuados} * 100\%}{\#\text{ total de capturas efectuadas}} \qquad \textit{Ecuación 11}$$

Se realizar 3 pruebas al día para las 4 personas registradas y 3 pruebas al día de una persona no registrada en la base de datos. En general se obtuvieron 45 pruebas durante los 3 días de prueba con las personas registradas y no registradas, de los cuales 36 pruebas pertenecen a usuarios registrados y 9 pertenece a un usuario no registrado. En tabla 31 se muestra la cantidad de pruebas realizadas obtenidas de la base de datos.

Tabla 31 Registros de asistencia en la base de los datos.

id	nombre	fecha	hora	código	similitud	Reconocimiento Facial	Reconocimiento Voz	Intentos Voz	Acceso
1	Javier	16/7/2022	8:40:12	1279	75.56	si	si	1	si
2	Carlos	16/7/2022	8:42:45	6489	78.89	si	si	1	si
3	Diego	16/7/2022	8:45:02	1052	89.52	si	si	1	si
4	Jesseni a	16/7/2022	8:50:18	4291	75.23	si	si	1	si
5	Desconocido	16/7/2022	8:55:23	-	75,23	si	no	2	no
6	Carlos	16/7/2022	13:25:17	6489	85.12	si	si	1	si
7	Javier	16/7/2022	13:28:14	1279	75.26	si	no	3	no

8	Diego	16/7/2022	13:30:45	1052	82.53	si	si	1	si
9	Jesseni a	16/7/2022	13:32:23	4291	79.21	si	si	1	si
10	Desconocido	16/7/2022	13:35:52	-	0	no	no	0	no
11	Carlos	16/7/2022	19:10:23	6489	70.50	si	si	1	si
12	Javier	16/7/2022	19:15:45	1279	72.52	si	no	2	no
13	Diego	16/7/2022	19:17:29	1052	80.12	si	si	1	si
14	Jesseni a	16/7/2022	19:20:11	4291	75.46	si	si	1	si
15	Desconocido	16/7/2022	19:32:12	-	0	no	no	0	no
16	Carlos	17/7/2022	10:45:45	6489	80.12	si	si	1	si
17	Javier	17/7/2022	10:47:12	1279	65.15	no	no	0	no
18	Diego	17/7/2022	10:50:18	1052	72.56	si	si	1	si
19	Jesseni a	17/7/2022	10:53:23	4291	75.89	si	si	1	si
20	Desconocido	17/7/2022	10:55:14	-	0	no	no	0	no
21	Carlos	17/7/2022	14:15:05	6489	80.56	si	si	1	si
22	Javier	17/7/2022	14:20:15	1279	85.42	si	si	1	si
23	Diego	17/7/2022	14:22:45	1052	79.52	si	si	1	si

24	Jesseni a	17/7/2 022	14:25 :50	429 1	78.12	si	si	1	si
25	Descon ocido	17/7/2 022	14:28 :56	-	82,56	si	no	2	no
26	Carlos	17/7/2 022	20:05 :06	648 9	70.12	si	si	1	si
27	Javier	17/7/2 022	20:08 :15	127 9	55,23	no	no	0	no
28	Diego	17/7/2 022	20:10 :55	105 2	75.59	si	si	1	si
29	Jesseni a	17/7/2 022	20:15 :32	429 1	78.12	si	si	1	si
30	Descon ocido	17/7/2 022	20:18 :52	-	0	no	no	0	no
31	Carlos	18/7/2 022	9:10: 52	648 9	75.56	si	si	1	si
32	Javier	18/7/2 022	9:12: 15	127 9	81.23	si	si	1	si
33	Diego	18/7/2 022	9:15: 53	105 2	89.12	si	si	1	si
34	Jesseni a	18/7/2 022	9:20: 52	429 1	76.23	si	no	3	no
35	Descon ocido	16/7/2 022	9:25: 05	-	0	no	no	0	no
36	Carlos	18/7/2 022	14:15 :05	648 9	78.23	si	si	1	si
37	Javier	18/7/2 022	14:20 :15	127 9	82.50	si	si	1	si
38	Diego	18/7/2 022	14:22 :45	105 2	71.52	si	si	1	si
39	Descon ocido	16/7/2 022	14:23 :25	-	0	no	no	0	no

40	Jesseni a	18/7/2 022	14:25 :50	429 1	79.52	si	si	1	si
41	Carlos	18/7/2 022	20:05 :06	648 9	75.12	si	si	1	si
42	Javier	18/7/2 022	20:08 :15	127 9	74.59	si	si	1	si
43	Diego	18/7/2 022	20:10 :55	105 2	77.54	si	si	1	si
44	Jesseni a	18/7/2 022	20:15 :32	429 1	50,26	no	no	0	no
45	Descon ocido	16/7/2 022	20:18 :23	-	0	no	no	0	no

Elaborado por: El Investigador

Con estos datos obtenidos durante tres días se elabora la siguiente tabla 32:

Tabla 32 Resultados de las pruebas realizadas durante 3 días.

Resultados	# casos	Porcentaje
Usuario Registrada - Identificada	30	66,66%
Usuario Registrada – No Identificada	6	13.33%
Usuario Desconocido – No Identificada	9	20%

Elaborado por: El Investigador

Como se puede observar en la tabla 32 de las 45 pruebas realizadas a las personas registradas solo 30 personas se registraron de manera segura y sin problemas obtenido un 66.6%. A diferencia de la segunda métrica en la cual solo 6 pruebas de las personas registradas fallaron esto debido a que realizo la identificación de una persona registrada, pero a una persona incorrecta, para solucionar este problema el usuario

debió nuevamente a realizar la identificación nuevamente desde el inicio para obtener el registro correcto. Cuando una persona que no está registrada en la base se acerca al dispositivo las pruebas fueron satisfactorias ya que marco desconocido en todas las pruebas realizadas a otras personas como se observa en la figura 105.

Figura 105 Prueba a una persona desconocida.

Elaborado por: Investigador

Como se observa en la tabla 33 ,según las pruebas realizadas se determinó que el dispositivo tiene una confiabilidad del 80.6% y una inconsistencia del 19.4%, si analizamos estos datos obtenidos con un estudio llamado Labeled Faces in the Wild en el cual menciona que un dispositivo basado en biometría el nivel de confiabilidad de un dispositivo en el mercado es del 95% y 99% y dando un margen de inconsistencia de un 5% el resultado es aceptable para poder implementarlo a más viviendas. [74]

Tabla 33 Porcentajes totales de efectividad del sistema.

	prototipo	**dispositivo venta**
% de efectividad	80,6%	98%
% de inconsistencia	19,4%	5%
% total	100%	100%

Elaborado por: El Investigador

Notificaciones aplicación telegram

Después de realizar la autentificación de las personas y el registro de este, la imagen captura en ese momento es enviada a la aplicación de mensajería telegram junto con los datos la fecha y hora. Como se observa en la figura 106 el sistema registra y notifica mediante un mensaje de texto el registro realizado en ese momento.

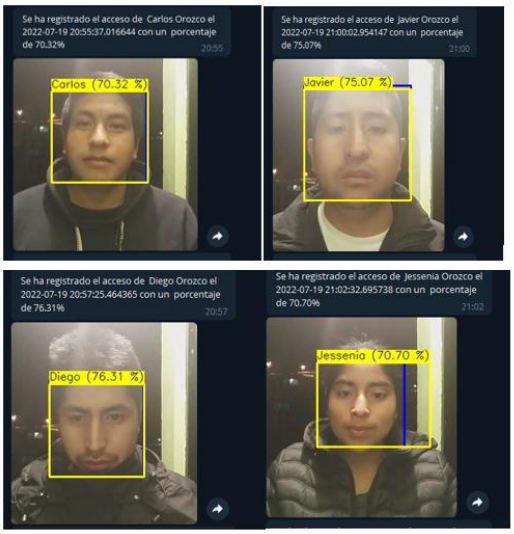


Figura 106 Datos en Aplicación de Mensajería.

Elaborado por: Investigador

Al igual, cuando detecta una persona desconocida el sistema tiene la capacidad de una alerta sobre posibles intrusos hacia el interior como se observa en la figura 107.Se presenta el código anexo K.

Figura 107 Datos en Aplicación de Mensajería.

Elaborado por: Investigador

Costos del Prototipo

Para la implementación del prototipo se requirieron los siguientes materiales, los cuales son detallados en la tabla 34. El costo de los materiales utilizados en la implementación del sistema de control de acceso tiene un valor de 321.16 dólares incluido IVA de cada uno de los materiales.

Tabla 34 Precios del Hardware del prototipo.

Ítem	Detalle	Cantidad	Precio Unitario(dólares)	Total (dólares)
1	Raspberry pi 4	1	157	$ 157,00
2	Raspberry Pi Pantalla	1	24	$ 24,00
3	Fuente de alimentación	1	10	$ 10,00
4	Case	1	10	$ 10,00
5	Cámara Usb	1	20	$ 20,00
6	Micrófono 3,5Mm	1	5	$ 5,00
7	Tarjeta de sonido USB	1	10	$ 10,00
8	Módulo relé	1	3	$ 3,00
9	Batería	1	2,5	$ 2,50
10	Cerradura Eléctrica	1	10	$ 10,00
11	Plug Audio 3,5Mn	1	1	$ 1,00
12	Amplificador PAM8403	1	3	$ 3,00
13	Estaño	1	0,75	$ 0,75
14	Caja plástica	1	7,5	$ 7,50
15	Disipadores	1	3	$ 3,00
16	Silicona	1	2	$ 2,00
17	Taype	1	2	$ 2,00
18	Conversor HDMI	1	3	$ 3,00
19	Parlante 3W	1	10	$ 10,00
20	Cables	10	0,1	$ 1,00
21	Cable Luz	2	1	$ 2,00
			Subtotal	$ 286,75
			IVA (12%)	$ 34,41
			Total	$ 321,16

Elaborado por: Investigador

Precio del Software

Los Softwares utilizados son gratuitos detallados a continuación:

- IDE Python
- Visual Studio Code
- Raspberrry PI OS
- LAMP

Precio de mano de obra

Además, al costo de total del prototipo se debe incluir los gastos de transporte que corresponde a 80 dólares, gastos de materiales de oficina que son 30 dólares y un sueldo básico de ingeniería que está en 800 dólares, este último costo se incluye ya que el investigador invirtió tiempo en la investigación y desarrollo del sistema, dando como resultado un valor total de 1286.04 dólares como se observa en la tabla 35.

Tabla 35 Valor de implementación del sistema biométrico multimodal.

Ítem	Precio(dólares)
Prototipo	376,04
Transporte	80
Materiales de oficina	30
Salario	800
Total	1286,04

Elaborado por: Investigador

En el lado comercial en la actualidad existen varios dispositivos que permiten realizar el reconocimiento de facial, pero la mayoría de estos tiene como segundo método de autentificación la huella digital o tarjeta RFID y no tiene el reconocimiento de voz, también estos biométricos comerciales no tiene la capacidad de notificaciones en tiempo real al usuario. Si analizamos esto para implementar un sistema se necesitaría comprar dos sistemas de, uno biométrico facial y un biométrico de voz (Alexa). Un ejemplo es el control de acceso con reconocimiento Facial de la marca Hikvision que

tiene un valor de $250 que ofrece un nivel de confiabilidad más segura pero no cuenta con el reconocimiento de voz para lo cual se debería adquirir otro sistema como por ejemplo un Alexa con un precio de $100 cómo se observa en la figura 108, que se encargaría de realizar el control automatizado de la entrada de ingreso. Estos son dispositivos muy sofisticas y más precisos, pero si se desea implementar se debería instalar ambos dispositivos por separado.

Figura 108 Asistente para el control de acceso.

A continuación, se muestra una comparativa de los sistemas biométricos más comerciales en la actualidad.

Tabla 36 Comparación de los sistemas biométricos más utilizados.

Nombre	Marca	costo	hardware	software	Imagen
Biométrico DS-K1T341AMF facial	HIKVISION	250,00	Cámara 2 MP con lente doble, de gran angular. Alimentación: 12 VCD - 2 Amp - 24 W Cerradura Eléctrica: 1 USB 2.0: 1 RS-485: 1 Botón de salida: 1	Soporta: 1,500 rostros. Soporta: 1,500 huellas. Soporta: 1,500 tarjetas (Mifare). Adopta un algoritmo DEEP LEARNING.	

				Comunicación IP (10/100/1000 Mbps)	
control de personal DHI-ASI4214Y	DAHUA	190,00	Voltaje de entrada 12V DC 2A. Consumo energético 12W. Display touch capacitivo 7". Doble lente de 2MP. Conexión RS-485.2 Entradas de alarma. 2 salidas de alarma. 1 puerto USB	Distancia de lectura de tarjeta 1cm – 5 cm. Altura de humanos 1.1mts – 2.4mts. Distancia de reconocimiento facial .3mt – 2mts. 2,000 huellas, rostros. 50 administradores.	
Biométrico K20	ZKTECO	195,00	Diseño Elegante y Moderno. Pantalla TFT LCD a Color de 2.8 Pulgadas. Cerradura Eléctrica Botón de Salida Alarma	Interfaz TCP/IP y Puerto USB Host. Salida de Reportes por USB en Formato de Excel. Multilenguaje. Incluye Software Lite para	

				Gestión de Asistencia	

Elaborado por: Investigador

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Se implementó un sistema del control de acceso de personal basado en reconocimiento facial y voz, donde se aplicó técnicas de aprendizaje automático como lo son las redes neuronales convolucionales. A pesar de que ya existen la aplicación redes neuronales entrenadas empleados en los trabajos referenciales, la creación una red desde cero permito obtener valores más precisos en una Raspberry pi 4 de 8GB, lo que permitió realizar su uso en tiempo real con la propia base de datos. Se presenta un sistema biométrico multimodal de control de bajo coste en comparación a los trabajos analizadas que solo presentan un método de autentificación.

- Usar el método de transformaciones afines para el procesamiento de imágenes ayudo considerablemente a reducir el trabajo de entrenamiento de la red a menos de tres horas, ya que este método permite modificar los parámetros de la imagen en rotación, escalado, recorte y escala grises , al igual que el uso del método de MFCC para el tratamiento de audios por medio de la librería LIBROSA permitió obtener 193 indicadores para cada audio en una matriz de datos, es facilito que el procesamiento y entrenamiento de la red neuronal sea mucho más rápido que el reconocimiento facial, ya que los que se entrena es una matriz de datos de audios y no una imagen procesada.

- Como conclusión se tiene que la red neuronal convolucional creada para el reconocimiento facial presenta un rendimiento de precisión de más de 95% y para el reconocimiento de voz con un precisión de más de 90%, logrando obtener valores más que confiables y aceptables para la implementación del sistema. A pesar de que ya existen maneras más sencillas de aplicar reconocimiento facial como los algoritmos de Eigenfaces, Fisherfaces, LBPH

aplicados en los trabajos referenciales, estos no son tan precisos ya realizan la detección de rostros donde no se encuentra ninguna persona, ese problema se reduce con las aplicaciones de un procesamiento de imágenes y las redes neuronales, logrando obtener así una mejor de precisión en el prototipo.

- En el presente trabajo se presentaron varios problemas durante la implementación. El problema radica en la compatibilidad de librerías y módulos a la hora de instalar las dependencias en una tarjeta de bajo coste, ya que el funcionamiento de una red neuronal en una computadora no es la misma en una tarjeta (Raspberry Pi), hay que tener mucho en cuenta que los tiempos y procesamiento de ejecución no son los mismos debidos a los recursos que posee cada dispositivo. Además, para una mejor captura de imágenes se utilizó una cámara con visión nocturna, pero por temas de compatibilidad de Raspberry Pi OS (bullseye) se descartó esa posibilidad.

4.2 Recomendaciones

- Se recomienda utilizar una tarjeta de desarrollo con las características más avanzadas de la actualidad para mejorar los tiempos de procesamiento y ejecución en tiempo real, o utilizar una computadora avanzada que reduzca el esfuerzo computacional durante el entrenamiento de las redes neuronales. Al utilizar una tarjeta de desarrollo se debe instalar un sistema de enfriamiento ya que el trabajo que realiza es muy pesado y esto causa que el sistema se recaliente.
- Para que el sistema tenga una mejor eficiencia se recomienda ejecutar el programa en una computadora avanzada como servidor local y colocar una cámara IP que sea capaz de transmitir audio y video de manera rápida.
- Se sugiere aumentar el tamaño de la data set a más usuarios, esto permitirá que la red neuronal puede reconocer más rostros y obtenga mejores resultados, ya que entre más datos tenga la red neuronal más resultados óptimos obtendrá.

- Se recomienda utilizar una cámara con visión nocturna que facilite la detección en la noche y se elimine la necesidad de depender de luz artificial en ese horario. Además, se debe tener en cuenta que la cámara sea compatible con los nuevos sistemas operativos Raspberry Pi OS.

BIBLIOGRAFÍA

[1] Chipantasi, M. D. J. (2019, 22 enero). Repositorio Universidad Técnica de Ambato: Registro de asistencia de alumnos por medio de reconocimiento facial utilizando visión artificial. https://repositorio.uta.edu.ec/handle/123456789/29179.

[2] Rosero, P. V. G. (2019, 18 mayo). Implementación de un control de acceso biométrico mediante reconocimiento facial. http://repositorio.espe.edu.ec/xmlui/handle/21000/20347.

[3] Auz, A. C. X. (2019, 20 septiembre). Repositorio Digital - EPN:, Repositorio Digital - EPN: Desarrollo de un sistema prototipo de acceso a los laboratorios de redes de la Facultad de Ingeniería Eléctrica y Electrónica (FIEE) de la Escuela Politécnica Nacional (EPN) basado en reconocimiento facial.

[4] Saxena, Navya, and Devina Varshney. "Smart Home Security Solutions using Facial Authentication and Speaker Recognition through Artificial Neural Networks." International Journal of Cognitive Computing in Engineering 2 (2021): 154-164.

[5] Shanthakumar, H. C. "Performance Evolution of Face and Speech Recognition system using DTCWT and MFCC Features." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12.3 (2021): 3395-3404.

[6] ". G. d. E. |. C. d. r. h.-d. (. e. 1. d. e. d. 2022).

[7] ". g. p. d. $. m. a. y. a. a. inversionistas".

[8] 5. d. l. h. h.-a.-c.-a.-e.-a. Robo a casas aumenta en Ambato – Diario La Hora. (2021.

[9] p. e. L. e. e. u. d. i. a. p. d. C.-1. (. 3. m. w. h.-p.-e.-l.-e.-e.-u.-d.-i.-a.-p.-d.-c.-1. Ecuador.

[10] P. A. López, Seguridad informática, Editex, 2010.

[11] Google Sites: Sign-in. https://sites.google.com/site/edmunmena/support/vf30?tmpl=/system/app/templates/print/&showPrintDialog=1 (accedido el 24 de agosto de 2022)..

[12] J. A. A. Sánchez, Sistema de Control de Acceso con RFID, México DF, 2008.

[13] L. Cosentino, Control de accesos. Elementos de identificación, 2015.

[14] S. Nogales. "Cómo instalar control de accesos con cerradura electrónica". Tienda de Electrónica Online. https://www.todoelectronica.com/blog-electronica/como-instalar-control-de-accesos-con-cerradura-electronica.html (accedido el 29 de agosto de 2022).

[15] "Sistema de Control de Acceso - Laarcom". Laarcom.com. https://www.laarcom.com/gestion-y-seguridad-con-el-sistema-de-control-de-acceso..

[16] "Control de Acceso .: Problemas más Comunes". Portada. http://www.tecnycomp.com.ec/2012-12-11-15-28-11/controles-de-acceso/102-seguridad-electronica/controles-de-acceso/102-control-de-acceso-problemas-mas-comunes#:~:text=Los%20problemas%20más%20comunes%20.

[17] Moreano, J. A. C., Pulloquinga, R. H. M., Lagla, G. A. F., Chisag, J. C. C., & Pico, O. A. G. (2017). Reconocimiento facial con base en imágenes. Revista Boletín Redipe, 6(5), 143-151.

[18] J. Lacort. "Las claves de los sistemas de reconocimiento facial: ¿cuál es su verdadero nivel de seguridad?" Xataka - Tecnología y gadgets, móviles, informática, electrónica. https://www.xataka.com/seguridad/las-claves-de-los-sistemas-de-reconocimiento-fac.

[19] Cabello Pardos, E. (2004). Técnicas de reconocimiento facial mediante redes neuronales (Doctoral dissertation, Informatica)..

[20] J. García. "Los algoritmos de reconocimiento facial y el sesgo racial: la mayoría tienen problemas al identificar a personas no caucásicas". Xataka - Tecnología y gadgets, móviles, informática, electrónica. https://www.xataka.com/inteligencia-artificial/a.

[21] "¿Qué entendemos por algoritmo?" UDE Universidad de la Empresa. https://ude.edu.uy/que-son-algoritmos/#:~:text=Se%20puede%20entender%20un%20algoritmo,pueden%20ver%20como%20un%20algoritmo. (accedido el 29 de agosto de 2022).

[22] D. E. E. Olguín, «Reconocimiento Facial,» 2015.

[23] ". H. d. G. O. h.-h.-d.-g.-o. (. e. 2. d. a. d. 2022)..

[24] J. SALAZAR, «Procesadores digitales de señal,Arquitecturas y criterios de selección,» UNIVERSIDAD POLITÉCNICA DE CATALUÑA. , España, 2019.

[25] Pérez, Carlos Roberto. "Transformaciones lineales, afines y fractales." (2007).

[26] Gutiérrez Soto, Roberto. "Transformaciones afines del plano y su aplicación en la construcción de superficies topológicas.".

[27] Pérez, Ángel Alejandro Juan, and Cristina Steegmann Pascual. "Transformaciones geométricas." Universitat Oberta de Catalunya.

[28] Clemente, M. R. (2019, 2 abril). idUS - Reconocimiento facial basado en redes neuronales convolucionales. idUS - Depósito de Investigación Universidad de Sevilla. https://idus.us.es/handle/11441/85086.

[29] Castrillon, W. A., Alvarez, D. A., & López, A. F. (2008). Técnicas de extracción de características en imágenes para el reconocimiento de expresiones faciales. Scientia et technica, 14(38), 7-12.

[30] Clemente, M. R. (2021, 7 abril). idUS - Detección, reconocimiento y seguimiento derostros aplicando Redes Neuronales Convolucionales. idUS - Depósito de Investigación Universidad de Sevilla. https://idus.us.es/handle/11441/106808.

[31] Pietikäinen, M. (2010). Local binary patterns. Scholarpedia, 5(3), 9775.

[32] Betancourt, Gustavo A. "Las máquinas de soporte vectorial (SVMs)." Scientia et technica 1.27 (2005).

[33] Gandhi, R. (2018, 5 julio). Support Vector Machine — Introduction to Machine Learning Algorithms. Medium. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[34] Arellano, M. E. E., & Briseño, J. B. A. (2003). Reconocimiento de voz. Conciencia Tecnológica, (22).

[35] Mascorro, G. A. M., & Torres, G. A. (2013). Reconocimiento de voz basado en MFCC, SBC y Espectrogramas. Ingenius, (10), 12-20.

[36] Jackson-Menaldi, M. C. A. (1992). La voz normal. Ed. Médica Panamericana.

[37] Miranda, Sonia Muslares. "Cualidades del sonido." Eufonía: Didáctica de la música 71 (2017): 71-72.

[38] Ortega, C. A. D. L., Romo, J. C. M., & González, M. M. (2006). Reconocimiento de voz con redes neuronales, DTW y modelos ocultos de Markov. Conciencia Tecnológica, (32), 0.

[39] Mascorro, G. A. M. (2013). Reconocimiento de voz basado en MFCC, SBC y Espectrogramas. Dialnet. https://dialnet.unirioja.es/servlet/articulo?codigo=5972819.

[40] Boden, M. A. (2017). Inteligencia artificial. Turner..

[41] Haugeland, J. (2001). La inteligencia artificial. Siglo XXI.

[42] Sandoval Serrano, L. J. (2018). Algoritmos de aprendizaje automático para análisis y predicción de datos. Revista Tecnológica; no. 11.

[43] Arteaga, H. C. (2015). Técnicas de aprendizaje supervisado y no supervisado para el aprendizaje automatizado de computadoras. In Memorias del primer Congreso Internacional de Ciencias Pedagógicas: Por una educación integral, participativa e incluyente, Instituto Superior Tecnológico Bolivariano..

[44] Salas, R. (2004). Redes neuronales artificiales. Universidad de Valparaıso. Departamento de Computación, 1, 1-7.

[45] J. Sanchez, «R E C O N O C I M I E N T O FA C I A L E N I M Á G E N E S,» Trabajo Especial de la Licenciatura en Ciencias de la Computación, Córdoba, 2018.

[46] Suárez, F., Rosales, L., & Hurtado, D. (2019). Red neuronal artificial para análisis de las emociones humanas. Tecnología e Innovación Industrial y sus Perspectivas, 96.

[47] Buitrago, D. G. S. (2011). Perceptrón auto-supervisado una red neuronal artificial capaz de replicación memética. Revista Educación en Ingeniería, 6(12), 90-101.

[48] Vásquez López, J. P. (2008). La redes neuronales artificiales feedforward como identificadoras de asociaciones espurias entre caminatas aleatorias.

[49] Torres, Luz Gloria. "Redes neuronales y aproximación de funciones." Boletín de Matemáticas 1.2 (1994): 35-58.

[50] Gómez, M. J. (2020, 20 abril). idUS - Sistema de reconocimiento facial basado en redes neuronales convencionales sobre el dispositivo Raspberry Pi. idUS - Depósito de Investigación Universidad de Sevilla. https://idus.us.es/handle/11441/95442.

[51] Martínez Llamas, Javier. "Reconocimiento de imágenes mediante redes neuronales convolucionales." (2018).

[52] "Lenguaje de ProgramaciÃ³n - Concepto, tipos y ejemplos". Concepto. https://concepto.de/lenguaje-de-programacion/ (accedido el 29 de agosto de 2022).

[53] Robledano, A. (2021, 24 agosto). Qué es Python: Características, evolución y futuro. OpenWebinars.net. https://openwebinars.net/blog/que-es-python/.

[54] Alberca, A. S. (2022, 12 mayo). La librería Numpy. Aprende con Alf. https://aprendeconalf.es/docencia/python/manual/numpy/.

[55] S. (2021, 25 julio). ¿Qué es OpenCV? Spiegato. https://spiegato.com/es/que-es-opencv.

[56] Team, T. A. I. (2021, 8 agosto). What is TensorFlow, and how does it work? Towards AI. https://towardsai.net/p/l/what-is-tensorflow-and-how-does-it-work.

[57] Keras backends. keras.io. Consultado el 23 de febrero de 2018.

[58] https://www.geeksforgeeks.org/python-convert-speech-to-text-and-text-to-speech/.

[59] Paquete de procesamiento de audio Python: instalación de Librosa - programador clic. (2020). https://programmerclick.com/article/12101226300/.

[60] Artola Moreno, Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python. Trabajo Fin de Grado. Universidad de Sevilla..

[61] Medina, Carlos Roberto Pérez. "TRANSFORMACIONES LINEALES, AFINES Y FRACTALES EN UN AMBIENTE COMPUTACIONAL".

[62] "ISO/IEC 19794-5:2011". ISO. https://www.iso.org/standard/50867.html (accedido el 21 de junio de 2022).

[63] Torres, I. H. P. Curso Procesamiento de Imágenes Digitales. Prácticas de Laboratorio Python y sus familiares.

[64] Dabhi, Mehul K., and Bhavna K. Pancholi. "Face detection system based on Viola-Jones algorithm." International Journal of Science and Research (IJSR) 5.4 (2016): 62-64.

[65] Plenilunia."Manejo de las emociones y enfermedades". https://plenilunia.com/portada/manejo-de-las-emociones-y-enfermedades/25453/ (accedido el 30 de agosto de 2022).

[66] ". e. u. t. d. d. m. p. I. -. C. L. C. L. h. (. e. 2. d. a. d. 2022).

[67] rsuagued. "ARDUINO". Blog de Tecnologías. https://www3.gobiernodecanarias.org/medusa/ecoblog/rsuagued/arduino/ (accedido el 29 de agosto de 2022).

[68] C. Rus. "Raspberry Pi 4 es oficial: una completa actualización con procesador Cortex-A72, hasta 4 GB de RAM y desde 35 dólares". Xataka - Tecnología y gadgets, móviles, informática, electrónica. https://www.xataka.com/ordenadores/raspberry-pi-4-caracteris.

[69] "BPI-M64 SINOVOIP - Ordenador uniplaca | RAM: 2GB; Flash: 8GB; ARM A53 Quad-Core; 5VCC; BANANA-PI-M64 | TME - Elektroniikka komponentit". Electronic components. Distributor, online shop – Transfer Multisort Elektronik. https://www.tme.eu/es/details/banana.

[70] "Amazon.com". Amazon.com. https://www.amazon.com/-/es/micrófono-computadora-videollamadas-conferencias-escritorio/dp/B087WT6L6B (accedido el 29 de agosto de 2022).

[71] "1080P HD USB Webcam with Built-In Microphone - Adesso Inc ::: Your Input Device Specialist :::". Adesso Inc. https://www.adesso.com/product/cybertrack-h4-1080p-hd-usb-webcam-with-built-in-microphone/ (accedido el 29 de agosto de 2022)..

[72] "Diagnóstico de la degradación en reboilers: un modelo en base al aprendizaje profundo". Repositorio Académico - Universidad de Chile. https://repositorio.uchile.cl/handle/2250/174589 (accedido el 16 de julio de 2022).

[73] "ISO 27001 - Certificado ISO 27001 punto por punto - Presupuesto Online". ISO 27001. https://normaiso27001.es/ (accedido el 19 de julio de 2022).

[74] "LFW Face Database : Main". Vision Lab : WELCOME. http://vis-www.cs.umass.edu/lfw/ (accedido el 19 de julio de 2022).

[75] E. Muñoz, Desarrollo de un sistema de control de acceso de personal empleando reconocimiento facial respaldado con técnicas de aprendizaje profundo, Quito: Universidad de las Fuerzas Armadas ESPE, 2021.

[76] Ortiz, M. M. (2013). Procesamiento digital de imágenes. Alfaomega & RA-MA, 10(21).

[77] Petrou, M., P. Bosdogianni. Image Processing: the fundamentals. ISBN 0-471-99883-4,USA: John Wiley and Sons, 1999.

[78] Gallant, S. I. (1990). Perceptron-based learning algorithms. IEEE Transactions on neural networks, 1(2), 179-191.

[79] Python: qué es, para qué sirve y cómo se programa | Informática Industrial. (s. f.). aula21 | Formación para la Industria. https://www.cursosaula21.com/que-es-python/.

[80] opencv. (s. f.). OpenCV: OpenCV installation overview. OpenCV documentation index. https://docs.opencv.org/4.x/d0/d3d/tutorial_general_install.html.

[81] Lopez, P. (2020, 9 de agosto). ¿Qué es Telegram y para qué sirve? - Definición. GEEKNETIC. https://www.geeknetic.es/Telegram/que-es-y-para-que-sirve.

[82] Sirkin, J. (2021, 22 de abril). ¿Qué es SQL (Structured Query Language o Lenguaje de consultas estructuradas)? - Definición en WhatIs.com. ComputerWeekly.es. https://www.computerweekly.com/es/definicion/SQL-Structured-Query-Language-o-Lenguaje-de-consulta.

[83] Escala de grises de imagen y realización con python - programador clic. (2019). https://programmerclick.com/article/23061330684/.

[84] R. (2017, 9 agosto). Conversion de imagenes RGB a escala de grises con Python 3. Python's eyes. https://pythoneyes.wordpress.com/2017/05/22/conversion-de-imagenes-rgb-a-escala-de-grises/.

[85] Mantoro, T., & Ayu, M. A. (2018, May). Multi-faces recognition process using Haar cascades and eigenface methods. In 2018 6th International Conference on Multimedia Computing and Systems (ICMCS) (pp. 1-5). IEEE.

[86] Melo, S. B. (2009). Transformaciones geométricas sobre imágenes digitales. Facultad de Ciencias-Carrera de Matemáticas. Universidad Distrital Francisco José de Caldas.

[87] Procesamiento de imagen OpenCV --- transformación afín - programador clic. (2019). https://programmerclick.com/article/7118251680/.

[88] Domínguez Pavón, S. (2017). Reconocimiento facial mediante el Análisis de Componentes Principales (PCA).

[89] Domínguez Pavón, S. (2017). Reconocimiento facial mediante el Análisis de Componentes Principales (PCA).

[90] Vikram, K., and S. Padmavathi. "Facial parts detection using Viola Jones algorithm." 2017 4th international conference on advanced computing and communication systems (ICACCS). IEEE, 2017.

[91] Deeba, Farah, et al. "LBPH-based enhanced real-time face recognition." International Journal of Advanced Computer Science and Applications 10.5 (2019).

[92] Guo, Shanshan, Shiyu Chen, and Yanjie Li. "Face recognition based on convolutional neural network and support vector machine." 2016 IEEE International conference on Information and Automation (ICIA). IEEE, 2016.

[93] "Fiscalía General del Estado | Cifras de robos". https://www.fiscalia.gob.ec/estadisticas-derobos/.

[94] "Robo a casas aumenta en Ambato". Diario La Hora | Noticias de Ecuador, sus regiones,provincias y Quito. https://www.lahora.com.ec/tungurahua/robo-a-casas-aumenta-en-ambato.

[95] Sutton, Oliver. "Introduction to k nearest neighbour classification and condensed nearest neighbour data reduction." University lectures, University of Leicester 1.

[96] S. (2022, 19 julio). Un paso más a la seguridad de los sistemas biométricos con la Norma ISO / IEC 19989–1:2020. Software ISO. https://www.isotools.org/2020/12/17/un-paso-mas-a-la-seguridad-de-los-sistemas-biometricos-con-la-norma-iso-iec-19989-12020/.

ANEXO A

EN ESTA PAGINA SE PRESENTA EL CÓDIGO UTILIZADO PARA LA CAPTURA DE ROSTROS EN EL LENGUAJE DE PROGRAMACIÓN PYTHON.

```python
import cv2
import os
import imutils
personName = 'Carlos'#Nombre del usuario a registrar
dataPath = './datset'# Cambia a la ruta donde hayas almacenado Data
personPath = dataPath + '/' + personName#Cambia a la ruta donde hayas almacenado Data
if not os.path.exists(personPath):
    print('Carpeta creada: ', personPath)
    os.makedirs(personPath)
cap = cv2.VideoCapture(1)#Inicializacion de la camara
faceClassif = cv2.CascadeClassifier(
    cv2.data.haarcascades+'haarcascade_frontalface_default.xml')#Inicio de modulo haarcascade
count = 0 #contador
while True:
    ret, frame = cap.read()
    if ret == False: break
    frame =  imutils.resize(frame, width=1080)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)#Conversion a escala grises
    auxFrame = frame.copy()
    faces = faceClassif.detectMultiScale(gray,1.3,5)
    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
        rostro = auxFrame[y:y+h,x:x+w]
        rostro = cv2.resize(rostro,(250,250),interpolation=cv2.INTER_CUBIC)#Formato de imagen de 250x250
        cv2.imwrite(personPath + '/rostro_{}.jpg'.format(count),rostro)#formato de imagen jpg
        count = count + 1
    cv2.imshow('frame',frame)
    k =  cv2.waitKey(1)
    if k == 27 or count >= 100:#Total de muestras
        break
cap.release()
cv2.destroyAllWindows()
```

ANEXO B

EN ESTA PAGINA SE PRESENTA EL CÓDIGO UTILIZADO PARA EL PROCESAMIENTO DE IMÁGENES APLICANDO LAS TRANSFORMACIONES AFINES EN EL LENGUAJE DE PROGRAMACIÓN PYTHON.

```
datset\
    label_name_A\
        label_name_A_001.jpg
        label_name_A_002.jpg
        label_name_A_003.jpg
        .
        .
    label_name_B\
        label_name_B_001.jpg
        label_name_B_002.jpg
        label_name_B_003.jpg
        .
        .
```

```python
def detect_face(img):
    #img = cv2.cvtColor(img, cv2.IMREAD)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img = cv2.resize(img,(168,192))
    return img
```

```python
dataset_folder = "./datset/"

names = []
images = []
for folder in os.listdir(dataset_folder):
    files = os.listdir(os.path.join(dataset_folder, folder))[:250]
    if len(files) < 50 :
        continue
    for i, name in enumerate(files):
        if name.find(".jpg") > -1 :
            img = cv2.imread(os.path.join(dataset_folder + folder, name))
            img = detect_face(img) # detect face using mtcnn and crop to 100x100
            if img is not None :
                images.append(img)
                names.append(folder)

                print_progress(i, len(files), folder)
```

```python
def img_augmentation(img):
    h, w = img.shape
    center = (w // 2, h // 2)
    M_rot_5 = cv2.getRotationMatrix2D(center, 5, 1.0)
    M_rot_neg_5 = cv2.getRotationMatrix2D(center, -5, 1.0)
    M_rot_10 = cv2.getRotationMatrix2D(center, 10, 1.0)
    M_rot_neg_10 = cv2.getRotationMatrix2D(center, -10, 1.0)
    M_trans_3 = np.float32([[1, 0, 3], [0, 1, 0]])
    M_trans_neg_3 = np.float32([[1, 0, -3], [0, 1, 0]])
    M_trans_6 = np.float32([[1, 0, 6], [0, 1, 0]])
    M_trans_neg_6 = np.float32([[1, 0, -6], [0, 1, 0]])
    M_trans_y3 = np.float32([[1, 0, 0], [0, 1, 3]])
    M_trans_neg_y3 = np.float32([[1, 0, 0], [0, 1, -3]])
    M_trans_y6 = np.float32([[1, 0, 0], [0, 1, 6]])
    M_trans_neg_y6 = np.float32([[1, 0, 0], [0, 1, -6]])

    imgs = []
    imgs.append(cv2.warpAffine(img, M_rot_5, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_rot_neg_5, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_rot_10, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_rot_neg_10, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_trans_3, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_trans_neg_3, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_trans_6, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_trans_neg_6, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_trans_y3, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_trans_neg_y3, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_trans_y6, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.warpAffine(img, M_trans_neg_y6, (w, h), borderValue=(255,255,255)))
    imgs.append(cv2.add(img, 10))
    imgs.append(cv2.add(img, 30))
    imgs.append(cv2.add(img, -10))
    imgs.append(cv2.add(img, -30))
    imgs.append(cv2.add(img, 15))
    imgs.append(cv2.add(img, 45))
    imgs.append(cv2.add(img, -15))
    imgs.append(cv2.add(img, -45))

    return imgs
```

ANEXO C

EN ESTA PAGINA SE PRESENTA EL CÓDIGO UTILIZADO PARA REALIZAR EL RECONOCIMIENTO DE ROSTROS BASADO EN HAAR CASCADE EN EL LENGUAJE DE PROGRAMACIÓN PYTHON.

```
# --------- load Haar Cascade model ------------
faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
dataPath = 'C:/Users/Acer/Desktop/tesis/datset/imagenes'
imagePaths = os.listdir(dataPath)
print('imagePaths=', imagePaths)
# --------- load Keras CNN model ------------
model = load_model("model-cnn-facerecognition_32x30.h5")
print("[INFO] finish load model...")
cap = cv2.VideoCapture(0)
while cap.isOpened() :
    ret, frame = cap.read()
    if ret:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = faceClassif.detectMultiScale(gray, 1.1, 5)
        for (x, y, w, h) in faces:
            face_img = gray[y:y+h, x:x+w]
            face_img = cv2.resize(face_img, (150, 150))
            face_img = face_img.reshape(1, 150, 150, 1)
            result = model.predict(face_img)
            idx = result.argmax(axis=1)
            print(result.max(axis=1))
            confidence = result.max(axis=1)
            print(confidence)
            if confidence > 0.7:
                id=idx[0]
                print(imagePaths[id])
                #label_text = "%s (%.2f %%)" % (imagePaths[idx],confidence)
                label_text = "%s (%.2f %%)" % (imagePaths[id], confidence)

            else :
                label_text = "N/A"
            frame = draw_ped(frame, label_text, x, y, x + w, y + h, color=(0,255,255), text_color=(50,50,50))
            #cv2.imwrite("test2.jpg",frame)

        cv2.imshow('Biometric', frame)
    else :
        break
    if cv2.waitKey(10) == ord('q'):
        break

cv2.destroyAllWindows()
cap.release()
```

ANEXO D

EN ESTA PAGINA SE PRESENTA LOS PARÁMETROS UTILIZADOS EN EL DISEÑO DE LA RED NEURONAL LENGUAJE EN PYTHON.

```python
def cnn_model(input_shape):
    model = Sequential()

    model.add(Conv2D(64,
                     (3,3),
                     padding="valid",
                     activation="relu",
                     input_shape=input_shape))
    model.add(Conv2D(64,
                     (3,3),
                     padding="valid",
                     activation="relu",
                     input_shape=input_shape))

    model.add(MaxPool2D(pool_size=(2, 2)))

    model.add(Conv2D(128,
                     (3,3),
                     padding="valid",
                     activation="relu"))
    model.add(Conv2D(128,
                     (3,3),
                     padding="valid",
                     activation="relu"))
    model.add(MaxPool2D(pool_size=(2, 2)))

    model.add(Flatten())

    model.add(Dense(128, activation="relu"))
    model.add(Dense(64, activation="relu"))
    model.add(Dense(len(labels)))  # equal to number of classes
    model.add(Activation("softmax"))

    model.summary()

    model.compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics = ['accuracy'])

    return model
```

ANEXO E

EN ESTA PAGINA SE PRESENTA LOS RESULTADOS DEL ENTRENAMIENTO DE LA RED NEURONAL UTILIZADOS POR EL SISTEMA.

```
input_shape = x_train[0].shape

EPOCHS = 10
BATCH_SIZE = 32

model = cnn_model(input_shape)

history = model.fit(x_train,
                    y_train,
                    epochs=EPOCHS,
                    batch_size=BATCH_SIZE,
                    shuffle=True,
                    validation_split=0.15   # 15% of train dataset will be used as validation set
                    )


model.save("model-cnn-facerecognition.h5")
```

```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 64)      640

 conv2d_1 (Conv2D)           (None, 146, 146, 64)      36928

 max_pooling2d (MaxPooling2D  (None, 73, 73, 64)       0
 )

 conv2d_2 (Conv2D)           (None, 71, 71, 128)       73856

 conv2d_3 (Conv2D)           (None, 69, 69, 128)       147584

 max_pooling2d_1 (MaxPooling  (None, 34, 34, 128)      0
 2D)

 flatten (Flatten)           (None, 147968)            0

 dense (Dense)               (None, 128)               18940032

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 4)                 260
...
Epoch 9/10
91/91 [==============================] - 336s 4s/step - loss: 4.5167e-08 - accuracy: 1.0000 - val_loss: 4.2724e-06 - val_accuracy: 1.0000
Epoch 10/10
91/91 [==============================] - 317s 3s/step - loss: 3.3618e-08 - accuracy: 1.0000 - val_loss: 3.8926e-06 - val_accuracy: 1.0000
```

ANEXO F

EN ESTA PAGINA SE PRESENTA EL CÓDIGO UTILIZADO PARA LA CAPTURA DE LOS CLIPS DE AUDIO.

```
##------Captura de voz:
r=sr.Recognizer()

with sr.Microphone() as source:
    print('Menciona tu numero de DNI....')
    audio= r.listen(source)
    print('realizado')

    try:
        voice=r.recognize_google(audio, language='es')
        print(voice)

    except Exception as e:
        print(e)

##-- Validación que el audio y el texto se haya guardado
print(audio)
print(voice)

##--Guardamos el audio en un carpeta.
with open('C:/Users/Acer/Documents/tesis 2022/voice_clasification_tesis/audios_enteros/audio100.wav', "wb") as f:
    f.write(audio.get_wav_data())


##--Cargamos y probamos el archivo de audio
ipd.Audio(audio[-1], rate=22050)
```

ANEXO G

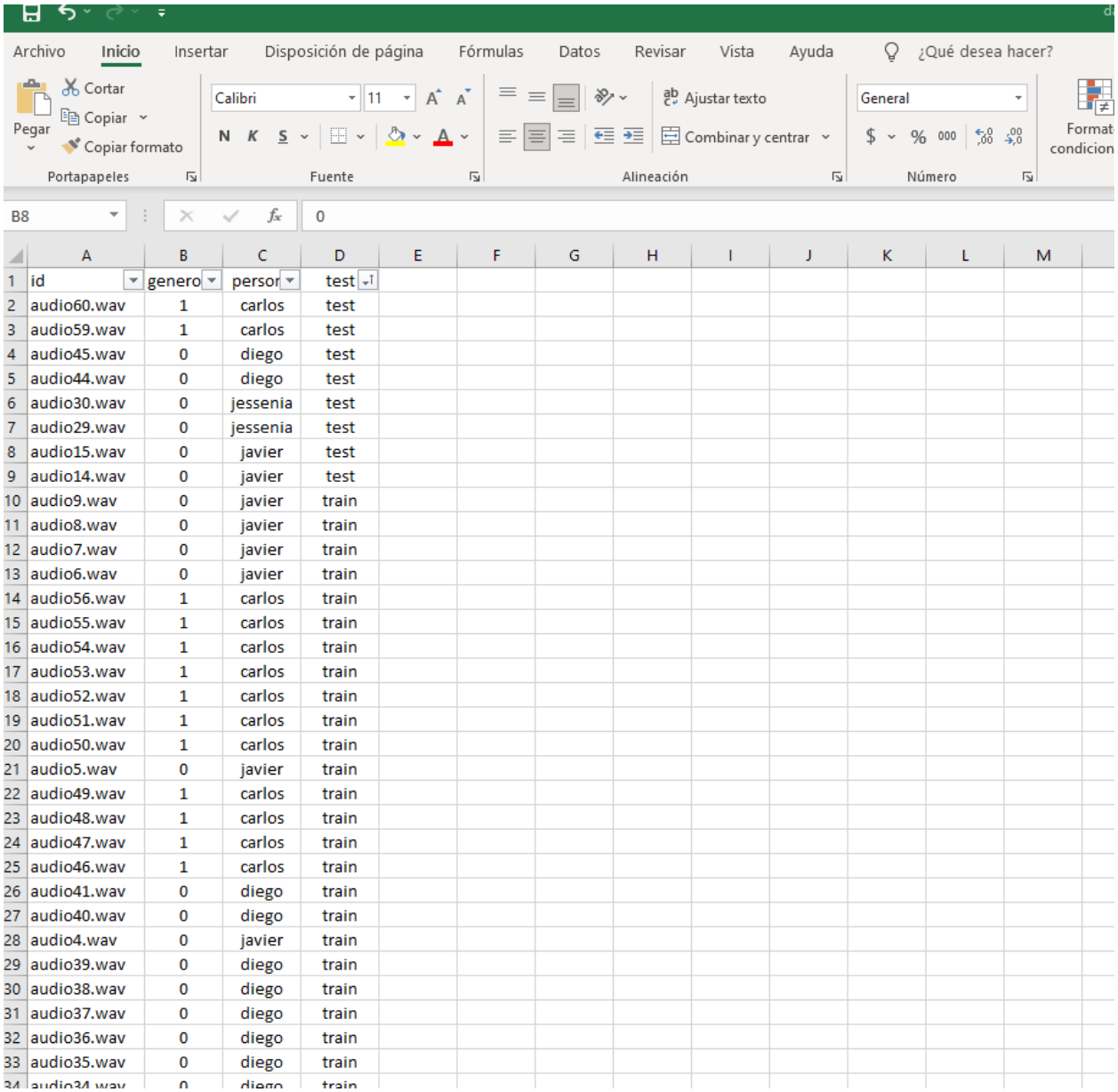
EN ESTA PAGINA SE PRESENTA EL CÓDIGO DE LA APLICACIÓN DEL PROCESAMIENTO MEDIANTE MFCC Y LOS RESULTADOS OBTENIDOS.

```python
##--Creamos una funcion para extraer las mediciones del audio.
def extract_features(files):
    # Establece el nombre de la ruta a donde están los archivos de audio en mi computadora
    file_name = os.path.join(os.path.abspath(data_dir)+'//audios_enteros//'+str(files.id))
    # Carga el archivo de audio como una serie de tiempo de coma flotante y asigna la frecuencia de muestreo predeterminada
    # Sample rate is set to 22050 by default
    # la serie de tiempo esta almacenada en [X]
    X, sample_rate = librosa.load(file_name, res_type='kaiser_fast')
    # genera Mel-frequency cepstral coefficients (MFCCs) de la serie de tiempo
    mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T,axis=0)
    # Genera una transformada de Fourier a corto plazo (STFT) para usar en chroma_stft
    stft = np.abs(librosa.stft(X))
    # Calcula un cromagrama a partir de una forma de onda o espectrograma de potencia.
    chroma = np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)
    # calcula un espectograma de mel-scaled
    mel = np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
    # Calcula el contraste espectral
    contrast = np.mean(librosa.feature.spectral_contrast(S=stft, sr=sample_rate).T,axis=0)
    # Calcula las características del centroide tonal (tonnetz)
    tonnetz = np.mean(librosa.feature.tonnetz(y=librosa.effects.harmonic(X),sr=sample_rate).T,axis=0)
    # Agregamos también las clases de cada archivo como una etiqueta al final
    label = files.persona
    # Pedimos que nos devuelva todos los indicadores mas el target
    return mfccs, chroma, mel, contrast, tonnetz,label
```

```
0     ([-324.5547, 109.65847, -24.028189, 32.481377,...
1     ([-329.36792, 114.05945, -21.196245, 51.867516...
2     ([-346.59256, 101.914566, -31.034883, 17.71996...
3     ([-378.7318, 91.71632, -48.53743, 26.312744, -...
4     ([-540.1187, 101.36175, 9.502917, 30.952785, 8...
5     ([-490.71887, 107.82947, 7.531805, 33.837063, ...
6     ([-453.1581, 91.98391, -3.6293182, 50.91851, 1...
7     ([-494.3419, 88.446846, -7.887439, 40.71482, 1...
8     ([-474.84958, 96.5753, -2.494228, 38.72428, 21...
9     ([-487.47357, 90.21612, -4.8630896, 50.20412, ...
10    ([-475.27838, 86.359695, -2.6666186, 42.22874,...
11    ([-465.4772, 107.78159, -16.584843, 51.510284,...
12    ([-347.6164, 112.12447, -14.493463, 37.768875,...
13    ([-343.21072, 112.60918, -13.15485, 42.31026, ...
14    ([-367.64044, 112.46898, -15.526094, 50.01483,...
15    ([-376.67856, 112.96906, -16.186237, 49.112873...
16    ([-348.18985, 114.997025, -12.768083, 48.40086...
17    ([-356.6978, 109.27251, -14.229045, 54.273075,...
18    ([-320.43735, 112.59219, -22.775988, 64.92516,...
19    ([-478.62045, 99.30916, -16.856077, 41.921047,...
20    ([-315.97934, 113.22996, -13.735534, 47.37032,...
21    ([-363.09854, 110.658936, -13.115851, 49.14202...
22    ([-342.0529, 118.82664, -18.573076, 57.82655, ...
23    ([-345.74704, 114.04286, -13.594094, 56.392147...
24    ([-352.75754, 102.99185, -67.208885, 20.48928,...
...
```

ANEXO H

EN ESTA PAGINA SE PRESENTA EL ARCHIVO DE DISTRIBUCIÓN DE AUDIOS EN EXCEL.



Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vista Ayuda ¿Qué desea hacer?

Pegar Cortar Copiar Copiar formato — Portapapeles

Calibri 11 — Fuente

Ajustar texto Combinar y centrar — Alineación

General — Número

B8 0

	A	B	C	D
1	id	genero	persor	test
2	audio60.wav	1	carlos	test
3	audio59.wav	1	carlos	test
4	audio45.wav	0	diego	test
5	audio44.wav	0	diego	test
6	audio30.wav	0	jessenia	test
7	audio29.wav	0	jessenia	test
8	audio15.wav	0	javier	test
9	audio14.wav	0	javier	test
10	audio9.wav	0	javier	train
11	audio8.wav	0	javier	train
12	audio7.wav	0	javier	train
13	audio6.wav	0	javier	train
14	audio56.wav	1	carlos	train
15	audio55.wav	1	carlos	train
16	audio54.wav	1	carlos	train
17	audio53.wav	1	carlos	train
18	audio52.wav	1	carlos	train
19	audio51.wav	1	carlos	train
20	audio50.wav	1	carlos	train
21	audio5.wav	0	javier	train
22	audio49.wav	1	carlos	train
23	audio48.wav	1	carlos	train
24	audio47.wav	1	carlos	train
25	audio46.wav	1	carlos	train
26	audio41.wav	0	diego	train
27	audio40.wav	0	diego	train
28	audio4.wav	0	javier	train
29	audio39.wav	0	diego	train
30	audio38.wav	0	diego	train
31	audio37.wav	0	diego	train
32	audio36.wav	0	diego	train
33	audio35.wav	0	diego	train
34	audio34.wav	0	diego	train

ANEXO I

EN ESTA PAGINA SE PRESENTA LOS PARÁMETROS DE LA RED NEURONAL PARA LA VOZ.

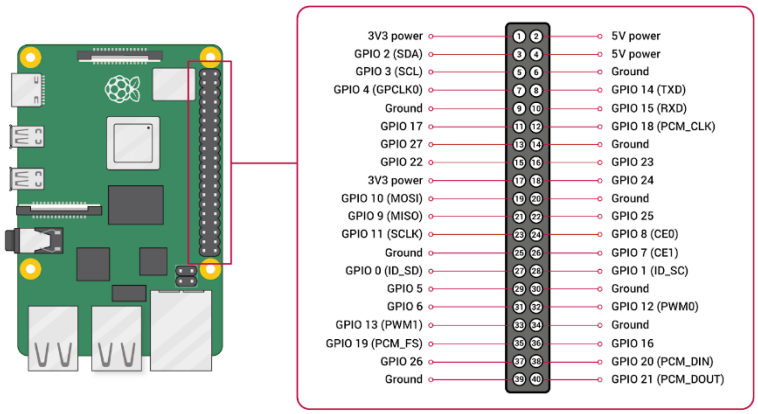
```
##--construimos el modelo CNN
model = Sequential()
model.add(Dense(193, input_shape=(193,), activation = 'relu'))
model.add(Dropout(0.1))
model.add(Dense(128, activation = 'relu'))
model.add(Dropout(0.25))
model.add(Dense(128, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(4, activation = 'softmax'))
model.compile(loss='categorical_crossentropy', metrics=['accuracy'], optimizer='adam')
early_stop = EarlyStopping(monitor='val_loss', min_delta=0, patience=100, verbose=1, mode='auto')


##--revisamos su notación
model.summary()

history = model.fit(X_train, y_train, batch_size=785, epochs=200,validation_data=(X_val, y_val))##,callbacks=[early_stop])

model.save("model-cnn-speechrecognition.h5")
```

```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_36 (Dense)            (None, 193)               37442

 dropout_27 (Dropout)        (None, 193)               0

 dense_37 (Dense)            (None, 128)               24832

 dropout_28 (Dropout)        (None, 128)               0

 dense_38 (Dense)            (None, 128)               16512

 dropout_29 (Dropout)        (None, 128)               0

 dense_39 (Dense)            (None, 4)                 516

=================================================================
```

```
Epoch 1/200
1/1 [==============================] - 1s 702ms/step - loss: 1.3956 - accuracy: 0.1591 - val_loss: 1.3848 - val_accuracy: 0.5
000
Epoch 2/200
1/1 [==============================] - 0s 34ms/step - loss: 1.3820 - accuracy: 0.3182 - val_loss: 1.3826 - val_accuracy: 0.37
50
Epoch 3/200
1/1 [==============================] - 0s 35ms/step - loss: 1.3795 - accuracy: 0.3182 - val_loss: 1.3802 - val_accuracy: 0.25
00
Epoch 4/200
1/1 [==============================] - 0s 35ms/step - loss: 1.3743 - accuracy: 0.2955 - val_loss: 1.3777 - val_accuracy: 0.25
00
Epoch 5/200
1/1 [==============================] - 0s 39ms/step - loss: 1.3657 - accuracy: 0.2727 - val_loss: 1.3744 - val_accuracy: 0.25
00
Epoch 6/200
1/1 [==============================] - 0s 40ms/step - loss: 1.3743 - accuracy: 0.3636 - val_loss: 1.3715 - val_accuracy: 0.62
50
```

ANEXO J

EN ESTA PAGINA SE PRESENTA LOS MATERIALES UTILIZADOS EN EL PROTOTIPO.

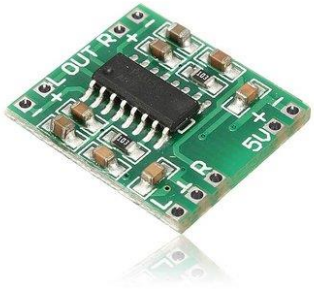


PROCESADOR	**ARM Cortex-A72**
FRECUENCIA DE RELOJ	1,5 GHz
GPU	VideoCore VI (con soporte para OpenGL ES 3.x)
MEMORIA	1 GB / 2 GB / 4 GB LPDDR4 SDRAM
CONECTIVIDAD	Bluetooth 5.0, Wi-Fi 802.11ac, Gigabit Ethernet
PUERTOS	GPIO 40 pines 2 x micro HDMI 2 x USB 2.0 2 x USB 3.0 CSI (cámara Raspberry Pi) DSI (pantalla tácil) Micro SD Conector de audio jack USB-C (alimentación)

Cámara full-hd

Características
-Videollamada Full HD 1080p (hasta 1920×1080 píxeles)
-Micrófono de reducción de ruido incorporado
-Corrección automática de poca luz
-El clip universal listo para trípode se adapta a computadoras portátiles y monitores LCD
-Compatible con Windows 10/8/7 -Transmisión y Conferencia web en vivo

Amplificador PAM8403.



ESPECIFICACIÓN Y CARACTERÍSTICAS
Chip Principal: PAM8403
Protección de corto circuito del PAM8403
Tipo de Amplificador: Clase D
Voltaje de alimentación: 2.5V a 5.5V
Número de canales : 2
Salida máxima: 3W por canal
Altavoces compatibles (impedancia de salida): 4Ω
Peso: 9 g
Dimensiones: 30 mm x 22 mm x 16 mm (Dimensiones sin potenciómetro)
Dimensiones del eje: 6 mm
Eficiencia: 83 %
Pines:
Audio:
Speaker Derecho: INT(+) e INT(-)
Speaker Izquierdo: INT(+) e INT(-)
Voltaje de Alimentación: INT(+) e INT(-)
Entrada de Audio (TSR)

ANEXO K

EN ESTA PAGINA SE PRESENTA EL CÓDIGO FINAL IMPLEMENTADO SOBRE LA RASPBERRY PI

```
#modulos para el reconocimiento de rostros

import os

import cv2

import itertools

import numpy as np

import matplotlib.pyplot as plt

from keras.models import load_model

import speech_recognition as sr

import time

#modulos para el reconocimiento de voz

import pyttsx3

from playsound import playsound

import IPython.display as ipd

import pandas as pd

import librosa

import librosa.display

import numpy as np

import random

import os

from sklearn import preprocessing

from sklearn.preprocessing import LabelEncoder

from sklearn import decomposition
```

```
from sklearn import datasets

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# --------- load Haar Cascade model ----------

faceClassif =
cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')

dataPath = 'C:/Users/Acer/Documents/tesis 2022/voice_clasification_tesis/datset'

imagePaths = os.listdir(dataPath)

print('imagePaths=', imagePaths)

cap = cv2.VideoCapture(1)



# --------- load Keras CNN model ----------

model1 = load_model("model-cnn-facerecognition.h5")

model2 = load_model("model-cnn-speechrecognition.h5")


print("[INFO] finish load model...")



df=pd.read_excel("C:/Users/Acer/Documents/tesis
2022/face_recognition/dataset.xlsx")

df.head()

##--dirección raiz

data_dir=('C://Users//Acer//Documents//tesis 2022//voice_clasification')
```

```
i = random.choice(df.index)
##--toma el ID de la fila seleccionada en el [i]
audio_name = df.id[i]
##--crea la ruta donde se encuentra el audio
path = os.path.join(data_dir, '//audios_enteros', str(audio_name) + '.wav')
##--imprime el nombre de la persona a quien le pertenece el audio



engine = pyttsx3.init()
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[2].id)
engine. setProperty('rate', 178)
engine.setProperty('volume', 0.7)
model = load_model("model-cnn-speechrecognition.h5")
lb = LabelEncoder()



def extract_features(files):
    # Establece el nombre de la ruta a donde están los archivos de audio en mi computadora
```

```
        file_name =
os.path.join(os.path.abspath(data_dir)+'//audios_enteros//'+str(files.id))

        # Carga el archivo de audio como una serie de tiempo de coma flotante y asigna
la frecuencia de muestreo predeterminada

        # Sample rate is set to 22050 by default

        # la serie de tiempo esta almacenada en [X]

        X, sample_rate = librosa.load(file_name, res_type='kaiser_fast')

        # genera Mel-frequency cepstral coefficients (MFCCs) de la serie de tiempo

        mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate,
n_mfcc=40).T,axis=0)

        # Genera una transformada de Fourier a corto plazo (STFT) para usar en
chroma_stft

        stft = np.abs(librosa.stft(X))

        # Calcula un cromagrama a partir de una forma de onda o espectrograma de
potencia.

        chroma = np.mean(librosa.feature.chroma_stft(S=stft,
sr=sample_rate).T,axis=0)

        # calcula un espectograma de mel-scaled

        mel = np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)

        # Calcula el contraste espectral

        contrast = np.mean(librosa.feature.spectral_contrast(S=stft,
sr=sample_rate).T,axis=0)

        # Calcula las características del centroide tonal (tonnetz)

        tonnetz =
np.mean(librosa.feature.tonnetz(y=librosa.effects.harmonic(X),sr=sample_rate).T,axi
s=0)
```

```
        # Agregamos también las clases de cada archivo como una etiqueta al final
        label = files.persona
        # Pedimos que nos devuelva todos los indicadores mas el target
        return mfccs, chroma, mel, contrast, tonnetz,label


def reconocimientoaudio():
    voice='221221'
    with sr.Microphone() as source:
        talk('Menciona tu DNI despues del tono')


        #time.sleep(1)
        playsound('C:/Users/Acer/Documents/tesis
2022/voice_clasification/tono/note.wav')


        print('Menciona tu numero de DNI....')
        nuevo= r.listen(source)
        print('realizado')



    try:
        voice=r.recognize_google(nuevo, language='es')
        voice = voice.replace(' ', '')


        print(voice)
```

```
except Exception as e:

    print(e)

with open('C:/Users/Acer/Documents/tesis
2022/voice_clasification/audios_enteros/nuevo.wav', "wb") as f:
    f.write(nuevo.get_wav_data())
audio=r'C:/Users/Acer/Documents/tesis
2022/voice_clasification/audios_enteros/nuevo.wav'
##--Cargamos y probamos el archivo de audio
ipd.Audio(audio, rate=22050)
data= pd.DataFrame(columns=('id', 'genero', 'persona', 'test'))
data.loc[len(data)]=['nuevo.wav',0,'nuevo','nuevo']
nuevo = data.apply(extract_features, axis=1)
features_new = []
for i in range(0, len(nuevo)):
    features_new.append(np.concatenate((nuevo[i][0], nuevo[i][1],
            nuevo[i][2], nuevo[i][3],
            nuevo[i][4]), axis=0))
##--separamos los valores X y los valores Y
```

```
Xnuevo = np.array(features_new)

X_Xnuevo=preprocessing.normalize(Xnuevo,norm='l2')

# We get our predictions from the test data

predict_x=model2.predict(X_Xnuevo)

preds=np.argmax(predict_x,axis=1)

#preds = model.predict_classes(X_Xnuevo)

# # Transformamos nuestras predicciones a los ID..

print(preds)

#preds = lb.inverse_transform(preds)

print(preds)


print('tu eres  ' , preds[0])

identificador = preds[0]

prov_str=str(identificador)


print(type(prov_str))

if prov_str== '0' and voice == '6489':

   talk('Gracias por identificarte '  +' puedes pasar')

elif prov_str=='1' and voice=='1052':

   talk('Gracias por identificarte ' + preds[0] +' puedes pasar')

elif prov_str=='2' and voice == '1279':

   talk('Gracias por identificarte ' + preds[0] +' puedes pasar')

elif prov_str=='3' and voice == '4291':

   talk('Gracias por identificarte ' + preds[0] +' puedes pasar')
```