



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

“IMPLEMENTACIÓN DE UN SISTEMA PARA EL CONTROL DE DISPOSITIVOS ELÉCTRICOS UTILIZANDO RECONOCIMIENTO DE LENGUAJE DE SEÑAS DACTILOLÓGICO MEDIANTE VISIÓN ARTIFICIAL.”

Trabajo de Integración Curricular

Tipo: Proyecto de Investigación

Presentado para optar al grado académico de:

INGENIERO EN TELECOMUNICACIONES

AUTOR:

KEVIN EDGARDO SAAVEDRA DELGADO

Riobamba – Ecuador

2023



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

“IMPLEMENTACIÓN DE UN SISTEMA PARA EL CONTROL DE DISPOSITIVOS ELÉCTRICOS UTILIZANDO RECONOCIMIENTO DE LENGUAJE DE SEÑAS DACTILOLÓGICO MEDIANTE VISIÓN ARTIFICIAL.”

Trabajo de Integración Curricular

Tipo: Proyecto de Investigación

Presentado para optar al grado académico de:

INGENIERO EN TELECOMUNICACIONES

AUTOR: KEVIN EDGARDO SAAVEDRA DELGADO

DIRECTOR: Ing. JEFFERSON ALEXANDER RIBADENEIRA RAMÍREZ, PHD.

Riobamba – Ecuador

2023

© 2023, Kevin Edgardo Saavedra Delgado

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Kevin Edgardo Saavedra Delgado, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 28 de marzo del 2023



Kevin Edgardo Saavedra Delgado


080316169-4

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; tipo: Propuesta Tecnológica, **"IMPLEMENTACIÓN DE UN SISTEMA PARA EL CONTROL DE DISPOSITIVOS ELÉCTRICOS UTILIZANDO RECONOCIMIENTO DE LENGUAJE DE SEÑAS DACTILOLÓGICO MEDIANTE VISIÓN ARTIFICIAL"**, realizado por el señor: **KEVIN EDGARDO SAAVEDRA DELGADO**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA aa-mm-dd
Ing. Fabricio Javier Santacruz Sulca PRESIDENTE DEL TRIBUNAL		2023-03-28
Ing. Jefferson Alexander Ribadeneira Ramírez DIRECTOR DE TRABAJO DE ITEGRACIÓN CURRICULAR		2023-03-28
Ing. Diego Ramiro Ñacato Estrella ASESOR DE TRABAJO DE INTEGRACION CURRICULAR		2023-03-28

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS	viii
ÍNDICE DE ILUSTRACIONES.....	ix
ÍNDICE DE ECUACIONES	xi
RESUMEN.....	¡Error! Marcador no definido.
SUMMARY	¡Error! Marcador no definido.
INTRODUCCIÓN	1

CAPÍTULO I

1. PROBLEMA DE INVESTIGACIÓN	2
1.1 Antecedentes del Problema	2
1.2 Planteamiento Del Problema.....	3
1.2.1 <i>Formulación del Problema</i>	4
1.2.2 <i>Sistematización del Problema</i>	4
1.3 <i>Justificación</i>	4
1.3.1 <i>Justificación Teórica</i>	4
1.3.2 <i>Justificación Práctica</i>	5
1.4 Objetivos	6
1.4.1 <i>Objetivo General</i>	6
1.4.2 <i>Objetivos Específicos</i>	6
1.5 Hipotesis	6

CAPITULO II

2. MARCO TEÓRICO	7
2.1 Antecedentes De La Investigación	7
2.2 Referencias Teóricas.....	9
2.2.1 <i>Lenguaje Dactilológico</i>	9
2.2.2 <i>Sistema de Control</i>	10
2.2.3 <i>Mediapipe</i>	11
2.2.4 <i>Reconocimiento de manos</i>	11
2.2.5 <i>Python</i>	12
2.2.6 <i>Tensorflow</i>	12

2.2.7	<i>OpenCv</i>	13
2.2.8	<i>PyCharm Community</i>	13
2.2.9	<i>Mosquitto</i>	15
2.2.10	<i>Inteligencia Artificial (IA)</i>	16
2.2.11	<i>Visión Artificial</i>	16
2.2.12	<i>Red Neuronal</i>	16
2.2.13	<i>Anaconda</i>	18
2.2.14	<i>Módulo Wifi ESP8266</i>	18
2.2.15	<i>Módulo de Relé</i>	19
2.2.16	<i>Transistor 2N222</i>	19
2.2.17	<i>Regulador de 5V</i>	20
2.2.18	<i>Software para el diseño y simulación de circuitos eléctricos</i>	20
 CAPITULO III		 22
 3.	 MARCO METODOLÓGICO	 22
3.1	Introducción	22
3.2	Tipos de investigación	22
3.2.1	<i>Investigación Experimental</i>	22
3.2.2	<i>Investigación Aplicativa</i>	23
3.3	Equipos y materiales	23
3.3.1	<i>Hardware</i>	23
3.3.2	<i>Software</i>	24
3.4	Estudio de Factibilidad	25
3.5	Desarrollo del Prototipo	25
3.5.1	<i>Diseño del sistema</i>	27
3.5.2	<i>Instalación de Anaconda</i>	28
3.5.3	<i>Descarga de librerías para el funcionamiento del Prototipo</i>	30
3.5.4	<i>Modelo Personalizado</i>	32
3.5.5	<i>Entrenamiento del modelo</i>	35
3.5.6	<i>Proceso de detección en tiempo real</i>	37
 CAPITULO IV		
 4.	 MARCO DE ANALISIS E INTERPRETACION DE RESULTADOS	 44
4.1	Resultados	44
4.1.1	<i>Recolección de Datos</i>	44

4.1.2 Evaluación del Prototipo	46
CONCLUSIONES	58
RECOMENDACIONES	60
BIBLIOGRAFÍA	
ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-3: Escala de Likert de las tecnologías	25
---	----

ÍNDICE DE ILUSTRACIONES

Ilustración 1-1: Porcentajes de discapacidades en grupos etarios de la ciudad de Riobamba.....	2
Ilustración 1-2: Movimientos gestuales que definen el lenguaje dactilológico en español.....	10
Ilustración 2-2: Puntos de referencia en la mano para su reconocimiento por MediaPipe.....	12
Ilustración 3-2: Esquema de una neurona artificial.....	17
Ilustración 4-2: Regulador de voltaje.....	20
Ilustración 1-3: Esquema de funcionamiento del sistema para el control de dispositivos eléctricos.....	26
Ilustración 2-3: Esquema de funcionamiento del sistema (publicador).....	27
Ilustración 3-3: Esquema de funcionamiento del sistema desde el suscriptor.....	28
Ilustración 4-3: Instalación de Anaconda3 exitosa.....	29
Ilustración 5-3: Creación del ambiente TESIS2.....	30
Ilustración 6-3: Tabla correspondiente a los puntos de las imágenes de entrenamiento.....	31
Ilustración 7-3: Dependencias adicionales para el funcionamiento del prototipo.....	32
Ilustración 8-3: Etiquetas asignadas a las señas entrenadas.....	33
Ilustración 9-3: Función para seleccionar el modo.....	33
Ilustración 10-3: Tracking de la mano, dedo pulgar e índice.....	34
Ilustración 11-3: Tracking de la mano, puntos clave.....	34
Ilustración 12-3: Función para capturar las señas y escribir los puntos en keypoint.csv.....	34
Ilustración 13-3: Datos de entrada para realizar el entrenamiento.....	35
Ilustración 14-3: Parámetros de la red neuronal para el entrenamiento del modelo.....	36
Ilustración 15-3: Parámetros de la red neuronal para el entrenamiento del modelo.....	36
Ilustración 16-3: Método fit para el entrenamiento.....	37
Ilustración 17-3: Preparación para la cámara web.....	37
Ilustración 18-3: Preparación para la cámara web.....	38
Ilustración 19-3: Parámetros para la detección.....	38
Ilustración 20-3: Parámetros de detección en tiempo real.....	39
Ilustración 21-3: Instancia MQTT variable que contiene la seña identificada.....	39
Ilustración 22-3: Condiciones para publicar en los diferentes tópicos.....	40
Ilustración 23-3: Parámetros para el funcionamiento del ESP8266.....	41
Ilustración 24-3: Parámetros para el funcionamiento del ESP8266.....	41
Ilustración 25-3: Función para la recepción de datos del dispositivo cerradura.....	42
Ilustración 26-3: Circuito de amplificación para la cerradura eléctrica.....	43
Ilustración 27-3: Circuito de amplificación para la cerradura eléctrica.....	43

Ilustración 28-3: Circuito de amplificación para la cerradura eléctrica.	43
Ilustración 1-4: Tabla del conjunto datos para el entrenamiento.....	45
Ilustración 2-4: Total de datos recolectados para la clase Alarma.....	45
Ilustración 3-4: Inicio de entrenamiento	46
Ilustración 4-4: Fin de entrenamiento (pausado por EarlyStopping).	47
Ilustración 5-4: Evaluación del modelo.....	47
Ilustración 6-4: Resultados del modelo entrenado	47
Ilustración 7-4: Predicción del modelo entrenado.....	48
Ilustración 8-4: Resultado de predicción del modelo entrenado	48
Ilustración 9-4: Matriz de confusión.	49
Ilustración 10-4: Reporte de clasificación.....	50
Ilustración 11-4: Detección de la clase 0.....	51
Ilustración 12-4: Detección de la clase 1.....	51
Ilustración 13-4: Detección de la clase 2.....	52
Ilustración 14-4: Detección de la clase 3.....	52
Ilustración 15-4: Detección de la clase 4.....	53
Ilustración 16-4: Detección de la clase 5.....	53
Ilustración 17-4: Activación de foco	54
Ilustración 18-4: Desactivación de foco.....	55
Ilustración 19-4: Activación de cerradura eléctrica.....	55
Ilustración 20-4: Desactivación de cerradura eléctrica	56
Ilustración 21-4: Activación de alarma sonora.....	56
Ilustración 22-4: Desactivación de alarma sonora.....	57

ÍNDICE DE ECUACIONES

Ecuación 1-4: Matriz de confusión.....	49
---	----

RESUMEN

El objetivo del presente estudio fue implementar un sistema de control de dispositivos eléctricos mediante el reconocimiento de lenguaje de señas dactilológico, implementado con visión artificial. El proyecto está enfocado a personas con discapacidad auditiva puesto que incentiva a la inclusión, convivencia afectiva, interactividad humano-tecnología y fortalecimiento de la investigación y desarrollo. La ejecución del presente trabajo se realizó mediante una investigación experimental y aplicada, a la vez que se sistematizó, practicó e implementó los conocimientos adquiridos con la utilización del hardware y software apropiado. Para empezar, se instaló Arduino 1.8.19 para programar un módulo wifi ESP8266 e instalar posteriormente el entorno de Anaconda para crear el ambiente de programación en Python. Luego, se programó el modelo en Jupyter Notebook para llevar a cabo el entrenamiento de los gestos, con PyCharm Community como entorno de desarrollo integrado (IDE) y con Python como lenguaje de programación, se creó el script para lograr la detección en tiempo real utilizando visión artificial y MediaPipe. La conexión del módulo wifi al bróker mosquitto se realizó mediante el protocolo de comunicación MQTT para finalmente conectar los dispositivos eléctricos: módulo relé, alarma sonora, cámara web, foco y cerradura eléctrica al módulo wifi. Los resultados obtenidos presentan una tasa de precisión cercana al 90%, lo que demuestra la viabilidad de utilizar esta tecnología como alternativa para el control de dispositivos eléctricos en situaciones donde el control manual no es posible o para personas con discapacidad auditiva. Se recomienda continuar con el proyecto para la mejora continua con diversas contribuciones prácticas y metodológicas del prototipo orientándolo al futuro lanzamiento al mercado.

Palabras clave: <INTELIGENCIA ARTIFICIAL>, <VISIÓN ARTIFICIAL>, <DISPOSITIVOS ELÉCTRICOS>, <COMUNICACIÓN INALÁMBRICA>, <CONTROL REMOTO>, <LENGUAJE DACTILOLÓGICO>.

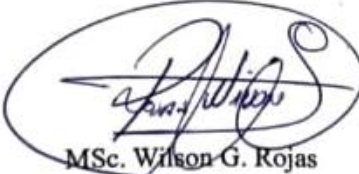


0580-DBRA-UPT-2023

SUMMARY

The objective of this study was to implement a control system for electrical devices through the recognition of dactylological sign language, implemented with artificial vision. The project is focused on people with hearing disabilities since it encourages inclusion, affective coexistence, human-technology interactivity and strengthening of research and development. The execution of this work was carried out through an experimental and applied research, while the knowledge acquired was systematized, practiced and implemented with the use of appropriate hardware and software. To begin with, Arduino 1.18.19 was installed to program an ESP8266 Wi-Fi module and later installed the Anaconda environment to create the programming environment in Python. Then, the model was programmed in Jupyter Notebook to carry out the gesture training, with PyCharm Community as integrated development environment (IDE) and with Python as programming language, the script was created to achieve real-time detection using computer vision and MediaPipe. The connection of the Wi-Fi module to the mosquito broker was carried out using the MQTT communication protocol to finally connect the electrical devices: relay module, audible alarm, webcam, spotlight and electric lock to the Wi-Fi module. The results obtained present a precision rate close to 90%, which demonstrates the feasibility of using this technology as an alternative for the control of electrical devices in situations where manual control is not possible for people with hearing disabilities. It is recommended to continue with the project for continuous improvement with various practical and methodological contributions of the prototype, guiding it to the future launch on the market.

Keywords: <ARTIFICIAL INTELLIGENCE>, <ARTIFICIAL VISION>,
<ELECTRICAL DEVICES>, <WIRELESS COMMUNICATION>, <REMOTE
CONTROL>, <FINGER LANGUAGE>



MSc. Wilson G. Rojas

NOMBRE Y FIRMA PROFESOR

C.I 0602361842

INTRODUCCIÓN

La discapacidad auditiva es una limitación que afecta a millones de personas en todo el mundo. Esta condición puede limitar la capacidad de una persona para comunicarse e interactuar con su entorno, lo que a su vez afecta su calidad de vida. Uno de los problemas específicos que enfrentan las personas con discapacidad auditiva es la dificultad para controlar dispositivos eléctricos mediante comandos de voz, lo que puede limitar su independencia y autonomía en el hogar y otros entornos.

Para abordar esta problemática, se ha propuesto el uso de la visión artificial y el reconocimiento de lenguaje de señas dactilológico como una forma de permitir a las personas con discapacidad auditiva controlar dispositivos eléctricos utilizando gestos manuales. Este enfoque tiene el potencial de mejorar significativamente la accesibilidad para las personas con discapacidad auditiva, permitiéndoles controlar los dispositivos eléctricos de manera independiente y eficiente.

El objetivo de esta tesis es implementar un sistema para el control de dispositivos eléctricos utilizando reconocimiento de lenguaje de señas dactilológico mediante visión artificial. El sistema propuesto constará de una cámara que capturará los gestos manuales del usuario, y un algoritmo de visión artificial que procesará la información de la cámara para identificar los gestos y traducirlos en comandos para los dispositivos eléctricos.

El desarrollo de este sistema es de gran importancia ya que permitiría a las personas con discapacidad auditiva controlar sus dispositivos eléctricos de manera independiente y efectiva, mejorando así su calidad de vida. Además, el uso de la visión artificial y el reconocimiento de lenguaje de señas dactilológico tiene el potencial de ser una solución más eficiente y efectiva que otras alternativas disponibles en la actualidad, como el uso de interruptores de palanca o controles remotos. Sin embargo, este proyecto presenta desafíos técnicos y de costo debido a la necesidad de una tecnología avanzada y costosa para su implementación. Por lo tanto, es importante explorar opciones para hacer que esta tecnología sea más accesible y asequible para las personas con discapacidad auditiva en el país.

En resumen, el presente trabajo busca mejorar la inclusión y accesibilidad tecnológica de las personas con discapacidad auditiva. Sin embargo, se enfrentan a desafíos técnicos y de costo que deben ser abordados para lograr este objetivo. Por tanto, es necesario considerar los medios más efectivos para su implementación.

CAPÍTULO I:

1. PROBLEMA DE INVESTIGACIÓN

1.1 Antecedentes del Problema

De acuerdo con el Consejo Nacional para la Igualdad de Discapacidades (CONADIS), la minusvalía auditiva es una de las más comunes en el Ecuador. El poder ejecutivo ha estado trabajando en mejorar la vida de las personas con este tipo de incapacidad a través de políticas públicas y programas de apoyo. Estos incluyen la provisión de dispositivos de asistencia para mejorar la comunicación, la inclusión en la educación y el empleo, así como la sensibilización pública sobre la importancia de la inclusión de las personas discapacitadas. Sin embargo, aún existen muchos desafíos por superar, incluyendo la discriminación y la falta de acceso a servicios adecuados. Es importante continuar trabajando en esta dirección para mejorar la vida de estas personas y garantizar su pleno derecho a la participación plena en la sociedad.

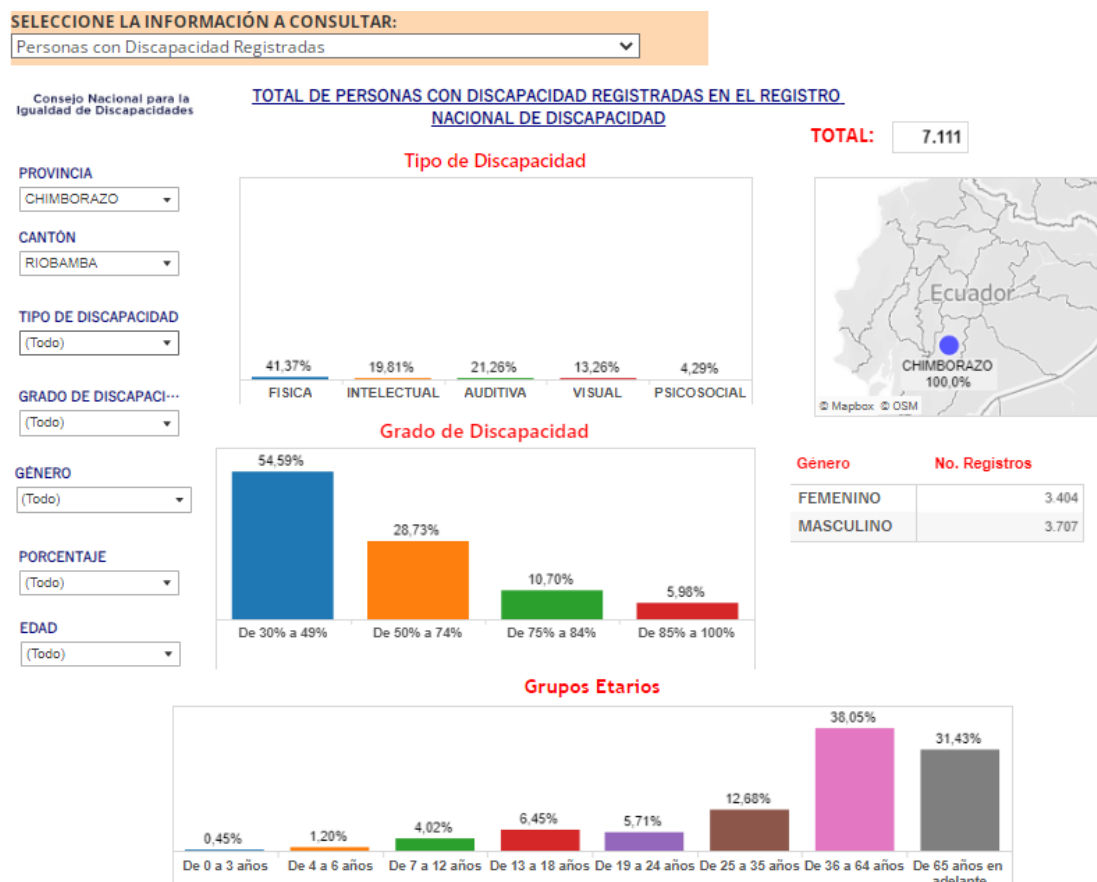


Ilustración 1-1-1: Porcentajes de discapacidades en grupos etarios de la ciudad de Riobamba.

Fuente: (Consejo Nacional para la Igualdad de Discapacidades CONADIS, 2022)

En la Ilustración 1-1 se presenta los datos publicados por el Ministerio de Salud Pública, apreciando que la discapacidad auditiva es la tercera predominante en el territorio de Riobamba, siendo el grupo masculino de 36 a 64 años quienes poseen mayormente esta condición, con un grado de discapacidad del 54,59%.

En Ecuador, existen diversas instituciones y organizaciones que trabajan en el desarrollo de tecnologías accesibles y eficaces para las personas con discapacidad auditiva, lo que muestra un interés creciente por mejorar la calidad de vida de esta población. Entre estas instituciones, se pueden mencionar la Secretaría Nacional de Discapacidades (SENADIS), el Instituto de Discapacidades Físicas y Mentales (IDIFIM) y la Federación Ecuatoriana de Personas con Discapacidad Auditiva (FEDEA). Además, en el ámbito internacional, hay empresas e instituciones que ofrecen soluciones de tecnología de visión artificial para el reconocimiento de lenguaje de señas, que podrían estar disponibles en el país. En este sentido, es importante destacar que estas tecnologías emergentes de reconocimiento de imágenes con inteligencia artificial y asistencia por computadora se encuentran en constante evolución y mejora, lo que permite la adaptación a las necesidades de personas con discapacidad.

1.2 Planteamiento Del Problema

En el país no existen medios o sistemas para controlar dispositivos eléctricos que puedan ayudar a personas con discapacidades motoras y auditivas debido a la falta de investigación y desarrollo. Esta falta de accesibilidad tecnológica contribuye a varias desigualdades en la sociedad. Por lo tanto, es importante considerar el desarrollo de un sistema que permita el control de dispositivos eléctricos mediante el reconocimiento del lenguaje de señas dactilológico a través de la visión artificial. Asimismo, existe un gran número personas en la población con discapacidades motoras o de habla que tienen dificultades para controlar dispositivos eléctricos mediante controles físicos o dispositivos como asistentes virtuales por comando de voz, de ahí que un sistema de control basado en señas dactilológicas es una alternativa accesible para ellos.

El Ecuador es un país que aún se encuentra en desarrollo, por tanto, la interacción humano-computadora es limitada. La utilización de controles físicos o dispositivos de asistencia puede resultar incómoda o poco intuitiva para el usuario. Por esta razón, un sistema de control basado en señas dactilológicas podría mejorar significativamente la experiencia de los usuarios, proporcionando una forma más natural y fluida de interactuar con los dispositivos eléctricos. Además, en ciertas aplicaciones industriales, como la robótica, la automatización y la medicina, el control preciso y rápido de los dispositivos eléctricos es crucial. Actualmente, el sistema

existente es insuficiente y la implementación de un sistema de control basado en señas dactilológicas podría ser una solución eficaz para estos escenarios.

Por otra parte, es importante destacar que en el país los hogares en su gran mayoría no tienen un sistema inteligente que salvaguarde la integridad física de su núcleo familiar. Por tanto, los denominados “hogares inteligentes” con los “dispositivos eléctricos inteligentes” se están convirtiendo cada vez más en parte integral de la vida cotidiana. Este tipo de sistemas al ser adaptado a hogares u otros establecimientos proporcionan una forma más intuitiva y cómoda de interactuar con los dispositivos eléctricos del lugar donde se encuentre.

1.2.1 Formulación del Problema

¿Cómo implementar un sistema para el control de dispositivos eléctricos utilizando reconocimiento de lenguaje de señas dactilológico mediante visión artificial?

1.2.2 Sistematización del Problema

- ¿Qué requisitos y características debe tener el sistema a desarrollar?
- ¿Qué detalles técnicos deberá contener el sistema para dar soluciones a las personas con deficiencia motora auditiva?
- ¿Cómo entrenar una red neuronal para el diseño del modelo?
- ¿Cómo aprobar el prototipo para justificar los resultados logrados?

1.3 Justificación

1.3.1 Justificación Teórica

El presente proyecto busca implementar un sistema de control de dispositivos eléctricos mediante el reconocimiento de lenguaje de señas dactilológico utilizando visión artificial.

Para lograr este objetivo, se requerirá el uso de varias tecnologías, incluyendo el procesamiento de imágenes, el reconocimiento de patrones, la inteligencia artificial, las redes neuronales y la programación en Python haciendo uso de IDE's (Entorno de desarrollo integrado). Además, se utilizará la biblioteca de código abierto de Google, MediaPipe, que proporciona herramientas y bibliotecas para la construcción de aplicaciones de visión artificial en tiempo real.

Desde una perspectiva teórica, el uso de visión artificial para el reconocimiento de lenguaje de señas dactilológico es una técnica viable y bien establecida. La literatura científica cuenta con diversos estudios que demuestran la capacidad de la visión artificial para reconocer con precisión los gestos y movimientos de las manos y dedos que conforman el lenguaje de señas dactilológico. Además, las redes neuronales son una herramienta poderosa y efectiva para la clasificación y el reconocimiento de patrones, lo que las convierte en una opción óptima para el desarrollo del sistema de reconocimiento de lenguaje de señas dactilológico. La programación en Python también proporciona una amplia gama de bibliotecas y herramientas que son altamente eficientes y escalables, lo que permite el desarrollo rápido y eficaz del sistema de control de dispositivos eléctricos.

En resumen, este proyecto cuenta con una base teórica sólida y una variedad de tecnologías que son adecuadas y efectivas para su implementación.

1.3.2 Justificación Práctica

El sistema propuesto en el presente trabajo permite mejorar la calidad de vida de las personas con discapacidad auditiva, al proporcionarles una herramienta que les permita controlar dispositivos eléctricos de forma autónoma y sin la necesidad de utilizar dispositivos tradicionales como controles remotos.

Este sistema puede ser utilizado en diferentes contextos, como en el hogar, en la oficina o en lugares públicos, lo que amplía su alcance y su impacto en la sociedad.

Además, el sistema de reconocimiento de lenguaje de señas dactilológico utilizando visión artificial puede ser fácilmente integrado en diferentes tipos de dispositivos eléctricos, como interruptores de luz, televisores, aires acondicionados, entre otros, lo que permite que las personas con discapacidad auditiva tengan un mayor control y autonomía sobre su entorno.

Otro aspecto importante de este proyecto es que la tecnología utilizada, como el procesamiento de imágenes, el reconocimiento de patrones y las redes neuronales, se encuentra en constante evolución y mejora, lo que permite la actualización y mejora del sistema a medida que avanza la tecnología.

En conclusión, la implementación de este sistema para el control de dispositivos eléctricos utilizando el reconocimiento de lenguaje de señas dactilológico mediante visión artificial se puede

llevar a la práctica de forma significativa y con un alto impacto social, ya que ofrece una solución innovadora y efectiva para mejorar la calidad de vida de las personas con discapacidad auditiva.

1.4 Objetivos

1.4.1 Objetivo General

Implementar un sistema de control de dispositivos eléctricos para el reconocimiento de lenguaje de señas dactilológico mediante visión artificial.

1.4.2 Objetivos Específicos

- Fundamentar el estudio a través de un marco teórico para la validación de la investigación acorde al desarrollo de un sistema de control de dispositivos eléctricos y su relación con el reconocimiento de lenguaje de señas dactilológicas por visión artificial mediante fuentes secundarias de información.
- Realizar un estudio y selección de las tecnologías para el reconocimiento óptimo y adecuado de imágenes.
- Desarrollar un sistema donde se efectúe la correcta aplicación de los requerimientos para la evaluación, control del funcionamiento e implementación.

1.5 Hipotesis

Se puede implementar un sistema para el control de dispositivos eléctricos utilizando reconocimiento de lenguaje de señas dactilológico mediante visión artificial.

CAPITULO II

2. MARCO TEÓRICO

2.1 Antecedentes De La Investigación

El avance de la tecnología se ve impulsado por diversos factores, siendo la investigación y el desarrollo los más importantes. Estos procesos son esenciales para la creación de nuevos métodos y para mejorar los mecanismos existentes. En particular, la investigación permite la identificación de problemas y oportunidades, mientras que el desarrollo se enfoca en la creación de soluciones prácticas. Gracias a estos esfuerzos, se pueden crear tecnologías innovadoras que satisfagan las necesidades de las personas y promuevan el progreso en diferentes áreas de la sociedad.

Además de la investigación y el desarrollo, existen otros factores importantes que influyen en el proceso de avance tecnológico, como la financiación, el personal y los recursos adecuados. Es importante destacar que la ciencia y la tecnología están estrechamente relacionadas, y los avances en un área a menudo impulsan el desarrollo en la otra. No obstante, en el contexto de la realidad nacional, la economía y el comercio también juegan un papel fundamental en el desarrollo de proyectos tecnológicos. Por lo tanto, es necesario recopilar información relevante para la evaluación adecuada de las circunstancias económicas y comerciales previas al inicio del proyecto. De esta forma, se pueden abordar adecuadamente los desafíos y oportunidades que se presenten durante el proceso de investigación y desarrollo.

Conforme con el proyecto “Diseño e Implementación de un Sistema Traductor de Lenguaje de Señas de Manos a un Lenguaje de Texto Mediante Visión Artificial en un Ambiente Controlado” presentado por (Chiguano y Moreno, 2011) afirman que:

El proyecto ofrece a las personas que no tienen conocimiento del lenguaje, la opción de un entrenador para que aprendan cada uno de los símbolos y adquieran la habilidad necesaria para utilizar la aplicación. Este proyecto se desarrolló en varias etapas: adquisición de imágenes, donde se debe utilizar un fondo negro para lograr mejor contraste y una iluminación detrás de la cámara para que la mano tenga características bien definidas sus características, evitando la presencia de sombras y ruido. En el procesamiento digital de imágenes se aplicaron algunos filtros y operaciones morfológicas para resaltar las características de la imagen y eliminar información

innecesaria como ruido. Con la ayuda del Vision Assistant de Labview se elaboraron las bases de datos, para llevar a cabo la comparación con la imagen recortada y de esta manera asignar la clase (letra) correspondiente a cada imagen para después formar el texto que se muestra en forma escrita en la pantalla o a su vez se puede enviar a un documento de Word. Además, es posible reproducir mediante voz el texto formado con la ayuda del reproductor de Windows. (p.6)

Por otra parte, en la investigación denominada “Reconocimiento e Interpretación del Alfabeto Dactilológico de la Lengua de Señas mediante Tecnología Móvil Y Redes Neuronales Artificiales” presenta un sistema de reconocimiento e interpretación del alfabeto dactilológico mediante una red neuronal artificial desarrollada en TensorFlow y siguiendo la metodología de visión artificial apoyada por la librería OpenCV. Para la construcción de la red neuronal se utilizó la técnica de transferencia de aprendizaje reentrenando una red neuronal convolucional previamente entrenada. Posteriormente se optimizó este modelo de red neuronal para ser implementado en una aplicación para dispositivos móviles con sistema operativo Android. Finalmente se hicieron experimentos con la aplicación móvil para evaluar la fiabilidad del sistema, del cual, se obtuvieron resultados satisfactorios bajo ciertas condiciones controladas de distancia e iluminación. (Aquino, 2018, p.5)

En este sentido, El estudio consistente en el “Desarrollo de un Intérprete Básico del Lenguaje de Señas para Dactilología empleando Inteligencia Artificial” muestra una metodología para el desarrollo de un sistema intérprete de lenguaje de señas básico empleando inteligencia artificial, utilizando diferentes algoritmos en MATLAB para el procesamiento de imágenes y desarrollo de código para las capturas de imágenes y evaluación de resultados en C#. Primeramente, se realizó el análisis del algoritmo para el reconocimiento de patrones. Se entrenó una red neuronal capaz de reconocer imágenes de lenguaje de señas, posteriormente se ejecuta el mismo análisis, pero con imágenes obtenidas con el sensor Kinect y finalmente, se integra el diseño del algoritmo de la red neuronal con el sensor Kinect para formar un sistema de visión artificial. (Rivas, 2019, p.5)

Como aporte importante en la publicación “Diseño e Implementación de una Interfaz para la Traducción de Caracteres del Alfabeto Dactilológico a Palabras Utilizando Sensores Musculares y Reconocimiento De Patrones” se indica que se desarrolló un sistema dirigido a personas con discapacidad auditiva que hacen uso del alfabeto dactilológico para su comunicación. Este dispositivo permite la traducción de los símbolos del alfabeto convencional representadas en señas a letras utilizando modelos de aprendizaje automático supervisado. La implementación, procesamiento y clasificación se ejecuta sobre una plataforma embebida FPGA PYNQ-ZI, la misma que es desarrollada bajo el sistema de código abierto Linux compatible con el programa Python y sus diferentes librerías. El proceso se desarrolla inicialmente con la adquisición de los

datos por medio del sensor muscular Myo Armband, luego se realiza el procesamiento de las señales electromiográficas EMG a través de Análisis Multirresolución de la Transformada Wavelet Discreta, con su wavelet madre Daubechies 4 y nivel de descomposición 2. Una vez procesadas las señales se extrae información útil, por medio de la extracción de características Valor Absoluto Medio (MAV), Raíz Cuadrática Media (RMS) y Longitud de la forma de Onda (WL) de cada uno de los subconjuntos de datos. Posteriormente, con las características obtenidas, se prepara el conjunto de datos para los modelos de aprendizaje supervisado seleccionados: red neuronal Multi Layer Perceptron, Árbol de decisión y Random Forest. (Ñacato, 2019, p.16)

Con dichos aportes documentales es posible el desarrollo de un sistema que controle un dispositivo eléctrico mediante el reconocimiento del lenguaje de señas dactilológico por visión artificial.

2.2 Referencias Teóricas

2.2.1 *Lenguaje Dactilológico*

Es un lenguaje de señas generado por los movimientos gestuales de la mano, en donde se busca interpretar cada letra del alfabeto a través de diferentes figuras o formas. Este lenguaje varía de comunidad en comunidad, donde cada una maneja un gentilicio para su idioma gestual, por ejemplo, las personas con discapacidad auditiva en comunidades de habla francesa utilizan el lenguaje des signes française conocido como LSF, en los Estados Unidos de América (USA) este lenguaje se conoce como american sign language (ASL), y en el idioma español en donde se enmarca el lenguaje de signos colombiano, este se conoce como lengua de signos española (LSE). (Betancur, Vélez y Peña, 2013, p.1) En la Ilustración 1-2 se observa los gestos para la comunidad de hispana.



Ilustración 1-2: Movimientos gestuales que definen el lenguaje dactilológico en español.

Fuente: (Betancur, et al., 2013)

La dactilología es la representación manual de cada una de las letras que componen el alfabeto. A través de ella se puede transmitir a la persona sorda cualquier palabra que se desee comunicar, por complicada que ésta sea. El deletreo es una parte importante del sistema de comunicación de las personas sordas. Se trata, sencillamente, de la escritura del alfabeto castellano ejecutada en el aire en lugar de un papel. Existen veintinueve posiciones con sus variantes de movimiento de mano, algunas de las cuales son la representación exacta de la letra. El deletreo manual es usado en combinación con el lenguaje de signos para sustantivos, nombres propios, direcciones y palabras para las cuales no existe un ideograma o signo creado o es poco conocido por la comunidad signante, como ocurre con signos de reciente creación (neologismos) o palabras poco usuales. Su importancia no puede ser subestimada; es por tanto esencial para la persona que se inicie en la lengua de signos, concentrarse en desarrollar tanto las habilidades receptivas como las expresivas, con el fin de adquirir experiencia. Para realizar la dactilología, se utiliza la mano dominante (derecha para los diestros, e izquierda para los zurdos). Se ejecuta principalmente a la altura de la barbilla. Su realización se complementa con la articulación oral, por lo que es necesario que la cara y la boca sean visibles. (Vilches, 2005, p.1)

2.2.2 *Sistema de Control*

Se puede considerar un sistema como una caja o diagrama de bloques que cuenta con una entrada y una salida, en donde lo importante no es lo que sucede dentro de la caja sino solo la relación

entrada- salida. Para predecir la forma en la que los sistemas se comportan cuando tienen entradas, es necesario elaborar modelos que relacionen la entrada con la salida; así como predecir cómo variará en el tiempo un sistema con una entrada dada. A los sistemas de medición, por lo general, se les puede considerar para crear los tres elementos básicos: sensor, acondicionador de señal y visualizador. Existen dos formas básicas de sistema de control: sistema lazo abierto y sistema lazo cerrado. Con el sistema de lazo cerrado hay retroalimentación, es un sistema que cuenta con un elemento de comparación, un elemento de control, un elemento de corrección, un elemento de proceso y la retroalimentación que incluye un elemento de medición. (Bolton, 2017, p.26)

2.2.3 *Mediapipe*

Con una denominación más técnica, se define a MediaPipe como una framework, para crear canalizaciones de aprendizaje automático para datos de series temporales, como video y audio. Está diseñado para los profesionales del aprendizaje automático (ML), incluyendo investigadores, estudiantes y desarrolladores de software, y su principal uso es el prototipado rápido de puntos de referencia en imágenes con modelos de inferencia y otros componentes reutilizables (Lugaresi et al., 2019). Este marco multiplataforma es compatible con escritorio/servidor, Android,iOS y dispositivos integrados como Raspberry Pi y Jetson Nano. (Sánchez, 2022)

Este framework contiene modelos de ML para detección de rostros, seguimiento de manos, segmentación de cabello, detección y seguimiento 2D y 3D de objetos y detección y seguimiento de posturas humanas entre muchos otros. MediaPipe está desarrollado para que pueda trabajar en Android, IOS, Python, C++, Javascript y Coral. Se podría pensar que la utilización de modelos ML en dispositivos con Android o IOS, que mayormente son smartphones o tabletas. (Cáceres, 2021, p.32)

Los fotogramas por segundo o frames per second (FPS) que utiliza el algoritmo MediaPipe de acuerdo con Cáceres (2021): "... hasta los 30.2 FPS de media en el caso de MediaPipe. Es por ello por lo que, el algoritmo que utiliza MediaPipe sería el de utilización más sencilla y cómoda, siendo también la más precisa..." (p.3)

2.2.4 *Reconocimiento de manos*

(Sung et al., citado en Sánchez, 2022) manifiestan que "El reconocimiento de gestos con la mano (HGR) es una forma natural e intuitiva de interacción persona-ordenador (HCI), que ha sido objeto de numerosos estudios; con una amplia gama de dispositivos de entrada y enfoques, el HGR basado

en el esqueleto se convirtió en una opción popular debido a su resistencia a las fluctuaciones del fondo y la luz” (p.8).

“Entre las herramientas utilizadas para el reconocimiento de gestos con la mano, MediaPipe dispone de una solución que emplea ML para inferir 21 puntos de referencia en 3D de una mano, a partir de un solo fotograma” MediaPipe (Sánchez, 2022, p.8) como se presenta en la Ilustración 2-2.



Ilustración 2-2: Puntos de referencia en la mano para su reconocimiento por MediaPipe.

Fuente: MediaPipe (Sánchez, Leandro, 2022)

2.2.5 Python

Python es un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido. La sintaxis de Python es tan sencilla y cercana al lenguaje natural que los programas elaborados en Python parecen pseudocódigo. Por este motivo se trata además de uno de los mejores lenguajes para comenzar a programar. Python no es adecuado sin embargo para la programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico. Algunos casos de éxito en el uso de Python son Google, Yahoo, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles. (González, 2011, p.32)

2.2.6 Tensorflow

Gómez (Vera, 2019) afirma que es una biblioteca de código abierto dirigida al aprendizaje automático a través de una serie de tareas. Ha sido desarrollado por Google para satisfacer las necesidades de sistemas capaces de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos. Actualmente es

utilizado tanto para la investigación como para la producción de productos de Google, reemplazando el rol de su predecesor de código cerrado, DistBelief. TensorFlow fue originalmente desarrollado por el equipo de Google Brain para uso interno en Google antes de ser publicado bajo la licencia de código abierto Apache 2.0 el 9 de noviembre de 2015. (p.26)

2.2.7 OpenCv

La librería OpenCv (Caballero, 2017) (Open Source Computer Vision) es una biblioteca de visión artificial iniciada por Intel en 1999. Esta biblioteca multiplataforma se centra en el procesamiento de imágenes en tiempo real e incluye implementaciones libres de patentes de los últimos algoritmos de visión computarizada. Esta librería está escrita en C y C++ y cuenta con aproximadamente quinientas funciones que abarcan diferentes procesos de la visión artificial. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real. La librería permite el manejo eficiente de imágenes y matrices, reconocimiento de objetos y análisis de imágenes utilizando la técnica de Eigenfaces y el algoritmo AdaBoost

OpenCv cuenta con la clase de FaceRecognizer para el reconocimiento facial, Los algoritmos disponibles son:

- a) Eigenfaces
- b) Fisherfaces
- c) Local Binary Patterns Histograms

OpenCv nos permite detectar un rostro mediante el uso de clasificadores en cascada, y realizar el reconocimiento de un rostro específico utilizando Eigenfaces. Por lo anterior se eligió esta librería para el desarrollo de la aplicación para el reconocimiento de una persona. (p.32)

Jupiter-Notebook: De acuerdo con (Cabrera y Díaz citado en Marín, 2019) “Es una herramienta que posibilita generar documentos interactivos, ofrece un entorno rico en contenidos que facilita el aprendizaje, integrando de manera natural simulaciones realizadas en Python” (p.26).

2.2.8 PyCharm Community

Es uno de los entornos de Desarrollo más completos. Cuenta con entornos para construir Código en distintos lenguajes como PHP o Ruby. Pycharm ofrece distintos servicios y en la versión community es totalmente gratuito: (Moreno, 2020, p.5)

- 1) Asistencia inteligente a Python. Pycharm proporciona finalización del Código inteligente, inspecciones del Código, indicación de errores sobre la marcha y arreglos rápidos, así como refactorización de Código automática y completas funcionalidades de navegación.
- 2) Marcos de trabajo de Desarrollo web. Pycharm ofrece una gran compatibilidad con marcos de trabajo de Desarrollo web modern como Django, Flask, Google App Engine...
- 3) Herramientas científicas. Cuenta con una consola de Python interactive y es compatible con Anaconda y varios paquetes científicos como matplotlib y NumPy.
- 4) Desarrollo multitecnología. Además de Python, PyCharm es compatible con JavaScript, HTML/CSS...
- 5) Capacidades para desarrollo remoto. Ejecuta, depura, prueba y desarrolla aplicaciones en máquinas virtuales, Herramienta de Reconocimiento de Imágenes etc.
- 6) Herramientas de Desarrollo. Una gran colección de herramientas listas para usar, como un depurador integrado y ejecutor de pruebas.
- 7) IDE multiplataforma. PyCharm Community funciona en Windows, Mac OS o Linux. Es posible instalar y ejecutar PyCharm en tantas máquinas como se tenga, y usar el mismo entorno y la misma funcionalidad.
- 8) Cabe destacar que PyCharm community está diseñado por programadores y para programadores, con el fin de proporcionarles herramientas que son necesarias para un desarrollo productivo de Python.
- 9) Este lenguaje de programación está presente en muchas aplicaciones y sistemas operativos, como iOS, Windows, Linux, Mac o Android, debido a que tiene una curva de aprendizaje moderada, con un Código legible. Es un Código versátil multiplataforma que se destaca por su Código limpio. Su éxito principalmente se debe a que es un Código abierto, que permite su uso desde cualquier rincón del mundo. Es importante destacar que

en este software se trata con la version Python 3.8 (32 bits), actualmente la más reciente. Tiene como características (Moreno, 2020, p.6):

- 10) Lenguaje multiplataforma. Se puede usar en distintos sistemas operativos.
- 11) Ideal para trabajar con volúmenes de datos muy grandes, favorece el procesamiento, siendo también el lenguaje utilizado por las empresas de Big Data. Algunas de sus ventajas son: Sencillez y presteza, selecto, legible, fácil de aprender, organizado, portable y con un gran número de usuarios que participan activamente en su Desarrollo.
- 12) Imperativo. Describen el estado del programa y permiten la modificación mediante instrucciones de Código. Se describe paso a paso un conjunto de instrucciones.
- 13) Funcional. Variación del programa mediante la mutación y el cambio de variables. Así, se opera con datos de entrada y salida.
- 14) Orientado a Objetos. Los objetos manipulan a otros, para ofrecer una salida específica. También permite juntar librerías.
- 15) Dinámico. Una variable puede tomar distintos valores de diferentes tipos en distintos momentos.

2.2.9 Mosquitto

Permite una escalabilidad amplia en diversos dispositivos gracias a ser liviano. Son considerados Herramientas de “Cola de mensajes Se basa de una arquitectura de poseer un núcleo y subproceso único que permite admitir dispositivos integrados. Se ejecuta en cualquier microcontrolador de baja potencia debido a ser liviano y compacto. Provee líneas de comando simple de publicación y suscripción, Soporta QoS 0,1 y 2 y Poseen hilos de seguridad ante ataques Los dos broker son compatibles entre sí, EMQ admite otros protocolos de comunicación y acceso.(Amaguaya, 2020, p.26)

Topic: Es el tema del mensaje, es decir, a quien va dirigido ese mensaje o de quien se quiere recibir tal mensaje. El topic sirve como identificador al broker para poder encaminar cada uno de los mensajes recibidos hacia los clientes adecuados que también se hayan suscrito al topic. Los topicos son case sensitive, es decir, distinguen entre mayúsculas y minúsculas. Estos nos brindan la posibilidad de poder jerarquizarlos.(Mahedero, 2020, p.26)

2.2.10 *Inteligencia Artificial (IA)*

Es importante resaltar que la idea de inteligencia es también un concepto abstracto y al crear programas inteligentes en los ordenadores, se debe decidir si el programa pensará y actuará como un humano o lo hará de forma racional y de igual manera, diseñar los comandos para que el programa lo haga. Entre las aplicaciones de la IA se tiene el análisis de grandes cantidades de datos, búsqueda de patrones, optimización de procesos, resolución de problemas, por mencionar algunos. (Sánchez, 2022, p.4)

2.2.11 *Visión Artificial*

Conforme con Branch & Olague (Caballero, 2017) La visión artificial por computadora es una disciplina de gran crecimiento en los últimos años, su futuro más prometedor se encuentra en la creación de máquinas autónomas capaces de interactuar de forma inteligente con el entorno. Para lograr la interacción de las máquinas con su entorno, es necesario desarrollar un sistema que tenga la capacidad de percibirlo e interpretarlo. (p.8)

2.2.12 *Red Neuronal*

La unidad de una red neuronal artificial es un procesador elemental llamado neurona que posee la capacidad limitada de calcular, en general, una suma ponderada de sus entradas y luego le aplica una función de activación para obtener una señal que será transmitida a la próxima neurona. Estas neuronas artificiales se agrupan en capas o niveles y poseen un alto grado de conectividad entre ellas, conectividad que es ponderada por los pesos. A través de un algoritmo de aprendizaje supervisado o no supervisado, las ANN (Artificial Neural Network) ajustan su arquitectura y parámetros de manera de poder minimizar alguna función de error que indique el grado de ajuste a los datos y la capacidad de generalización de las ANN. (Salas, 2004, p.1)

Para definir un modelo de red neuronal como se muestra en la en la Ilustración 3-2. MediaPipe toma un complemento función ya que los gráficos no definen las partes internas de una red neuronal, sino que especifican conductos a mayor escala en los que uno o los modelos modelo están incrustados. McCulloch and Pitts en 1943 [MP43] concibieron un modelo abstracto y simple de una neurona artificial, este es el elemento básico de procesamiento en una red neuronal artificial.

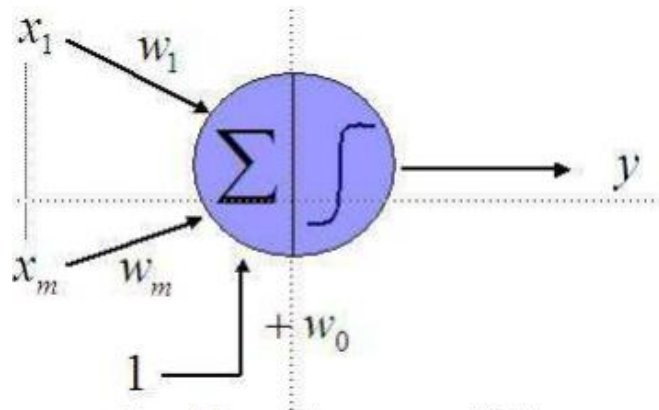


Figura 1: Esquema de una neurona artificial

Ilustración 3-2: Esquema de una neurona artificial.

Fuente: (Salas, 2004)

Las Redes Neuronales Artificiales según (Granger, citado en Aristizábal, 2006) son modelos computacionales que tratan de replicar, de manera simplificada, el complejo funcionamiento del cerebro humano. Su capacidad de aprendizaje a través de ensayos repetidos las ha hecho muy populares en una amplia variedad de aplicaciones en todas las ciencias. Su reciente implementación en economía se debe al hecho de que en las series económicas y financieras es más probable que aparezcan relaciones no lineales que lineales. (p.1)

En el desarrollo de un sistema de control de dispositivos eléctricos que utiliza reconocimiento de lenguaje de señales dactilológico, una red neuronal secuencial con capas de entrada dropout y capas densas fue una herramienta valiosa para procesar y analizar los datos capturados por la placa de visión artificial. La red neuronal secuencial es un tipo de modelo de red neuronal que consiste en una secuencia lineal de capas. Cada capa procesa la información de la capa anterior y pasa el resultado a la capa siguiente. Esta arquitectura permite a la red aprender una gran cantidad de información y realizar tareas complejas de análisis de datos.

Las capas de entrada dropout son una técnica de regularización que ayuda a prevenir el sobreajuste, un problema común en el entrenamiento de redes neuronales. Dropout consiste en ignorar aleatoriamente una parte específica de los nodos de una capa durante el entrenamiento, lo que ayuda a evitar que la red dependa demasiado de cualquier característica en los datos de entrenamiento. No obstante, las capas densas, por otro lado, son capas completamente conectadas en las que cada nodo de una capa está conectado con todos los nodos de la siguiente capa. Estas capas son utilizadas para aprender y extraer patrones complejos en los datos y son una parte clave de muchos modelos de red neuronal.

En sí, la combinación de una red neuronal secuencial con capas de entrada dropout y capas densas puede ser una herramienta efectiva para procesar y analizar los datos capturados por la placa de visión artificial en el desarrollo de un sistema de control de dispositivos eléctricos utilizando reconocimiento de lenguaje de señales dactilológico. La red puede aprender y extraer patrones complejos en los datos y utilizarlos para realizar tareas de reconocimiento de señales dactilológicas con una alta precisión.

2.2.13 Anaconda

Anaconda (García, 2023) Es un sistema y entorno de gestión de paquetes de código abierto, sistema de administración que se ejecuta en Windows, Mac OS y Linux. Conda instala, ejecuta y actualizar rápidamente paquetes y sus dependencias. Conda crea, guarda, carga y cambia fácilmente entre entornos en el equipo local. fue creado para programas Python, pero puede empaquetar y distribuir software para cualquier lenguaje Como administrador de paquetes le ayuda a encontrar e instalar paquetes. sí necesita un paquete que requiere una versión diferente de Python, no se necesita cambiar a un entorno diferente porque Conda también administra diferentes medios con tan solo unos pocos comandos, puede configurar un entorno totalmente separado para ejecutar esa versión diferente de Python, mientras continúa ejecutándose su versión habitual de Python en su entorno normal. (p.11)

2.2.14 Módulo Wifi ESP8266

El módulo WiFi ESP8266 es un microcontrolador que incorpora un chip WiFi y se utiliza principalmente para la conectividad inalámbrica de dispositivos electrónicos. Este módulo es pequeño y económico, lo que lo hace ideal para proyectos de Internet de las cosas (IoT) y automatización doméstica, en donde Rentería, Ruelas y Ochoa (2017) indican que “Un módulo Wifi ESP8266 de bajo costo es ideal para aplicaciones de internet de las cosas (IOT) y hay que saber cómo elegir el más adecuado para nuestros futuros proyectos, enfocados en el internet de las cosas” (p.24). En este sentido Granados (2017) indica que “Programando directamente un ESP8266 para que opere de forma independiente, se puede utilizar como punto de acceso. La finalidad es poder acceder a toda la información que nos pueda aportar, sin necesidad de acceso a internet” (p.1). Por lo tanto, El ESP8266 puede funcionar como un dispositivo cliente o como un servidor, lo que significa que puede conectarse a una red WiFi existente o crear su propia red WiFi para que otros dispositivos se conecten a él. Además, puede ser programado para realizar diferentes tareas, como controlar dispositivos electrónicos, recopilar y transmitir datos, y mucho más.

En resumen, el módulo WiFi ESP8266 es una solución potente y versátil para proyectos de IoT que requiere conectividad inalámbrica y procesamiento de datos en tiempo real.

2.2.15 Módulo de Relé

Un módulo de relé es un dispositivo electrónico que se utiliza para controlar la energía eléctrica que fluye a una carga, como una lámpara o un motor. Está compuesto por un relé, que es un interruptor eléctrico controlado por una señal de corriente, y un circuito de control que le permite activarse o desactivarse. Para dicha situación, en el desarrollo de un sistema de control de dispositivos eléctricos utilizando reconocimiento de lenguaje de señales dactilológico mediante visión artificial con el módulo ESP8266, puede utilizarse un relé de estado sólido con optoacopladores Márquez (2018)

Este relé combina las desconexiones de estado sólido con los optoacopladores para separar el circuito de control del circuito de carga, lo que aumenta la seguridad y la confianza del sistema. Además, debido a que se utiliza visión artificial para el reconocimiento de lenguaje de señales dactilológico, es probable que la corriente y el voltaje necesarios para controlar los dispositivos eléctricos sean relativamente bajos, por lo que un relé de estado sólido con optoacopladores es una buena opción.

En cuanto al optoacoplador, es un dispositivo en el que se utiliza un LED para acoplar un fotodiodo o un fototransistor en un solo paquete. En un fototransistor la corriente en la base se produce cuando la luz choca con la región de la base. Se expone la Unión pn colector-base a la luz incidente mediante la abertura de una lente incluida en el transistor. Cuando no hay luz incidente, sólo hay una pequeña corriente generada térmicamente del colector al emisor. Se utilizan optoacopladores para aislar secciones de un circuito que son incompatibles en términos de niveles de voltaje o corrientes requeridas. Por ejemplo, para aislar circuitos de señalización o de control de baja corriente, de circuitos de suministro de potencia ruidosos, o de circuitos de máquinas y motores de alta intensidad de corriente. (Márquez, 2018, p.52)

2.2.16 Transistor 2N222

El transistor 2N222 es un tipo de transistor NPN que se utiliza comúnmente como amplificador o conmutador en aplicaciones de electrónica. En el desarrollo de un sistema de control de dispositivos eléctricos utilizando reconocimiento de lenguaje de señales dactilológico mediante visión artificial, fue necesario utilizar un transistor 2N222 para controlar los dispositivos

eléctricos a través de un relé, puesto que funciona como un interruptor que puede controlar la corriente eléctrica que fluye a través de un circuito.

El uso de un transistor 2N222 para controlar los dispositivos eléctricos puede ser una buena opción si se requiere una fuente de control con corriente y voltaje relativamente bajo para activar un relé. Cuando se aplica una corriente pequeña al terminal de base, se produce una corriente mucho mayor en el terminal de colector, lo que permite controlar un circuito de carga conectado al terminal de colector.

2.2.17 Regulador de 5V

Un regulador de 5 voltios contribuye en el desarrollo del prototipo al proporcionar una fuente estable y confiable de energía para los componentes electrónicos del sistema. En este tipo de sistema, se utilizan componentes electrónicos que requieran una alimentación de 5 voltios, en donde se haya la relación con el módulo ESP8266. Si la alimentación varía en voltaje o está contaminada con ruido eléctrico, puede afectar el funcionamiento de estos componentes y aumentar el riesgo de errores y caídas en el sistema. En la figura 4-2 se visualiza la forma y tamaño del regulador de 5V.



Ilustración 4-2: Regulador de voltaje.

Fuente: (Mahedero,F., 20)

Este tipo de regulador se encarga de convertir la alimentación eléctrica disponible a un voltaje constante de 5 voltios, lo que proporciona una fuente de energía estable para los componentes electrónicos. Esto asegura que el sistema tenga una alimentación confiable y reduzca el riesgo de fallas relacionadas con la alimentación eléctrica.

2.2.18 Software para el diseño y simulación de circuitos eléctricos

El software para el diseño y simulación de circuitos eléctricos es una herramienta que permite a los usuarios diseñar, simular y probar circuitos eléctricos antes de construirlos en el mundo real. Estos software suelen incluir una amplia gama de componentes eléctricos, desde resistencias y condensadores hasta microcontroladores y dispositivos de entrada/salida, que pueden ser utilizados para crear modelos de circuitos complejos, por lo que Pérez et al. (2022) afirma:

Es un software para diseño y simulación de circuitos eléctricos permite visualizar, construir y realizar simulaciones interactivas de circuitos eléctricos mediante una interfaz gráfica. De esta forma es posible aprender cómo funcionan los circuitos eléctricos, cuáles son los parámetros para tener en cuenta en el diseño de un determinado circuito de control y cómo afectan los cambios en los diferentes elementos. Todo esto ayuda a mejorar el proceso de enseñanza-aprendizaje contrastando los conocimientos teóricos estudiados con los prácticos mediante el método de la simulación. (p.157-172)

Además, los softwares para el diseño y simulación de circuitos eléctricos suelen incluir herramientas de análisis, como el análisis de corriente y voltaje, el análisis de transitorios y el análisis de ruido, que permiten a los usuarios evaluar el comportamiento y las propiedades eléctricas de los circuitos simulados. Estos son muy utilizados en la industria, la investigación y la educación, y permiten a los usuarios crear y probar diseños de circuitos antes de construirlos y ahorrar tiempo y recursos en la construcción y prueba de prototipos.

CAPITULO III

3. MARCO METODOLÓGICO.

3.1 Introducción

Este capítulo se enfoca en describir de manera clara y organizada todos los procesos esenciales para la implementación del sistema para el control de dispositivos eléctricos utilizando reconocimiento de lenguaje de señas dactilológico mediante visión artificial de manera fácil e intuitiva. De esta forma, los usuarios y personas interesadas podrán hacer uso de este sistema de manera apropiada.

3.2 Tipos de investigación

Dentro de la investigación es importante entender todos los procedimientos que se ejecutaron para la implementación del sistema, al igual que la sistematización de la información extraída. En este estudio se optó por la investigación experimental y aplicada debido a su pertinencia para abordar el estudio de las tecnologías para reconocimiento de imágenes y evaluar el funcionamiento del sistema. De tal forma que se logre alcanzar un conocimiento profundo y sistematizado sobre el tema. Se llevaron a cabo varias de pruebas para recopilar datos y analizarlos, lo que posibilita la extracción de conclusiones y recomendaciones concretas sobre la tesis.

3.2.1 *Investigación Experimental*

Un experimento es una investigación en la cual el investigador manipula y controla una o más variables independientes y observa la o las variables dependientes para medir las variaciones concomitantes. En el estudio se utilizó este tipo de investigación para evaluar la eficacia y eficiencia del sistema en términos de su capacidad para detectar y reconocer con precisión las señas dactilológicas y controlar dispositivos eléctricos. La investigación experimental posibilitó experimentar y comparar con diferentes algoritmos de reconocimiento y señas dactilológicas, así como también evaluar el impacto del entorno y las condiciones ambientales en la precisión del reconocimiento. Esto permitiría mejorar la eficacia y la eficiencia del sistema y asegurarse de que funcione de manera efectiva y confiable en un entorno real.

3.2.2 *Investigación Aplicativa*

Para Murillo (2008), la investigación aplicada recibe el nombre de “investigación práctica o empírica”, que se caracteriza porque busca la aplicación o utilización de los conocimientos adquiridos, a la vez que se adquieren otros, después de implementar y sistematizar la práctica basada en investigación. El uso del conocimiento y los resultados de investigación que da como resultado una forma rigurosa, organizada y sistemática de conocer la realidad.

Para este sistema se enfoca en la implementación y evaluación práctica de una solución utilizando métodos y técnicas específicas para el desarrollo del sistema. La investigación aplicada es adecuada para este tipo de proyecto porque se requiere una solución práctica y efectiva para el control de dispositivos eléctricos mediante reconocimiento de lenguaje deseñas utilizando visión artificial.

3.3 Equipos y materiales

Para la ejecución de esta investigación se emplearon diversos materiales. A continuación, se detalla la utilidad de cada material en la implementación del proyecto:

3.3.1 *Hardware*

El hardware destinado para este sistema que controla dispositivos eléctricos utilizando reconocimiento de lenguaje de señas dactilológico mediante visión artificial contiene los siguientes elementos:

- **Computador:** Laptop marca, ASUS STRIX RYZEN 7 - 16GBRAM – 512 GB SSD, Windows 11, GPU RTX 3050. Este computador se emplea como servidor que a través de una cámara web está censando los movimientos de las manos y activa los dispositivos eléctricos. Por otra parte, también se encarga del procesamiento de imágenes y envió de datos al servidor mosquito.
- **Módulo Wifi ESP8266:** Se utiliza para que los dispositivos eléctricos tengan una conexión a internet y de esta forma pueda recibir las señales desde el servidor Mosquito para que posteriormente se accionen los dispositivos eléctricos.
- **Módulo Relé:** Se usa el voltaje de 110V de los dispositivos eléctricos a través de un módulo relé que hace uso de un transistor 2N2222. Dicho transistor recibe 3V en la base

mediante el módulo ESP8266 e inyecta 5V en el emisor para activar el módulo relé y de esta forma accionar el dispositivo eléctrico.

- **Alarma Sonora 110V:** Dispositivo electrónico que emite un sonido o señal para alertar o avisar sobre un evento o situación determinada. Normalmente estas alarmas están conectadas a una fuente de alimentación eléctrica de 110V. Se emplea como actuador que se acciona al reconocer un gesto.
- **Foco:** Dispositivo de iluminación que se utiliza para iluminar un área o para iluminar objetos. En el presente trabajo, este dispositivo se utiliza como actuador que se acciona al reconocer un gesto.
- **Cerradura Eléctrica:** Tipo de cerradura que funciona mediante un mecanismo eléctrico y se activa y desactiva con una señal eléctrica. Se emplea como actuador, que se acciona al reconocer un gesto
- **Regulador de 5V:** Estabilizar la tensión de entrada y proporciona una tensión de salida constante de 5 voltios. Se utiliza como fuente de energía estable y regulada a diferentes a diferentes componentes electrónicos. En el proyecto alimenta a el módulo ESP8266 y el módulo Relé.
- **Cámara WEB:** Este sirve como sensor de señas que se utiliza para capturar los gestos para que posteriormente procese estas imágenes en el computador y se envíe a través del protocolo MQTT para finalmente accionar los dispositivos.

3.3.2 *Software*

- **Arduino 1.8.19:** En esta plataforma se desarrolló la programación del módulo ESP8266 para conectarse a una red local y enviar datos al servidor Mosquitto.
- **Anaconda:** En el proyecto se hizo uso de esta herramienta para manejar un ambiente aislado y así evitar problema al momento de instalar librerías y dependencias en Python.
- **PyCharm Community:** En este entorno integrado se llevó a cabo la escritura del código para la detección en tiempo real.

- **Mosquitto:** Se emplea mosquitto para enviar señales desde el computador hasta el módulo ESP8266 y este accione los dispositivos finales.
- **Jupyter-notebook:** En este proyecto se utiliza para escribir el código que contiene los parámetros para llevar a cabo el entrenamiento.

3.4 Estudio de Factibilidad

En esta sección, se selecciona la tecnología más adecuada para el proyecto, haciendo una evaluación comparativa utilizando la escala de Likert, dónde 5 es la puntuación de mejor percepción y 1 es la paupérrima correspondientemente, tal como se presenta en la tabla 1-3.

	MediaPipe	Object Detection	Action Recognition
Tiempo de entrenamiento	5	2	3
Tasa de Precisión	5	2	3
Integración	5	5	5
Rapidez	2	5	4
Versatilidad	5	5	5
Total	22	19	20

Tabla 1-3: Escala de Likert de las tecnologías

Realizado por: Saavedra Kevin, 2023

En la tabla 3-1 se puede argumentar que la tecnología más rápida es MediaPipe por encima de las otras tecnologías, además de tener mejor precisión. Por lo cual, fue selecciona como la más adecuada para la implementación del proyecto.

3.5 Desarrollo del Prototipo

El sistema de control de dispositivos eléctricos basado en reconocimiento de lenguaje de señas dactilológico utilizando visión artificial, consta de un computador que conectado a una cámara captura las señas, para posteriormente procesar esas imágenes con el algoritmo de MediaPipe Hand Tracking, el cual fue modificado para este proyecto. MediaPipe a través de redes neuronales artificiales entrenadas procesa las imágenes y posibilita la detección en tiempo real. Este entrenamiento se lleva a cabo en Jupyter-Notebook haciendo uso de imágenes recolectadas previamente, de esta forma la red neuronal artificial puede predecir de forma acertada por medio de librerías y dependencias que ofrece el lenguaje de programación Python para el entrenamiento de redes neuronales.

Una vez realizada la detección, el sistema mediante condicionales envía datos (señales) al servidor bróker MQTT. Este servidor hace de intermediario entre el suscriptor, dispositivo final, y el publicador, dispositivo que contiene los datos necesarios para hacer funcionar el sistema. Por medio de tópicos el computador tiene la posibilidad de publicar si el sensor reconoció algún gesto conocido. El sistema hace uso de tres tópicos, dentro de cada tópico se encuentra asociado un dispositivo final, llámese este; foco, sirena o cerradura. Cuando éste reconozca un gesto entrenado previamente enviará (Publicador) una señal, en este caso un carácter al tópico correspondiente, el dispositivo final (Suscriptor) asociado a dicho tópico consumirá dicha señal y de esta forma se accionará el dispositivo eléctrico.

Cuando el módulo ESP8266 (que está alimentado por un regulador de 5V dentro de una caja negra) recibe la señal desde el servidor MQTT haciendo uso de una conexión Wi.-Fi, este a través de un módulo relé que contiene un transistor 2N2222 que está alimentando al emisor con 5V, recibe esta señal de 3.3V proveniente del módulo ESP8266 y la amplifica, de esta forma se activa un relé que puede controlar dispositivos con voltaje de 110V.

Para desarrollar todo este sistema fue necesario hacer uso del software anaconda, debido a que este permite el manejo de ambientes aislados. De esta forma se evita problemas de compatibilidad con diferentes librerías, dependencias y otros paquetes ya instalados dentro de un mismo ambiente.

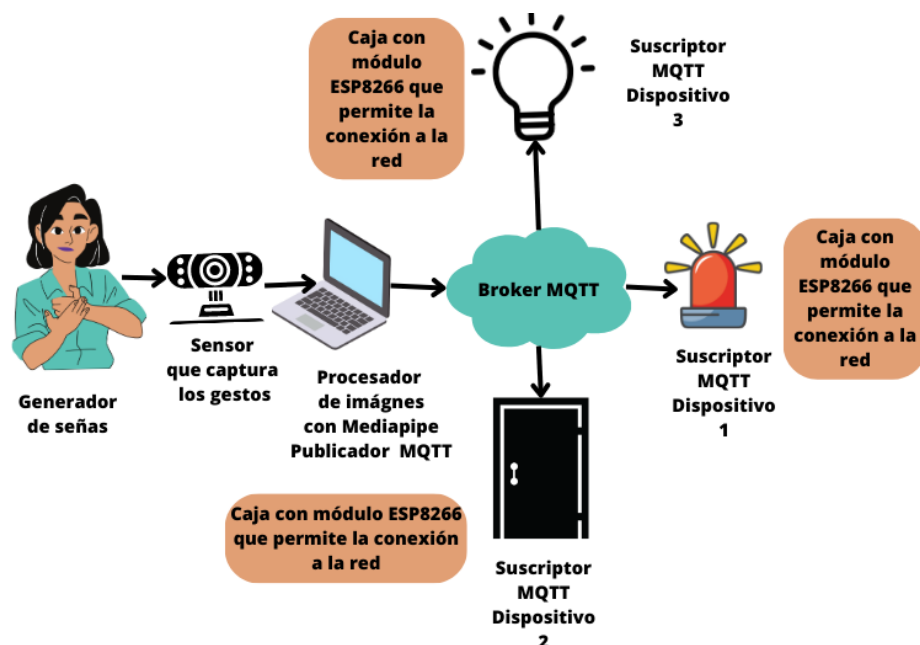


Ilustración 1-3: Esquema de funcionamiento del sistema para el control de dispositivos eléctricos.

Elaborado por: Saavedra Delgado, Kevin, 2023

3.5.1 Diseño del sistema

Para el diseño del sistema que se muestra en la Ilustración 1-3 se segmentó en dos etapas, la primera detalla el diseño desde el punto de vista del publicador como se observa en la Ilustración 2-3. Es aquí donde se realiza la mayor parte del procesamiento, debido a que aquí se lleva a cabo el reconocimiento de señas, es decir aquí se procesan todos los datos necesarios para activar los dispositivos eléctricos.

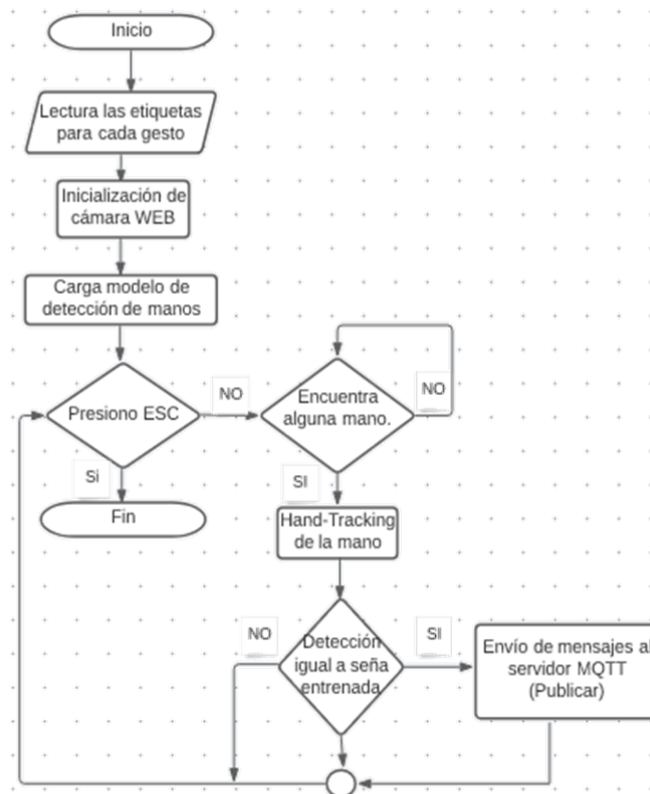


Ilustración 2-3: Esquema de funcionamiento del sistema (publicador)

Elaborado por: Saavedra Delgado, Kevin, 2023

Dentro de esta primera etapa está inmerso el entrenamiento del modelo, el cual fue diseñado de tal forma que a través del script principal existe una tecla que permita acceder a un modo que posibilita capturar imágenes en tiempo real y procesar estas imágenes en forma de tabla de datos en texto plano. Finalmente, estos datos son procesados a través de un modelo de red neuronal secuencial que se entrena con 1000 épocas, tamaño de lote 128, activación relu y optimizador Adam.

La segunda parte está compuesta por los dispositivos eléctricos, esta sección detalla la conexión de los dispositivos a la red y al servidor MQTT para escuchar los mensajes correspondientes a los respectivos tópicos y accionar los dispositivos como se muestra en la Ilustración 3-3.

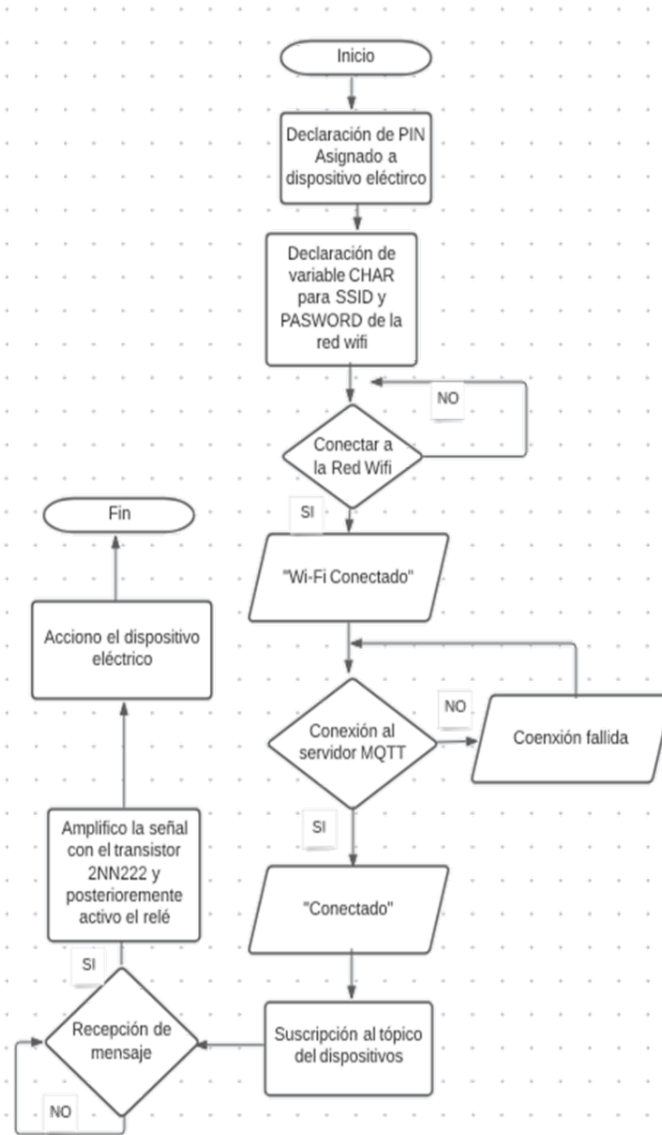


Ilustración 3-3: Esquema de funcionamiento del sistema desde el suscriptor

Elaborado por: Saavedra Delgado, Kevin, 2023

3.5.2 Instalación de Anaconda

Como se mencionó previamente, anaconda es un software muy importante dentro de este proyecto, haciendo uso de ambientes aislados. Gracias a estos ambientes se evitan conflictos entre librerías y dependencias no compatibles, garantizado así el funcionamiento del proyecto. Por otra parte, Anaconda desde su interfaz gráfica, tiene la facilidad de ejecutar aplicaciones desde otros

ambientes y trabajar con múltiples ambientes de manera simultánea. La instalación de este software es muy básica, fácil e intuitiva.

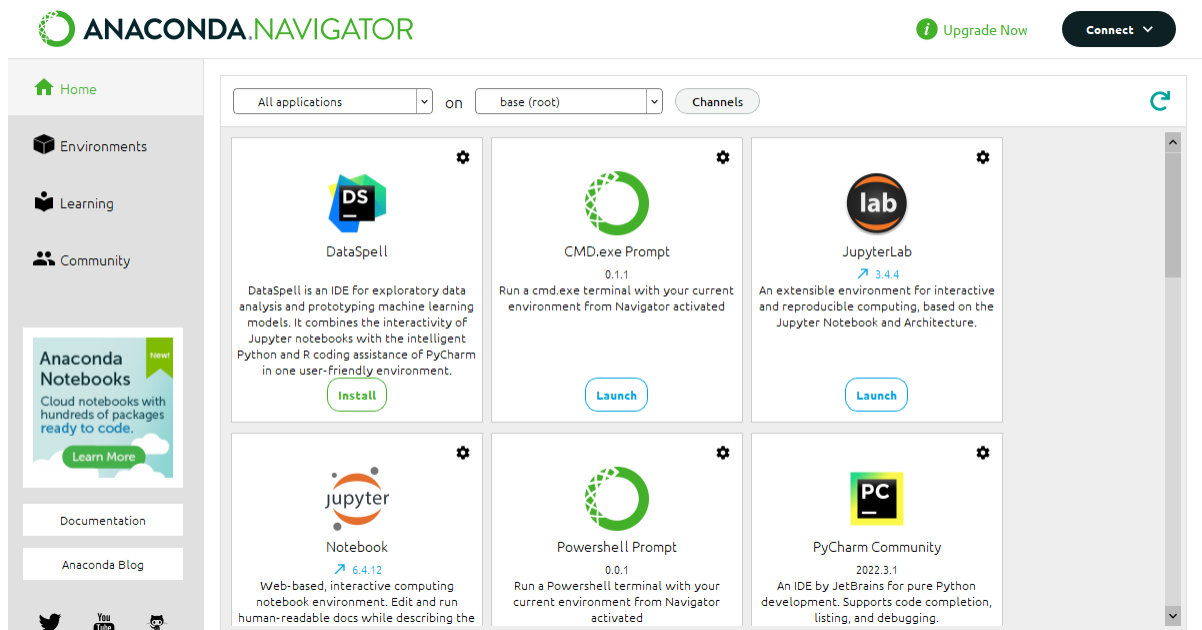


Ilustración 4-3: Instalación de Anaconda3 exitosa

Elaborado por: Saavedra Delgado, Kevin, 2023.

Luego de instalar Anaconda de manera exitosa como se observa en la Ilustración 4-3 se procede a la creación del ambiente aislado de trabajo donde se instalarán todas las librerías, paquetes y dependencias necesarias para la ejecución del proyecto. La creación de este ambiente se puede realizar muchas maneras, sin embargo, para este proyecto se optó por utilizar CMD debido a su popularidad y porque permite crear ambientes con cualquier versión de Python. El comando que se usa “*conda create -n TESIS2 python= 3.8.*” En la Ilustración 5-3 se puede apreciar la creación del ambiente TESI2 que es donde se alojará el proyecto.

```

Proceed ([y]/n)? y

Downloading and Extracting Packages
tzdata-2022g          | 114 KB | ##### | 100%
libffi-3.4.2         | 109 KB | ##### | 100%
xz-5.2.10            | 520 KB | ##### | 100%
#####xz-5.2.10      | 520 KB | ##### | 100%
sqlite-3.40.1        | 889 KB | ##### | 100%
ite-3.40.1           | 889 KB | ##5    | 29%
#####sqlite-3.40.1 | 889 KB | #####9 | 100%
#####sqlite-3.40.1 | 889 KB | ##3    | 29%
sqlite-3.40.1        | 889 KB | #####9 | 100%
python-3.10.9        | 15.8 MB | ##### | 29%
ca-certificates-2023 | 121 KB | ##### | 100%
setuptools-65.6.3    | 1.2 MB | ##### | 100%
pip-22.3.1           | 2.8 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate TESIS2
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#
Retrieving notices: ...working... done
C:\Users\SYSTEMarket>conda activate TESIS2
(TESIS2) C:\Users\SYSTEMarket>

```

Ilustración 5-3: Creación del ambiente TESIS2

Elaborado por: Saavedra Delgado, Kevin, 2023

3.5.3 Descarga de librerías para el funcionamiento del Prototipo.

Dentro del ambiente TESIS2 se procede a instalar las librerías más importantes para el funcionamiento del proyecto con el comando “*pip install*”. A continuación, se detalla estas librerías.

MediaPipe 0.8.1: Esta librería permite el procesamiento imágenes y videos, la detección de objetos y reconocimiento de gestos en tiempo real utilizando técnicas de aprendizaje automático y de visión por computadora. Para este proyecto basado en el reconocimiento de gestos se hizo uso de esta librería y del algoritmo de Google Open-Source Hand-Tracking.

OpenCV 3.4.2 or later: Esta librería permite el uso de la visión por computadora, para de esta forma utilizar la cámara como sensor de gestos a través de reconocimiento de patrones, seguimiento de movimiento, etc.

Tensorflow 2.3.0 or later: Esta librería permite la creación y entrenamiento de modelos de aprendizaje automático, así como su integración con otras plataformas. Para este proyecto se utiliza para crear y entrenar el modelo de reconocimiento de gestos para la activación de dispositivos eléctricos

Scikit-learn 0.23.2 or later: Esta librería proporciona herramientas para resolver problemas de aprendizaje automático, como la clasificación, la regresión y la agrupación y selección de

características. Para este proyecto es muy útil para agrupar características de un mismo gesto y clasificar los gestos de manera eficiente además de ayudar al muestreo de la matriz de confusión.

Matplotlib 3.3.2 or later: Esta librería permite la creación de gráficos en una amplia gama de aplicaciones. Para este proyecto se utiliza para los gráficos correspondiente para la matriz de confusión.

Paho.mqtt.client: Esta librería permite que permite que el cliente se conecten a servidores MQTT y publiquen y suscriban mensajes. Para este proyecto se utiliza para que el computador publique y los dispositivos finales se suscriban a los diferentes tópicos y de esta forma puedan accionarse cuando se detecte una señal a través del servidor broker.

KeyPointClassifier; Una clase importada desde una carpeta llamada model que permite clasificar los puntos clave del tracking de una mano facilitando así el entrenamiento.

CSV: Esta librería permite hacer uso de un archivo de texto plano para representar tablas de datos en los que cada línea representa una fila y los valores de la fila están separados por comas. En el presente proyecto se usa para representar los datos a través de tablas de la descripción de puntos de las imágenes capturadas para el entrenamiento en un archivo de extensión CSV como se muestra en la Figura 3-6, además, también para hacer una tabla de las etiquetas a las que responde cada imagen entrenada como se observa en la Ilustración 6-3.

6914	3, 0.0, 0.0, -0.2196078431372549, -0.03137254901960784, -0.4196078431372549, -0.12941176470588237, -0.5725490196078431,
6915	3, 0.0, 0.0, -0.2076923076923077, -0.019230769230769232, -0.4, -0.11923076923076924, -0.5461538461538461, -0.22692307692
6916	3, 0.0, 0.0, -0.2099236641221374, -0.026717557251908396, -0.40076335877862596, -0.1297709923664122, -0.5458015267175572
6917	3, 0.0, 0.0, -0.19133574007220217, -0.04332129963898917, -0.36101083032490977, -0.16606498194945848, -0.480144404332129
6918	3, 0.0, 0.0, -0.17525773195876287, -0.058419243986254296, -0.32989690721649484, -0.20618556701030927, -0.43298969072164
6919	3, 0.0, 0.0, -0.17562724014336917, -0.08243727598566308, -0.32616487455197135, -0.25448028673835127, -0.408602150537634
6920	4, 0.0, 0.0, -0.34306569343065696, -0.27007299270072993, -0.6788321167883211, -0.4233576642335766, -0.9708029197880292,
6921	4, 0.0, 0.0, -0.3546099290780142, -0.2553191489361702, -0.6879432624113475, -0.41134751773049644, -0.9787234042553191, -
6922	4, 0.0, 0.0, -0.35507246376811596, -0.26811594202898553, -0.7028985507246377, -0.43478260869565216, -1.0, -0.47826086956
6923	4, 0.0, 0.0, -0.37857142857142856, -0.2571428571428571, -0.7142857142857143, -0.42857142857142855, -1.0, -0.5, -1.0, -0.42
6924	4, 0.0, 0.0, -0.36879432624113473, -0.28368794326241137, -0.7092198581560284, -0.46099290780141844, -1.0, -0.47517730496
6925	4, 0.0, 0.0, -0.35251798561151076, -0.2517985611510791, -0.7050359712230215, -0.4244604316546763, -1.0, -0.4964028776978
6926	4, 0.0, 0.0, -0.36231884057971014, -0.2753623188405797, -0.7101449275362319, -0.4492753623188406, -1.0, -0.4927536231884
6927	4, 0.0, 0.0, -0.3560606060606061, -0.26515151515151514, -0.696969696969697, -0.4318181818181818, -0.9848484848484849, -0
6928	4, 0.0, 0.0, -0.3560606060606061, -0.25, -0.7045454545454546, -0.4393939393939394, -1.0, -0.5151515151515151, -1.0, -0.416
6929	4, 0.0, 0.0, -0.3695652173913043, -0.2753623188405797, -0.7101449275362319, -0.47101449275362317, -1.0, -0.5289855072463
6930	4, 0.0, 0.0, -0.3695652173913043, -0.2753623188405797, -0.7101449275362319, -0.463768115942029, -1.0, -0.514492753623188
6931	4, 0.0, 0.0, -0.36363636363636365, -0.26573426573426573, -0.7132867132867133, -0.42657342657342656, -1.0, -0.46853146853

Ilustración 6-3: Tabla correspondiente a los puntos de las imágenes de entrenamiento

Elaborado por: Saavedra Delgado, Kevin, 2023.

Además de las librerías ya conocidas a detalle, cada librería hace uso de paquete adicionales y dependencias que garantizan el correcto funcionamiento del prototipo como se observa en la Ilustración 7-3.

Package	Version	Latest version
numpy	1.22.4	▲ 1.23.5
oauthlib	3.2.1	3.2.1
opencv-contrib-python	4.6.0.66	
opencv-python	4.5.5.64	
openssl	1.1.1s	1.1.1s
opt_einsum	3.3.0	3.3.0
packaging	21.3	▲ 22.0
paho-mqtt	1.6.1	
pandas	1.4.4	▲ 1.5.2
pandocfilters	1.5.0	1.5.0
parso	0.8.3	0.8.3
patsy	0.5.3	0.5.3
pickleshare	0.7.5	0.7.5
pillow	9.3.0	9.3.0
pip	22.2.2	▲ 22.3.1
prometheus_client	0.14.1	0.14.1
prompt-toolkit	3.0.20	▲ 3.0.36
protobuf	3.9.2	▲ 3.20.3
psutil	5.9.0	5.9.0
pure_eval	0.2.2	0.2.2
pyasn1	0.4.8	0.4.8
pyasn1-modules	0.2.7	▲ 0.2.8
pycparser	2.21	2.21

Ilustración 7-3: Dependencias adicionales para el funcionamiento del prototipo

Elaborado por: Saavedra Delgado, Kevin, 2023.

3.5.4 Modelo Personalizado

El modelo se realizó para capturar las imágenes de entrenamiento y detección en tiempo real, dentro del mismo script las etiquetas seleccionadas para este proyecto fueron cuatro las cuales activan o desactivan los dispositivos a partir de reconocimiento de gestos. Se tiene la letra **A** que activa una alarma sonora, se seleccionó esta señal debido a que es la letra inicial del dispositivo a activar y para desactivar el dispositivo se seleccionó el símbolo que significa enfriar del lenguaje de señas. Por otra parte, Se tiene la letra **F** para la activación del foco, mientras que para desactivación es la palma abierta simulando la señal stop. Finalmente, se tiene que para la activación de la cerradura eléctrica se tiene la letra C del lenguaje de señas y para la desactivación el puño de frente.

Después de seleccionar las señas con las que se activarán o desactivarán los dispositivos se genera un documento con extensión CSV donde se almacenan las etiquetas con las que cada señal se guardará para posteriormente mostrarse en pantalla. Dicho documento está almacenado dentro de:

hand-gesture-recognition-mediapipe/model/keypoint_classifier/keypoint_classifier_label.csv y se utiliza igual en la ejecución del script principal, para dar nombre a las señas aprendidas.

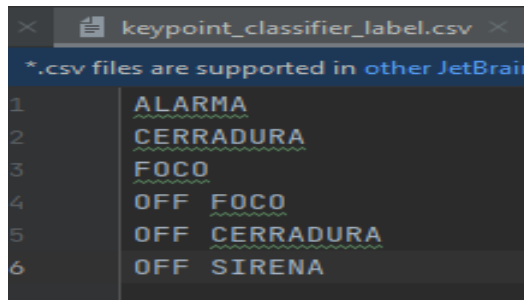


Ilustración 8-3: Etiquetas asignadas a las señas entrenadas

Elaborado por: Saavedra Delgado, Kevin, 2023.

En el script principal se realiza el entrenamiento mediante una función que permite entrar y salir del modo de captura de puntos clave, utilizando una variable denominada "mode". Al presionar la tecla "K", que se almacena en la variable "key", se activa el modo de captura de puntos clave durante la ejecución del programa. Además, gracias a la variable "number", que contiene el número máximo de imágenes, es posible capturar las imágenes necesarias para el entrenamiento en el programa.

```
def select_mode(key, mode):  
    number = -1  
    if 48 <= key <= 57: # 0 ~ 9  
        number = key - 48  
    #if key == 110: # n  
        #mode = 0  
    if key == 107: # k  
        mode = 1  
    return number, mode
```

Ilustración 9-3: Función para seleccionar el modo

Elaborado por: Saavedra Delgado, Kevin, 2023.

En la Ilustración 9-3 se puede observar el cálculo que se realiza para almacenar en la variable "number" el número máximo de imágenes correspondiente, ya que estas teclas tienen un valor en ASCII. Esta variable está delimitada para que contenga valores de 0 a 9, siendo 10 el número máximo de etiquetas que se pueden almacenar en el proyecto. Sin embargo, es posible modificar esta línea para introducir más imágenes en el programa si se desea. Para capturar las imágenes, se utilizan las mismas teclas que contienen el número máximo de etiquetas, siendo la tecla "0" la utilizada para tomar capturas de la primera seña, que en este caso es la "A" correspondiente a la alarma sonora, y la tecla "5" para capturar la seña que desactiva dicha alarma.

Una vez capturada las imágenes durante la ejecución del programa, estás a través de la función *draw_landmark* que dibuja el tracking de las manos con una línea como se muestra en la Ilustración 10-3 y los puntos claves con un círculo pequeño como se observa en la Ilustración 11-3. Todo esto se escribe dentro del archivo *keypoint.csv* como se muestra en penúltima línea de la Ilustración 12-3.

```
# Key Points
for index, landmark in enumerate(landmark_point):
    if index == 0: # 手首1
        cv.circle(image, (landmark[0], landmark[1]), 5, (255, 255, 255),
                    -1)
        cv.circle(image, (landmark[0], landmark[1]), 5, (0, 0, 0), 1)
    if index == 1: # 手首2
        cv.circle(image, (landmark[0], landmark[1]), 5, (255, 255, 255),
                    -1)
        cv.circle(image, (landmark[0], landmark[1]), 5, (0, 0, 0), 1)
```

Ilustración 10-3: Tracking de la mano, dedo pulgar e índice.

Elaborado por: Saavedra Delgado, Kevin, 2023.

```
def draw_landmarks(image, landmark_point):
    if len(landmark_point) > 0:
        # Thumb
        cv.line(image, tuple(landmark_point[2]), tuple(landmark_point[3]),
                (0, 0, 0), 6)
        cv.line(image, tuple(landmark_point[2]), tuple(landmark_point[3]),
                (255, 255, 255), 2)
        cv.line(image, tuple(landmark_point[3]), tuple(landmark_point[4]),
                (0, 0, 0), 6)
        cv.line(image, tuple(landmark_point[3]), tuple(landmark_point[4]),
                (255, 255, 255), 2)
```

Ilustración 11-3: Tracking de la mano, puntos clave.

Elaborado por: Saavedra Delgado, Kevin, 2023.

```
def logging_csv(number, mode, landmark_list, point_history_list):
    if mode == 0:
        pass
    if mode == 1 and (0 <= number <= 9):
        csv_path = 'model/keypoint_classifier/keypoint.csv'
        with open(csv_path, 'a', newline='') as f:
            writer = csv.writer(f)
            writer.writerow([number, *landmark_list])
    return
```

Ilustración 12-3: Función para capturar las señas y escribir los puntos en *keypoint.csv*

Elaborado por: Saavedra Delgado, Kevin, 2023

3.5.5 Entrenamiento del modelo

El entrenamiento del modelo se realiza a través de una red neuronal secuencial, como datos de entrada se tiene la tabla de datos previamente almacenada donde se guardan todos los puntos clave de las capturas de las imágenes utilizadas para el entrenamiento, la cantidad de clases que utiliza este modelo es 6 como se muestra en la Ilustración 13-3. Este modelo de entrenamiento hace uso de un `RANDOM_SEED`, que es una constante numérica que se utiliza para la inicialización del generador de números aleatorios en un modelo de entrenamiento de una red neuronal, que permite controlar la aleatoriedad de las operaciones en el modelo dando como consecuencia resultados repetibles y comparables.

```
RANDOM_SEED = 42

DATOS DE ENTRADA

In [2]: dataset = 'model/keypoint_classifier/keypoint.csv'
        model_save_path = 'model/keypoint_classifier/keypoint_classifier.hdf5'

NUMERO DE CLASES

In [3]: NUM_CLASSES = 6
```

Ilustración 13-3: Datos de entrada para realizar el entrenamiento

Elaborado por: Saavedra Delgado, Kevin, 2023.

En la Ilustración 14-3 se observa la construcción del modelo de red neuronal de TensorFlow de 5 capas, se utiliza una secuencia de capas consecutivas que procesan los datos de entrada. La primera capa, `tf.keras.layers.Input`, acepta una entrada de forma $(21 * 2)$, que hace referencia a la forma o tamaño de un tensor o vector, que en este caso tiene 42 elementos. El hecho de que se multiplique 21 por 2 significa que cada punto de referencia de la mano (presumiblemente 21 puntos) se está representando con dos valores, las coordenadas x e y del punto en la imagen. Posteriormente hay una capa de Dropout con una tasa de 0.2 y 42 neuronas que se usa para prevenir el sobreajuste. Después hay una capa densa con 20 neuronas y una función de activación `relu`. Luego hay otra capa de Dropout con una tasa de 0.4 con 20 neuronas. A continuación, hay otra capa densa con 10 neuronas y una función de activación `relu`. Finalmente, la última capa densa tiene 6 neuronas y una función de activación `softmax`. Esta capa es usada para realizar la clasificación.

```

model = tf.keras.models.Sequential([
    tf.keras.layers.Input((21 * 2, )),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(20, activation='relu'),
    tf.keras.layers.Dropout(0.4),
    tf.keras.layers.Dense(10, activation='relu'),
    tf.keras.layers.Dense(NUM_CLASSES, activation='softmax')
])

```

Ilustración 14-3: Parámetros de la red neuronal para el entrenamiento del modelo

Elaborado por: Saavedra Delgado, Kevin, 2023.

Posteriormente, a través de Keras se configuró el proceso de entrenamiento del modelo de aprendizaje profundo. Se define la función de pérdida, el optimizador y las métricas que se utilizarán para evaluar el rendimiento del modelo durante entrenamiento.

El optimizador Adam se utilizó para actualizar los pesos del modelo durante el entrenamiento. La función de pérdida fue empleada para calcular la diferencia entre las predicciones del modelo y las etiquetas verdaderas, comúnmente utilizada para problemas de clasificación con múltiples clases. La métrica que se aplicó para evaluar el rendimiento del modelo durante el entrenamiento fue la métrica de precisión, que mide la proporción de predicciones correctas del modelo en relación con las etiquetas verdaderas como se observa en la Ilustración 15-3.

```

]: # Compilación del modelo
model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

```

Ilustración 15-3: Parámetros de la red neuronal para el entrenamiento del modelo

Elaborado por: Saavedra Delgado, Kevin, 2023.

Finalmente, para concluir el entrenamiento como se observa en la Ilustración 16-3 se ajustó algunos parámetros para la ejecución. Utilizando el método *fit* del modelo de red neuronal, que entrena el modelo con los datos (X_{train} y y_{train}) por 1000 épocas con un tamaño de lote de 128 muestras. Además, se proporciona una validación con los datos de prueba (X_{test} y y_{test}). También se pasan dos "callbacks", *cp_callback* y *es_callback*, los cuales son objetos que permiten realizar tareas adicionales durante el entrenamiento, como salvar pesos o detener el entrenamiento temprano.

```

model.fit(
    X_train,
    y_train,
    epochs=1000,
    batch_size=128,
    validation_data=(X_test, y_test),
    callbacks=[cp_callback, es_callback]
)

```

Ilustración 16-3: Método fit para el entrenamiento

Elaborado por: Saavedra Delgado, Kevin, 2023.

3.5.6 *Proceso de detección en tiempo real.*

Una vez instalados todos los requisitos necesarios, es posible iniciar el proceso de reconocimiento de señas en tiempo real, para lo cual primero se inicializa la cámara, para posteriormente ejecutar un bucle en donde se llevan a cabo la delimitación del cuadro, el cálculo de FPS y la escritura de etiquetas en pantalla. Dentro de todo esto está inmerso el hand-tracking y el reconocimiento de señas.

3.5.6.1 *Preparación de la cámara*

En la Ilustración 17-3 se establecen y ajustan las dimensiones de la imagen que se capturará con el objeto "cap". Esto es útil si se desea capturar imágenes con un tamaño específico para un procesamiento posterior.

```

# PREPARACIÓN DE CÁMARA
cap = cv.VideoCapture(cap_device)
cap.set(cv.CAP_PROP_FRAME_WIDTH, cap_width)
cap.set(cv.CAP_PROP_FRAME_HEIGHT, cap_height)

```

Ilustración 17-3: Preparación para la cámara web.

Elaborado por: Saavedra Delgado, Kevin, 2023

A continuación, como se visualiza en la Ilustración 18-3 se prosigue con la configuración de algunos parámetros para la apropiada percepción a través de la cámara. En primer lugar, se lee una imagen de la cámara "cap", que previamente se creó usando la clase "cv. VideoCapture". La variable "ret" indica si la lectura fue exitosa y "image" contiene la imagen capturada. Si la lectura no es exitosa, se detiene el procesamiento. Luego, se invierte la imagen horizontalmente para reflejarla en la pantalla. Después se hace una copia profunda de la imagen capturada en la variable. Más adelante la bandera de la imagen se establece en "False" para evitar que sea modificada. A

continuación, se procesa la imagen con el objeto "hands" para detectar manos en ella. El resultado se almacena en la variable *results* y finalmente se establece la bandera de la imagen en "True" para permitir que sea modificada en el futuro.

```
# Camera capture #####
ret, image = cap.read()
if not ret:
    break
image = cv.flip(image, 1) # Mirror display
debug_image = copy.deepcopy(image)

# Detection implementation #####
image = cv.cvtColor(image, cv.COLOR_BGR2RGB)
image.flags.writeable = False
results = hands.process(image)
image.flags.writeable = True
```

Ilustración 18-3: Preparación para la cámara web.

Elaborado por: Saavedra Delgado, Kevin, 2023

3.5.6.2 *Proceso de detección de manos en tiempo real*

Para la detección en tiempo real como se observa en la Ilustración 19-3, se hace uso de la herramienta MediaPipe a través de la importación de la solución "Hands", luego se crea un objeto usando la clase "mp_hands.Hands" y pasándole argumentos que verifican si la imagen es estática, el número máximo de manos que detecta, el mínimo de confianza para la detección de manos y el nivel mínimo de confianza para el seguimiento de manos.

```
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(
    static_image_mode=use_static_image_mode,
    max_num_hands=1,
    min_detection_confidence=min_detection_confidence,
    min_tracking_confidence=min_tracking_confidence,
```

Ilustración 19-3: Parámetros para la detección.

Elaborado por: Saavedra Delgado, Kevin, 2023

La detección se realiza mediante un bucle que solo finaliza cuando se presiona la tecla **ESC**. Si se ha detectado manos, el ciclo **for** recorre dos listas al mismo tiempo, una contiene las coordenadas de los landmarks de cada mano detectada y la otra información sobre la mano izquierda o derecha.

El bucle realiza todas estas acciones para cada mano detectada. Empieza calculando el cuadro de delimitación que rodea la mano detectada. A continuación, se hace una lista de landmarks de la mano detectada. Después se normaliza las coordenadas de los landmarks para que sean relativas al tamaño de la imagen y se puedan usar para entrenar el modelo de aprendizaje automático.

```
if results.multi_hand_landmarks is not None:
    for hand_landmarks, handedness in zip(results.multi_hand_landmarks,
                                          results.multi_handedness):
        # Calculando cuadro de delimitación
        brect = calc_bounding_rect(debug_image, hand_landmarks)
        # Landmark calculation
        landmark_list = calc_landmark_list(debug_image, hand_landmarks)
        # Conversion to relative coordinates / normalized coordinates
        pre_processed_landmark_list = pre_process_landmark(
            landmark_list)
        pre_processed_point_history_list = pre_process_point_history(
            debug_image, point_history)
        # Guardar en el archivo que contiene los datos
        logging_csv(number, mode, pre_processed_landmark_list,
                   pre_processed_point_history_list)
```

Ilustración 20-3: Parámetros de detección en tiempo real

Elaborado por: Saavedra Delgado, Kevin, 2023

Finalmente se llama a la función `logging_csv` para guardar los datos en un archivo CSV que se observa en la Ilustración 20-3. Esta función toma como argumentos el número de la imagen, el modo de captura y la lista de landmarks preprocesados. Esta función almacena los datos en un archivo CSV para su uso posterior en la creación de un modelo de aprendizaje automático como se muestra en la Ilustración 12-3.

3.5.6.3 Envío de datos al servidor MQTT

El primer paso, para el envío de datos mediante MQTT es crear una instancia del cliente MQTT para el envío de datos a través de la red. Luego se llama a la función “`keypoint_classifier`”, que es una función personalizada que utiliza los datos de los puntos de referencia de la mano y determina qué señal de mano se está haciendo. El resultado de esta clasificación se almacena en la variable “`hand_sign_id`.” Tal y como se observa en la segunda línea de la Ilustración 21-3

```
# Clasificación de los signos de mano
client = mqtt.Client()
hand_sign_id = keypoint_classifier(pre_processed_landmark_list)
print(hand_sign_id)
```

Ilustración 21-3: Instancia MQTT variable que contiene la señal identificada.

Elaborado por: Saavedra Delgado, Kevin, 2023

Una vez identificada la variable que contiene el resultado de la clasificación se procede a condicionar esta variable y de esta forma activar o desactivar los dispositivos eléctricos, tal y como se observa en la Ilustración 22-3. Cada dispositivo se activa con carácter diferente y todos se apagan con el carácter F.

```
if hand_sign_id == 0:
    client.connect("test.mosquitto.org", 1883, 60)
    client.publish("prueba/tres", "A")
if hand_sign_id == 1:
    client.connect("test.mosquitto.org", 1883, 60)
    client.publish("prueba/dos", "C")
if hand_sign_id == 2:
    client.connect("test.mosquitto.org", 1883, 60)
    client.publish("prueba/uno", "B")
if hand_sign_id == 3:
    client.connect("test.mosquitto.org", 1883, 60)
    client.publish("prueba/uno", "F")
if hand_sign_id == 4:
    client.connect("test.mosquitto.org", 1883, 60)
    client.publish("prueba/dos", "F")
if hand_sign_id == 5:
    client.connect("test.mosquitto.org", 1883, 60)
    client.publish("prueba/tres", "F")
```

Ilustración 22-3: Condiciones para publicar en los diferentes tópicos.

Elaborado por: Saavedra Delgado, Kevin, 2023

3.5.6.4 Recepción de datos desde el servidor MQTT

En el dispositivo final se encuentra el módulo ESP8266 que se conecta a una red Wi-Fi para recibir los datos desde el servidor MQTT que para este caso es “test.mosquitto.org. El módulo se conecta a un celular en cuya red define como usuario usuario: “Micel” y contraseña “1234567890g.” El dispositivo final está conectado al Pin D7 que en la programación se define como 13, tal como se visualiza en Ilustración 23-3. El pin D7 se conecta a un módulo relé que hace uso de un transistor 2N2222 y trabaja como amplificador, para que finalmente se active el relé y accione el dispositivo eléctrico. El módulo ESP8266 trabaja con velocidad de 115200 baud/s y hace uso del puerto 1883 y usa funciones clave para la recepción de datos. Estas funciones se detallan a continuación.

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.

const char* ssid = "Micel";
const char* password = "1234567890g";
const char* mqtt_server = "test.mosquitto.org";
int cerradura = 13;
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];
int value = 0;

```

Ilustración 23-3: Parámetros para el funcionamiento del ESP8266

Elaborado por: Saavedra Delgado, Kevin, 2023.

Reconnect: es una función que intenta conectarse a un servidor MQTT (Message Queuing Telemetry Transport) a través de un cliente ESP8266. La función intenta conectarse al servidor y si la conexión falla, espera 5 segundos antes de intentar nuevamente. Si la conexión es exitosa, la función publica un mensaje en el tema "outTopic" y se suscribe al tema "prueba/dos". El cliente ID es generado aleatoriamente para cada intento de conexión como se observa en la Ilustración 24-3.

```

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.publish("outTopic", "hello world");
      // ... and resubscribe
      client.subscribe("prueba/dos");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

```

Ilustración 24-3: Parámetros para el funcionamiento del ESP8266

Elaborado por: Saavedra Delgado, Kevin, 2023.

Callback: Es una función que es llamada cuando se recibe un mensaje MQTT en el cliente ESP8266. La función imprime el tema y el mensaje recibido en la consola serial. Además, la función controla el estado de un pin de salida (llamado "cerradura" en el código) en función del primer carácter del mensaje recibido. Si el primer carácter es una "C", se establece el pin en estado alto y si no lo es, se establece en estado bajo. En otras palabras, la función enciende o apaga un dispositivo (una cerradura) según el mensaje recibido a través de MQTT, tal y como se aprecia en la Ilustración 25-3.

```
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();

  // Switch on the LED if an 1 was received as first character
  if ((char)payload[0] == 'C') {
    digitalWrite(cerradura, HIGH); // Turn the LED on (Note that LOW is the voltage level
    // but actually the LED is on; this is because
    // it is active low on the ESP-01)
  } else {
    digitalWrite(cerradura, LOW); // Turn the LED off by making the voltage HIGH
  }
}
```

Ilustración 25-3: Función para la recepción de datos del dispositivo cerradura.

Elaborado por: Saavedra Delgado, Kevin, 2023.

3.5.6.5 *Diseño y simulación de circuito amplificador*

Por último, se tiene el diseño y simulación del circuito amplificador de voltaje. Todo parte de una fuente de alimentación de corriente alterna de 110V, esta se conecta a un regulador de voltaje de 5v y al foco para que este pueda encenderse. El regulador de voltaje de 5V se conecta al ESP8266 y al módulo relé que contiene un transistor 2N2222. A la salida del ESP8266 se tiene un voltaje de 3.3V que está conectado a la base del transistor 2N2222. Este componente eléctrico tiene conectado en el emisor el positivo del regulador de 5v proporcionando una amplificación a la salida del transistor. El relé se activa al recibir los 5v desde el emisor haciendo que el interruptor cambie a normalmente abierto y de esta forma se active el dispositivo eléctrico que opera con

110V. En la Ilustración 26-3, 27-3 y 28-3 se presenta el esquema de amplificación para cada dispositivo eléctrico.

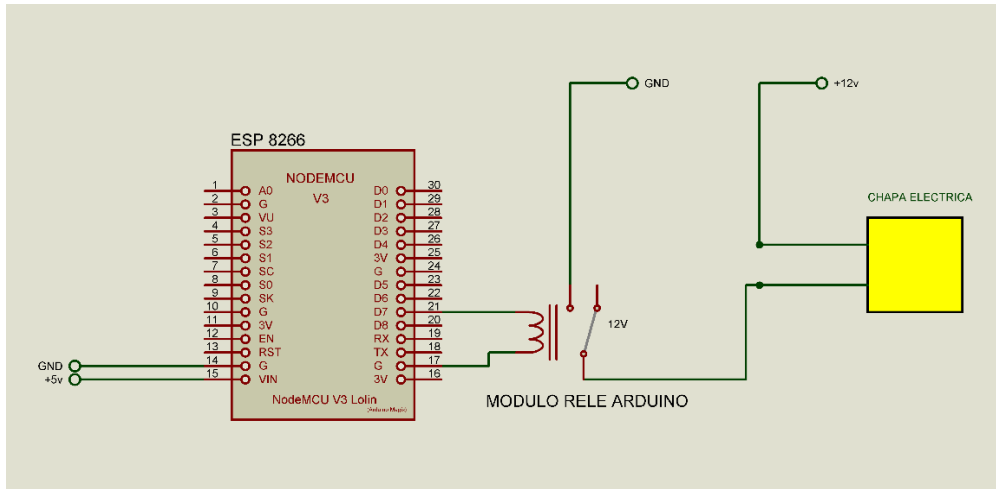


Ilustración 26-3: Circuito de amplificación para la cerradura eléctrica.

Elaborado por: Saavedra Delgado, Kevin, 2023.

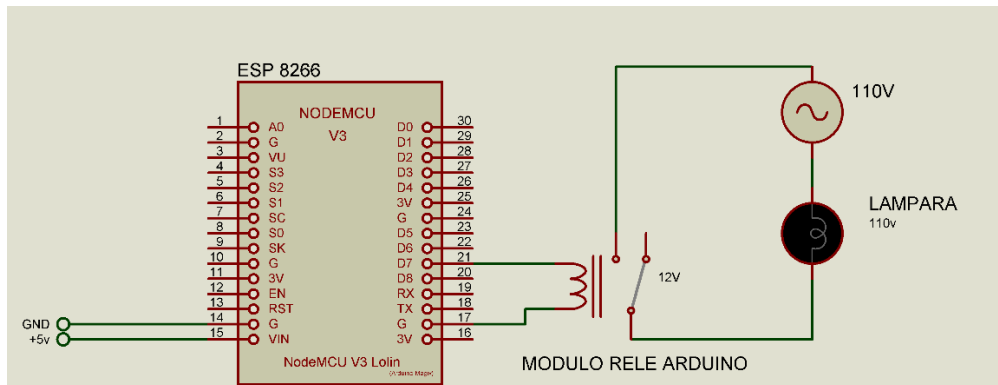


Ilustración 27-3: Circuito de amplificación para la cerradura eléctrica.

Elaborado por: Saavedra Delgado, Kevin, 2023.

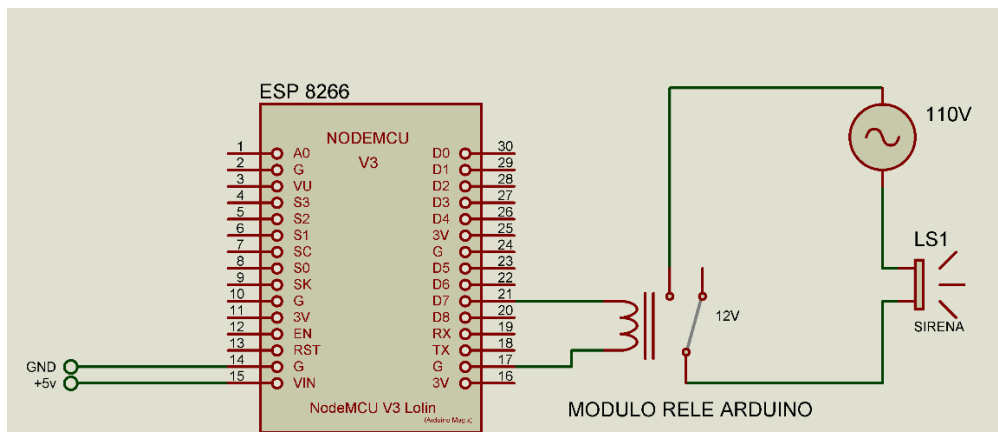


Ilustración 28-3: Circuito de amplificación para la cerradura eléctrica.

Elaborado por: Saavedra Delgado, Kevin, 2023.

CAPITULO IV

4. MARCO DE ANALISIS E INTERPRETACION DE RESULTADOS.

En esta sección se presentará el marco de análisis e interpretación de resultados para la implementación de un sistema de control de dispositivos eléctricos utilizando reconocimiento de lenguaje de señas dactilológico mediante visión artificial. En primer lugar, se describen las metodologías utilizadas para la recolección de datos y la evaluación del desempeño del sistema. Luego, se presentan los resultados obtenidos a partir de la aplicación del sistema en diferentes escenarios y se analizan los resultados en términos de eficacia, eficiencia y usabilidad. Finalmente, se discuten las limitaciones y desafíos encontrados durante la implementación y se propone posibles mejoras y futuras líneas de investigación en el área.

4.1 Resultados

Los resultados del estudio incluyen una evaluación de la eficacia del sistema en términos de su capacidad para reconocer los diferentes gestos realizados por el usuario en el lenguaje dactilológico. Además, se evaluó la eficiencia del sistema midiendo los tiempos de respuesta desde que el usuario realiza un gesto hasta que el dispositivo eléctrico responde a la acción solicitada.

Por otro lado, la funcionalidad del sistema también fue otro parámetro evaluado, midiendo la capacidad del sistema para controlar diferentes dispositivos eléctricos y la compatibilidad con diferentes sistemas operativos y plataformas. Posteriormente la estabilidad y fiabilidad evaluando la capacidad que tiene el sistema para mantener un funcionamiento estable y fiable durante un periodo prolongado de tiempo.

Finalmente, se presenta el análisis detallado de los resultados obtenidos, identificando fortalezas y debilidades del sistema proponiendo posibles mejoras y futuras líneas de investigación.

4.1.1 *Recolección de Datos*

Luego de la construcción del modelo de entrenamiento de 5 capas detallado en el capítulo anterior, se procede a describir a detalle el conjunto de datos utilizados para el entrenamiento. Como se puede apreciar en la Ilustración 1-4 en el entrenamiento se capturó aproximadamente 8572 imágenes. Mientras se captura cada imagen con los dígitos ingresados por teclado, Al instante a través de landmarks de MediaPipe y la librería CSV se escribe en una tabla de los puntos de cada

imagen capturada, para posteriormente ser utilizada como dato de entrada del modelo de entrenamiento. Bajo esta modalidad de recolección de datos se evita almacenar gran cantidad de imágenes, y de esta forma el proyecto se vuelve mucho más liviano.

8553	4	0	0	0.35862069	0.13793103	0.63448276	0.26896552	0.71034483	0.45517241	0.6	0.57241379	0.6137931	0.5806452
8554	4	0	0	0.3862069	0.13793103	0.64137931	0.30344828	0.71034483	0.52413793	0.57241379	0.6137931	0.5806452	0.5806452
8555	4	0	0	0.33548387	0.11612903	0.58709677	0.25806452	0.69032258	0.46451613	0.65806452	0.5806452	0.5806452	0.5806452
8556	4	0	0	0.36538462	0.17307692	0.59615385	0.31410256	0.68589744	0.48076923	0.56410256	0.54487179	0.54487179	0.54487179
8557	4	0	0	0.39751553	0.13664596	0.64596273	0.27329193	0.72049689	0.45962733	0.58385093	0.53416149	0.53416149	0.53416149
8558	4	0	0	0.41975309	0.13580247	0.69135802	0.25308642	0.72839506	0.39506173	0.58024691	0.47530864	0.47530864	0.47530864
8559	4	0	0	0.41104294	0.14723926	0.66871166	0.26993865	0.68711656	0.43558282	0.52760736	0.5398773	0.5398773	0.5398773
8560	4	0	0	0.4251497	0.11377246	0.68862275	0.21556886	0.73652695	0.36526946	0.62275449	0.46706587	0.46706587	0.46706587
8561	4	0	0	0.41818182	0.0969697	0.67878788	0.20606061	0.73939394	0.38181818	0.61818182	0.5030303	0.5030303	0.5030303
8562	4	0	0	0.42038217	0.13375796	0.67515924	0.26751592	0.68789809	0.44585987	0.50955414	0.54140127	0.54140127	0.54140127
8563	4	0	0	0.43589744	0.13461538	0.68589744	0.25641026	0.71153846	0.41666667	0.57051282	0.48717949	0.48717949	0.48717949
8564	4	0	0	0.41614907	0.13664596	0.63975155	0.28571429	0.67080745	0.49689441	0.50931677	0.62732919	0.62732919	0.62732919
8565	4	0	0	0.38607595	0.15189873	0.60126582	0.3164557	0.58227848	0.53164557	0.36708861	0.62658228	0.62658228	0.62658228
8566	4	0	0	0.39354839	0.12903226	0.61935484	0.27096774	0.61290323	0.46451613	0.43225806	0.59354839	0.59354839	0.59354839
8567	4	0	0	0.36912752	0.14765101	0.61744966	0.29530201	0.67114094	0.48322148	0.5033557	0.56375839	0.56375839	0.56375839
8568	4	0	0	0.40718563	0.1257485	0.67664671	0.22155689	0.75449102	0.35928144	0.66467066	0.46107784	0.46107784	0.46107784
8569	4	0	0	0.45138889	0.09722222	0.78472222	0.15277778	0.84027778	0.26388889	0.67361111	0.35416667	0.35416667	0.35416667
8570	4	0	0	0.45255474	0.11678832	0.70072993	0.23357664	0.69343066	0.40875912	0.47445255	0.49635036	0.49635036	0.49635036
8571	4	0	0	0.40243902	0.06707317	0.73170732	0.04268293	0.90853659	-0.02439024	1	-0.14634146	-0.14634146	-0.14634146
8572	4	0	0	0.46043165	0.0647482	0.76258993	0.10791367	0.89208633	0.1942446	0.87769784	0.24460432	0.24460432	0.24460432

Ilustración 1-4: Tabla del conjunto de datos para el entrenamiento.

Elaborado por: Saavedra Delgado, Kevin, 2023.

Esta recolección de datos en forma de puntos está distribuida en 6 clases distintas, dichas clases se identifican a través de un número que corresponde a cada gesto para activación y desactivación de los dispositivos eléctricos. La clase 0 corresponde a la activación de una sirena y para esta se capturaron 1371 datos como se observa en la Ilustración 2-4. Las imágenes se capturaron con tres personas distintas, con distintos tonos de piel, tamaño y forma de la mano.

	A	B	C	D	E	F	G	H	I
1		0	0	-0.31343284	-0.06467662	-0.60199005	-0.26368159	-0.8159204	-0.45273632
2	0	0	0	-0.31188119	-0.06435644	-0.5990099	-0.26237624	-0.81683168	-0.45544554
3	0	0	0	-0.31	-0.06	-0.6	-0.26	-0.82	-0.455
4	0	0	0	-0.31067961	-0.0631068	-0.59708738	-0.26213592	-0.81553398	-0.45631068
5	0	0	0	-0.29901961	-0.05882353	-0.59313725	-0.25980392	-0.81372549	-0.44607843
6	0	0	0	-0.29756098	-0.05853659	-0.5902439	-0.26341463	-0.81463415	-0.44878049
7	0	0	0	-0.30392157	-0.07352941	-0.59313725	-0.2745098	-0.81372549	-0.45098039
8	0	0	0	-0.30188679	-0.06603774	-0.58962264	-0.25943396	-0.81132075	-0.43867925
9	0	0	0	-0.31219512	-0.06829268	-0.6	-0.26829268	-0.8195122	-0.45365854
10	0	0	0	-0.30882353	-0.07843137	-0.59313725	-0.28431373	-0.81372549	-0.46078431

Ilustración 2-4: Total de datos recolectados para la clase Alarma

Elaborado por: Saavedra Delgado, Kevin, 2023.

De la misma manera se llevó a cabo la recolección de datos para las demás clases. Generando para la clase 1 que corresponde a la activación de la cerradura un total de 1907 imágenes, 1904

para la clase 2 que corresponde a la activación del foco, 1438 para la clase 3 que corresponde a la desactivación del foco, 1067 para la clase 4 que corresponde a la desactivación de la cerradura y finalmente 885 para la clase 5 que corresponde a la desactivación de la sirena. Para definir la cantidad de imágenes para cada clase se utilizó el método prueba y error y visualizando la matriz de confusión.

4.1.2 Evaluación del Prototipo

4.1.2.1 Entrenamiento

El entrenamiento se ejecuta sobre CPU debido a que es poca la cantidad de clases para la que está diseñado el presente proyecto. El modelo de entrenamiento permite que éste se detenga de manera temprana a través de un objeto *EarlyStopping*, que es una callback que detiene el entrenamiento si no se observa una mejora en la métrica seleccionada (como la precisión o la pérdida) durante un cierto número de épocas, tal y como se observa en la Ilustración 4-4.

El entrenamiento tiene una duración de aproximadamente 27 segundos, mientras la ejecución del script que contiene el entrenamiento y lo demás parámetros de evaluación en conjunto duran alrededor de 36 segundos. Este modelo de entrenamiento presenta un rendimiento bastante aceptable. En la Ilustración 3-4 se visualiza a detalle el entrenamiento.

```
Epoch 1/1000
29/51 [=====>.....] - ETA: 0s - loss: 1.7969 - accuracy: 0.1
910
Epoch 1: saving model to model/keypoint_classifier\keypoint_classifier.hdf5
51/51 [=====] - 1s 7ms/step - loss: 1.7432 - accuracy:
0.2305 - val_loss: 1.5776 - val_accuracy: 0.3626
Epoch 2/1000
51/51 [=====] - ETA: 0s - loss: 1.5431 - accuracy: 0.3
529
Epoch 2: saving model to model/keypoint_classifier\keypoint_classifier.hdf5
51/51 [=====] - 0s 3ms/step - loss: 1.5431 - accuracy:
0.3529 - val_loss: 1.3524 - val_accuracy: 0.4862
Epoch 3/1000
50/51 [=====>.] - ETA: 0s - loss: 1.3921 - accuracy: 0.4
323
Epoch 3: saving model to model/keypoint_classifier\keypoint_classifier.hdf5
51/51 [=====] - 0s 4ms/step - loss: 1.3923 - accuracy:
0.4318 - val_loss: 1.1880 - val_accuracy: 0.5786
Epoch 4/1000
25/51 [=====>.....] - ETA: 0s - loss: 1.2931 - accuracy: 0.4
```

Ilustración 3-4: Inicio de entrenamiento

Elaborado por: Saavedra Delgado, Kevin, 2023.

```

682
Epoch 124: saving model to model/keypoint_classifier\keypoint_classifier.hdf5
51/51 [=====] - 0s 3ms/step - loss: 0.6235 - accuracy:
0.7682 - val_loss: 0.4095 - val_accuracy: 0.8651
Epoch 125/1000
29/51 [=====>.....] - ETA: 0s - loss: 0.6009 - accuracy: 0.7
777
Epoch 125: saving model to model/keypoint_classifier\keypoint_classifier.hdf5
51/51 [=====] - 0s 3ms/step - loss: 0.5993 - accuracy:
0.7760 - val_loss: 0.3992 - val_accuracy: 0.8647
Epoch 126/1000
30/51 [=====>.....] - ETA: 0s - loss: 0.5999 - accuracy: 0.7
880
Epoch 126: saving model to model/keypoint_classifier\keypoint_classifier.hdf5
51/51 [=====] - 0s 3ms/step - loss: 0.6181 - accuracy:
0.7738 - val_loss: 0.4134 - val_accuracy: 0.8577
Epoch 126: early stopping

```

Out[11]: <keras.callbacks.History at 0x2904a042440>

Ilustración 4-4: Fin de entrenamiento (pausado por EarlyStopping).

Elaborado por: Saavedra Delgado, Kevin, 2023.

4.1.2.1.1 Evaluación del modelo

Para las pruebas de razonamiento se realiza primero la evaluación del modelo a través de los valores de pérdidas y precisión, en un conjunto de datos de validación. La función “*model.evaluate(X_test, y_test, batch_size=128)*” Se utiliza para evaluar el desempeño de la red neuronal en el conjunto de datos de validación utilizando un tamaño de lote de 128, como se aprecia en la Ilustración 5-4.

```

# EVALUACION DEL MODELO
val_loss, val_acc = model.evaluate(X_test, y_test, batch_size=128)

```

Ilustración 5-4: Evaluación del modelo.

Elaborado por: Saavedra Delgado, Kevin, 2023.

Durante el entrenamiento se evaluaron 17 lotes de datos, cada evaluación tardó 2ms/step La pérdida promedio del modelo en el conjunto de datos de prueba durante la evaluación es de 0.4134, esto mide qué tan incorrectas son las predicciones del modelo en comparación con los valores reales.

La Ilustración 6-4 muestra la precisión promedio del modelo en el conjunto de datos de prueba durante la evaluación es de 0.8577, esto mide que tan precisas son las predicciones del modelo en comparación con los valores reales. La precisión que nos brinda el modelo es del 85.77%.

```

17/17 [=====] - 0s 2ms/step - loss: 0.4134 - accuracy:
0.8577

```

Ilustración 6-4: Resultados del modelo entrenado

Elaborado por: Saavedra Delgado, Kevin, 2023.

El modelo ya entrenado toma la primera imagen del conjunto de datos de prueba, la introduce en la red neuronal previamente entrenada y devuelve la predicción de la clase correspondiente a esa imagen como se visualiza en la Ilustración 7-4.

```
# PRUEBAS DE RAZONAMIENTO
predict_result = model.predict(np.array([X_test[0]]))
print(np.squeeze(predict_result))
print(np.argmax(np.squeeze(predict_result)))
```

Ilustración 7-4: Predicción del modelo entrenado

Elaborado por: Saavedra Delgado, Kevin, 2023.

En la Ilustración 8-4 se observa la salida que indica que el modelo ha realizado una predicción sobre un conjunto de datos de un solo elemento y que ha predicho que el elemento pertenece a la clase 0 con una probabilidad del 76.4%.

```
1/1 [=====] - 0s 166ms/step
[7.6397479e-01 7.5030100e-04 9.8550841e-03 1.9029286e-02 8.1064336e-02
 1.2532626e-01]
0
```

Ilustración 8-4: Resultado de predicción del modelo entrenado

Elaborado por: Saavedra Delgado, Kevin, 2023.

4.1.2.1.2 Matriz de confusión

La matriz de confusión permite evaluar el rendimiento del modelo en la clasificación de los gestos realizados por los usuarios y determinar en qué medida se están cometiendo errores y qué clases están siendo confundidas con mayor frecuencia. Estos resultados serán fundamentales para identificar posibles áreas a mejorar en el modelo y en el sistema en general, con el objetivo de lograr un mejor desempeño en el reconocimiento de gestos y, por ende, en el control de dispositivos eléctricos.

La matriz de confusión se puede calcular a partir de una fórmula matemática, este cálculo se realiza con una determinada cantidad de detecciones. Posteriormente, se llamará *verdaderos positivos* a la cantidad de resultados reales reconocidos correctamente con la predicción, *falsos positivos*, a los valores reales negativos que fueron predichos como verdaderos, *verdaderos negativos*, a los valores reales negativos que fueron predichos correctamente como negativos, y por último *falsos negativos* a los valores reales positivos que fueron predichos como negativos. La Ecuación 1-4 presenta la formula.

$$\text{Precisión} = \frac{VP}{VP + FP}$$

Ecuación 1-4: Matriz de confusión.

Elaborado por: Saavedra Delgado, Kevin, 2023.

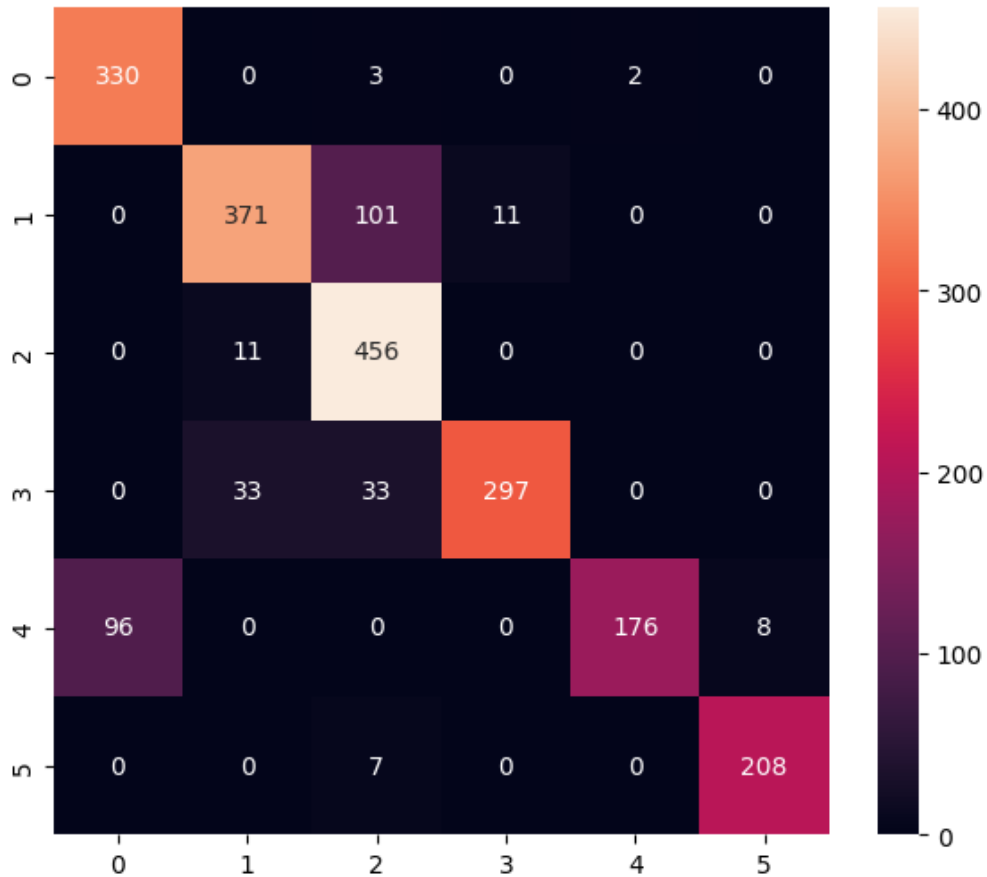


Ilustración 9-4: Matriz de confusión.

Elaborado por: Saavedra Delgado, Kevin, 2023.

Acorde con la matriz de confusión presentada en la Ilustración 9-4, el presente proyecto indica confusión en la clase 0 perteneciente al dispositivo activar alarma con la clase 4 correspondiente a la desactivación de la cerradura. De la misma manera en la clase 1 que corresponde a la activación de la cerradura con la clase activación del foco y la desactivación de foco. Por último, la clase 2 correspondiente a la activación a la activación de un foco existe mayor confusión con la activación de cerradura y desactivación de foco. Todo esto tiene lógica debido a la similitud de los gestos. Sin embargo, la cantidad de imágenes utilizadas en el entrenamiento dan como resultado un gran porcentaje de precisión en cada clase, lo cual hace eficiente el proyecto.

Por otra parte, la matriz de confusión también presente un informe donde resume las métricas de desempeño de clasificación para un conjunto de datos de prueba. Aquí se muestra la precisión, la

recuperación y la puntuación F1 para cada clase de destino en el conjunto de datos de prueba (véase la Ilustración 10-4).

La precisión se refiere a la fracción de instancias clasificadas correctamente para una clase, la recuperación se refiere a la fracción de instancias de la clase que se clasificaron correctamente, y la puntuación F1 es una medida de la combinación de la precisión y la recuperación. La macro media y la media ponderada son la media de las métricas para cada clase. El support indica el número de muestras en el conjunto de datos de prueba que pertenecen a cada clase.

Classification Report					
	precision	recall	f1-score	support	
0	0.77	0.99	0.87	335	
1	0.89	0.77	0.83	483	
2	0.76	0.98	0.85	467	
3	0.96	0.82	0.89	363	
4	0.99	0.63	0.77	280	
5	0.96	0.97	0.97	215	
accuracy			0.86	2143	
macro avg	0.89	0.86	0.86	2143	
weighted avg	0.88	0.86	0.86	2143	

Ilustración 10-4: Reporte de clasificación

Elaborado por: Saavedra Delgado, Kevin, 2023.

4.1.2.2 *Detección en Tiempo Real.*

En esta sección se presenta la detección en tiempo real de la implementación del sistema se describen los detalles de la implementación y se presentan los resultados de la evaluación del proyecto en términos de precisión y velocidad de detección.

La detección en tiempo real nos brinda gran precisión y velocidad de detección en *ms*, mientras el programa corre desde 14 a 20 FPS cuando no se realiza la conexión a MQTT. Cabe recalcar que la detección se está realizando en CPU, en vista que el algoritmo de MediaPipe para la detección en tiempo real no admite su ejecución en GPU. Por otra parte, el entrenamiento también se llevó a cabo mediante CPU.

Para la detección de la clase 0 que se muestra en la Ilustración 11-4. Esta corresponde a la activación de la alarma sonora, siempre que se coloque de forma adecuada la señal brinda el 100%

de precisión, si la mano se gira en forma de puño entonces es confundida. Para 10 detecciones las predicciones fueron acertadas en su totalidad.

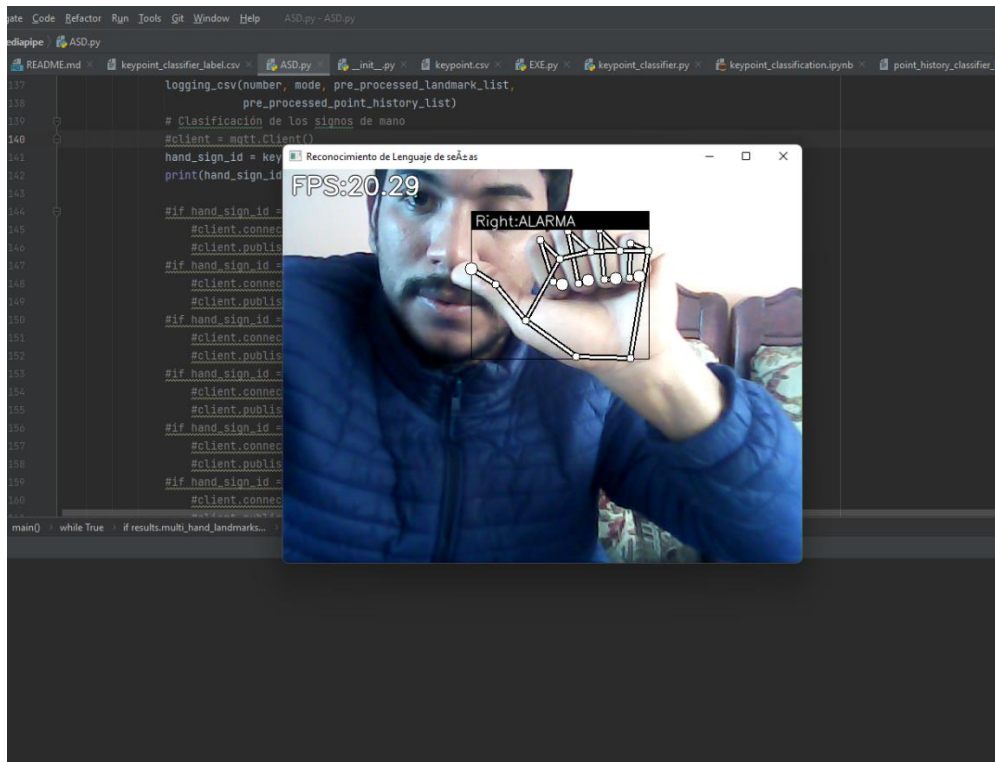


Ilustración 11-4: Detección de la clase 0

Elaborado por: Saavedra Delgado, Kevin, 2023.

Para la detección de la clase 1 correspondiente a la activación de la cerradura, acorde a la matriz de confusión presenta un 89% de precisión como se muestra en la Ilustración 12-4. Sin embargo, la detección es casi perfecta excepto cuando se levanta la muñeca hacia arriba o se inclina un poco hacia la derecha.

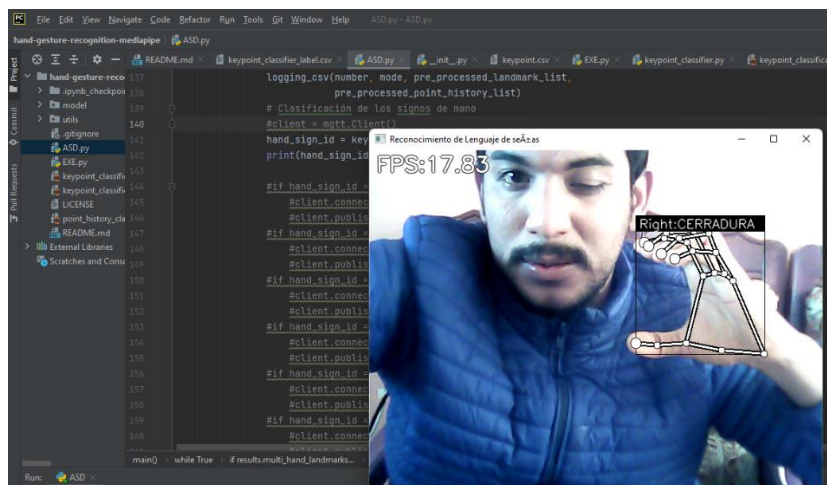
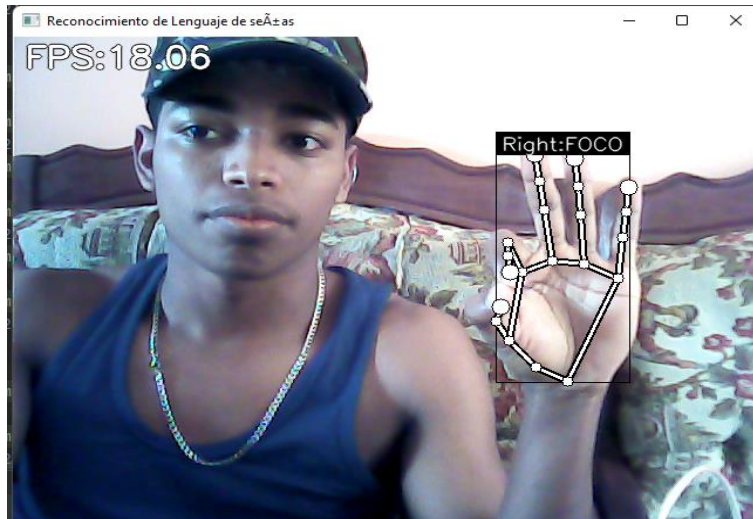


Ilustración 12-4: Detección de la clase 1

Elaborado por: Saavedra Delgado, Kevin, 2023.

Para la detección de la clase 2 que se visualiza en la Ilustración 13-4, correspondiente a la activación del foco, se presenta un 100% de detección a menos que se baje alguno de los 3 dedos de la seña, en ese caso se confunde con la cerradura.

Ilustración 13-4: Detección de la clase 2



Elaborado por: Saavedra Delgado, Kevin, 2023.

En la Figura 14-4 se presenta la detección de la clase 3 correspondiente a la desactivación del foco, acorde a la matriz de confusión presentada en la figura 4-10 nos brinda un 96% de precisión. Sin embargo, la detección es casi perfecta excepto cuando se inclina la mano hacia la izquierda o derecha pero rara vez. En este contexto de 12 detecciones inclinadas solo 1 fue errónea.

Ilustración 14-4: Detección de la clase 3

Elaborado por: Saavedra Delgado, Kevin, 2023.



En la Ilustración 15-4 se visualiza la detección de la clase 4 correspondiente a la desactivación de la cerradura, acorde a la matriz de confusión presenta un 99% de precisión como se visualiza en la Ilustración 10-4. Se presentaron 15 detecciones y todas fueron acertadas para esta seña.



Ilustración 15-4: Detección de la clase 4

Elaborado por: Saavedra Delgado, Kevin, 2023.

Para la última detección, la clase 5 correspondiente a la desactivación de la sirena que se visualiza en la Ilustración 16-4, acorde a la matriz de confusión presenta un 96% de precisión como se observa en la Ilustración 10-4. Se presentaron 10 detecciones y las 10 fueron acertadas para esta seña.



Ilustración 16-4: Detección de la clase 5

Elaborado por: Saavedra Delgado, Kevin, 2023.

4.1.2.2.1 Activación de Dispositivos.

La activación de dispositivos eléctricos es una parte clave en el análisis e interpretación de resultados de este trabajo. En esta sección, se examinaron los resultados obtenidos en las pruebas realizadas para evaluar la eficacia del sistema propuesto. En esta sección, se muestra cómo la combinación de la visión artificial y el control de dispositivos eléctricos puede ofrecer una solución innovadora y eficaz para personas con discapacidades físicas.

El sistema propuesto al conectarse al servidor MQTT tiene una caída considerable de FPS que corre desde 2 a los 5 FPS. A pesar de que el resultado de detección de 2 a 5 FPS es relativamente bajo, se ha demostrado que es suficiente para el control de dispositivos eléctricos en tiempo real, esto solo causa un retardo de 1 segundo en la activación y desactivación de dispositivos eléctricos, por tanto, no afecta al sistema.

Para la activación del foco cuya activación se ejecuta cuando se ejecuta la señal correspondiente a la letra F y se desactiva con la palma de la mano abierta. Como se puede apreciar en las Ilustraciones 17-4 y 18-4 el foco se enciende al censar la F y se apaga al censar la mano abierta.



Ilustración 17-4: Activación de foco

Elaborado por: Saavedra Delgado, Kevin, 2023.



Ilustración 18-4: Desactivación de foco

Elaborado por: Saavedra Delgado, Kevin, 2023.

Para la activación de la cerradura eléctrica se utiliza la letra C y se desactiva con el puño de frente. Como se puede notar en las Ilustraciones 19-4 y 20-4. El relé se enciende al ejecutar la seña en tiempo real.

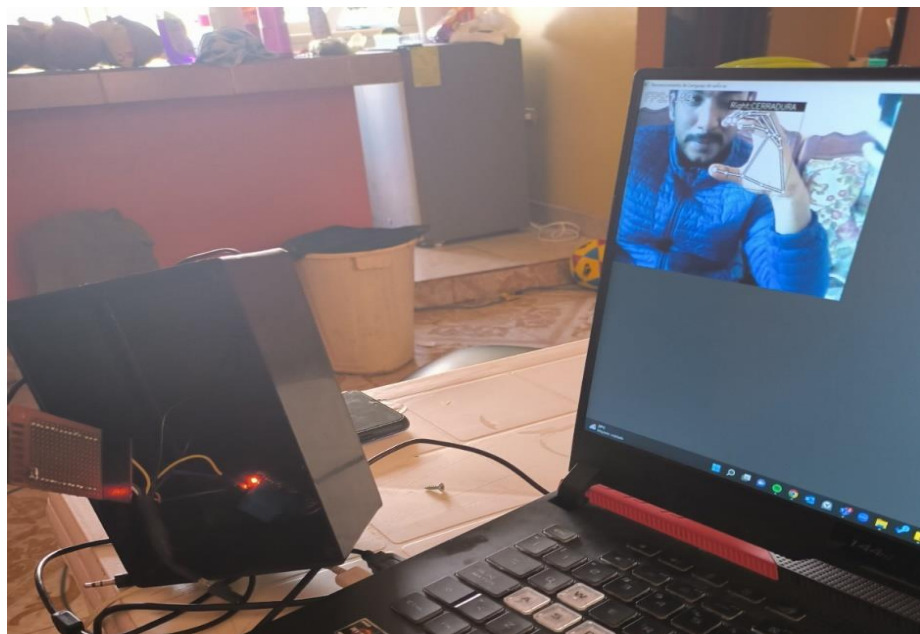


Ilustración 19-4: Activación de cerradura eléctrica

laborado por: Saavedra Delgado, Kevin, 2023.

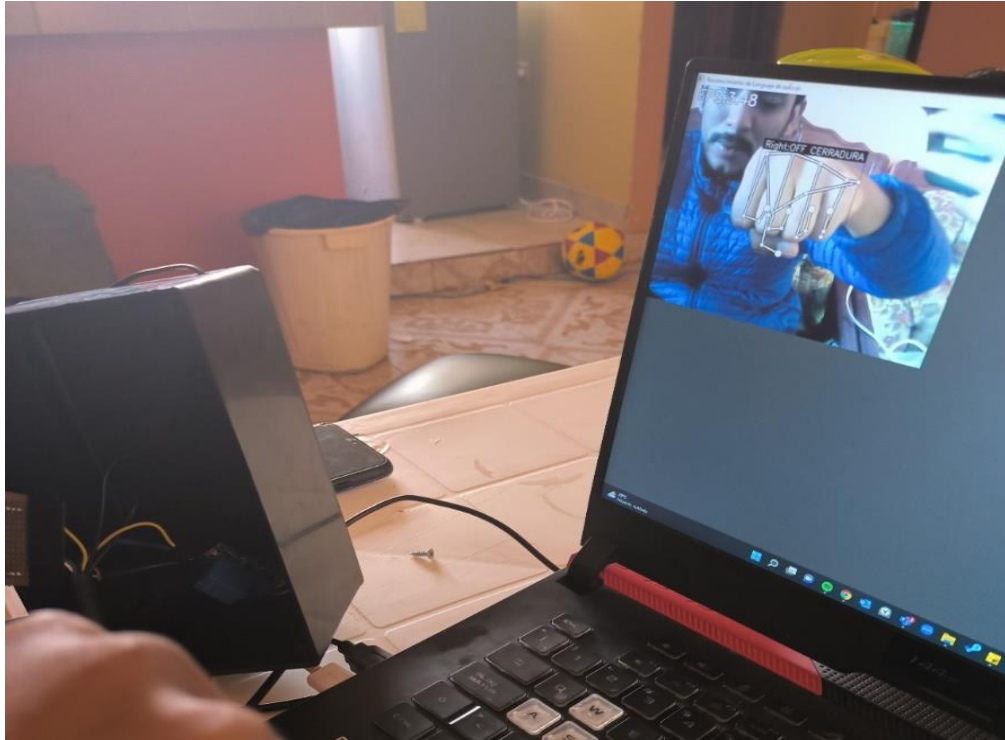


Ilustración 20-4: Desactivación de cerradura eléctrica

Elaborado por: Saavedra Delgado, Kevin, 2023.

Para la activación de la alarma sonora se utiliza la letra A y se desactiva con la seña llámame. Como se puede notar en la Ilustración 21-4 y 22-4. El relé se enciende al ejecutar la seña en tiempo real y se desactiva al censar la otra seña.



Ilustración 21-4: Activación de alarma sonora

Elaborado por: Saavedra Delgado, Kevin, 2023.



Ilustración 22-4: Desactivación de alarma sonora.

Elaborado por: Saavedra Delgado, Kevin, 2023.

CONCLUSIONES

Se logró implementar un sistema de control de dispositivos eléctricos mediante reconocimiento de lenguaje de señas dactilológico y visión artificial, seleccionando la tecnología de reconocimiento de imágenes más adecuada tras un estudio exhaustivo. El sistema permitió activar un foco, una cerradura eléctrica y una alarma sonora mediante diferentes gestos en lenguaje de señas, obteniendo una alta tasa de acierto cercana al 90%. Estos resultados demuestran la viabilidad de utilizar esta tecnología como alternativa para el control de dispositivos eléctricos en situaciones donde el control manual no es posible o para personas con discapacidad auditiva.

Aunque se logró una tasa de acierto cercana al 90% en la detección de gestos en esta tesis, se observó que, en algunas situaciones con baja iluminación o movimientos bruscos, el sistema no lograba detectar correctamente el gesto.

El sistema propuesto alcanzó una precisión media del 88% en la detección de gestos de lenguaje de señas dactilológico. La tasa de precisión más alta se logró fue el gesto "Off Cerradura" con un 99%, mientras que la tasa de precisión más baja se observó para el gesto "Foco" con un 76%. Para los gestos "Cerradura", "Alarma", "Off Sirena", "Off Foco", se obtuvo 89%, 77%, 96% y 96% Respectivamente. De la misma manera los datos presentados en la matriz confusión corroboran un mínimo de confusión en cada gesto, siendo el gesto "Alarma" y foco los que presentan mayor grado de confusión. Sin embargo, en comparación con otros trabajos relacionados en el campo, se observa que esta precisión y confusión es similar o ligeramente superior a los sistemas existentes.

El proceso de entrenamiento del modelo de reconocimiento de gestos se vuelve mucho más sencillo y eficiente, puesto que las funciones y sentencias utilizadas en la programación reducen el tiempo de entrenamiento a 30 segundos para un total de 8572 imágenes. Todo esto es posible gracias a que la biblioteca tiene modelos pre entrenados y una gran cantidad de herramientas de procesamiento de imagen y video.

El modelo de aprendizaje automático utilizado en el proyecto presenta diversas ventajas y mejoras en relación a modelos tradicionales. Una de las principales características es la capacidad de entrenar los datos como puntos almacenados en una hoja de cálculo, lo cual reduce significativamente el tiempo de entrenamiento y permite que el modelo se entrene desde cualquier computadora. Además, el modelo de entrenamiento implementa un objeto EarlyStopping, que detiene el entrenamiento si no se observa una mejora en la métrica seleccionada. Esto permite que el modelo se detenga tempranamente en caso de que el entrenamiento no esté produciendo

resultados efectivos, evitando así el desperdicio de recursos computacionales y sobre entrenamiento.

El resultado de detección del sistema propuesto corre desde 14 a 20 FPS sin la conexión y corre desde 2 a 5 FPS con la conexión al servidor MQTT. Esto ocurre debido a que el servidor MQTT utiliza el protocolo TCP en la capa de transporte, lo que ralentiza el envío y recepción de datos en tiempo real. Aunque los frames con la conexión al servidor MQTT son relativamente bajos, se ha demostrado que es suficiente para el control de dispositivos eléctricos en tiempo real.

El sistema propuesto es una importante contribución al campo de la visión artificial y la accesibilidad para personas con discapacidad. La utilización de visión artificial para detectar y reconocer gestos de lenguaje de señas dactilológico es una tecnología emergente que puede mejorar significativamente la calidad de vida de las personas con discapacidad auditiva. El sistema puede permitir a las personas con discapacidad controlar de manera autónoma dispositivos eléctricos de su hogar, lo que puede aumentar su independencia y su calidad de vida.

.

RECOMENDACIONES

Evaluar la eficacia del sistema con diferentes grupos de usuario.

Mejorar la eficiencia del sistema: se pueden explorar técnicas de optimización para mejorar la eficiencia del sistema en términos de velocidad y consumo de energía.

Validar el sistema en un entorno real: se puede realizar una prueba piloto del sistema en un entorno real para evaluar su efectividad y usabilidad en situaciones cotidianas.

Mantener las manos fuera del sensor, ya que pueda activar o desactivar los dispositivos eléctricos de manera involuntaria.

Es importante seguir ampliando el conjunto de gestos para lograr un mayor control de dispositivos eléctricos y una mayor accesibilidad para las personas con discapacidad auditiva. En futuras investigaciones, se debe considerar la integración del sistema con otros dispositivos inteligentes y trabajar en mejoras en la usabilidad para lograr una experiencia de usuario más amigable e intuitiva.

BIBLIOGRAFÍA

AMAGUAYA RAMOS, Richard Xavier. Análisis comparativo a nivel transaccional de brokers mqtt (mosquitto, mosca y emq) con respecto a la disponibilidad en infraestructuras IoT ante ataques DDoS. [en línea] (Trabajo de titulación). (Ingeniería) Escuela Politécnica Nacional, Quito, Ecuador. 2020. pp. 19 [Consulta: 2023-02-15]. Disponible en: https://bibdigital.epn.edu.ec/bitstream/15000/21374/1/CD_10446.pdf.

AQUINO, Antonio. Reconocimiento e interpretación del alfabeto dactilológico de la lengua de señas mediante tecnología móvil y redes neuronales artificiales [en línea] (Trabajo de titulación). (Licenciatura) Universidad Mayor de San Andrés. 2018. pp. 5 [Consulta: 2023-02-15] Disponible en: <https://repositorio.umsa.bo/bitstream/handle/123456789/17474/T-3406.pdf?sequence=1&isAllowed=y>.

ARISTIZÁBAL, María. *Evaluación asimétrica de una red neuronal artificial: aplicación al caso de la inflación en Colombia* [en línea]. Medellín-Colombia: Publicado por la Universidad de Antioquia, 2006. [Consulta: 15 de febrero 2023] Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-25962006000200003

BARRÓN RUIZ, Mariano. Uso didáctico del software de ayuda al diseño electrónico “Proteus” [en línea]. (working papper) Universidad del País Vasco. 1970. pp. 1 [Consulta: 2023-02-15] Disponible en: <http://e-spacio.uned.es/fez/eserv/taee:congreso-2004-1034/S1F05.pdf>.

BETANCUR, Daniel; VÉLEZ Mateo & PEÑA Alejandro. "Traducción automática del lenguaje dactilológico de sordos y sordomudos mediante sistemas adaptativos". Revista Ingeniería Biomédica Publicado por Escuela de Ingeniería de Antioquia CES-, Universidad CES [en línea], 2013, 7(13), pp. 18-30 [Consulta 15 febrero 2023] ISSN 1909-9762. Disponible en: http://www.scielo.org.co/scielo.php?pid=S1909-97622013000100003&script=sci_abstract&tlng=es

BOLTON, William, 2017. *Mecatrónica*. [en línea]. 6. Colombia: Alfaomega Grupo Editor S.A de C.V, 2017. [Consulta: 15 febrero 2021] Disponible en: <https://books.google.es/books?hl=es&lr=&id=b6FxEAAQBAJ&oi=fnd&pg=PR5&dq=que+es+un+sistema+de+control+electrónico&ots=tnB8pxWpB6&sig=h0BQA-BIylZwPqCCEZ2tPbERk7A#v=onepage&q=que+es+un+sistema+de+control+electrónico&f=false>.

CABALLERO, E. Aplicación práctica de la visión artificial para el reconocimiento de rostros en

una imagen, utilizando redes neuronales y algoritmos de reconocimiento de objetos de la biblioteca OpenCV (Trabajo de titulación) (Ingeniería). [en línea]. Universidad Distrital Francisco José de Caldas Facultad de Ingeniería. Bogotá, Colombia. 2017. pp. 7-60. [Consulta: 15 febrero 2023] Disponible en: <https://repository.udistrital.edu.co/bitstream/handle/11349/6104/CaballeroBarrigaEdisonRene2017.pdf?sequence=1&isAllowed=y>.

CÁCERES BORDALLO, Javier. Sistema de reconocimiento de gestos utilizando técnicas de Machine Learning en un entorno quirúrgico (Trabajo de titulación) (Ingeniería). [en línea]. E.T.S.I. y Sistemas de Telecomunicación (UPM), Madrid-España. 2021. pp. 1-105. [Consulta: 15 febrero 2023] Disponible en: <https://oa.upm.es/70354/>.

CHIGUANO, E. y MORENO, N. Diseño e implementación de un sistema traductor de lenguaje de señas de manos a un lenguaje de texto mediante visión artificial en un ambiente controlado. (Trabajo de titulación) (Ingeniería). [en línea]. Escuela Politécnica Nacional, Quito-Ecuador. 2011. pp. 1-134 [Consulta: 15 febrero 2023] Disponible en: <https://bibdigital.epn.edu.ec/handle/15000/3915>

Consejo Nacional para la Igualdad de Discapacidades CONADIS, 2022. *Estadísticas de Discapacidad*. [en línea]. Ecuador: 2022. [Consulta: 15 febrero 2023] Disponible en: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>.

GARCÍA SANUNGA, Jefferson Tobías. Implementación de un prototipo de emisión de alertas en zonas de peligro para el cuidado de niños de 1 a 5 años mediante visión artificial. (Trabajo de titulación) (Ingeniería). Escuela Superior Politécnica de Chimborazo, Riobamba-Ecuador. 2023. pp. 1-114.

GONZÁLEZ, R. *Python para Todos* [en línea]. Elliott Hauser, Sue Blumenberg repositario.uci.cu, 2011 [Consulta: 15 febrero 2023] Disponible en: https://repositorio.uci.cu/bitstream/123456789/10206/1/Python_para_todos.pdf.

GRANADOS, C.U. Programando directamente un ESP8266. (Memoria) [en línea]. Escola Tècnica Superior d'Enginyeria Industrial de Barcelona. Barcelona-España. 2017. pp. 1-52 [Consulta: 15 febrero 2023] Disponible en: https://upcommons.upc.edu/bitstream/handle/2117/105042/Memoria_TFG_Carles_Ubach.pdf?sequence=1&isAllowed=y.

GUERRA, A.; MOROCHO, M.J. & VALDIESO, C., 2012. Control mediante joystick de tarjeta AVR BUTTERFLY (con microcontrolador ATMEGA169) usando módulos de radiofrecuencia para comunicación con tarjeta LPC1769 controladora de motor BLDC. (working paper) [en línea] Escuela Superior Politécnica del Litoral. Quito, Ecuador. 2012. pp. 1-5. [Consulta: 15 febrero 2023] Disponible en: https://www.dspace.espol.edu.ec/bitstream/123456789/24402/1/Paper_Guerra-Morocho.pdf.

MAHEDERO, F. Desarrollo de una aplicación iot para el envío de imágenes mediante el protocolo MQTT (Trabajo de titulación) (Ingeniería) [en línea] Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València. Valencia-España. 2020. pp. 1-51. [Consulta: 15 febrero 2023] Disponible en: <https://riunet.upv.es/bitstream/handle/10251/152408/Mahedero - Desarrollo de una aplicación IoT para el envío de imágenes mediante el protocolo MQTT..pdf?sequence=1&isAllowed=y>.

MARÍN, A. *Integración de la programación en el aprendizaje de economía: el uso de Jupyter Notebook.* [blog]. 2019 [Consulta: 15 febrero 2023] Disponible en: <http://www.aliciamarin.es/>

MÁRQUEZ AGUILAR, J.I. Diseño de un relevador de estado sólido para la protección de equipo eléctrico (Trabajo de titulación) (Maestría) [en línea] Instituto Tecnológico de Ciudad Madero. Madero, Tamaulipas. 2018. pp. 1-90 [Consulta: 15 febrero 2023] Disponible en: <https://rinacional.tecnm.mx/handle/TecNM/3117>

MORENO, S. 2020. Herramienta de Reconocimiento de Imágenes en Python (Trabajo de titulación) (Ingeniería) [en línea]. Escuela Técnica Superior de Ingeniería Universidad de Sevilla. Sevilla-España. 2020. pp. 1-88 Disponible en: <https://idus.us.es/bitstream/handle/11441/102090/TFG-2877-MORENO FERNANDEZ.pdf?sequence=1&isAllowed=y>.

MURILLO HERNÁNDEZ, W.J. La investigación científica. [blog] 2008 [Consulta: 14 febrero 2023]. Disponible en: <https://www.monografias.com/trabajos15/invest-cientifica/invest-cientifica>.

ÑACATO, A. Diseño e implementación de una interfaz para la traducción de caracteres del alfabeto dactilológico a palabras utilizando sensores musculares y reconocimiento de patrones. (Trabajo de titulación) (Ingeniería) [en línea]. Escuela Superior Politécnica de Chimborazo. Riobamba-Ecuador. 2019. pp. 1-99 [Consulta: 14 febrero 2023]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/13706>

PÉREZ MARTÍNEZ, M.; RAMOS GUARDARRAMA, J.; RODRÍGUEZ VALDÉS, J.A.; SANTOS BARANDA, J. & LÓPEZ COLLAZO, Z.S. 2022. "La simulación como método para mejorar el proceso de enseñanza-aprendizaje de los circuitos eléctricos". Referencia Pedagógica [en línea], 2022, (La Habana-Cuba) 10(1), pp. 157-172. [Consulta: 15 febrero 2023]. Disponible en: http://scielo.sld.cu/scielo.php?pid=S2308-30422022000100157&script=sci_arttext&tlng=en.

RENTERÍA, R., RUELAS, R. y OCHOA, G. 2017. "Módulo ESP8266 y sus aplicaciones en el internet de las cosas". Revista de Ingeniería Eléctrica [en línea], 2017, (Perú) 1(2) pp. 32-72. [Consulta: 16 febrero 2023] Disponible en: https://www.ecorfan.org/republicofperu/research_journals/Revista_de_Ingenieria_Electrica/vol1num2/ECORFAN_Revista_de_Ingenieria_Electrica_VI_N2.pdf#page=31.

RIVAS, P. Desarrollo de un intérprete básico del lenguaje de señas para dactilología empleando inteligencia artificial (Trabajo de titulación) (Tecnólogo) [en línea]. TECNOLÓGICO NACIONAL DE MÉXICO. 2019. pp. 1-98 [Consulta: 16 febrero 2023] Disponible en: <https://www.reev.us/pdfs/rivas2019desarrollo.pdf>.

SALAS, R. *Redes Neuronales Artificiales* [blog]. 2004. [Consulta: 16 febrero 2023] Disponible en: https://www.academia.edu/24633757/Redes_Neuronales_Artificiales

SÁNCHEZ, Leandro. "Inteligencia artificial y visión por computadora aplicada a la educación". Revista Odigos [en línea], 2022 (Ecuador) 3(2), pp. 61-73. [Consulta: 16 febrero 2023] Disponible en: <https://revista.uisrael.edu.ec/index.php/ro/article/view/587>

VERA, Y. Desarrollo e Implementación de una aplicación móvil para la gestión de seguros de autos en Pacífico Seguros utilizando Tensorflow (Trabajo de titulación) (Tecnólogo) [en línea] Universidad Tecnológica del Perú. (Perú). 2019. pp. 1-123 [Consulta: 16 febrero 2023] Disponible en: file:///C:/Users/GENESIS/Downloads/Yerko_Vera_Trabajo_de_Suficiencia_Profesional_Titulo_Profesional_2019.pdf.

VILCHES, J. La dactilología, ¿qué, cómo, cuándo...? (working paper) [en línea]. Escuela Universitaria de Magisterio "Sagrado Corazón" de Córdoba (Córdoba-España). 2005. pp.1-5 [Consulta: 16 febrero 2023] Disponible en: http://www.uco.es/~fe1vivim/alfabeto_dactilologico.pdf.

ANEXOS



