



Implementación de un Sistema de Control Remoto para la Inspección de un Proceso Inmótico sobre un Sistema Embebido a través de Internet para la Contingencia del COVID 19.

Trabajo de grado para optar al título de
Ingeniero en Mecatrónica

Trabajo de Grado

Autor:
Jhan Carlos Roperó Delgado

Tutor:
PhD. Cristhian Ivan Riaños Jaimes

16 de Junio de 2021

Formando **líderes** para la
construcción de un nuevo
país en paz

Ingeniería Mecatrónica

Implementación de un Sistema de Control Remoto para la Inspección de un Proceso Inmótico sobre un Sistema Embebido a través de Internet para la Contingencia del Covid19.

Autor

Jhan Carlos Roperero Delgado

Tutor

PhD. Cristhian Ivan Riaños Jaimes
Universidad de Pamplona



Trabajo de grado para optar al título de Ingeniero en Mecatrónica

Universidad de Pamplona
Facultad de Ingenierías y Arquitectura
Pamplona, Colombia
16 de Junio de 2021

Dedico este trabajo a...

A mi papá, Diomedes Ropero Uribe y a mi mamá Sonia Delgado Criado quiénes me apoyaron durante todo el proceso de formación universitaria, quiénes que con su amor incondicional y sabios consejos me ayudaron a no rendirme y a superarme cada día sin importar que tan grande eran los obstáculos.

A mi hermano y hermana, Fabián Ropero Delgado y Daniela Ropero Delgado, por su cariño y estar para mí siempre.

A mis familiares y amigos, quienes siempre creyeron en mí, me aconsejaron y me apoyaron.

Agradecimientos

Este trabajo no habría sido posible sin el apoyo y el estímulo de mi tutor, MSc. Cristhian Ivan Riaño Jaimes, bajo cuya supervisión escogí este tema y comencé la tesis. Además, fue mi consejero en cada etapa que contempla mi trabajo de grado, también ha sido generosamente servicial por permitirse leer, en cuanto estuvieron disponible cada avance del presente libro.

No puedo terminar sin agradecer a mi familia, colegas y amigos:

Mi padre, Diomedes Ropero Uribe, uno de mis motores en la vida que poco a poco me ha inspirado a seguir adelante sin importar los obstáculos que se encuentren en el camino.

Mi madre, Sonia Delgado Criado, una mujer emprendedora que a pesar de tantas adversidades ha luchado hombro a hombro para darme el privilegio de seguir estudiando.

Mis hermanos, Fabián Ropero Delgado y Daniela Ropero Delgado, jóvenes ambiciosos de crecer y mejorar día a día, con ello, trato de ser un ejemplo de vida cercano y así puedan seguir mis pasos para crecer profesionalmente.

Mi abuela, Noema Uribe Ortiz, una mujer que desde el cielo me acompaña y me protege. Un día le prometí en convertirme en un profesional y enorgullecerla pero Dios se la llevó. Gracias a ella es que sigo luchando por alcanzar éste sueño de ser Ingeniero Mecatrónico.

Mi tío y pareja, Alvaro Delgado Criado y Rosa Cecilia Zafra, una pareja que me han brindado todo su apoyo incondicional, siempre estando para mí sin importar las circunstancias.

A mis colegas, Jasyd David Caballero Quintero, Leyder Dalmir Quintero Bayona y Hanuar Antonio Rubio Cárdenas; personas que me brindaron su confianza y amistad. Eternamente agradecido por todo el apoyo incondicional que me dieron en los buenos y malos momentos.

A mi amiga, Ana Camila Lozada Peña, una mujer que entre risas, llanto, compañía y consejos; me ha ayudado a no desistir, a tener la mirada en alto para seguir luchando y llegar a la cima.

Y a todos mis amigos por apoyarme e inspirarme con su valiosa compañía. A todos ellos mi cariño y profundo agradecimiento por su significativa presencia y apoyo incondicional. Es a ellos a quien dedico este trabajo de grado.

Si consigo ver más lejos es porque he conseguido auparme a hombros de gigantes

-Isaac Newton-

Resumen

En este documento se presenta el desarrollo de un sistema inmótico para el control de accesos a un recinto; este control se realiza remotamente a través de internet. El proyecto recorrió varias fases, inicialmente se realiza un estudio del estado del arte para identificar los tipos de sistemas inmóticos que hay en la actualidad y disponer de la alternativa más adecuada con el propósito de cumplir con los objetivos estipulados. Posteriormente, se estudió el funcionamiento de técnicas de visión artificial, métodos y equipos utilizados para la implementación. Un sistema de visión artificial y una base de datos específica de imágenes (data sets), se emplean para desarrollar un código de programación para el reconocimiento facial que identifique el uso de tapabocas en individuos que se acerquen al lugar. Seguidamente se crea la red inmótica, la cual, enlazará inalámbricamente todos los puntos que formen parte del sistema ordenando los parámetros de comunicación y configuración. Resaltando los elementos fundamentales para la creación de esta red inalámbrica se usará una placa Raspberry Pi; para el avance del sistema de control implicó, además, el diseño y construcción de un punto o nodo de control de red y otro del servidor de dicha placa, de tal manera que se pueda conectar y controlar los elementos del recinto; tales como: cámara, motor, luz ultravioleta. Además, se desarrolló un tablero de control eléctrico con salidas de voltajes estándares para el funcionamiento de los elementos finales de control como motores eléctricos y la luz ultravioleta. Una aplicación web (Interfaz) en REACT (Biblioteca de JavaScript Software Libre) es creada para que el usuario acceder al sistema y pueda controlar, permitiendo la interacción entre cada uno de los dispositivos que contemplan el proceso. Para finalizar, se evaluará el sistema de control verificando el funcionamiento y el cumplimiento de los objetivos anteriormente descritos.

Palabras Claves: Inmótica, Sistema de Control, Sistema Embebido, Visión Artificial, Aplicación WEB.

Abstract

The project went through several phases. Initially, a study of state of the art is carried out to identify the types of inmotoc systems currently existing and have the most appropriate alternative to meet the stipulated objectives. Subsequently, the operation of artificial vision techniques, methods, and equipment used for implementation were studied. An artificial vision system and a specific database of images (data sets) are used to develop a programming code for facial recognition that identifies masks in individuals who approach the place. Next, the inmotoc network is created, which will wirelessly link all the system points, ordering the communication and configuration parameters—highlighting the fundamental elements for creating this wireless network; a Raspberry Pi board will be used. In the control system development, it was necessary to design and build a network control point or node and another for the server on said board so that the enclosure elements such as camera, motor, and ultraviolet light can be connected and controlled. In addition, an electrical control board with standard voltage outputs was developed for the operation of the aforementioned final control elements. A web application (Interface) in REACT (Free Software JavaScript Library) is created to access the system and control it, allowing interaction between each of the devices that contemplate the process. Finally, the control system will be evaluated, verifying the operation and compliance with the objectives described above.

Keywords: Innmotics, Control System, Embedded System, Artificial Vision, WEB Application.

Índice general

Agradecimientos	vii
Resumen	xi
Abstract	xiii

Capítulos	Página
1. Introducción	1
1.1. Justificación	3
1.2. Objetivos	5
1.2.1. Objetivo General	5
1.2.2. Objetivos específicos	5
1.3. Estructura del Libro	6
2. Marco Teórico	7
2.1. Introducción	7
2.2. Antecedentes Investigativos	7
2.2.1. Visión Artificial - Reconocimiento Facial	7
2.2.2. Redes Neuronales Convolucionales	8
2.2.3. Inmótica	9
2.2.4. Aplicaciones Raspberry Pi	9
2.3. Fundamentos Teóricos	10
2.3.1. Visión Artificial	10
2.3.2. Inteligencia artificial (AI)	14
2.3.3. Reconocimiento de Patrones	24
2.3.4. Inmótica	26
2.3.5. Sistema Embebido - Raspberry Pi	27
2.3.6. Transfer Learning	28
2.4. Síntesis	29
3. Metodología	31
3.1. Introducción	31
3.2. Metodología y Fase de Investigación	32
3.3. Síntesis	34

4. Desarrollo de la Propuesta	35
4.1. Introducción	35
4.2. Composición del Dataset de Imágenes	35
4.3. Estructura de la Red Neuronal a Implementar	37
4.3.1. Arquitectura de la CNN	37
4.3.2. Dataset para el Entrenamiento de la CNN	38
4.3.3. Parámetros de entrenamiento de la MobileNet	39
4.3.4. Verificación del funcionamiento de la Red Neuronal	42
4.4. Interfaz para la Comunicación Sistema - Usuario	46
4.4.1. Programación en REACT	47
4.5. Implementación de la Red Neuronal Entrenada en la Interfaz Sistema - Usuario	52
4.6. Programación del Sistema Embebido	58
4.7. Tablero Eléctrico	61
4.8. Síntesis	64
5. Resultados	67
5.1. Introducción	67
5.2. La arquitectura MobileNet Entrenamiento	67
5.3. Verificación del funcionamiento usando OpenCV	68
5.4. Verificación del Funcionamiento del Modelo en el Prototipo	71
5.5. Otras Pruebas de Funcionamiento	75
5.5.1. Pruebas del Prototipo - Usando Accesorios	75
5.6. Recomendaciones	84
5.7. Síntesis	85
6. Conclusiones	87
Bibliografía	89
A. Apéndice	93
A.1. Programación de la Interfaz Gráfica	95
A.2. Programación Sistema Embebido	98

Índice de figuras

2.1. Método Secuencial Para Reconocimiento de Imágenes.	11
2.2. Componentes de un Sistema de Visión Artificial.	12
2.3. Clasificación de un Sistema de Reconocimiento.	14
2.4. Esquema de una neurona artificial.	16
2.5. Estructura de una neurona biológica.	17
2.6. Características aprendidas de una red neuronal convolucional.	19
2.7. Ejemplo de una operación de convolución con un kernel de tamaño 3×3 , con desplazamiento de valor 1.	20
2.8. Representación de la funcionalidad de ReLU.	21
2.9. Representación del max pooling y el average pooling.	21
2.10. Fully-connected layer	22
2.11. El modelo MobileNet puede ser aplicado a varias tareas de reconocimiento de forma eficiente.	22
2.12. El sistema de detección YOLO.	23
2.13. Imagen con grilla de entrada, cuadros delimitadores, mapa de probabilidades y detección del algoritmo YOLO.	23
2.14. Paradigmas del Reconocimiento de Patrones.	25
2.15. Esquema general de un Sistema de Reconocimiento de Patrones.	26
3.1. Metodología y Fase de Investigación.	31
4.1. Imágenes CON Mascarilla del Dataset.	36
4.2. Imágenes SIN Mascarilla del Dataset.	37
4.3. Esquema de la Red Neuronal.	38
4.4. Imágenes del Dataset CON y SIN Mascarilla.	38
4.5. Diagrama de Flujo para Entender la Lógica de Programación del Algoritmo para la Verificación del Funcionamiento del Modelo.	43
4.6. Bosquejo de la Interfaz Gráfica para Implementarla en REACT.	47
4.7. Diagrama de Flujo para Entender la Lógica de Programación del Algoritmo para la Interfaz.	48
4.8. Visualización de las Líneas de Código en REACT	50
4.9. Primera Visualización de la Interfaz Gráfica.	51
4.10. Diagrama de Flujo para Entender la Lógica de Programación del Algoritmo 4.4	53
4.11. Primera Visualización del Reconocimiento Facial.	56
4.12. Modelo Neuronal en la Interfaz Gráfica para el Reconocimiento Facial.	57

4.13. Diagrama de Flujo para Entender la Lógica de Programación del Algoritmo 4.5	59
4.14. Bosquejo del Tablero Eléctrico en una Fase Preliminar.	62
4.15. Tablero Eléctrico en una Fase Real	63
5.1. Gráfico precisión de MobileNet vs épocas y Pérdidas vs épocas.	67
5.2. Imágenes para la verificación del funcionamiento de la red con OpenCV	68
5.3. Funcionamiento de la red con OpenCV	69
5.4. Verificación del Funcionamiento del Modelo con una Mujer.	71
5.5. Funcionamiento del Modelo y Verificación de los Pilotos Luminosos . .	72
5.6. Funcionamiento Completo del Modelo.	72
5.7. Verificación del Funcionamiento del Modelo en el Prototipo.	73
5.8. Carga de 110v Conectada a la Salida de Voltaje.	74
5.9. Desempeño de la Red Neuronal - Autor con Gorra y Tapabocas.	75
5.10. Desempeño de la Red Neuronal - Autor con Gorra y sin Tapabocas. . .	76
5.11. Desempeño de la Red Neuronal - Autor con Gorra, Gafas y sin Tapabocas.	76
5.12. Desempeño de la Red Neuronal - Autor con Gorra, Gafas y Tapabocas.	77
5.13. Desempeño de la Red Neuronal - Autor con Bufanda y sin Tapabocas.	77
5.14. Desempeño de la Red Neuronal - Autor con Bufanda y Tapabocas. . .	78
5.15. Desempeño de la Red Neuronal - Autor sin Tapabocas.	79
5.16. Desempeño de la Red Neuronal - Autor con Tapabocas.	79
5.17. Desempeño de la Red Neuronal - Autor con Gorra y Tapabocas.	80
5.18. Desempeño de la Red Neuronal - Autor con Bufanda y sin Tapabocas.	80
5.19. Desempeño de la Red Neuronal - Autor con Gafas, Bufanda y sin Tapabocas.	81
5.20. Desempeño de la Red Neuronal - Autor Bufanda y Tapabocas.	81
5.21. Desempeño de la Red Neuronal - Primer Caso Hipotético.	82
5.22. Desempeño de la Red Neuronal - Segundo Caso Hipotético.	83
5.23. Desempeño de la Red Neuronal - Tercer Caso Hipotético.	83
A.1. Diagrama de Flujo - Algoritmo de Reconocimiento Facial.	93
A.2. Diagrama de Flujo - Funcionamiento de la Interfaz Gráfica.	94

Índice de tablas

2.1. Especificaciones Técnicas de la Raspberry Pi	30
4.1. Resumen de la arquitectura MobileNet	39

Índice de Códigos

4.1. Algoritmo para el entrenamiento de la MobileNet	40
4.2. Algoritmo para la verificación del funcionamiento del modelo.	44
4.3. Algoritmo inicial para la Interfaz Sistema-Usuario	49
4.4. Algoritmo para la implementación de la Red Neuornal en la Interfaz . .	52
4.5. Algoritmo Base para el Sistema Embebido	58
A.1. Algoritmo Completo Programación Interfaz	95
A.2. Ejemplo - Algoritmo una Función de la Interfaz Gráfica	96
A.3. Algoritmo Completo Programación Sistema Embebido	98

1. Introducción

La creciente importancia que tiene la tecnología en el mundo actual y su continuo desarrollo y evolución, conllevan a ir mejorando o implementando estrategias para la eficiencia de los procesos. Es por ello que surge la temática de la inmótica, la cual consiste en automatizar inmuebles o edificaciones de cualquier tipo, aportando una buena gestión energética, seguridad y bienestar; este tipo de sistemas están integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, cuyo control se satisface a cierta ubicuidad dentro o fuera del recinto de automatización[1]. El resultado se puede decir que es óptimo, pero muy dependientes de factores como la calidad de materiales, costos de los mismos y hasta las diversas técnicas de programación que se lleven a cabo.

Es un hecho que la inmótica está generando un huella en la humanidad, y si bien es algo que entra gradualmente a la industria, en un futuro no muy alejado se convertirá en un estilo mundial, ofreciendo muchas ventajas. Y con el constante hábito de las diferentes tecnologías como tabletas y celulares, nos veremos cada vez más atraídos por estas nuevas tecnologías que faciliten nuestro continuo existir.

La pandemia del coronavirus covid 19 es una de las principales crisis humanitarias recientemente, han puesto a prueba la limitada medida de consulta de muchos sistemas de sanidad ante situaciones extremas, ya que estos han colapsado ante la enorme solicitud de análisis, inclusive en países con sistemas de sanidad que se pueden reconocer como ejemplares[2]. El covid 19 ha afectado actualmente a un sinnúmero de personas en el mundo entero, y ha matado a cerca de 276 mil víctimas[3]. Esta problemática supera los medios sanitarios de nuestra humanidad, ha reducido considerablemente los recursos mundiales y forzado a los gobiernos a destinar enormes cantidades de fondos para su reivindicación[4], a excepción del inmenso costo general y el martirio de las familias y comunidades de las personas o individuos afectados.

Un aspecto relevante en esta investigación es la aplicación de visión artificial, el cual, incluye el uso de hardware y software para procesar y a su vez analizar fotografías en un entorno específico; de dicha manera se genera información al instante para una posterior toma de decisiones[5]. Al integrar esta aplicación, se obtienen ventajas como adaptabilidad a la estructura de las edificaciones, alta precisión en el proceso, facilidad para la instalación y/o mantenimiento, datos accesibles y un cómodo manejo para el sistema.

Teniendo ésta problemática presente y ventajas que pueden otorgar los métodos a usar, la finalidad es establecer un sistema de control remoto para la inspección de un proceso inmótico y reducción de la propagación del presente virus, haciendo uso de la creación de una página web como interfaz para el control interno/externo de la edificación, con el uso de un sistema embebido, el cual nos proporcione la comunicación y el control de las variables a tratar. Además busco aportar o dar una nueva perspectiva como solución a la propagación del presente virus, contando con la ayuda de visión artificial, su función es observar a través de una cámara de definición HD el rostro de las personas que se acercan al recinto; con ello se busca hacer un reconocimiento facial, el cual, nos informe sí el individuo está haciendo uso del tapabocas (Primera medida de Bioseguridad) para otorgarle el acceso al lugar. En este trabajo se presenta el desarrollo de un sistema de control confiable, con el fin de brindarle al usuario y a los trabajadores un espacio seguro que cumpla con las medidas estándares de bioseguridad. [6]

1.1. Justificación

Los esfuerzos de las entidades de salud nacionales e internacionales en encontrar una cura para erradicar el virus aun no es efectiva. Actualmente no existe un tratamiento útil validado para el covid 19. Partiendo de ésto, las medidas que se han acogido hasta el instante se pueden ordenar en tres fuentes primordiales: “Medidas Sanitarias y de Acontecimiento Sanitaria”, “Medidas de Hecho Nacional, Económica y Ecológica” y “Medidas de Precepto General y otras de Forma Común”. Lo antepuesto sin pérdida de dar la razón que existen relaciones estrechas entre todo el conjunto de éstas normas[7].

Este estudio se realiza con el fin de otorgar un soporte de información sobre variables, procesos, control, equipos, visión artificial, canales de comunicación, parámetros que son esenciales para el correcto desempeño del sistema de control remoto.

Efectuar este trabajo de investigación, no sólo implica el análisis estructural para el desarrollo del mismo, sino, también en pensar sobre el cuidado de las personas que frecuentan el recinto. Con esto, me refiero a los trabajadores que laboran día a día en ese lugar; brindar un espacio seguro y de bienestar es un objetivo abstracto en la implementación del sistema inmótico.

La noción de eficacia de vida profesional tiene que contemplar con la dicha, la sanidad, la paz y la seguridad del empleado, y asimismo con todo lo concerniente con su ambiente de trabajo. Es una percepción que pretende mediar aspectos de la labor que tienen que distinguir con las experiencias individuales y con los objetivos organizacionales. Por consiguiente, un pensamiento multidimensional, formado por indicadores objetivos y subjetivos que tienen en recuento tanto al humano como el enlace en el que se desenvuelve. (Elizur Shye, 1990; Flores et al., 2010).

Las razones que justifican elaborar una tesis como la que aquí se propone son el beneficio visto para todos los factores intervinientes en la interacción profesional o laboral, en el cual se privilegia la seguridad, al momento que se impulsa el crecimiento individual del trabajador, sin desatender los objetivos, la disminución de la propagación del contagio mediante un sistema de inspección remoto.

Por lo anterior, se busca dejar un libro documentando el desarrollo del sistema de control y a su vez, un prototipo compacto y estándar. Siguiendo esta premisa, llevar acabo esta investigación es de suma importancia debido a que se puede aportar una solución viable ante ésta problemática.

Han habido desarrollos como por ejemplo “Sistema de control remoto para aplicaciones domóticas a través de internet”, que han tratado de solventar una problemática; teniendo en cuenta estos trabajos como punto de partida, se desarrolló un sistema capaz de notificar al usuario el funcionamiento en tiempo real del sistema de control. Imágenes tomadas de una cámara, serán procesadas mediante técnicas de inteligencia artificial como las redes neuronales convolucionales, para la estimación de reconocimiento facial y uso de tapabocas. Además, usando las salidas análogas-digitales de la placa Raspberry Pi se puede continuar con el proceso como es la activación para la apertura de una puerta y demás subprocesos ligados a éste.

1.2. Objetivos

1.2.1. Objetivo General

- Desarrollar un sistema de control remoto (Aplicación web o App) para la inspección de un proceso inmótico de bajo costo, basado en un código abierto, con ayuda de un sistema embebido.

1.2.2. Objetivos específicos

- Diseñar y elaborar un tablero de control eléctrico, el cual nos proporciones salidas de voltajes estándares para la conexión de elementos finales de control.
- Desarrollar un sistema inmótico que permita el control de sensores y actuadores, lo cual le posibilita al usuario controlar de forma fácil el proceso inmótico.
- Desarrollar una interfaz (Página Web) en REACT (Biblioteca de JavaScript Software Libre) para la visualización y control del sistema.
- Implementar algoritmos de visión artificial para el reconocimiento facial de una dataset de imágenes de personas con y sin tapabocas.
- Programar el sistema embebido (Raspberry Pi), el cual tiene un servidor que se comunicará con la interfaz para el control Usuario-Interfaz.
- Validar el funcionamiento del sistema de manera experimental.

1.3. Estructura del Libro

Este trabajo consta de 6 capítulos distribuidos de la siguiente manera:

- **Capítulo 1:** Corresponde con la investigación y el trazado del problema existente en el presente, del mismo modo el argumento por el cual se efectúa un modelo de sistema de reconocimiento del aspecto facial y se plantean objetivos que servirán como apoyo en el progreso de la tesis.
- **Capítulo 2:** Muestra el marco teórico. Se explican los aspectos teóricos para conocer los diversos sistemas de visión artificial en la actualidad, realizando una minuciosa indagación de estas técnicas de identificación del rostro, en el cual se analiza el conjunto de las etapas que implica ésta identificación. Finalmente se presentan los conceptos relacionados con redes neuronales convolucionales.
- **Capítulo 3:** Se expone la metodología a seguir, se identifica el esquema de investigación, el resumen de indagación, señalando las tareas necesarias y obligatorias para formar la propuesta de desenlace.
- **Capítulo 4:** Se demuestra de forma detallada el proceso del arquetipo de método de identificación del rostro para la determinación de personas con el utilización o no de la mascarilla, agregando la parte de redes neuronales y la realización del tablero eléctrico. Determinando el esquema de flujo del sistema, el conjunto de componentes, programación, pruebas de marcha y funcionalidad.
- **Capítulo 5:** Se muestran los resultados obtenidos durante todo el desarrollo del trabajo de grado. Además, se especifican las posibles mejoras del prototipo.
- **Capítulo 6:** Estipula las conclusiones obtenidas en el desarrollo del sistema inmótico.

2. Marco Teórico

2.1. Introducción

El marco teórico que fundamenta éste estudio proporcionará al lector un concepto objetivo acerca de este proyecto. Tendrá una noción de los factores más relevantes de discusión y encontrará los conceptos básicos, complementarios y específicos de cada área abordada, sin olvidar un repaso bibliográfico sobre proyectos similares para tratar de tener una base de investigación y comparación que soporte nuestros resultados.

2.2. Antecedentes Investigativos

Dentro de la revisión del estado del arte se han encontrado varios escritos referentes a los diversos mecanismos de detalles existentes hoy por hoy, al mismo tiempo de trabajos de búsqueda con ciertas similitudes o apariencias, las cuales sirvieron de pautas para el desarrollo de la actual tesis.

2.2.1. Visión Artificial - Reconocimiento Facial

Enfoques en la identificación fisiológico según el ente de datos usados. Según la clase de los datos usados en el transcurso de inspección, existen dos enfoques: a partir de mapas de dominio y mapas de cavidad. Los basados en mapas de dominio realizan el estudio de fotografías digitales de la cara (imágenes 2D), sin embargo éstos presentan algunas desventajas como la susceptibilidad a los cambios de vista, de gradilla, luminosidad y el evento de suplir a un usuario a partir de otra fotografía 2D. Los sistemas basados en mapas de cavidad (imágenes 3D), realizan la determinación de la información de cavidad y/o la analítica de la cara. Algunas de las ventajas de estos sistemas son la indulgencia a los cambios de plano, a cambios de luminosidad y a las locuciones faciales. Por otro lado la adquisición de la información tridimensional del gesto no es tan ordinario, de manera que estos sistemas son robustos ante el intento de suplantación o falsificación[8].

En base a la exploración de Juan José Toro Agudelo sobre de los modelos de biometría basados en esquemas faciales del individuo. Debido a esto se puede plasmar la caracterización y seguidamente la identificación de personas, para admitir su acceso a ciertos bienes, pero este tipo de sistemas no se limita precisamente a dominar la

identidad del personal, sino que permite un progreso en el espacio de la videovigilancia, en la actividad con la identificación de procesos, por referir algunas aplicaciones, en dicho compromiso en el cual se establece un extenso examen de varias técnicas de inspección fisonómico existentes, enfocándose principalmente en los estándares faciales de las personas[9].

El artículo de Luis Blázquez en fundamento al espacio de identificación fisonómico está basado en el examen de puntos característicos del rostro en entornos no controlados, principalmente enfocado al desarrollo de un medio maquina de localización y finura de puntos faciales los cuales son característicos de las personas que están mal enmarcados, y mediante el uso de buses de datos controlados e incontrolados de autónoma dirección, de este modo se realiza un estudio de cada uno de los rasgos faciales y asimismo desarrollaron experimentos en cada curso del procedimiento que se ha ajustado en observar la viabilidad del detector y del verificador de los puntos faciales internamente en su prototipo[10].

Los campos de la visión por computadora y el procesamiento de imágenes digitales incluyen el reconocimiento facial, la identificación biométrica, Internet de las cosas (IoT), las investigaciones de delitos y las firmas están surgiendo. Identificación de patrones, análisis de documentos digitales, etiquetas inteligentes de Toll Plaza y muchos más. Ambas tecnologías se utilizan para capturar un análisis de imagen y video en tiempo real para estudios y pronósticos completos. La visión por computadora está ampliamente implementada en diversos sistemas, incluye procesamiento de imágenes 2D y 3D, pronósticos de emociones, reconocimiento de puntos, interacción humano-computadora (HCI), reconocimiento facial, robots móviles, interpretación de expresiones, percepción y expresión, visión estéreo, movimiento análisis, patrones de movimiento, dirección del movimiento, supervisión de la dirección de la actividad y otros sistemas[11].

2.2.2. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) se utilizan ampliamente para la clasificación de imágenes. Las imágenes ruidosas reducen el rendimiento de clasificación de las redes neuronales convolucionales y aumentan el tiempo de entrenamiento de las redes. En este artículo, se propone una red neuronal convolucional resistente al ruido (NR-CNN) para clasificar las imágenes ruidosas sin ningún preprocesamiento para la eliminación de ruido y mejorar el rendimiento de clasificación de imágenes ruidosas en redes neuronales convolucionales. En el NR-CNN propuesto, se agregan una capa de mapa de ruido y una capa de cambio de tamaño adaptativo a la arquitectura de la red neuronal convolucional. Además, el problema del ruido se considera en diferentes componentes de NR-CNN, de modo que la capa convolucional, la capa de agrupación y la función de pérdida de la red neuronal convolucional se mejoran para la robustez

de CNN al ruido. El aumento de datos adaptativo basado en el mapa de ruido se introduce para mejorar el rendimiento de clasificación del NR-CNN propuesto. Los resultados experimentales demuestran que el NR-CNN propuesto mejora la clasificación de imágenes ruidosas y la velocidad de entrenamiento de la red[12].

2.2.3. Inmótica

Villacampa realizó el diseño, montaje y puesta en movimiento de un método de inmótica para un edificio, diseñado para lograr los niveles de confianza y comodidad deseados, que consideraba, desde la protección frente a intrusos hasta sonidos ambientales, pasando por control de incendios. Afrontaba considerablemente el inconveniente de centralizar dispositivos, pero no contaba con la tecnología de Internet ni con plataforma de software abierto. Establece un solo controlador para funciones y acciones de sensores y actuadores, con la tecnología aportada concretamente por los módulos de inmótica y programa una simulación de software mediante Visual Basic, el cual permite ver cómo se comporta el sistema en una instalación real[13].

M. Lledó, realizó un trabajo de grado basado en Arduino. Allí deja ver que el proyecto se encarga de brindar los conocimientos básicos para entender qué es y cómo funciona un sistema domótico y cómo, utilizando el hardware libre de Arduino, se puede crear un sistema estable con un presupuesto muy inferior al de las viviendas de alta categoría[14].

Barrera en el artículo: “Análisis y diseño de un modelo de sistema domótico de inferior costo”, da a tratar los diferentes protocolos que pueden ser implementado en el intercambio de dispositivos idóneos para los sistemas inmóticos, mostrando las grandes ventajas ofrecidas por los protocolos que uso, al mismo que permiten establecer la posibilidad de diseñar tarjetas de control con elementos de fácil adquisición[15].

ATT lanzó al mercado en el mes de abril de 2013 el proyecto denominado Digital Life. En primera instancia tuvo una cobertura de 15 mercados a nivel mundial. Este servicio se encarga de la provisión y soporte de tecnología en lo que se refiere a automatización laboral. ATT considera a la inmótica como una oportunidad grande de mercado ya que existe un 35% de oficinas con algún tipo de servicio de seguridad y menos del 8% tiene servicios de automatización real[16].

2.2.4. Aplicaciones Raspberry Pi

Roger Gimeno y Josep Ramos hablan acerca de las técnicas de identificación de patrones, se desarrolla en sistemas que se aplican rectamente justo las representaciones pictóricas sin ejecutar la utilización de modelos 3D, en el cual los objetos están representados en acto de las diferentes vistas del propio, el primordial objetivo de los diferentes algoritmos es el de catalogar las diferentes subcaras de un cuerpo en

un nuevo sub lugar, y esto es obligatorio para guiar el medio de descubrimiento de caracteres faciales y de objetos[17].

A finales del año 2013, se estipula una técnica de protección que puede ser inspeccionado en cualquier sector de la tierra, esto lo permite actuar ya que está acoplado a internet, además, se ve la práctica de IoT, y esto genera progreso científico. Se utiliza una cámara inalámbrica, una sonaja y un sensor PIR, que son los dispositivos de recepción, mientras que los elementos de salida son; una reflector LCD y la mampara electromagnética, estos elementos, están interconectados a la Raspberry Pi y ésta está conectada a internet para gozar una mejor destreza e interés hacia internet de las cosas(IoT) por medio de la nube[18].

En su trabajo, Miranda plasma las facilidades ofrecidas por las tarjetas de adquisición de datos Raspberry y sus diversos módulos de integración, las cuales permiten la programación de dispositivos de manera más fácil e intuitiva. Adicionalmente, mediante el uso de sistemas de software libre, logra el diseño de una interfaz gráfica e interactiva que puede ser ejecutada en cualquier dispositivo[19].

En 2018, en el Instituto Politécnico Salesiana del Ecuador, desarrollaron un método de entrada usando ciencias aplicadas RfID e identificación del rostro. Tiene una base de datos donde se encuentran cerca de 310 imágenes almacenadas y asimismo lograr efectuar la comparación para al instante proyectar el alcance de la comprobación. Además, el hardware que se utiliza para este procedimiento es una Raspberry Pi 3 B+ y un lector de tarjetas de RfID, que es el encargado de mandar la información de la placa a la Raspberry y precisamente para conseguir el acceso del individuo[20].

2.3. Fundamentos Teóricos

2.3.1. Visión Artificial

La visión artificial es el compuesto de técnicas científicas que a través de una intervención computacional usando algoritmos de caracterización permite la toma, estudio y procesamiento de fotografías para conseguir información específica, la cual es procesada por un computador, semejante que los humanos los cuales obtienen indicaciones del medio a través de los ojos[5]. Los sistemas de visión artificial intentan formar un objeto equivalente obteniendo datos útiles de una o varias fotografías mediante un módulo electrónico programable que realice un autoridad de maquina. Los sistemas de visión artificial completan tareas de observación con un profundo grado de variabilidad y repetitividad; jamás existe agotamiento, desgana y descuido y pueden ocuparse en ambientes donde los individuos no podrían cumplir un correcto registro visual.

En un ente mortal, los ojos proporcionan al cerebelo explicación del círculo que lo rodea, apoyado en experiencias previas con cuerpos similares, interpreta la mutualidad entre las fotografías recibidas y es apto de obtener decisiones. De carácter relacionado, los sistemas de visión artificial ven al cuerpo a través de una cámara, interpretan y procesan la efigie mediante una práctica en un sistema computarizado. Las aplicaciones de visión artificial se encuentran principalmente en tareas de supervisión y ensamblaje, ya que estos son considerados trabajos repetitivos [21].

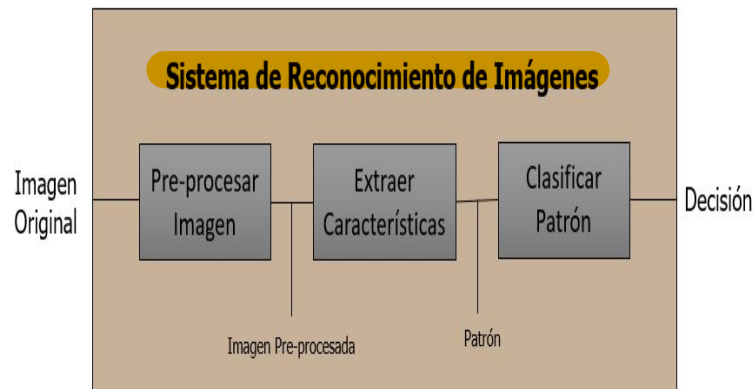


Figura 2.1: Método Secuencial Para Reconocimiento de Imágenes.

Fuente: Autor.

- **Ventajas de la Visión Artificial:** La visión artificial tiene obvias ventajas en tareas que se ocupa a grandes velocidades y en las que existe una profunda condición de reiteración, como representación en procesos de intervención visual de un perfil de ensamblaje que trabaje sin intermitencia. En procesos en los que se necesita ejecutar una verificación visual de grandes cantidades de objetos, los márgenes de desliz se ven reducidos debido a que se evitan factores externos los cuales son capaces de ejecutar actividades repetitivas con poder y velocidad, permitiendo la tipificación de detalles externamente del alcance del ojo humano, entre algunas ventajas tenemos: Mejoramiento de eficacia en tiempo, Reducción de costes generales, Superior lapso de utilización en correlación al individuo, Posibilidad de calcular con precisión características en las muestras, Bajos costos de implementación, Evita fallas Humanas, Aplicación en lugares inaccesibles[22].
- **Desventajas de la Visión Artificial:** Considerando la cabida sensorial de los ojos y el cerebro del ser humano, los modelos artificiales también poseen ciertas desventajas las cuales se evalúan en referencia al nivel de implementación de dicho

modelo, entre las cuales se puede indicar: Son sistemas en inspección y mejora, Susceptibles a problemas de luminosidad, Rapidez de deducción limitado a la del ser humano[22].

Componentes de un Sistema de Visión Artificial

La utilización de modelos de visión por computadora ha exagerado su inercia en la manufactura, en la cual su primordial función es en las tareas repetitivas y otras dificultosas para un especialista. Dichos sistemas de carácter básico poseen diferentes elementos como lo son: Fuente de luminosidad, cámara y prismáticos, placa de adquisición, computadora - procesador de visión y algoritmos para el procesamiento a ras de software como se evidencia a continuación[23].

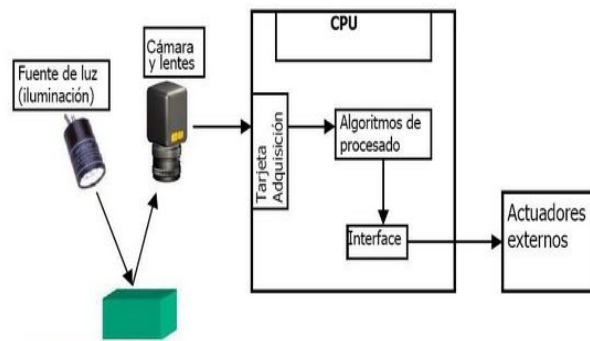


Figura 2.2: Componentes de un Sistema de Visión Artificial.

Fuente: Etitudela

- **Fuente de Luminosidad:** La luminosidad es alguno de los aspectos más críticos de éstas aplicaciones. Si el cuerpo no se ilumina adecuadamente, se puede extraviar información y rendimiento. Una habilidad de luminaria utilizada es una principio de claridad y su buena distancia con proporción a la sección y la cámara[23].
- **Cámara y Lentes:** Las cámaras y los prismáticos que se utilizan en visión artificial poseen una cadena de características específicas, como el dominio del tiro de la cámara para rescatar las piezas que pasan por al frente del dispositivo puntualmente en la posición requerida. Las cámaras de visión artificial son crecidamente sofisticadas que las convencionales, pues brindan un justo dominio de los tiempos, señales de video, rapidez de captura y susceptibilidad. Estos son factores fundamentales en modelos tecnológicos como en la manufactura. Este procedimiento puede exponer en un monitor la visualización de la ilustración, un

grabador de vídeo para almacenarla o un computador para visualizar, guardar y procesar la fotografía. Las cámaras de video tienen un rápido avance en las últimas décadas, desde las primeras cámaras hasta las actuales[23].

- **Computador Central y Procesador de Visión:** El computador es el red fundamental, está compuesto por hardware y software siendo fundamental al instante de tomar, examinar y procesar los datos a través de algoritmos, obteniendo un logro y excelente desempeño del sistema cumpliendo también los objetivos para los cuales fueron elaborado. El procesamiento de visión es el módulo para obtener información de una retrato digital y colocarla externamente en un sistema asentado en PC o internamente en un sistema de visión artificial independiente. El procesamiento se realiza a través de un software y conlleva varios pasos. Inicialmente, la fotografía se obtiene del sensor. En algunos casos es inevitable preprocesar la fotografía para optimizarla y garantizar que las características necesarias sobresalen. Inmediatamente, el software ubica las características específicas, realiza mediciones y las compara con la descripción. Posteriormente, se toma una disposición y se comunican los resultados. Mientras muchos de los componentes físicos de un sistema de visión artificial (como la luminosidad) ofrecen especificaciones similares, son los algoritmos del sistema de visión artificial lo que realmente los distingue[23].

Algoritmos para el Análisis y Reconocimiento de Imágenes

Los algoritmos para el determinación e identificación son técnicas de enfoque por computadora que varían en dependencia a los objetivos y técnicas que se utilizan, siguiendo un apropiado conducto[25].

- **Pre-Procesamiento(Filtrado):** El filtrado de una fotografía es la maniobra de iconografía que son adquiridas con un método de captura y las mismas no pueden ser utilizadas claramente adentro de un régimen de visión debido a que poseen cierta transición de rigor de la misma que es afectada por diversas fuentes como lo son: sonido, luminosidad y el escaso contraste al instante de su localización y desarrollo, haciéndose obligatorio la actuación de un pre-procesamiento con el único propósito de salvar dicha dificultad, ayudando igualmente a la tipificación de ciertas características específicas buscadas en las iconografía facilitando así la práctica de las mismas en etapas posteriores[25].
 - **Segmentación de la Imagen:** La segmentación de la fotografía es la división de las regiones y características de provecho de las fotografías una vez estas han sido capturadas y filtradas, y esto se logra dividiendo en dos regiones la condición para la localización de bordes y otros elementos en las mismas. La
-



Figura 2.3: Clasificación de un Sistema de Reconocimiento.

Fuente: Autor.

maniobra de segmentación adecuadamente depende de la condición y el arquetipo de información que se busque en dicha fotografía[25].

- **Extracción de Características:** La separación e identificación de características en una fotografía es la toma de medidas de los diferentes objetos estudiados una vez que fueron definidos, a estos se los extrae en carácter de vector[25].
- **Comparación y Resultado:** El proceso de comparación en muchas ocasiones depende de un algoritmo (Redes Neuronales) basado en rasgos predeterminados, con el fin de determinar la similitud que se busca para después con un porcentaje de apariencia poder tomar una decisión[25].

2.3.2. Inteligencia artificial (AI)

La inteligencia artificial es la rama de las ciencias de la computación que lidia con el estudio y el diseño de agentes inteligentes que perciben su entorno y toman decisiones para maximizar sus posibilidades de éxito [26]. Las características que poseen estos agentes son las que nosotros asociamos con el comportamiento inteligente de los seres humanos, características tales como percepción, procesamiento del lenguaje natural, resolución de problemas y planeación, aprendizaje y adaptación al entorno [27].

El machine learning es una de las áreas de estudio de la inteligencia artificial, área la cual ha estado en constante crecimiento a lo largo de los últimos años, usado comúnmente en tareas que requieren extraer información de grandes sets de datos

[28]. Su uso consiste principalmente en dos pasos, la primera fase donde se usan datos de entrada (como por ejemplo imágenes de gatos y perros para realizar una tarea de clasificación) para encontrar los parámetros que mejor solucionan la problemática, y la segunda fase, donde se usan los parámetros encontrados para realizar la tarea de clasificación, fase llamada también “inferencia”[29].

El deep learning es un subcampo del machine learning que basa sus modelos en componentes básicos llamados “neuronas”, las cuales están inspirados en las neuronas biológicas. Estas están organizadas en capas sucesivas y conectadas entre si, conexiones a las cuales se les atribuye un peso que es ajustado en la fase de aprendizaje. Cada neurona mapea su entrada a una salida con una función de transferencia, simulando el comportamiento de las neuronas del cerebro.

Después de la fase de aprendizaje, la red neuronal es capaz de separar los datos de entrada en jerarquías de características, representando múltiples capas de abstracción. Un ejemplo es el reconocimiento de rostros, la primera capa identifica patrones elementales como líneas, bordes y esquinas; y las capas siguientes se encargan de encontrar patrones mas grandes como labios, cejas y ojos. Existen dos tipos de redes neuronales especialmente conocidos, las redes neuronales convolucionales (especialmente para tareas de visión computacional) y las redes neuronales recurrentes (usadas en aplicaciones de procesamiento de lenguaje) [29].

Historia de las redes neuronales artificiales

Sus inicios se remontan a 1957, cuando Frank Rosenblatt comenzó sus desarrollos en el perceptrón, la red neuronal más antigua. Un modelo capaz de predecir patrones similares a los cuales se les había presentado en su entrenamiento, pero al ser un modelo tan simple, no era capaz de resolver problemas no linealmente separables como la función XOR (OR-exclusiva).

Luego en 1960, Bernard Widrow y Marcian Hoff hicieron la primera implementación exitosa de una red neuronal a un problema de la vida real con su modelo Adaline (ADAPtative LINear Elements), usado como filtro para la eliminación de ecos en líneas telefónicas. Las redes neuronales artificiales tuvieron un fuerte declive en 1969 cuando Marvin Minsky y Seymour Papert probaron matemáticamente que el perceptrón no era capaz de resolver problemas no linealmente separables, los cuales eran relativamente fáciles, demostrando que el perceptrón era débil dado que la no-linealidad está ampliamente presente en problemas de la vida real.

El renacimiento de las neuro-redes lo trajeron John Hopfield en 1985 con su libro “Computación neuronal de decisiones en problemas de optimización” y David Rumelhart junto con G. Hinton, quienes redescubrieron el algoritmo de propagación hacia atrás (backpropagation) usado hasta el día de hoy para el entrenamiento de redes neuronales [30].

Redes Neuronales Artificiales

“Las redes neuronales artificiales (RNA) surgen como un intento para emular el funcionamiento de las neuronas de nuestro cerebro” [31]. Estas neuro-redes están enfocadas en modelar la forma de procesamiento de la información en sistemas nerviosos biológicos, principalmente basándose en el funcionamiento del cerebro humano, el cual es un sistema altamente complejo, capaz de realizar muchas operaciones de forma paralela, a diferencia de las computadoras convencionales, las cuales son de tipo secuencial o una operación a la vez.

Una red neuronal se puede describir como un procesador de información, constituido por unidades sencillas e interconectadas llamadas neuronas, además es capaz de almacenar conocimiento a través de la experiencia, tal como lo hace el cerebro humano, y tener un comportamiento no lineal, teniendo la capacidad de procesar información que presente una no-linealidad.

Las neuronas artificiales reciben entradas las cuales son multiplicadas por los pesos asociados a cada conexión, dichos pesos indican la importancia de cada entrada de la neurona, posteriormente se realiza la suma de todas estas multiplicaciones, el resultado es evaluado en una función de activación la cual puede ser del tipo escalón, sigmoideal, entre otras, siendo el valor devuelto por esta función, la salida de la neurona [32].

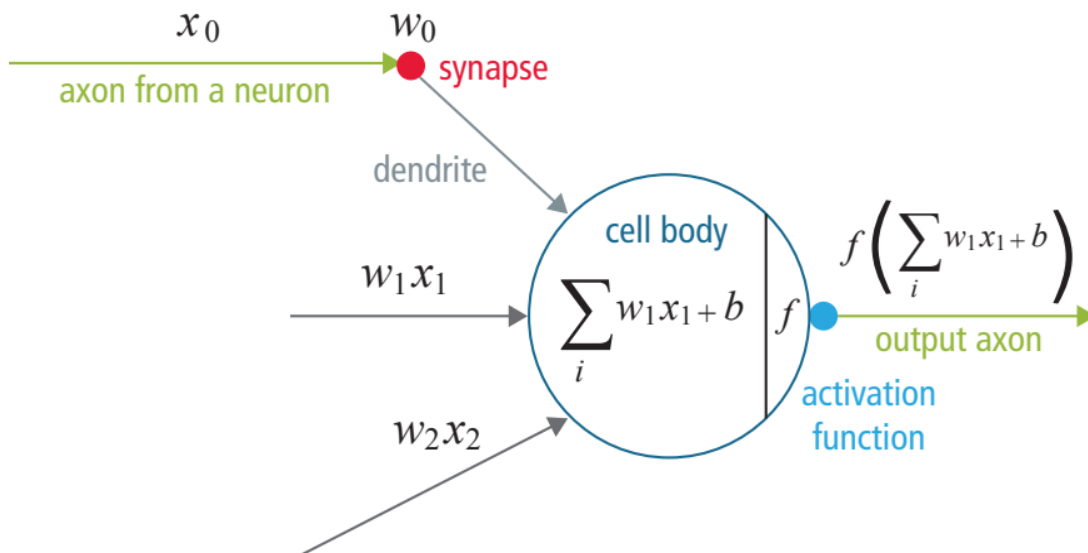


Figura 2.4: Esquema de una neurona artificial.

Fuente: Pigou and Sander

Dentro del cerebro se presenta una red de células (neuronas) las cuales presentan una gran cantidad de conexiones entre sí, la información que recibe cada uno de estos componentes a través de las dendritas, pasa a través del soma el cuales el órgano de cómputo y va al axón, para transmitir la información de salida a otras unidades a las que esté conectada, se estima que en el cerebro hay alrededor de cien mil millones de neuronas.

La unión entre 2 neuronas o más recibe el nombre de sinapsis, esta conexión es uni-direccional. Cada neurona recibe impulsos eléctricos a través de las dendritas y éstas están a su vez conectadas a otras para producir la sinapsis [34].

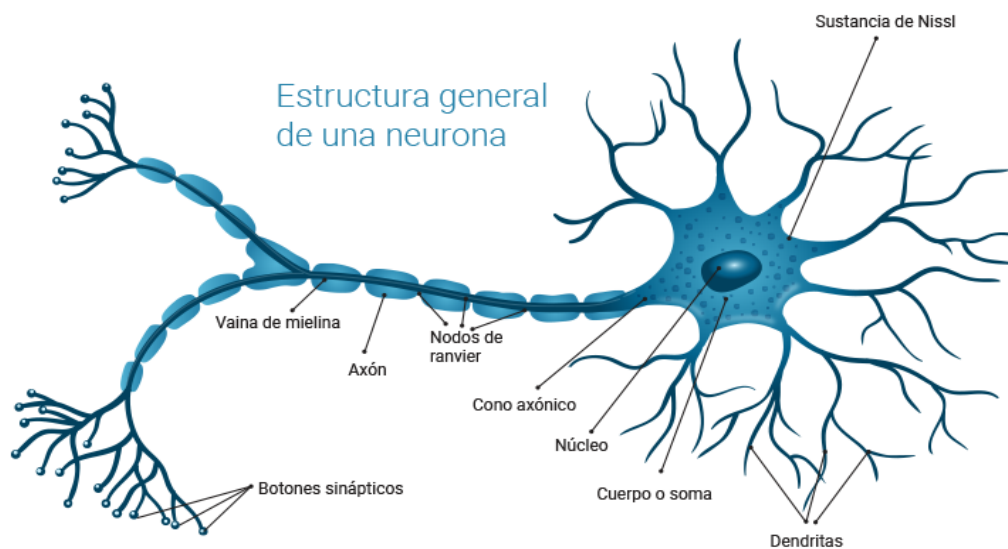


Figura 2.5: Estructura de una neurona biológica.

Fuente: Campos

Estructura de una Red Neuronal Artificial

Como se ha mencionado anteriormente, una red neuronal artificial está compuesta por elementos simples, llamados neuronas; tenemos varias de estas unidades ordenadas por capas, la capa de entrada, la cual no realiza ningún proceso, su única función es entregar los valores de entrada a los siguientes nodos; las capas ocultas, reciben los valores de entrada y se ocupan de proporcionar mayor complejidad a la red, permitiendo un mejor aprendizaje, éstas capas pueden o no estar presentes en una red, dependiendo de la topología escogida; por último la capa de salida de encarga de proporcionar la salida del sistema.

Ventajas de las Redes Neuronales Artificiales

- Cada una de las neuronas realiza un procesamiento, dependiendo de las entradas y pesos asociados a estas, además de la función de activación asignada. Dicho procesamiento se realiza de manera paralela y proporcionan una respuesta al mismo tiempo.
- Los pesos sinápticos son ajustados usando reglas de aprendizaje como el algoritmo de backpropagation, enseñando a la red lo que necesita para funcionar correctamente.
- Las redes neuronales son tolerantes a fallos, pueden seguir operando si parte de la red deja de funcionar, solo dejará de funcionar para los patrones en los que dicha región desempeñaba un papel importante.
- Las redes neuronales tienen la capacidad de predecir patrones que no han sido mostrados anteriormente (patrones que no fueron dados a la red en el periodo de entrenamiento). El único requisito es que el nuevo patrón sea similar a los de entrenamiento.
- La velocidad de respuesta de las redes neuronales es casi inmediata una vez que han sido entrenadas [36].

Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales han tenido gran protagonismo en la última década en gran variedad de campos relacionados con el reconocimiento de patrones, desde reconocimiento de imágenes hasta reconocimiento de voz. Debido al reducido número de parámetros requeridos, los desarrolladores se han enfocado en aumentar el tamaño de los modelos para dar solución a tareas complejas que no eran posibles con las redes neuronales artificiales clásicas.

Otro aspecto importante acerca de las CNNs es obtener características abstractas cuando la entrada se propaga a las capas más profundas. Por ejemplo, en la clasificación de imágenes, los bordes son detectados por las primeras capas, las formas simples son detectadas en las capas secundarias y las características de alto nivel como las caras, son detectadas en las capas más profundas como se puede ver en la figura 2.6[37].

Estructura de una CNN

La arquitectura de una CNN está compuesta por varios bloques como capas de convolución (convolutional layers), capas de agrupación (pooling layers) y capas conectadas. Típicamente una CNN consiste en la repetición de bloques con varias capas de convolución y capas de agrupación seguidos de una o más capas conectadas [38].

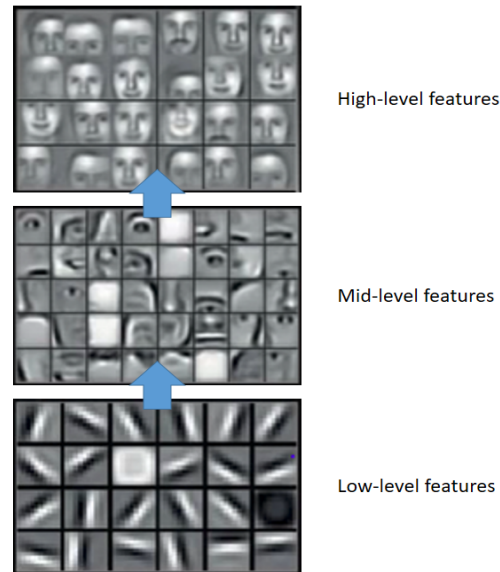


Figura 2.6: Características aprendidas de una red neuronal convolucional.

Fuente: Albawi

Convolutional Layer

Las capas de convolución son los elementos principales en una CNN que desarrolla la extracción de patrones, que consiste en una combinación de operaciones lineales y no lineales, que son las convoluciones y las funciones de activación.

Una Convolución es un tipo de operación lineal usada para extracción de características, donde un arreglo de números llamado kernel, es aplicado a la entrada, que es un arreglo de números llamado tensor. El valor de una posición dada del tensor de salida, es obtenido mediante la suma de los productos de cada uno de los elementos del kernel y cada una de las posiciones del tensor de entrada, a este tensor de salida se le llama mapa de características (ver figura 2.5). El proceso se repite múltiples veces para obtener varios mapas de características, difiriendo cada uno por el kernel usado, estos varían de tamaño, los más comunes son 3×3 , 5×5 y 7×7 . La operación de convolución está definida por el tamaño del kernel y la profundidad del mapa de características [38].

Non-linear Layers

Las salidas de operaciones lineales como las convoluciones son luego pasadas a funciones de activación no lineales. La capa ReLU es una de las más usadas ya que tiene la ventaja de entrenar las CNN mucho más rápido. En ReLU se implementa la función $y = \max(x, 0)$, el tamaño de entrada y salida son el mismo. Esta incrementa

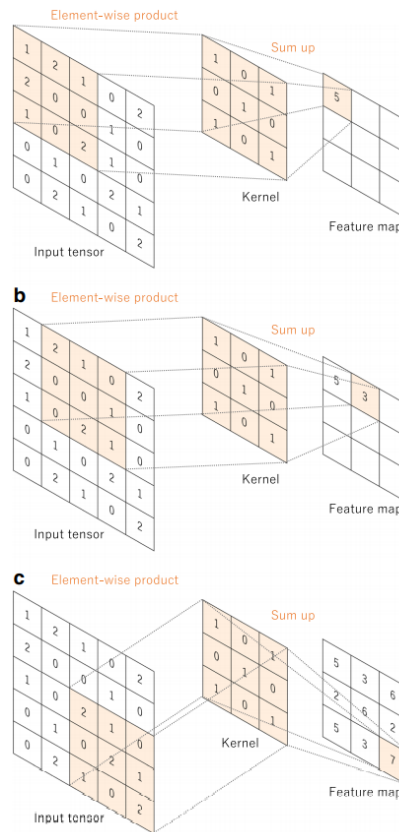


Figura 2.7: Ejemplo de una operación de convolución con un kernel de tamaño 3×3 , con desplazamiento de valor 1.

Fuente: Yamashita

las propiedades no lineales de la función de decisión y de la red en general sin afectar los campos receptivos de la capa convolucional [33].

Pooling Layers

Las capas de agrupación (Pooling layers), se encargan de reducir la resolución de los mapas de características, haciéndolos robustos frente a ruido y distorsión. Hay dos tipos de estas capas, agrupación máxima (Max pooling) y agrupación promedio (Average pooling). La figura 2.7 muestra el proceso de agrupación que realizan estos dos tipos de capas, la entrada tiene un tamaño de 4×4 , y la salida en ambos casos es de 2×2 . La entrada es dividida en matrices de tamaño 2×2 . En el caso de max pooling, se toma cada una de las subdivisiones de la entrada y se toma el valor máximo de los 4 valores y en el caso de average pooling, el resultado es el promedio entre los 4 valores del arreglo [33].

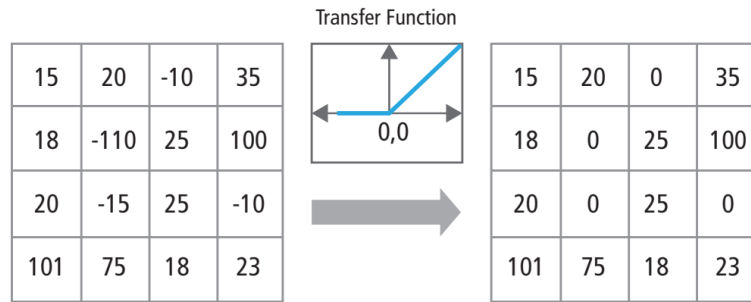


Figura 2.8: Representación de la funcionalidad de ReLU.

Fuente: Pigou and Sander

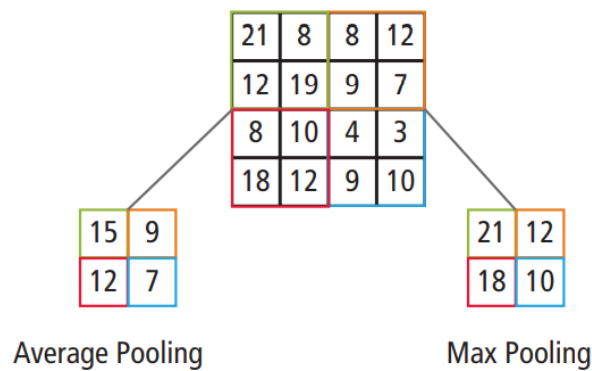


Figura 2.9: Representación del max pooling y el average pooling.

Fuente: Pigou and Sander

Fully-Connected Layers

Las capas completamente conectadas (Fully connected layers) son la última estación en la topología de una CC, consiste en un arreglo multicapa de neuronas (ver figura 2.8) [39]. Esta capa tiene conexiones con todas las activaciones de la capa previa. Sus activaciones se computan con operaciones de matrices teniendo en cuenta el desplazamiento “bias” [40].

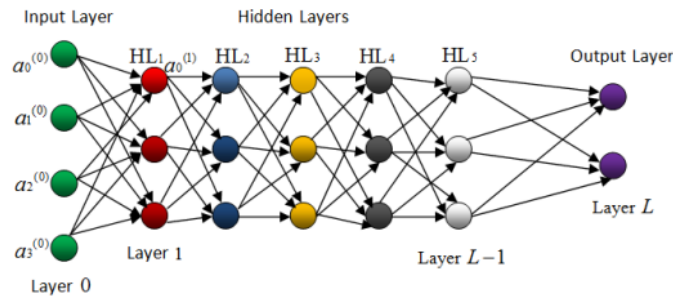


Figura 2.10: Fully-connected layer

Fuente: Sakib2018

MobileNet

La MobileNet es un modelo creado especialmente para aplicaciones de visión artificial para teléfonos móviles y sistemas embebidos, por lo tanto el modelo no requiere de altas prestaciones en cuanto a hardware para realizar los cálculos de forma rápida y en tiempo real.

La arquitectura usa convoluciones separables en profundidad el cual es un tipo de convolución mas rápida que las usada en otras arquitecturas.

MobileNet puede ser usada en varias tareas de reconocimiento de imágenes tales como reconocimiento de puntos de referencia, detección de objetos (MobileNet puede clasificar hasta 1000 objetos) y reconocimiento de rostros[41].



Figura 2.11: El modelo MobileNet puede ser aplicado a varias tareas de reconocimiento de forma eficiente.

Fuente: Howard.

'You Only Look Once (YOLO)

YOLO es una arquitectura capaz de detectar objetos de distintas clases en una imagen "viéndola" una sola vez, lo que le permite ser más rápida que otras arquitecturas desarrolladas anteriormente. La rapidez de YOLO le permite detectar objetos en tiempo real, llegando hasta los 30 FPS a un pequeño costo de exactitud.

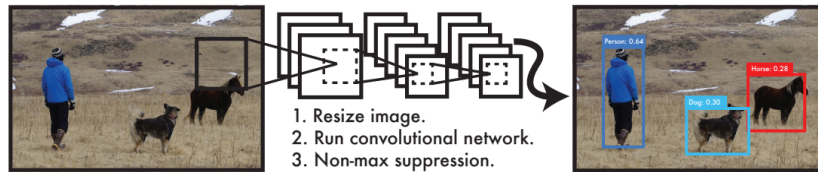


Figura 2.12: El sistema de detección YOLO.

Fuente: Redmon

La detección de objetos se lleva a cabo dividiendo la imagen de entrada con una cuadrícula de cierto tamaño S . Posteriormente, para cada una de las celdas se predicen N cuadros delimitadores y se calcula la probabilidad para cada uno de ellos. Después se eliminan los cuadros que están por debajo de un límite de probabilidad definido y de los restantes se toman los que tengan la máxima probabilidad con el fin de eliminar varios cuadros que estén sobre el mismo elemento, dejando el que lo encierre de la manera más exacta. El proceso puede visualizarse de forma gráfica con la figura 2.13[42].

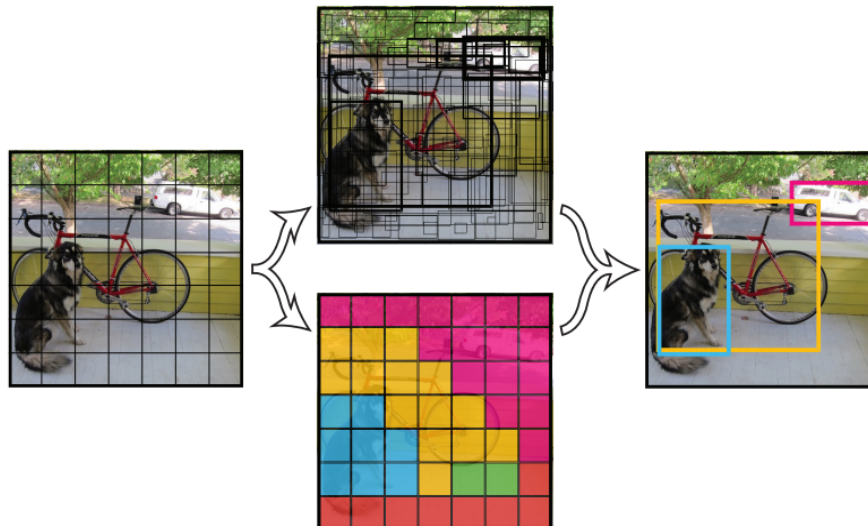


Figura 2.13: Imagen con grilla de entrada, cuadros delimitadores, mapa de probabilidades y detección del algoritmo YOLO.

Fuente: Redmon

2.3.3. Reconocimiento de Patrones

Con el avance de los seres humanos, como otros organismos, han desarrollado importantes mecanismos y habilidades para obtener la información del espacio y a partir de ello adquirir decisiones. La medida de inspeccionar diferentes situaciones y obtener conclusión pertinente es un medio esencial de la humanidad, aparentemente este trabajo se aprecia como una sencilla labor cotidiana, sin embargo implica un transcurso cognoscitivo dificultoso.

Los humanos obtienen la información del espacio mediante de los sentidos como la vista, el tacto, el gusto, el olfato y la escucha; en proporción los sistemas inteligentes la obtienen a mediante sensores y presentan los datos en representación idóneo para su procesamiento e inferencia en ordenadores. Los datos registrados son llamados patrones y éstos pueden ser representados como señales, fotografías o tablas de datos[43].

Una descripción puntual de Identificación de Patrones es la consecuente: Es la categorización de datos de ingreso en clases identificadas, intermedio de la separación de características significativas o atributos de los datos extraídos de un espacio que contiene detalles irrelevantes. Matemáticamente hablando, la categorización consiste en el fraccionamiento del área n-dimensional determinado por las características de un cuerpo, en varias regiones, en el cual cada zona corresponde a un curso[44].

Conceptos Básicos del Reconocimiento de Patrones

- **Patrones:** Un patrón es la idea que puede ser identificable de acuerdo a sus particularidades o propiedad, éste puede ser genérico o definido. Los patrones genéricos son ideas abstractas mientras que un patrón definido es la escritura mecánica de un cuerpo, por ejemplo los símbolos (letras, caligrafías), fotografías digitales, fotografías tridimensionales, firmas, señales de audio, etc[45].
 - **Clases:** Un género es un etapa o clase de los patrones. Los patrones que pertenecen al mismo género comparten los mismos emblemas y conductas. En un dirección genérica, una clase, se puede explicar como una muestra o modelo que idealiza un cuerpo. El conjunto de todas los géneros se le conoce como área de paráfrasis[45].
 - **Características:** Las particularidades describen a los patrones, de acuerdo al importe de sus emblemas, y determinan el área de características de un género[45].
 - **Similitud:** La conciencia de similitud, es el fragmento abierto en el transcurso del RP, se refiere a los valores parecidos de un emblema en dos o más cuerpos. En el transcurso de identificación, se evalúa la semejanza entre un cuerpo y un tipo que idealiza al género al que compete[45].
-

Paradigmas del Reconocimiento de Patrones

En el problema del Reconocimiento de Patrones existen diferentes paradigmas para la solución de un problema, aunque no existe un consenso de como categorizarlos, una de las aproximaciones se describe en el siguiente esquema[46]:



Figura 2.14: Paradigmas del Reconocimiento de Patrones.

Fuente: Autor

Éstos modelos se basan según su maniobra en uno de los enfoques de investigación de similitud, las cuales se les conoce como supuesto. Las hipótesis pueden ser obtenidas de dos maneras: ilustración supervisada e ilustración no supervisado.

- **Aprendizaje Supervisado o Hipótesis Inductiva:** Define un conjunto de funciones a partir de datos de entrenamiento o clases previamente definidas. Un patrón desconocido es evaluado en las funciones y de acuerdo al resultado es clasificado en uno o más clases[46].
- **Aprendizaje no Supervisado o Hipótesis Deductiva:** En este tipo de aprendizaje no existe ninguna clasificación previa del objeto y en algunas ocasiones ni siquiera se han definido las clases[46].

Diseño de un Sistema de Reconocimiento de Patrones.

Un método de identificación de patrones, independientemente de la norma que implementa, está emparejado por distintos modelos que operan de modo sistemático referente a los patrones. El sucesivo bosquejo corresponde a la distribución de un método propio de un sistema de identificación de patrones[46].

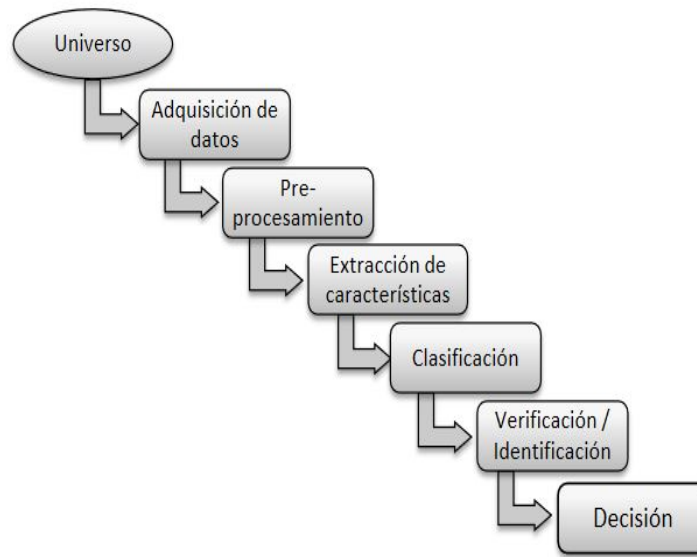


Figura 2.15: Esquema general de un Sistema de Reconocimiento de Patrones.

Fuente: Autor

2.3.4. Inmótica

La palabra inmótica es una variación de la domótica que está compuesta por domo proveniente del latín *domus* cuyo significado es casa, y, tica, que algunos la relacionan con automática proveniente del griego y otros con la palabra informática (De Russis, 2011). Aunque no exista una definición exacta de la inmótica el concepto en general se relaciona con el funcionamiento de un edificio o ambiente automatizado. Por este motivo la inmótica es una tecnología inteligente escalable a diferentes entornos (Bticino, 2012).

El término inmótica representa o se utiliza para denominar varios subconceptos relacionados a esta tecnología por ejemplo, la fusión entre la electrónica y la informática para integrar el control y supervisión de dispositivos existentes en un ambiente laboral, como edificios y oficina "Smart Environment", o también, como para cualquier tipo de viviendas donde se utiliza el nombre de "Home Automation" (Ryan, 1988). Otros, al hablar de inmótica lo relacionan con el término de edificio inteligente, un término que tiende a ser aplicado en el ámbito de los grandes bloques de oficinas, bancos, universidades y edificios industriales (Delgado, Moreno, Véliz, Leyton, 2007).

Si bien es cierto, el término inmótica engloba de manera general la posibilidad de automatizar procesos habituales y tener control sobre ellos, la misma como tal no logró desarrollarse fiamente hasta la introducción gradual y acelerada de la informática y las telecomunicaciones en ambientes frecuentes para la gente común (Sandetel, 2011).

La introducción de las TIC's, en procesos de automatización es la que ahora permite dotar de "inteligencia" a cualquier ambiente que el usuario desee.

En la planificación de un sistema inmótico se deben considerar los siguientes aspectos o áreas donde se espera optimizar los recursos (Gómez Paradells, 2010): disminuir los costos energéticos mediante la automatización, adquirir información para toma de decisiones en el área de seguridad, optimizar el confort utilizando el control de climatización e iluminación, integración con la nube y nuevas tecnologías y monitorización en diferentes medios y tiempo real de las instalaciones.

Esto significa que desde un punto de vista general, los sistemas inteligentes, con el fin de lograr los objetivos de automatización, requieren tener acceso y controlar las siguientes áreas críticas de la infraestructura del ambiente donde se instalará: sistema de iluminación, sistema de climatización, el suministro de energía, accesos a los locales, circuitos de video, sensores de incendio, sistema de seguridad accesos y antirrobo y seguridad informática (Bticino, 2012; Junestrud, Assare, Vasquez, 2005).

2.3.5. Sistema Embebido - Raspberry Pi

Con el deseo de compartir la diversión de la informática y la necesidad de que más personas conocieran cómo funcionaban las computadoras, Eben Upton creó una computadora pequeña y barata. Con la ayuda del Dr. Rob Mullins, el profesor Alan Mycroft y Jack Lang de la Universidad de Cambridge; Pete Lomas, un experto en hardware; y David Braben, la Fundación Raspberry Pi, una organización benéfica que se propuso promover el estudio de la informática y temas relacionados, especialmente a nivel de escuela, y poner de nuevo la diversión en el aprendizaje de la informática. La Fundación Raspberry Pi tenía como objetivo abrir el mundo de la informática mediante la creación de un dispositivo de hardware que era asequible al bolsillo, por ende era accesible a todo el mundo, y no habría necesidad de preocuparse por experimentar con ella. Estaba sin caja para que sea fácil de jugar[47].

En 2011, después de cinco años de intensa ingeniería, el primer prototipo de computadora Raspberry Pi se produjo. Después de un artículo sobre el blog de Rory Cellan-Jones, el periodista de tecnología Rory Cellan-Jones se volvió viral, la Fundación se preguntó si estaban en las primeras etapas de algo más grande de lo que esperaban. Después de una intensa ingeniería se realizó una placa reducida para permitir que la Raspberry Pi se construya lo suficientemente barato para ser vendido por 25 euros, un lote inicial de 10.000 salió a la venta el 29 de febrero de 2012 a las 6 am algunos minutos más tarde, se habían agotado. dieciocho meses después, 1,75 millones habían sido vendidos en todo el mundo (Robinson Cook, 2014).

La Tarjeta Raspberry Pi viene con algunos puertos de salida tales como HDMI para hacer uso del monitor, puertos USB para conectar dispositivos de memoria portátil,

mouse, teclado; puerto de audio, led indicadores de encendido de la tarjeta, puerto de alimentación, puerto para la tarjeta SD; los puertos GPIO que pueden ser utilizados para conectar diversos actuadores, además cuenta con Bluetooth 4.1, Bluetooth Low Energy y 802.11n Wireless Lan (Wi-fi)[47].(Ver Tabla 2.1)

2.3.6. Transfer Learning

Existen diferentes estrategias y técnicas de transfer learning que pueden aplicarse en función del dominio, la tarea en cuestión y la disponibilidad de datos. Los métodos de transfer learning se pueden clasificar según el tipo de algoritmos tradicionales de machine learning involucrados, como:

- ***Inductive Transfer learning:*** En este escenario, los dominios de origen y destino son los mismos, pero las tareas de origen y destino son diferentes entre sí. Los algoritmos intentan utilizar los sesgos inductivos del dominio de origen para ayudar a mejorar la tarea de destino. Dependiendo de si el dominio de origen contiene datos etiquetados o no, esto se puede dividir en dos subcategorías, similar al aprendizaje multitarea y el aprendizaje autodidacta, respectivamente[48].
- ***Unsupervised Transfer Learning:*** Esta configuración es similar a la transferencia inductiva en sí misma, con un enfoque en tareas no supervisadas en el dominio de destino. Los dominios de origen y destino son similares, pero las tareas son diferentes. En este escenario, los datos etiquetados no están disponibles en ninguno de los dominios[48].
- ***Transductive Transfer Learning:*** Hay similitudes entre las tareas de origen y de destino, pero los dominios correspondientes son diferentes. En esta configuración, el dominio de origen tiene muchos datos etiquetados, mientras que el dominio de destino no tiene ninguno. Esto puede clasificarse aún más en subcategorías, en referencia a configuraciones donde los espacios de características son diferentes o las probabilidades marginales[48].

Modelos Pre-entrenados

Un modelo pre-entrenado es un modelo creado por alguien más y entrenado con abundantes datos para resolver un problema similar. En lugar de construir un modelo desde cero para resolver un problema similar, utiliza el modelo entrenado en otro problema como punto de partida. Por ejemplo, si se desea construir un sistema de reconocimiento de plantas. Puede dedicar meses a construir un algoritmo de reconocimiento de imagen aceptable desde cero o puede tomar un modelo de inicio (un modelo pre-entrenado) de Google que se construyó sobre datos de ImageNet para identificar imágenes en esas fotos. Un modelo pre-entrenado puede no ser 100% preciso en su aplicación, pero ahorra enormes esfuerzos necesarios para reinventar los mismos conceptos una y otra vez[48].

El objetivo de entrenar una red neuronal es identificar los pesos correctos para la red mediante múltiples iteraciones hacia adelante y hacia atrás. Al usar modelos pre-entrenados que han sido entrenados previamente en grandes conjuntos de datos, podemos usar directamente los pesos y la arquitectura obtenidos y aplicar el aprendizaje en nuestra declaración del problema. Esto se conoce en inglés como Transfer Learning. Transferimos el aprendizaje del modelo pre-entrenado a nuestro enunciado del problema específico[48].

2.4. Síntesis

En el primer capítulo de esta Tesis, enunciábamos como propósito principal el estudio de Visión Artificial, ya que, con ello se puede realizar múltiples trabajos. De éste tema, surge el reconocimiento facial, para poder llevarlo a cabo se deben integrar factores como son las redes neuronales, es un medio que nos facilita todo el análisis del mismo para otorgarnos una respuesta del patrón que se busca. Partiendo de todo eso se concluye el capítulo de marco teórico como un análisis bibliográfico de antecedentes que involucren la misma iniciativa y definiciones sobre temas particulares para el apoyo del mismo.

Componentes	Descripción
Procesador	Chipset Broadcom BCM28387RIFBG y 1,2 GHz de cuatro núcleos ARM Cortex-A53
GPU	Dual Core VideoCore IV, Multimedia Co-procesador. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264 de alto perfil de decodificación.
RAM	1GB LPDDR2.
Conectividad	Ethernet socket Ethernet 10/100 BaseT, Salida de vídeo, HDMI rev 1.3 y 1.4, Salida de audio, USB 4 x Conector USB 2.0, 40-clavijas de 2,54 mm (100 milésimas de pulgada) de expansión: 2x20 tira

Tabla 2.1: Especificaciones Técnicas de la Raspberry Pi

Fuente: Autor.

3. Metodología

3.1. Introducción

Partiendo del compromiso social que tiene cada individuo con su entorno y el aportar una nueva estrategia para la reducción de la propagación del presente virus surge la iniciativa de éste trabajo de grado. Por consiguiente, se debe indagar sobre la temática expuesta para generar un buen estado del arte; siendo específico se hace incapié en temas como visión artificial (Reconominento Facial), redes neuronales convolucionales, aspectos de programación y página web. De acuerdo con lo anterior se puede plantear una estrategia de desarrollo del proyecto que tenga un impacto social considerable logrando los objetivos principales del mismo. Para éste trabajo de investigación se realiza un estudio con la finalidad de responder a los objetivos establecidos y dando un enfoque mixto como se aprecia en el siguiente esquema.

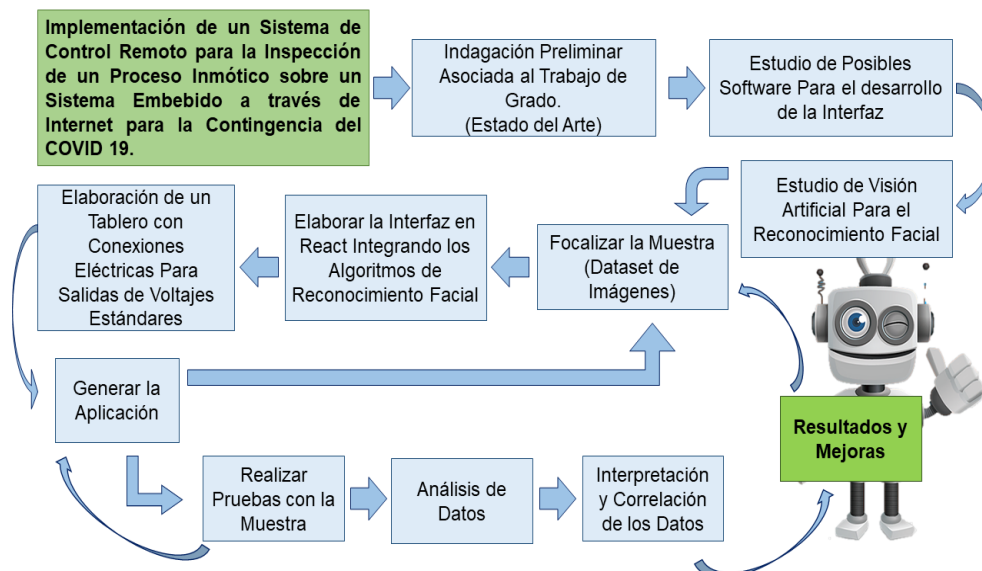


Figura 3.1: Metodología y Fase de Investigación.

Fuente: Autor

3.2. Metodología y Fase de Investigación

Indagación Preliminar Asociada al Trabajo de Grado (Estado del Arte)

Este es uno de los pasos fundamentales de la investigación, es el punto de partida del estudio, ya que, se aplica un estudio correlacional, es decir, relacionar los resultados obtenidos por estudios anteriores con los resultados a obtener. Los estudios en los que se referencia la investigación están en el apartado del Estado del arte, es aquí donde vamos a discernir las variables que se pueden identificar en nuestro trabajo, ya que, pueden ser cualitativas y cuantitativas convirtiéndolo en un enfoque mixto.

Estudio de Posibles Software para el Desarrollo de la Interfaz

Al comenzar con este proyecto nos encontramos con un factor de programación de alto grado. Esta fase se incluye para tener un contexto de los diversos programas o extensiones de los mismos que se implementan para todo el proceso. Además, la finalidad es encontrar un software que se mezcle a las habilidades de programación del tesista, con ello, se puede avanzar más rápido en ciertos aspectos y detenerse en otros.

Estudio de Visión Artificial Para el Reconocimiento Facial

Cómo lo había mencionado en la fase anterior, éste trabajo de grado tiene un cierto nivel de programación y no sólo en la interfaz para la interacción sistema - usuario, sino, en el reconocimiento facial para detectar el uso de tapabocas (Mascarillas). Partiendo de todo esto, se busca un lenguaje de programación ameno a las habilidades del tesista con el fin de solucionar éste ítem en el trabajo de investigación.

Focalizar la Muestra (Dataset de Imágenes)

En esta etapa del proyecto se hace una revisión de la población específica a la cual será dirigida el proyecto, teniendo en cuenta los diferentes aspectos que puedan presentar los grupos. Teniendo la proyección de las posibles problemáticas que se puedan presentar para el correcto reconocimiento facial como: Personas con alguna prenda que cubra la cara, el uso de bufandas, etc. Ante esto se focaliza la muestra a un dataset de imágenes para poder trabajar sobre una población determinada y obtener resultados idóneos.

Elaborar la Interfaz en React Integrando los Algoritmos de Reconocimiento Facial

Una vez terminada la fase de investigación y estado del arte, se procede al desarrollo del mecanismo. Sabiendo cual es la muestra y encontrando un lenguaje de programación que nos ayude con esto, se trata de integrar los conceptos. Siendo específico, React, es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página; con éste software

se desarrollará toda la parte visual y de interacción con el proceso. Además, Python se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional; con éste software se desarrollará la aplicación de reconocimiento facial, el cual, conlleva el uso de redes neuronales. Al finalizar las etapas de programación de forma independiente, se busca la integración de los mismos para la culminación del proyecto.

Elaboración de un Tablero con Conexiones Eléctricas para Salidas de Voltajes Estándares

Los tableros eléctricos son los encargados de resguardar los componentes o dispositivos de mando y de control de cualquier régimen eléctrico a partir un circuito primordial en un domicilio incluso el de una maquina manufacturera. En estos se puede reunir los dispositivos de conexión, maniobra, protección, etc.

Generar la Aplicación

La generación de la aplicación consiste en plasmar de forma física todo el proceso de trabajo. Este puede llegar a ser una de las fases más arduas y difíciles de la investigación, ya que, puede tomar tiempo para la implementación, como para conseguir los materiales requeridos para el correcto funcionamiento del prototipo.

Realizar Pruebas con la Muestra

En este apartado se lleva a cabo los ensayos técnicos del proceso obtenido de la investigación (en sus primeras etapas ya que este requiere de mejoras continuas) donde se evalúan parámetros de compatibilidad con el entorno y los participantes, propiedades de los materiales, entre otros. Haciendo un enfoque prioritario a la respuesta que tiene el sistema con los objetivos planteados al inicio del trabajo de grado, y de esta manera avanzar o tomar las medidas necesarias para replantear la aplicación generada o continuar con el proceso.

Análisis de Datos

Se analizarán las variables de enfoque mixto que se seleccionen de estudios anteriores para contar con un criterio basado en los resultados obtenidos de investigaciones similares que sean relevantes en el tema, con esto se pueden establecer mejoras.

Interpretación de los Datos

Este paso es donde se toman e interpretan los datos obtenidos, es decir las variables las cuales soportan los resultados de la investigación.

Resultados y Mejoras

En esta sección del trabajo de grado se espera obtener los resultados acordes a los objetivos anteriormente planteados. Además, se hace una revisión técnica y digital para identificar su viabilidad, logrando concluir el proyecto y plantear mejoras futuras. Se presentarán las conclusiones más relevantes que pueden ser consideradas como contribución a la comunidad científica para estudios similares y posteriores.

3.3. Síntesis

El capítulo de Metodología, señala un orden a implementar para lograr con los objetivos expuestos al inicio del libro. Además nos identifica que tipo de investigación se debe seguir, como también las actividades necesarias para el desarrollo de la propuesta de forma específica.

4. Desarrollo de la Propuesta

4.1. Introducción

Éste capítulo de investigación y desarrollo puede otorgarse como uno de los capítulos más completos y largos durante el proceso de implementación, ya que, de forma específica se busca darle solución a cada aspecto que conforman a las actividades de la metodología. En primer lugar se hablará de la composición del dataset de imágenes de personas con uso o no de mascarillas, de éste aspecto se deriva el resto de actividades a seguir. Teniendo un dataset como el que se explicará a continuación, nos libera de una limitante y específica la población con la que se trabaja para el desarrollo de los códigos de programación. Por consiguiente, al tener la base de datos de imágenes con la que se va a trabajar, se procede al aplicar técnicas de reconocimiento facial involucrando las redes neuronales como entes de aprendizaje y respuesta del mismo; teniendo éstas actividades culminadas, se puede continuar con el proceso, el cual, sería la implementación de una interfaz WEB para interacción Sistema - Usuario y la programación del sistema embebido como controlador de activación o desactivación de los elementos finales de control.

4.2. Composición del Dataset de Imágenes

La notación para la detección del empleo de la mascarilla en tiempo real, está elaborado utilizando herramientas de ilustración automático intrínsecamente de una norma que se conoce como Visión por Ordenador. La finalidad es que la utilización que contiene este algoritmo pueda divisar si una persona lleva o no la mascarilla en uso. Muestra que podría resaltar para nuestro cerebro una labor simple, reviste de alguna pluralidad, y se hace utilizando unos métodos conocidos como Aprendizaje Profundo (Deep Learning) en el cual resaltan las CNN. En límite de procesos inicialmente se le enseña a nuestro algoritmo a que “aprenda” (Inteligencia Artificial) y logre diferenciar si lo que ve es un individuo y en seguida si un individuo lleva o no el uso de mascarilla. ¿Cómo se hace? Esto se hace con fotografías de personas con y sin tapabocas, posteriormente de ver muchas veces las fotografías nuestro método habrá capturado las particularidades de las fotografías. Se hace vigor en la utilización de un algoritmo de ilustración maquina supervisado y por consiguiente nuestra red neuronal aprenderá de retratos previamente etiquetadas en el cual ya se han clasificado “de previo” las particularidades que a futuro vamos a exigir que la notación reconozca.

Este conjunto de datos consta de 3835 imágenes que pertenecen a dos clases:

- **CON Mascarrilla o Tapabocas:** 1916 imágenes.
- **SIN Mascarrilla o Tapabocas:** 1919 imágenes.

Las imágenes utilizadas fueron imágenes reales de rostros con y sin mascarilla. Las imágenes fueron recolectadas de las siguientes fuentes:

- API de Bing Search con este script se recolectan las imágenes de las web directamente, en este caso con el buscador de Microsoft “BING”
- Conjuntos de datos de Kaggle



Figura 4.1: Imágenes CON Mascarrilla del Dataset.

Fuente: Kaggle



Figura 4.2: Imágenes SIN Mascarilla del Dataset.

Fuente: Kaggle

4.3. Estructura de la Red Neuronal a Implementar

En esta sección se describe el entrenamiento de la red neuronal convolucional a cargar en la tarjeta Raspberry Pi. Además se describen los algoritmos para la implementación de esta red, la detección de rostros que se realiza con un modelo ya entrenado ofrecido por librerías de Python.

Para el entrenamiento de la red neuronal se usó la librería Keras, por la simplicidad que brinda a la hora de crear y entrenar modelos.

4.3.1. Arquitectura de la CNN

La arquitectura elegida para el desarrollo del proyecto fue la MobileNet debido al poco espacio que usa y su alta precisión. Se usó un modelo pre-entrenado y se ajustaron algunos parámetros usando el método de transfer-learning, con el cual se conservan los parámetros de las primeras capas, encargadas de clasificar características simples como líneas y figuras simples, y se entrenan solo las capas superiores, encargadas de clasificar características más específicas como partes del cuerpo, texturas, partes de animales o plantas, etc [49].

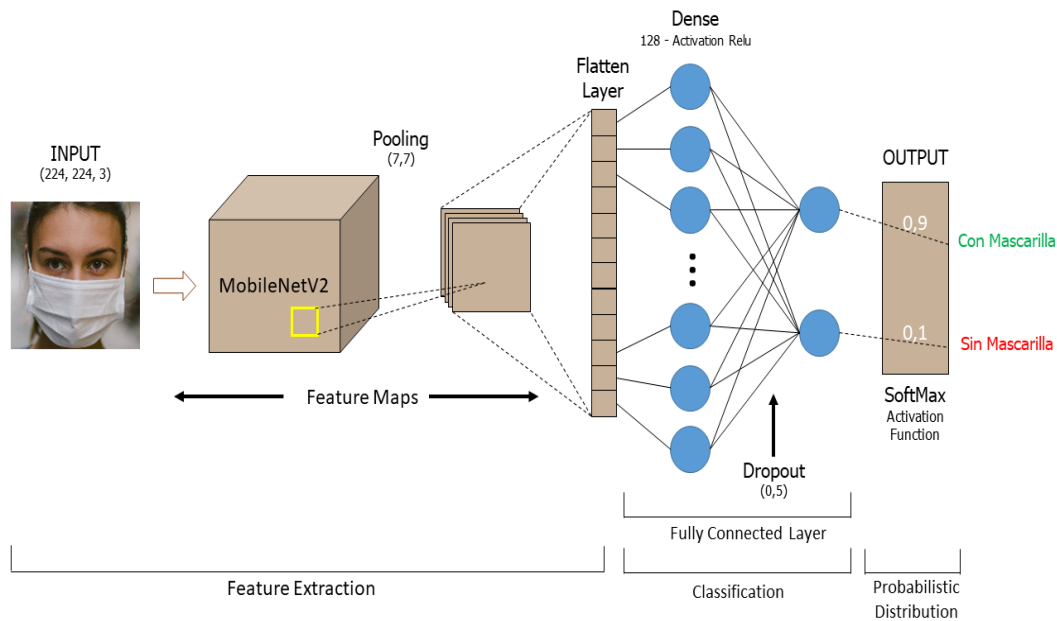


Figura 4.3: Esquema de la Red Neuronal.

Fuente: Autor.

4.3.2. Dataset para el Entrenamiento de la CNN

El conjunto de datos usado para el entrenamiento de la red neuronal convolucional fue el provisto por la plataforma Kaggle para el reto de reconocimiento facial y el uso o no de mascarillas. En la figura 4.4 se pueden observar algunas imágenes del dataset con el cual se trabaja.



Figura 4.4: Imágenes del Dataset CON y SIN Mascarilla.

Fuente: Kaggle

Este conjunto de datos consta de imágenes en a color con un tamaño de 160 x 160 píxeles. Dónde se registran caras mas o menos centradas y que ocupan el mismo espacio

en cada imagen.

El conjunto de datos para el entrenamiento consta de 3835 imágenes, mientras que el conjunto de datos para la validación consta de 589 imágenes en las cuales se presentan una gran variedad de personas en género, color, raza y otros aspectos para el análisis de la validación[50].

En el desarrollo del proyecto se incluyeron fotos personales y familiares, para poder obtener una validación más real. Además el uso de más clases aumenta el número de parámetros de la red neuronal, provocando que el espacio que esta ocupa incrementa, el cual es limitado en la tarjeta.

Para aumentar el número de imágenes se usó la clase ImageDataGenerator de la librería Keras, con el cual se ingresa cada imagen y se obtienen imágenes con efecto espejo en ambos ejes, rotación sobre el eje z determinados grados, etc.

4.3.3. Parámetros de entrenamiento de la MobileNet

Se usó un modelo pre-entrenado de MobileNet con un tamaño de entrada de 224 x 224, por lo cual era necesario escalar las imágenes del dataset a dicho tamaño. Además se usó solo el modelo base, configurando la red para que no se tuvieran en cuenta las últimas capas, teniendo la posibilidad de agregar otras para la clasificación final entre las 2 nuevas clases.

Tipo de capa	Forma de salida	Parámetros
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
conv_pw_13_relu (MobileNet)	(None, 4, 4, 768)	0
global_average_pooling2d	(None, 768)	0
dense (Dense)	(None, 256)	196864
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
dense_2 (Dense/Softmax)	(None, 2)	645

Tabla 4.1: Resumen de la arquitectura MobileNet

Fuente: Autor.

Entrenamiento de la Red Neuronal

En primer lugar es importante mencionar las características del computador u ordenador en donde se realizó la etapa de entrenamiento de la red neuronal, para ello, se contó con un pc cuyas especificaciones técnicas son: procesador Intel I5 7200U dual core 2,5 GHz, una tarjeta RAM 8Gb 2133MHz y un disco duro sólido de 240Gb. A partir de ésto se pudo realizar el entrenamiento de la red con un tiempo de 2h aproximadamente.

Para el entrenamiento se usó el optimizador Adam con un factor de aprendizaje de 0.0001 y 20 épocas. Además se implementaron funciones para guardar el modelo (en formato 'json') con mejor desempeño durante el entrenamiento, de manera que si la precisión del modelo no mejoraba transcurrida cierta cantidad de iteraciones, el proceso se detenía o se reducía el factor de aprendizaje. En el código 4.1 se explica a detalle el funcionamiento por medio de comentarios.

Código 4.1: Algoritmo para el entrenamiento de la MobileNet

```

1
2 # USAGE
3 # python train_mask_detector.py --dataset dataset
4
5 # import the necessary packages
6 from tensorflow.keras.preprocessing.image import ImageDataGenerator
7 from tensorflow.keras.applications import MobileNetV2
8 from tensorflow.keras.layers import AveragePooling2D
9 from tensorflow.keras.layers import Dropout
10 from tensorflow.keras.layers import Flatten
11 from tensorflow.keras.layers import Dense
12 from tensorflow.keras.layers import Input
13 from tensorflow.keras.models import Model
14 from tensorflow.keras.optimizers import Adam
15 from tensorflow.keras.applications.mobilenet import preprocess_input
16 from tensorflow.keras.preprocessing.image import img_to_array
17 from tensorflow.keras.preprocessing.image import load_img
18 from tensorflow.keras.utils import to_categorical
19 from sklearn.preprocessing import LabelBinarizer
20 from sklearn.model_selection import train_test_split
21 from sklearn.metrics import classification_report
22 from imutils import paths
23 import matplotlib.pyplot as plt
24 import numpy as np
25 import argparse
26 import os
27
28 print(os.getcwd())
29 # construct the argument parser and parse the arguments
30 dataset_path = os.getcwd()+"//dataset"
31 model_path = os.getcwd()+"//model//mask_model"
32 plot_path = os.getcwd()+"//plot"
33
34
35 # initialize the initial learning rate, number of epochs to train for,
36 # and batch size
37 INIT_LR = 1e-4
38 EPOCHS = 20
39 BS = 32

```

```

40
41 # grab the list of images in our dataset directory, then initialize
42 # the list of data (i.e., images) and class images
43 print("[INFO] loading images...")
44 imagePaths = list(paths.list_images(dataset_path))
45 imagePaths = [imagePath.replace("\\", "/",-1) for imagePath in imagePaths]
46 data = []
47 labels = []
48
49 # loop over the image paths
50 for imagePath in imagePaths:
51     # extract the class label from the filename
52     label = imagePath.split("/")[-2]
53
54     # load the input image (224x224) and preprocess it
55     image = load_img(imagePath, target_size=(224, 224))
56     image = img_to_array(image)
57     image = preprocess_input(image)
58
59     # update the data and labels lists, respectively
60     data.append(image)
61     labels.append(label)
62
63 # convert the data and labels to NumPy arrays
64 data = np.array(data, dtype="float32")
65 labels = np.array(labels)
66
67 # perform one-hot encoding on the labels
68 lb = LabelBinarizer()
69 labels = lb.fit_transform(labels)
70 labels = to_categorical(labels)
71
72 # partition the data into training and testing splits using 75% of
73 # the data for training and the remaining 25% for testing
74 (trainX, testX, trainY, testY) = train_test_split(data, labels,
75     test_size=0.20, stratify=labels, random_state=42)
76
77 # construct the training image generator for data augmentation
78 aug = ImageDataGenerator(
79     rotation_range=20,
80     zoom_range=0.15,
81     width_shift_range=0.2,
82     height_shift_range=0.2,
83     shear_range=0.15,
84     horizontal_flip=True,
85     fill_mode="nearest")
86
87
88
89 # place the head FC model on top of the base model (this will become
90 # the actual model we will train)
91 model = Model(inputs=baseModel.input, outputs=headModel)
92
93 # loop over all layers in the base model and freeze them so they will
94 # *not* be updated during the first training process
95 for layer in baseModel.layers:
96     layer.trainable = False
97
98 # compile our model
99 print("[INFO] compiling model...")
100 opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
101 model.compile(loss="binary_crossentropy", optimizer=opt,
102     metrics=["accuracy"])
103

```

```

104 # train the head of the network
105 print("[INFO] training head...")
106 H = model.fit(
107     aug.flow(trainX, trainY, batch_size=BS),
108     steps_per_epoch=len(trainX) // BS,
109     validation_data=(testX, testY),
110     validation_steps=len(testX) // BS,
111     epochs=EPOCHS)
112
113 # make predictions on the testing set
114 print("[INFO] evaluating network...")
115 predIdxs = model.predict(testX, batch_size=BS)
116
117 # for each image in the testing set we need to find the index of the
118 # label with corresponding largest predicted probability
119 predIdxs = np.argmax(predIdxs, axis=1)
120
121 # show a nicely formatted classification report
122 print(classification_report(testY.argmax(axis=1), predIdxs,
123     target_names=lb.classes_))
124
125 # serialize the model to disk
126 print("[INFO] saving mask detector model... path: %s"%(model_path+".h5"))
127 model.save(model_path+".h5")
128
129 # plot the training loss and accuracy
130 N = EPOCHS
131 plt.style.use("ggplot")
132 plt.figure()
133 plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
134 plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
135 plt.plot(np.arange(0, N), H.history["accuracy"], label="train_acc")
136 plt.plot(np.arange(0, N), H.history["val_accuracy"], label="val_acc")
137 plt.title("Training Loss and Accuracy")
138 plt.xlabel("Epoch #")
139 plt.ylabel("Loss/Accuracy")
140 plt.legend(loc="lower left")
141 plt.savefig(plot_path)

```

4.3.4. Verificación del funcionamiento de la Red Neuronal

Después del entrenamiento, se cargó el modelo usando el código 4.2 para verificar su correcto funcionamiento usando imágenes tomadas de la cámara web y del dataset. Para el reconocimiento de rostros y obtención de la posición de los mismos se usó el método del clasificador en cascada, disponible mediante la librería de open-cv.

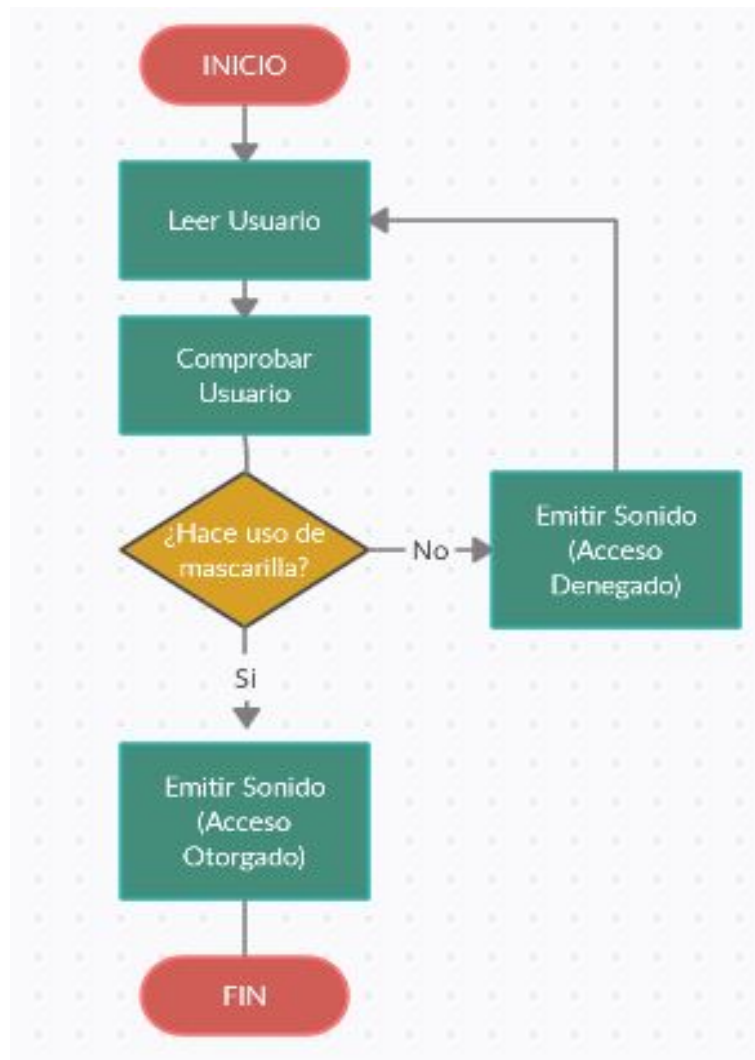


Figura 4.5: Diagrama de Flujo para Entender la Lógica de Programación del Algoritmo para la Verificación del Funcionamiento del Modelo.

Fuente: Autor.

Código 4.2: Algoritmo para la verificación del funcionamiento del modelo.

```

1 # import the necessary packages
2 from tensorflow.keras.applications.mobilenet import preprocess_input
3 from tensorflow.keras.preprocessing.image import img_to_array
4 from tensorflow.keras.models import load_model
5 from imutils.video import VideoStream
6 #from pygame import mixer
7 import numpy as np
8 import imutils
9 import time
10 import cv2
11 import os
12 import math
13
14 #system libraries
15 import os
16 import sys
17 from threading import Timer
18 import shutil
19 import time
20
21 detections = None
22 def detect_and_predict_mask(frame, faceNet, maskNet,threshold):
23     # grab the dimensions of the frame and then construct a blob
24     # from it
25     global detections
26     (h, w) = frame.shape[:2]
27     blob = cv2.dnn.blobFromImage(frame, 1.0, (300, 300),(104.0, 177.0, 123.0))
28
29     # pass the blob through the network and obtain the face detections
30     faceNet.setInput(blob)
31     detections = faceNet.forward()
32
33     # initialize our list of faces, their corresponding locations,
34     # and the list of predictions from our face mask network
35     faces = []
36     locs = []
37     preds = []
38     # loop over the detections
39     for i in range(0, detections.shape[2]):
40         # extract the confidence (i.e., probability) associated with
41         confidence = detections[0, 0, i, 2]
42
43         # filter out weak detections by ensuring the confidence is
44         # greater than the minimum confidence
45         if confidence > threshold:
46             # compute the (x, y)-coordinates of the bounding box for
47             # the object
48             box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
49             (startX, startY, endX, endY) = box.astype("int")
50
51             # ensure the bounding boxes fall within the dimensions of
52             # the frame
53             (startX, startY) = (max(0, startX), max(0, startY))
54             (endX, endY) = (min(w - 1, endX), min(h - 1, endY))
55
56             # extract the face ROI, convert it from BGR to RGB channel
57             # ordering, resize it to 224x224, and preprocess it
58             face = frame[startY:endY, startX:endX]
59             face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
60             face = cv2.resize(face, (224, 224))
61             face = img_to_array(face)
62             face = preprocess_input(face)

```

```

63     face = np.expand_dims(face, axis=0)
64
65     # add the face and bounding boxes to their respective
66     # lists
67     locs.append((startX, startY, endX, endY))
68     #print(maskNet.predict(face)[0].tolist())
69     preds.append(maskNet.predict(face)[0].tolist())
70     return (locs, preds)
71
72 # SETTINGS
73 MASK_MODEL_PATH=os.getcwd()+"\\model\\mask_model.h5"
74 FACE_MODEL_PATH=os.getcwd()+"\\face_detector"
75 SOUND_PATH=os.getcwd()+"\\sounds\\alarm.wav"
76 THRESHOLD = 0.5
77
78 # Load Sounds
79 #mixer.init()
80 #sound = mixer.Sound(SOUND_PATH)
81
82 # load our serialized face detector model from disk
83 print("[INFO] loading face detector model...")
84 prototxtPath = os.path.sep.join([FACE_MODEL_PATH, "deploy.prototxt"])
85 weightsPath = os.path.sep.join([FACE_MODEL_PATH, "res10_300x300_ssd_iter_140000.caffemodel"])
86 faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)
87
88 # load the face mask detector model from disk
89 print("[INFO] loading face mask detector model...")
90 maskNet = load_model(MASK_MODEL_PATH)
91
92 # initialize the video stream and allow the camera sensor to warm up
93 print("[INFO] starting video stream...")
94 vs = VideoStream(0).start()
95 time.sleep(2.0)
96
97
98 # loop over the frames from the video stream
99 while True:
100     # grab the frame from the threaded video stream and resize it
101     # to have a maximum width of 400 pixels
102     frame = vs.read()
103     frame = imutils.resize(frame, width=400)
104     original_frame = frame.copy()
105     # detect faces in the frame and determine if they are wearing a
106     # face mask or not
107     (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet, THRESHOLD)
108
109     # loop over the detected face locations and their corresponding
110     # locations
111     for (box, pred) in zip(locs, preds):
112         # unpack the bounding box and predictions
113         (startX, startY, endX, endY) = box
114         (mask, withoutMask) = pred
115
116         # determine the class label and color we'll use to draw
117         # the bounding box and text
118         label = "Mask" if mask > withoutMask else "No Mask"
119         #if(label=="No Mask") and (mixer.get_busy()==False):
120         # sound.play()
121         color = (0, 255, 0) if label == "Mask" else (0, 0, 255)
122
123         # include the probability in the label
124         label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
125
126         # display the label and bounding box rectangle on the output

```

```
127 # frame
128 cv2.putText(original_frame, label, (startX, startY - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
129 cv2.rectangle(original_frame, (startX, startY), (endX, endY), color, 2)
130 cv2.rectangle(frame, (startX, startY+math.floor((endY-startY)/1.6)), (endX, endY), color, -1)
131
132
133 cv2.addWeighted(frame, 0.5, original_frame, 0.5, 0, frame)
134
135 # show the output frame
136 frame= cv2.resize(frame, (860, 490))
137 cv2.imshow("Masks Detection", frame)
138 key = cv2.waitKey(1) & 0xFF
139
140 # if the 'q' key was pressed, break from the loop
141 if key == ord("q"):
142     break
143
144 # do a bit of cleanup
145 cv2.destroyAllWindows()
146 vs.stop()
```

4.4. Interfaz para la Comunicación Sistema - Usuario

La Interfaz Sistema - Usuario es una página web compuesta de elementos e iconos visuales, cuya función es presentar información al beneficiario y permitir la interacción con el sistema y la computadora.

Un diseño de interfaz de beneficiario es el fragmento primordial del sumario del diseño del método. Es esencial que la interfaz de usuario sea diseñada para ajustarse a las habilidades, expectativas, experiencias de los usuarios. Un buen diseño de una interfaz de beneficiario determina la confiabilidad del sistema.

Las interfaces básicas de usuario son aquellas que incluyen menús, ventanas, teclado, ratón, los beeps y algunos otros sonidos que la computadora hace, en general, todos aquellos canales por los cuales se permite el intercambio de comunicación entre el ser humano y el ordenador. La principal interacción humano-máquina a través de una adecuada interfaz (Interfaz de Usuario), que le brinde tanto facilidad, como eficacia mientras el proceso.

4.4.1. Programación en REACT

Como se había mencionado anteriormente, React es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. En éste trabajo de investigación, se decide iniciar desde este punto, es decir, comenzar con la línea de programación como es la Interfaz de comunicación y mando.

Inicialmente, cuando el proyecto se basaba en imágenes o bosquejos, nació una idea de como podría desarrollarse la interfaz y con ello lo que contemplaría en su escena gráfica y de mando.

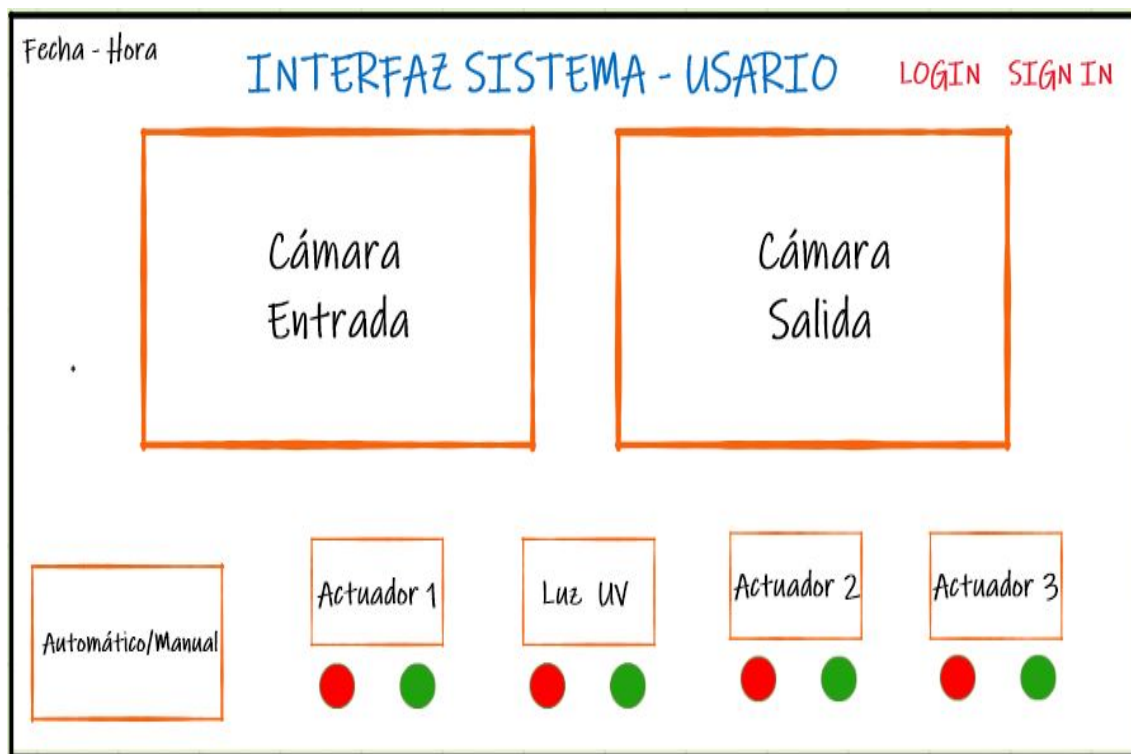


Figura 4.6: Bosquejo de la Interfaz Gráfica para Implementarla en REACT.

Fuente: Autor

Partiendo de la figura 4.6, se uso de referencia para programar la página web e integrar los elementos que la conformarían. En contexto, las líneas bases de éste código se muestran a continuación (Código 4.3). Antes de leer éste segmento de programación, mostraré un diagrama de flujo que abarca la lógica que se implementó.

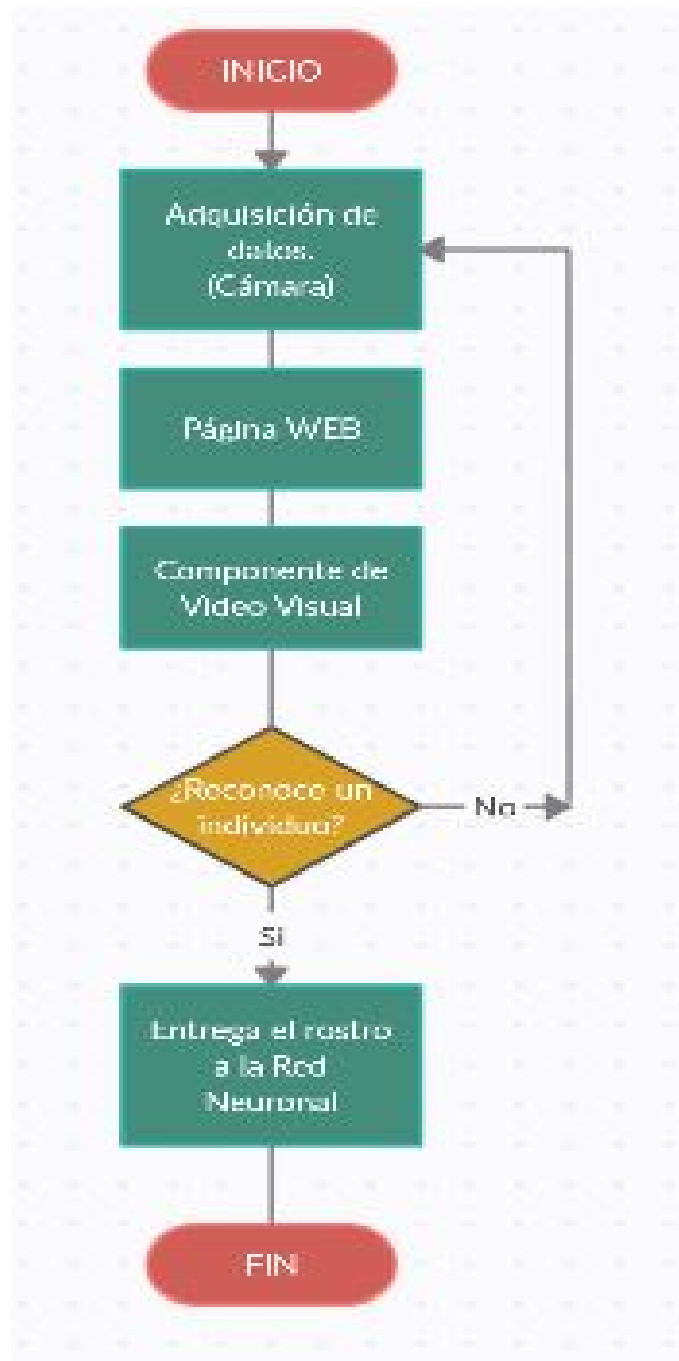


Figura 4.7: Diagrama de Flujo para Entender la Lógica de Programación del Algoritmo para la Interfaz.

Fuente: Autor.

Código 4.3: Algoritmo inicial para la Interfaz Sistema-Usuario

```

1 import './App.css';
2 import { useState } from 'react'
3 import { Container, Grid, makeStyles } from '@material-ui/core'
4
5 import Navbar from './components/Navbar';
6 import Controls from './components/Controls'
7 import StateConfiguration from './components/StateConfiguration'
8 import VideoCard from './components/VideoCard'
9 import Footer from './components/Footer'
10
11 const useStyles = makeStyles((theme) => ({
12   marginTop: {
13     marginTop: "30px"
14   }
15 })))
16
17 function App() {
18   const [isAuto, setIsAuto] = useState(false)
19   const classes = useStyles()
20
21   function handleChange(e) {
22     console.log(e.target.checked)
23     setIsAuto(e.target.checked)
24   }
25
26   return (
27     <>
28       <Navbar />
29       <Container maxWidth="lg">
30         <StateConfiguration callback={handleChange} />
31         <div className={classes.marginTop}>
32           <Grid container spacing={3} alignItems="center">
33             <Grid item xs={8}>
34               <VideoCard />
35             </Grid>
36             <Grid item xs={4}>
37               <Controls auto={isAuto} />
38             </Grid>
39           </Grid>
40         </div>
41       </Container>
42       <Footer />
43     </>
44   );
45 }
46
47 export default App;

```

Analizando éstas líneas del código principal, se puede decir que tiene dos divisiones. Una de ellas son las importaciones de los componentes, las cuales, cada uno tiene su index independiente y su respectiva programación. La segunda división del código se refiere a la estructura de la página WEB en sí.



Figura 4.9: Primera Visualización de la Interfaz Gráfica.

Fuente: Autor

El contexto inicial para la comunicación e interacción entre el individuo y el proceso se reduce a la Figura 4.9, en la cual, se prevee de dos estados (Automático y Manual). Además, posee una tarjeta de vídeo para mostrar las grabaciones de la cámara en tiempo real y el debido reconocimiento facial para el uso o no de mascarilla. Por consiguiente, tarjetas para la visualización del estado On/Off de los actuadores o elementos finales de control y Luz UV. Las acciones se realizan mediante manipulación directa, para facilitar la interacción del usuario con la computadora y el proceso inmótico.

4.5. Implementación de la Red Neuronal Entrenada en la Interfaz Sistema - Usuario

El desarrollo que demarca la metodología de investigación en el capítulo anterior, dispone de ciertas actividades para lograr los objetivos del trabajo de grado. Hasta el momento, se ha trabajado de forma independiente, es decir, los ítems o ramas que corresponden al desarrollo general se han implementado una a una. En ésta sección se busca unificar los códigos fuentes de programación para lograr un sistema compacto y automático.

En primer lugar, se busca incorporar el código de reconocimineto facial para el uso o no de mascarillas en las personas en la tarjeta de vídeo que brinda la interfaz, este procedimiento tiene la finalidad de dejar una página web funcional. Por consiguiente, al ir demarcando éstos pasos se puede visualizar todo el proceso inmótico desde una sola aplicación.

Código 4.4: Algoritmo para la implementación de la Red Neuornal en la Interfaz

```

1 import { useEffect } from 'react'
2 import { Paper, makeStyles } from "@material-ui/core";
3 // import * as tf from '@tensorflow/tfjs'
4 import * as blazeface from '@tensorflow-models/blazeface'
5
6 const useStyles = makeStyles((theme) => ({
7   root: {
8     height: "380px",
9     justifyContent: "center",
10  },
11  video: {
12    margin: "auto",
13    display: "inline-block",
14    height: "100%"
15  },
16  canvas: {
17    height: "380px",
18    left: "67px",
19    margin: "auto",
20    position: "relative",
21    top: "-384px"
22  }
23 })))
24
25
26 export default function VideoCard() {
27   const classes = useStyles()
28
29   useEffect(() => {
30     //Coded by oh yicong, visit my youtube channel for more programming tutorials :)
31     var model, mask_model, ctx, videoWidth, videoHeight, canvas;
32     const video = document.getElementById('video');
33
34     const state = {
35       backend: 'webgl'
36     };
37

```

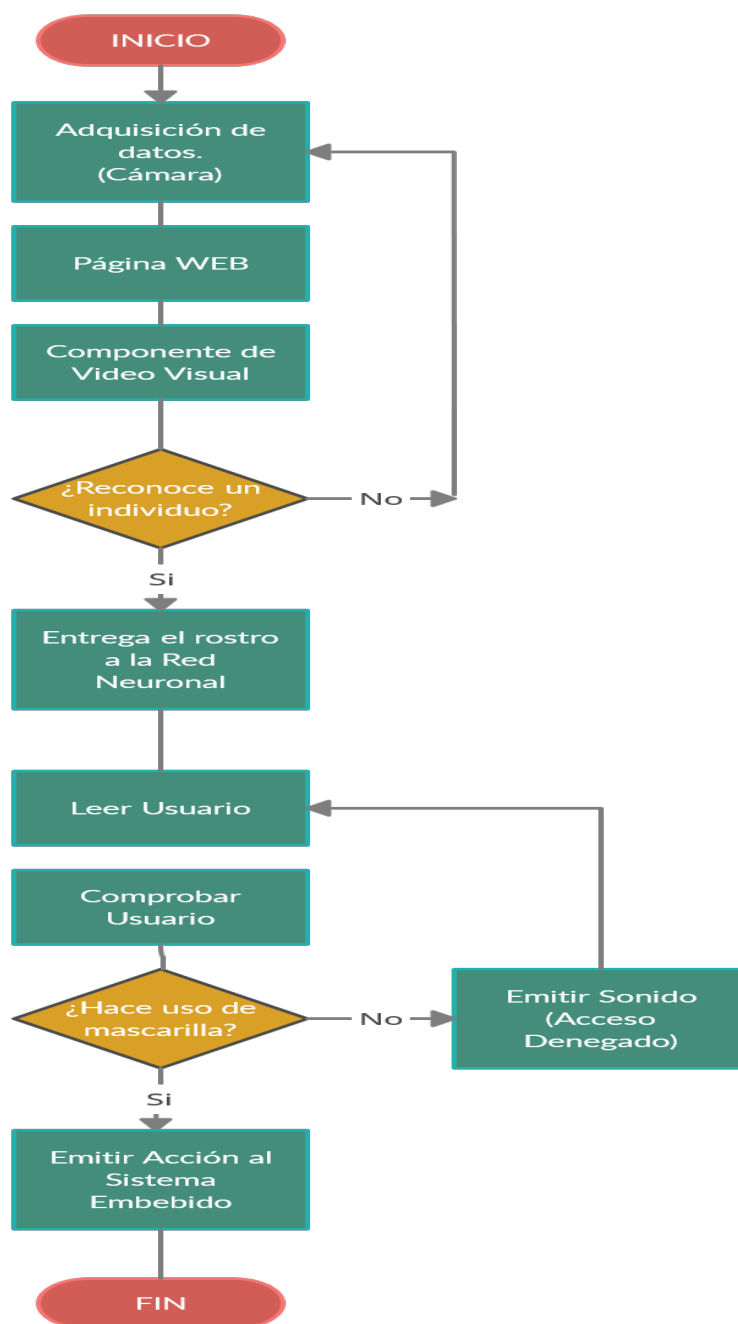


Figura 4.10: Diagrama de Flujo para Entender la Lógica de Programación del Algoritmo 4.4

Fuente: Autor

```

38  async function setupCamera() {
39      const stream = await navigator.mediaDevices.getUserMedia({
40          'audio': false,
41          'video': { facingMode: 'user' },
42      });
43
44      video.srcObject = stream;
45
46      return new Promise((resolve) => {
47          video.onloadedmetadata = () => {
48              resolve(video);
49          };
50      });
51  }
52
53  const renderPrediction = async () => {
54      try {
55          window.tf.engine().startScope()
56          ctx.clearRect(0, 0, canvas.width, canvas.height);
57          //estimatefaces model takes in 4 parameter (1) video, returnTensors, flipHorizontal, and annotateBoxes
58          const predictions = await model.estimateFaces(video, true, false, false);
59          const offset = window.tf.scalar(127.5);
60          //check if prediction length is more than 0
61          if (predictions.length > 0) {
62              //clear context
63
64              for (let i in predictions) {
65                  var text = ""
66                  var start = predictions[i].topLeft.arraySync();
67                  var end = predictions[i].bottomRight.arraySync();
68                  var size = [end[0] - start[0], end[1] - start[1]];
69                  if (videoWidth < end[0] || videoHeight < end[1] || start[0] < 0 || start[1] < 0) {
70                      console.log("image out of frame")
71                      continue
72                  }
73                  var inputImage = window.tf.browser.fromPixels(video).toFloat()
74                  inputImage = inputImage.sub(offset).div(offset);
75                  inputImage = inputImage.slice([parseInt(start[1]), parseInt(start[0]), 0], [parseInt(size[1]), parseInt(size[0]), 3])
76                  inputImage = inputImage.resizeBilinear([224, 224]).reshape([1, 224, 224, 3])
77                  let result = mask_model.predict(inputImage).dataSync()
78                  result = Array.from(result)
79                  ctx.beginPath()
80                  if (result[1] > result[0]) {
81                      //no mask on
82                      // await fetch('http://192.168.1.11:4000/0')
83                      ctx.strokeStyle = "red"
84                      ctx.fillStyle = "red";
85                      text = "No Mask: " + (result[1] * 100).toFixed(3).toString() + "%";
86                  } else {
87                      //mask on
88                      // await fetch('http://192.168.1.11:4000/1')
89                      ctx.strokeStyle = "green"
90                      ctx.fillStyle = "green";
91                      text = "Mask: " + (result[0] * 100).toFixed(3).toString() + "%";
92                  }
93                  ctx.lineWidth = "4"
94                  ctx.rect(start[0], start[1], size[0], size[1])
95                  ctx.stroke()
96                  ctx.font = "bold 15pt sans-serif";
97                  ctx.fillText(text, start[0] + 5, start[1] + 20)
98              }
99          }
100          //update frame
101          requestAnimationFrame(renderPrediction);

```

```
102     window.tf.engine().endScope()
103   } catch (error) {
104     console.log(error)
105   }
106 };
107
108 const setupPage = async () => {
109   await window.tf.setBackend(state.backend);
110   await setupCamera();
111   video.play();
112
113   videoWidth = video.videoWidth;
114   videoHeight = video.videoHeight;
115   video.width = videoWidth;
116   video.height = videoHeight;
117
118   canvas = document.getElementById('output');
119   canvas.width = videoWidth;
120   canvas.height = videoHeight;
121   ctx = canvas.getContext('2d');
122   ctx.fillStyle = "rgba(255, 0, 0, 0.5)";
123
124   model = await blazeface.load();
125
126   mask_model = await window.tf.loadLayersModel('http://localhost:3000/model/model.json');
127
128   renderPrediction();
129 };
130
131 setupPage();
132 }, [])
133
134
135 return (
136   <div className={classes.root}>
137     <video id="video" playsInline className={classes.video} ></video>
138     <canvas id="output" className={classes.canvas}></canvas>
139   </div>
140 )
141 }
```

A partir del código 4.4, se integra el reconocimiento facial en la página web, se vinculan estas dos ramas para lograr un alto grado de eficiencia en la práctica. Con ello, se empizan a realizar las primeras pruebas identificando personas externas al dataset de entrenamiento. Como ejemplo, presento mi rostro como primera visualización del reconocimiento facial y de los resultados entregados por la red neuronal.

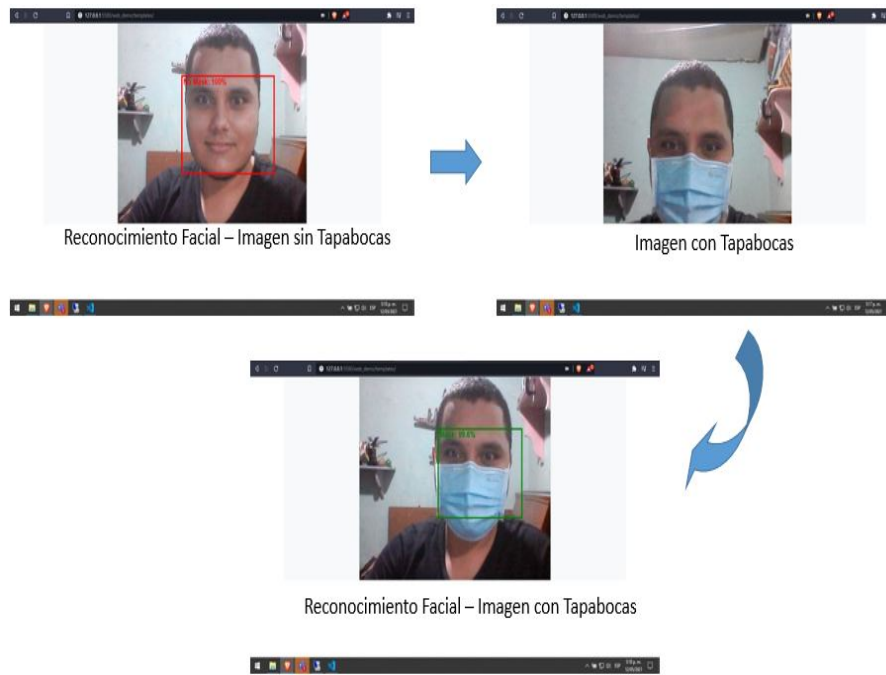


Figura 4.11: Primera Visualización del Reconocimiento Facial.

Fuente: Autor

Partiendo de la Figura 4.11, inicialmente se tenía el modelo de reconocimiento facial de forma independiente, es decir, al ejecutar las líneas de código se desprende un servidor local al cual se accede desde el navegador, seguidamente hace la petición del uso de la cámara para una transmisión instantánea. Con ello se visualiza nuestro perfil (Reconocimiento Facial - Imagen sin Tapabocas) En ésta primera foto, la red neuronal logra verificar mediante un cuadro rojo y un porcentaje de acierto al no uso de la mascarilla; posteriormente, hacemos uso de un tapabocas para un previo reconocimiento (Imagen con Tapabocas). Por último, se logra visualizar en el mismo vídeo que la red neuronal identifica el uso de mascarilla, implementando un cuadro verde y un porcentaje de acierto en el reconocimiento.

Teniendo éste sistema compacto, se implementa con individuos humanos no partícipes del banco de imágenes; se logra analizar que la red neuronal es capaz de identificar correctamente el uso o no de mascarillas en las personas.



Figura 4.12: Modelo Neuronal en la Interfaz Gráfica para el Reconocimiento Facial.

Fuente: Autor

Partiendo de lo descrito al inicio de la sección, se llega a la Figura 4.12. Se unifican las líneas de programación en el desarrollo del trabajo. En ésta figura se aprecia el resultado del reconocimiento facial, la interfaz sistema-usuario, los indicadores del estado de cada actuador, por último, se debe implementar el sistema embebido para el control de la activación o desactivación de los elementos finales de control y la comunicación con el tablero eléctrico.

4.6. Programación del Sistema Embebido

En ésta sección, se dará una breve explicación del código de programación para el sistema embebido, el cual, permite la comunicación de forma inalámbrica con el uso de la red local entre la interfaz y el tablero eléctrico. Además, con ayuda de éste sistema embebido como es la Raspberry Pi, nos otorga una gama amplia de aplicaciones, en éste trabajo de grado se usará como un ordenador para el control del prototipo real.

En primer lugar, la Raspberry Pi es un ordenador de bajo coste y tamaño reducido, tanto es así que cabe en la palma de la mano, pero puedes conectarle un televisor y un teclado para interactuar con ella exactamente igual que cualquier otra computadora. Es una placa o sistema embebido capaz de interactuar con diversos proyectos digitales o demás problemas de programación.

Para iniciar el proceso de programación de la placa embebida, se crea un servidor en el mismo, haciendo uso de Nodejs usando la librería Express para facilitar el desarrollo del mismo. Por consiguiente, con ayuda de la red local hogar se puede enlazar la comunicación con la interfaz gráfica sistema - usuario; de dicha manera, se puede obtener un preámbulo del funcionamiento del prototipo. Con ésta conexión inalámbrica, se gestiona el sistema de control para la inspección del proceso inmótico. Teniendo la respectiva comunicación, se hace una declaración de pines, los cuales son los encargados de llevar la señal del servidor al tablero eléctrico. En ésta declaración de pines se tiene en cuenta los pilotos luminosos, la activación de los elementos finales de control, etc.

Código 4.5: Algoritmo Base para el Sistema Embebido

```
1 import express from 'express'
2 import cors from 'cors'
3 import lowDb from 'lowdb'
4 import FileSync from 'lowdb/adapters/FileSync.js'
5 import bodyParser from 'body-parser'
6 import { Gpio } from 'onoff'
7
8 import puertasRoutes from './routes/puertas.js'
9 import luzUVRoutes from './routes/luzUV.js'
10 import pilotosRoutes from './routes/pilotos.js'
11 import allPinesRoutes from './routes/allPines.js'
12 import hacerSecuenciaRoute from './routes/secuencia.js'
13
14 export const db = lowDb(new FileSync('db.json'))
15
16 db.defaults({
17   puertas: [
18     { id: 0, pin: 4 },
19     { id: 1, pin: 5 },
20     { id: 2, pin: 6 }
21   ],
22   luzuv: { id: 1, pin: 7 }
23 }).write()
24
25 const app = express()
```

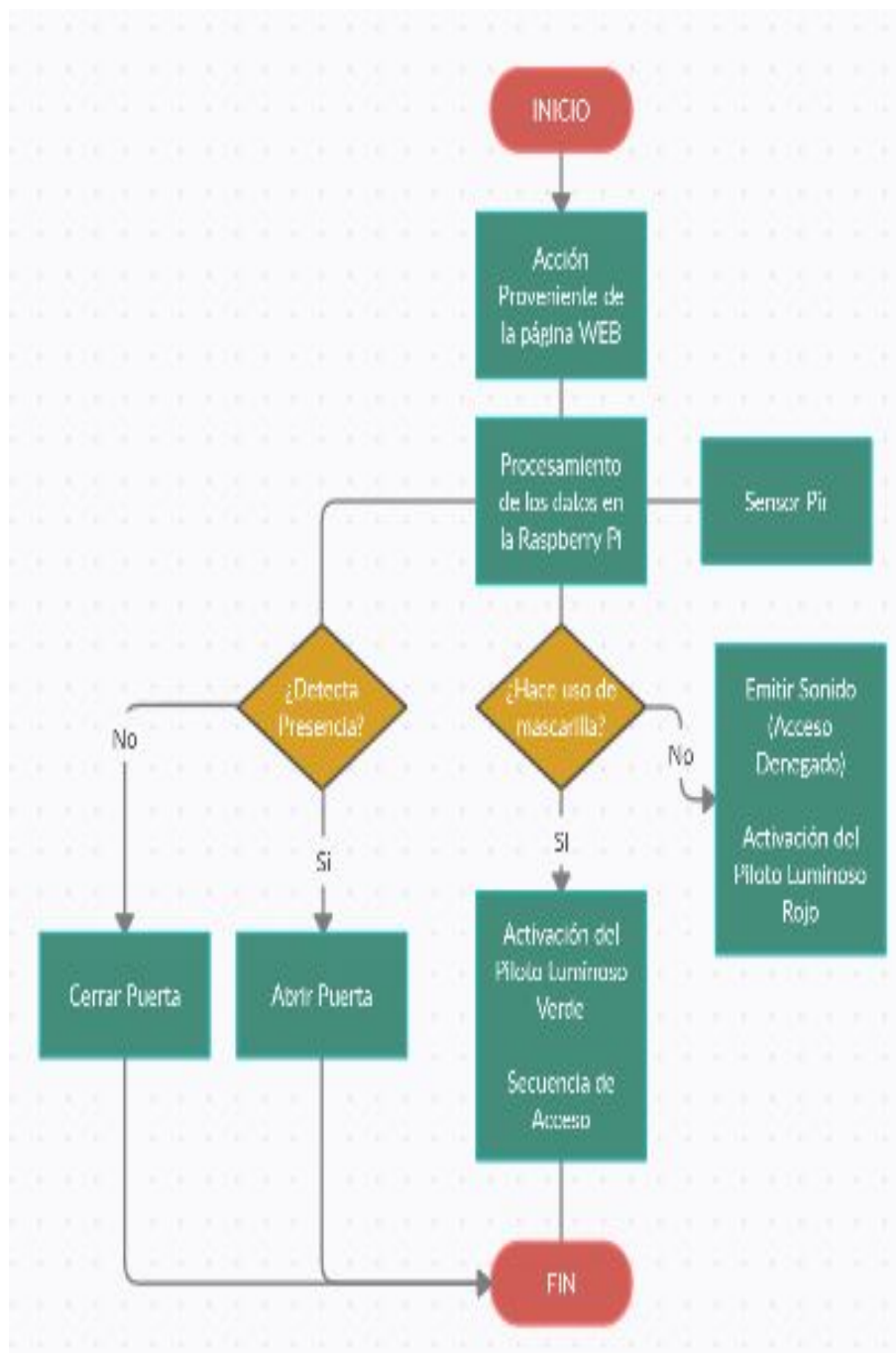



Figura 4.13: Diagrama de Flujo para Entender la Lógica de Programación del Algoritmo 4.5

Fuente: Autor

```
26
27 app.use(bodyParser.json())
28 app.use(cors())
29
30 const PORT = 4000
31
32 // inicializar puertas
33 const infoPuertas = db.get('puertas').value()
34 export const puertas = []
35 for (let i in infoPuertas) {
36   const { id, pin, ledActivo, ledInactivo } = infoPuertas[i]
37   const pines = {
38     id,
39     pin: new Gpio(pin, 'out'),
40     ledActivo: new Gpio(ledActivo, 'out'),
41     ledInactivo: new Gpio(ledInactivo, 'out'),
42   }
43   puertas.push(pines)
44 }
45
46 // inicializar Luz uv
47 const infoLuzUV = db.get('luzuv').value()
48 const { id, pin, ledActivo, ledInactivo } = infoLuzUV
49 export const luzUV = {
50   id,
51   pin: new Gpio(pin, 'out'),
52   ledActivo: new Gpio(ledActivo, 'out'),
53   ledInactivo: new Gpio(ledInactivo, 'out'),
54 }
55
56 // inicializar pilotos tapabocas
57 const infoPilotos = db.get('pilotos').value()
58 const { conTapabocas, sinTapabocas } = infoPilotos
59 export const pilotos = {
60   conTapabocas: new Gpio(conTapabocas, 'out'),
61   sinTapabocas: new Gpio(sinTapabocas, 'out'),
62 }
63
64 // Configuración inicial pines
65 puertas.forEach(puerta => {
66   const { pin, ledActivo, ledInactivo } = puerta
67   pin.writeSync(0)
68   ledActivo.writeSync(0)
69   ledInactivo.writeSync(1)
70 })
71
72 const { pin: pinluzuv, ledActivo: ledActivoluzuv, ledInactivo: ledInactivoluzuv } = luzUV
73 pinluzuv.writeSync(0)
74 ledActivoluzuv.writeSync(0)
75 ledInactivoluzuv.writeSync(1)
76
77 pilotos.conTapabocas.writeSync(0)
78 pilotos.sinTapabocas.writeSync(0)
79
80 app.listen(PORT, () => console.log('listening on port 4000'))
```

En contexto, el código 4.5 - Algoritmo Base para el Sistema Embebido, es el encargado de realizar la etapa de control, de ordenar la activación o desactivación de los elementos finales de control y de supervisar el estado del proceso.

4.7. Tablero Eléctrico

Otro aspecto a tener en cuenta en el desarrollo de la propuesta es la elaboración de un Tablero Eléctrico de Control, el cual, nos proporcionará salidas de voltajes para una previa conexión de elementos finales de control.

Los tableros eléctricos son los encargados de resguardar los componentes o dispositivos de mando y de control de cualquier sistema eléctrico desde un circuito básico en un domicilio incluso el de una maquina manufacturero. En estos se puede concentrar los dispositivos de conexión, maniobra, protección, etc. Que permitan que una red eléctrica funcione en óptimas condiciones.

En éste trabajo de grado, se implementa un tablero eléctrico de control de forma que sea un centro de control de motores. Este tipo de tableros se caracterizan por tener como componentes principales contactores, relevadores y todo tipo de componente que se utiliza para controlar a los motores eléctricos u otros elementos finales de control.

Antes de la implementación de éste prototipo, se deben conocer las partes que conforman a un tablero eléctrico de forma industrial, entre ellas tenemos:

- **Borneras** Cumplen la función de realizar las interconexiones entre los cables, lo que ayuda a que los empalmes se eviten, como también el uso indebido de la cinta aisladora. Los modelos y calibres variarán según sea el tipo de cable que se esté utilizando.[51]
 - **Cablecanal** Es de las partes de un tablero de inspección eléctrico que cumple con la ocupación de conducir los cables, lo que ayuda a que estos estén en orden. Existen al menos dos tipos de estos, los ranurados y los no ranurados. La alternativa entre ellos dependerá de la rutina que se le quiere proporcionar.[51]
 - **Seccionador porta fusible** Cumple con la función de seccionar un circuito (esto quiere decir abrirlo o cerrarlo), como también, mediante un fusible, proteger.[51]
 - **Pilotos luminosos** Los estados (si se encuentra en marcha, está en parada, o en estado de alarma) deben ser señalizados, y esa es la función de un piloto luminoso dentro de un circuito eléctrico.[51]
 - **Cables** Los cables son muy importantes, ya que, es la vía por la cual la electricidad es suministrada a todos los elementos finales de control.[51]
-

Teniendo conocimiento de las partes más importantes que conforman a un tablero eléctrico industrial, se procede a un diseño preliminar en donde se encuentren los componentes eléctricos a implementar y sobretodo que correlacionen a las partes anteriormente descritas.

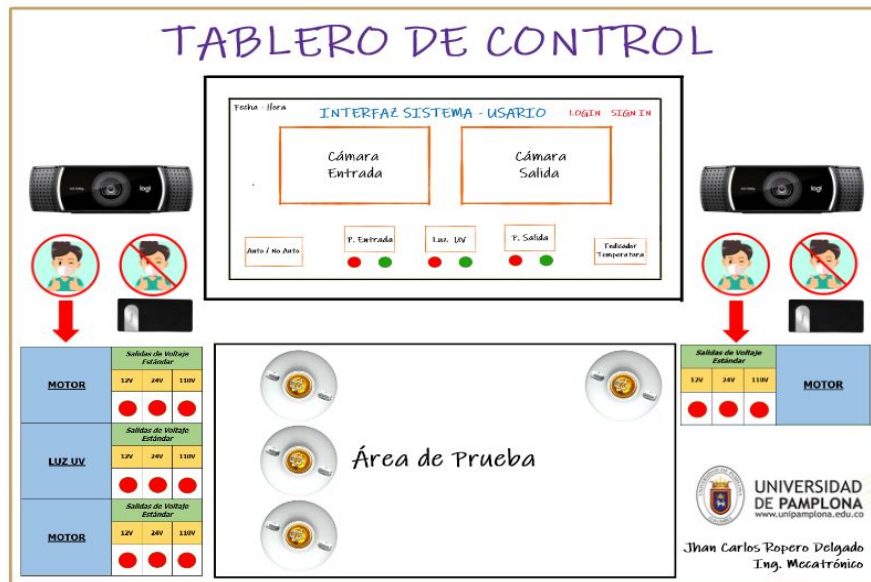


Figura 4.14: Bosquejo del Tablero Eléctrico en una Fase Preliminar.

Fuente: Autor

En primera instancia, según la Figura 4.14, se encuentra un bosquejo o diseño preliminar del como podría llevarse a cabo la implementación del Tablero Eléctrico de Control. En el, se encuentran las fases como:

- **Pilotos:** Son los encargados de señalar el estado del proceso, en nuestro caso, serán los indicadores del uso o no del tapabocas después del reconocimiento facial.
- **Borneras:** Éstos elementos nos permiten la fácil conexión entre el tablero eléctrico y los elementos finales de control. Además, contará con salidas previas de 12v ó 110v, con la finalidad de otorgar un plus ante los diversos actuadores estándares del mercado.
- **Zona de Prueba:** Para lograr que el dispositivo sea compacto, se postula una zona de prueba con plafones para conexiones de carga a 110v. Con ello, se logra darle un revestimiento al análisis de fallas que se puedan presentar en un futuro.
- **Zona de Interacción:** Se busca obtener en tiempo real la forma de visualizar el estado en general del proceso, de poder interactuar con los actuadores implicados y sobretodo el supervisar el estado de cada uno de los componentes en gestión.

Según lo enunciado anteriormente, la finalidad del bosquejo es lograr un acercamiento a la realidad para su debida implementación. En el camino del desarrollo de la propuesta se encontró con varios factores relevantes como: la adquisición de ciertos materiales, dispositivo como una CNC para el corte del molde principal, etc. Debido a las problemáticas que se dieron, el modelo se ajustó pero sin perder el orden de los objetivos.



Figura 4.15: Tablero Eléctrico en una Fase Real

Para implementar de forma real el prototipo del tablero eléctrico, se pasó por varias fases (Teniendo en cuenta la Figura 4.15), se tiene:

- **Diseño en Solidworks:** A través de un software CAD se puede sintetizar la idea, tener un diseño cercano a la realidad en dónde se involucren las medidas de los componentes a usar y con ello tener un modelo más compacto para su previa implementación. Haciendo uso de éste software (SOLIDWORKS) se consigue tener un diseño para luego proceder a los respectivos cortes y seccionado de pintura.
- **Primera Fase:** Al obtener el molde del tablero eléctrico se supervisan los cortes para evitar futuros errores con los materiales. Además, en ésta fase se implementan los "pilotos", los cuales son indicadores: Verde (Usa Mascarilla) y Rojo (No usa Mascarilla). Por consiguiente se adecuan las respectivas borneras para las salidas de voltajes para la conexión de los elementos finales de control.
- **Segunda Fase:** El modelo del tablero eléctrico, pasa de ser un bosquejo animado a hacer un prototipo físico. En ésta fase se implementan las debidas conexiones

exteriores entre la placa embebida, el sistema de control y la interfaz sistema - usuario. Recalcando que la implementación de la zona de prueba con bombillos de carga a 110v para una previa serie de pruebas del funcionamiento del tablero eléctrico y el proceso.

- **Fase Final:** Se implementan pilotos luminosos para corroborar el estado de cada actuador (Motor 1, Luz UV, Motor 2 y Motor 3) durante el proceso. Además se hacen pruebas de funcionamiento del prototipo completo para buscar averías y dar solución a ello. El tablero eléctrico está listo para cumplir con los objetivos planteados.

4.8. Síntesis

Culminando éste capítulo se logra obtener un sistema compacto para su previo en uso sistemas locales como micro empresas u otros sectores de la industria.

Cómo resultado se logra cumplir con los objetivos estipulados al inicio del proyecto.

- **Diseñar y elaborar un tablero de control eléctrico, el cual nos proporcione salida de voltajes estándares para la conexión de elementos finales de control.** Siendo éste el primer objetivo específico planteado, se logra identificar una investigación menuda sobre los diversos tipos de tableros eléctricos existentes, cómo también lo son las partes que lo conforman. Se desarrolló un diseño preliminar, al cual se le realizaron ajustes para luego llevarlo a un prototipo real. Dicho tablero eléctrico se ajusta a la necesidad demarcada, a la seguridad en cada componente y sobretodo a la funcionalidad correspondiente.
 - **Desarrollar un sistema inmótico que permita el control de sensores y actuadores, lo cual le posibilita al usuario controlar de forma fácil el proceso inmótico.** Cómo objetivo específico y a comparación de los demás, éste objetivo es el más general. Reúne toda la información del proyecto para poder darle cumplimiento al mismo. Para comenzar, un sistema inmótico es el conjunto de tecnologías aplicadas al control y la automatización inteligente de edificios no destinados a vivienda, como hoteles, centros comerciales, escuelas, universidades, hospitales y todos los edificios terciarios, permitiendo una gestión eficiente de los recursos. Describiendo las pautas de un sistema inmótico y haciendo uso de todas las herramientas implementadas en el trabajo de grado se logra desarrollar un sistema para la inspección y control de un proceso inmótico.
 - **Desarrollar una interfaz (Página Web) en REACT (Biblioteca de JavaScript Soft-ware Libre) para la visualización y control del sistema.** Aprender un lenguaje de programación puede presentarse como una tarea ardua
-

de implementación. Gracias a la información publicada en internet, se logra realizar una página WEB para la interacción y supervisión sistema - usuario.

- **Implementar algoritmos de visión artificial para el reconocimiento facial de un dataset de imágenes de personas con y sin tapabocas.** Éste es el objetivo que le da un valor agregado al trabajo de investigación, ya que, en la actualidad con la problemática sanitaria existente, impulsa a diversas ramas de la ciencia a buscar soluciones prácticas para evitar la propagación evolutiva del presente virus. Con ayuda de visión artificial se obtiene un plus para evitar el contagio por medio del tacto y se obtiene un margen de seguridad para cumplir con normas de bioseguridad. Al implementar un sistema de reconocimiento facial, trata de impulsar la cultura y educación de hacer uso de un elemento de protección, ya que, sin éste difícilmente tendrá acceso al recinto en mención. El trabajo de redes neuronales, fue un trabajo arduo en el sentido de que se requiere de un buen dispositivo electrónico para su debida programación e implementación. Gracias a la simplificación de trabajo que nos brinda el dataset de imágenes proporcionado por Kaggles, se pudo realizar un estudio para que el modelo neuronal aprendiera a reconocer las personas que usan o no mascarilla. Además, se realizó una etapa de verificación como resultado al porcentaje de aciertos para el reconocimiento facial, la cual será mostrado el siguiente capítulo.
- **Programar el sistema embebido (Raspberry Pi), el cual tiene un servidor que se comunicará con la interfaz para el control Usuario-Interfaz.** Al finalizar todo el proceso de programación e implementación, se llega al punto del equilibrio del trabajo de grado. Con el sistema embebido como es la Raspberry Pi, una placa base o también denominada en algunos texto como un micro computador; con éste dispositivo se busca llevar un control de todo el proceso, es el artefacto que nos brinda la comunicación real entre los elementos finales de control del proceso y la interfaz sistema - usuario. Además, las órdenes gestionadas por el usuario a través de la página web, serán captadas por ésta placa para su debido procesamiento en señales eléctricas y llevarlas al tablero eléctrico. Se logra programar éste dispositivo de forma eficiente para su previo uso en en el proyecto.

Cómo últimos objetivos para la demostración de su cumplimiento, se otorgará el desarrollo de los mismo en el siguiente capítulo, en el cual, se tratará de los resultados obtenidos como también, la validación del sistema de manera experimental.

5. Resultados

5.1. Introducción

En esta sección se exponen los resultados obtenidos del entrenamiento de la red neuronal usando la librería Keras/Tensorflow además de las pruebas de funcionamiento de la misma usando OpenCV para la verificación en el PC para posteriormente realizar la verificación final usando el prototipo. También se exponen imágenes de todo el proceso, obteniendo resultados optimos en cada etapa que contempla el trabajo de grado. Por consiguiente, se expresan pequeñas conclusiones relacionadas a las mejoras del prototipo.

5.2. La arquitectura MobileNet Entrenamiento

Tras numerosos intentos con distintas configuraciones de las capas agregadas a la arquitectura MobileNet se obtuvo la gráfica que representan la precisión de la red y los costos de la red a medida que el modelo iba siendo entrenado.



Figura 5.1: Gráfico precisión de MobileNet vs épocas y Pérdidas vs épocas.

Fuente: Autor.

5.3. Verificación del funcionamiento usando OpenCV

Una vez obtenido el modelo entrenado, se procedió a verificar su correcto funcionamiento con ayuda de la librería de visión computacional OpenCV para desarrollar la detección de rostros usando la cámara web e imágenes de distintos individuos. En la figura 5.2 se puede observar imágenes faciales, dichas imágenes no hacen parte del dataset de entrenamiento, son fotos familiares empleadas sólo para la verificación de la red neuronal y en la figura 5.3 se muestra el proceso para la identificación del uso o no del tapabocas en los individuos ya mencionados, junto con mas ejemplos para la verificación con todo el prototipo.



Figura 5.2: Imágenes para la verificación del funcionamiento de la red con OpenCV

Fuente: Autor.



Figura 5.3: Funcionamiento de la red con OpenCV
Fuente: Autor.

En primer lugar, la programación para el entrenamiento de la red neuronal, depende de un dataset de imágenes, el cual, contiene fotos de individuos con o sin mascarillas. Dicho dataset, también involucra a personas de ambos géneros, diferentes razas, edades y otras características.

Para poder verificar que dicho funcionamiento es óptimo se busca implementar éste modelo bajo personas fuera del dataset, para corroborar el funcionamiento del modelo en general. Partiendo de esto se encuentra la figura 5.2 (Imágenes para la verificación del funcionamiento de la red con OpenCV), éstas imágenes representan mi núcleo familiar, son individuos, los cuales, sus rostros no fueron usados para el entrenamiento anteriormente mencionado. Las fotos constan de personas de diversas edades y géneros, además, con y sin tapabocas para un previo procesamiento y obtener el resultado que estamos buscando.

Partiendo de la figura 5.3, se observa el procedimiento que se realiza internamente para obtener un adecuado reconocimiento facial. Consiste en obtener una detección de rostro en la entrada, luego se hace un escalado de ese rostro a un tamaño de 224px*224px, esto con el fin de que la red neuronal tenga la eficacia de realizar su trabajo más rápido y en el menor tiempo posible. Seguidamente de hacer el escalado y pasar la foto por varias líneas de código se llega al resultado general, en éste resultado esperamos dos tipos de respuesta dependientes de un porcentaje de acierto del mismo; ésta dos respuesta son: No mask 99% (No usa mascarilla en un 99% de acierto) ó Mask 99.9% (Usa mascarilla en un 99.9% de acierto). Dicho porcentaje de acierto depende mucho del enfoque de la cámara, de la iluminación del sector y de la calidad del dispositivo, aún así la verificación del modelo con la cámara usada genera buenos resultados.

5.4. Verificación del Funcionamiento del Modelo en el Prototipo

Tras entrenar el modelo con el mejor desempeño, se unificó el código según se expuso en el capítulo anterior. Tras varias pruebas se evidenció una fuerte influencia de la iluminación del entorno, el dispositivo funciona de manera ineficiente cuando no se cuenta con un buen nivel de luz, por lo que es recomendable usarlo solo cuando se tiene un nivel de luminosidad apropiado.

Siguiendo la recomendación mencionada en el párrafo anterior, se realizaron pruebas con el dispositivo con varios individuos (De género masculino y femenino) de edades comprendidas entre los 18 y los 55 años para verificar la robustez del modelo cargado en la tarjeta. Debido a la tarjeta no soporta todo el reconocimiento facial se opta por el uso de un monitor para el desarrollo de éste proceso.

Se puede observar que el modelo se desempeña de forma eficaz. Además, cumple los objetivos planteados, es capaz de realizar un perfecto reconocimiento facial para el uso de la mascarilla, como también, es capaz de unir la respuesta del modelo con los elementos finales de control que están conectados al tablero eléctrico.

Seguidamente, postularé imágenes de la verificación del funcionamiento del prototipo:

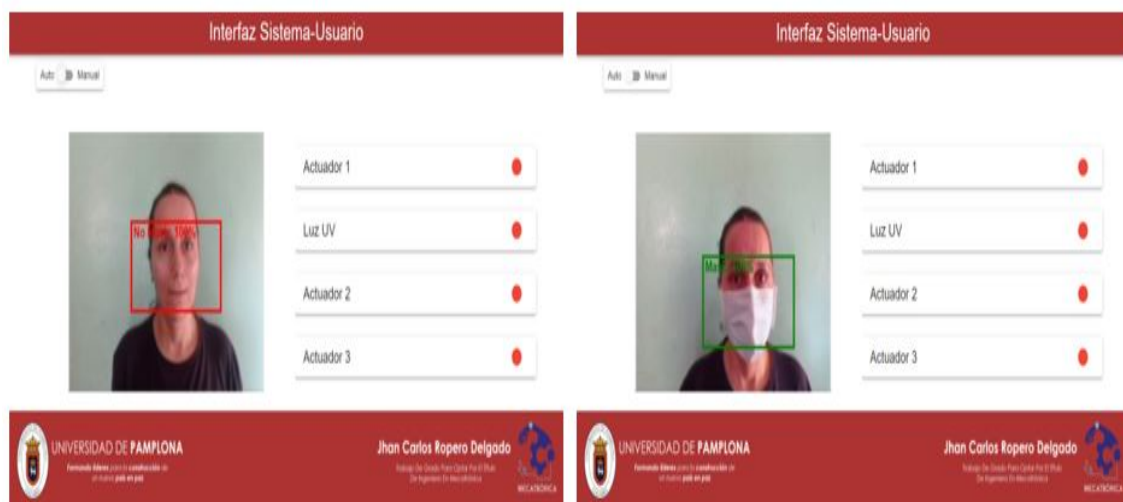


Figura 5.4: Verificación del Funcionamiento del Modelo con una Mujer.

Fuente: Autor.

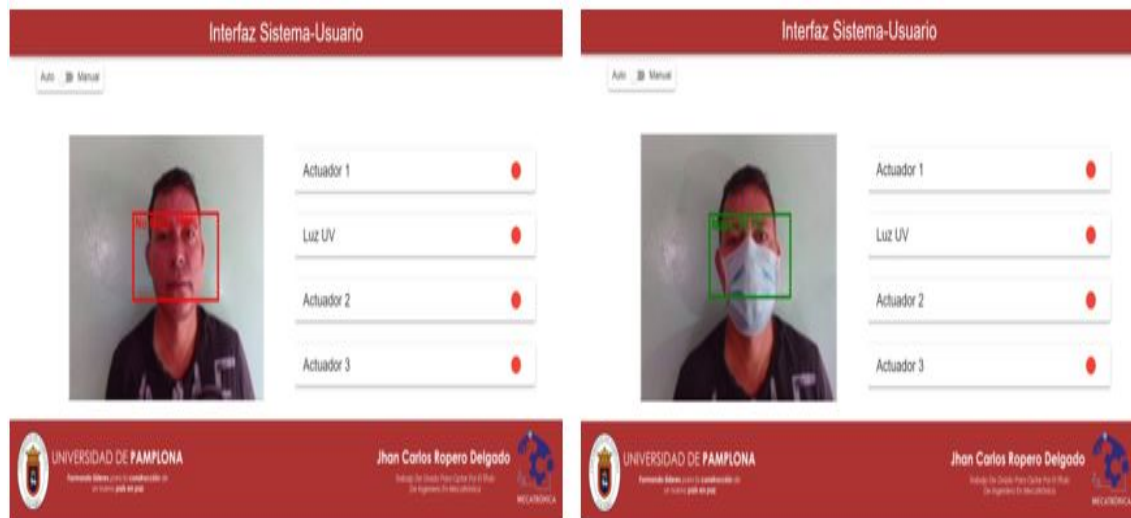


Figura 5.5: Funcionamiento del Modelo y Verificación de los Pilotos Luminosos
Fuente: Autor.



Figura 5.6: Funcionamiento Completo del Modelo.
Fuente: Autor.

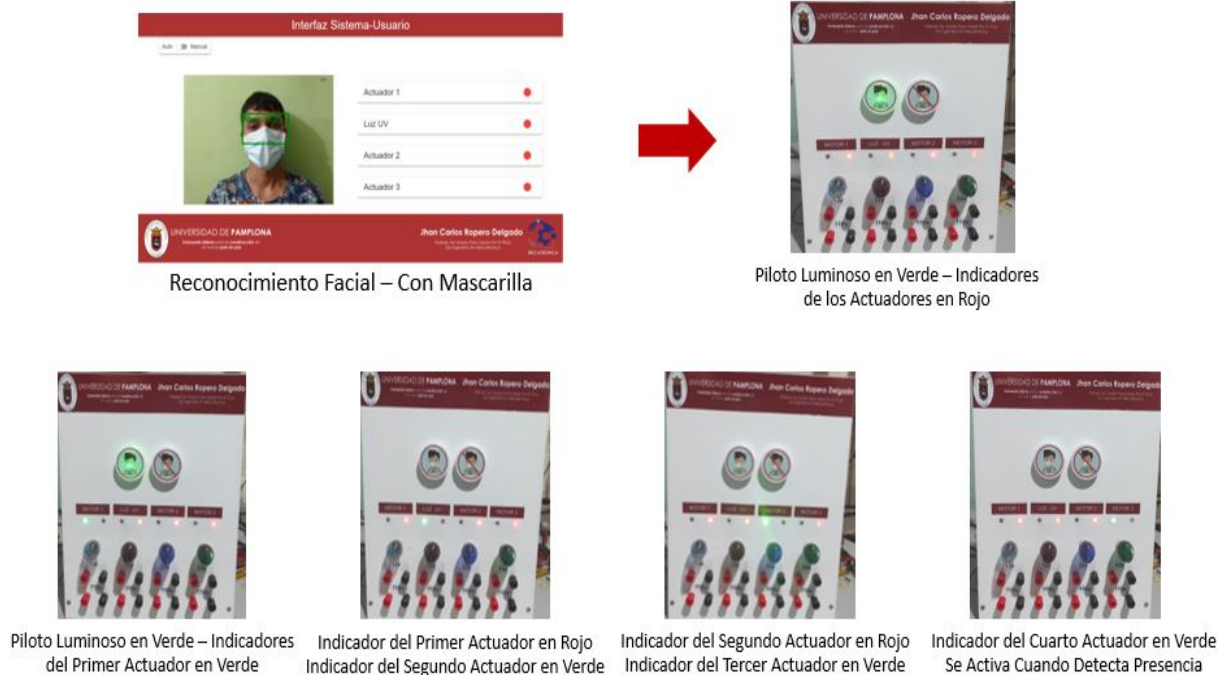


Figura 5.7: Verificación del Funcionamiento del Modelo en el Prototipo.

Fuente: Autor.

Postulando las imágenes anteriormente mostradas, se da un énfasis sobre la verificación del funcionamiento del modelo en el prototipo, pero de forma específica se obtiene:

La Figura 5.4 y Figura 5.5, éstas dos imágenes representan el funcionamiento de forma real de la interfaz, es decir, a través de una cámara para el enfoque del individuo (En éste caso un hombre y mujer) y la tarjeta de vídeo en la interfaz es capaz de simular dicho video y de reconocer el uso o no de mascarillas en las personas.

Por consiguiente, la figura 5.6, se muestra el mismo reconocimineto pero con el plus del enlace con el tablero eléctrico. Sí el individuo NO hace uso de una mascarilla, el sistema lo detecta y automática el tablero eléctrico enciende un piloto luminoso ROJO indicando el acceso denegado al recinto y que por favor haga uso de un tapabocas. Además, en el reconocimiento facial con mascarilla, se visualiza que el sistema detecta el uso del mismo y de igual manera el tablero eléctrico a través de un piloto luminoso nos indica que el acceso es otorgado al recinto en cuestión.

Por último, la Figura 5.7, representa el funcionamiento completo partiendo de la explicación anterior. De forma general, al identificar un usuario con el uso del tapabocas, se enciende el primer piloto luminoso que nos da el acceso al lugar y de forma simultanea el primer actuador (Representa la puerta de Entrada) se activa,

para ello tenemos un led de forma física que nos represente dicha acción y, además, las borneras o salidas de voltajes estarán en funcionamiento para conectar la carga respectiva. Seguidamente, se apaga el primer actuador para darle paso al segundo actuador, el cual tiene la misma explicación anterior, dicho actuador refiere a la luz UV que representa la zona de desinfección obligatoria para acceder al recinto. Continuando éste proceso, se apaga el segundo actuador para otorgar el funcionamiento del tercer actuador (Representa la puerta para entrar al recinto en sí), después de un tiempo dhico actuador se apaga para poder volver al inicio. Sin olvidar que al salir del lugar, se cuenta con un sensor de presencia PIR, el cual será activado cuando un individuo se acerque a la puerta de salida, en ese momento en el tablero eléctrico se visualiza el estado del actuador.



Figura 5.8: Carga de 110v Conectada a la Salida de Voltaje.

Fuente: Autor.

Mostrando la Figura 5.8, se puede observar la prueba de validación con una carga de 110v conectada a una salida de voltaje del tablero eléctrico. De ésta manera se puede tener una visión del funcionamiento a una escala real.

5.5. Otras Pruebas de Funcionamiento

Los resultados mostrados fueron tomados con imágenes ideales, es decir, fotografías con buena iluminación, excelente enfoque y rostro descubierto de accesorios cotidianos usados por las personas.

En éste apartado, se muestran otro tipo de pruebas, en el cual, se verifica el funcionamiento del prototipo; como también, el desempeño de la red neuronal ante otras situaciones que se puedan presentar al implementar el sistema de control remoto en un establecimiento laboral.

5.5.1. Pruebas del Prototipo - Usando Accesorios

Buena Iluminación

En primer lugar, se mostrará el desempeño de la red neuronal dentro de la página web. En estas pruebas se cuenta con accesorios como una gorra, gafas y bufanda; se pretende asimilar al comportamiento de muchos individuos al vestir y con ello tener otro énfasis de los resultados obtenidos.

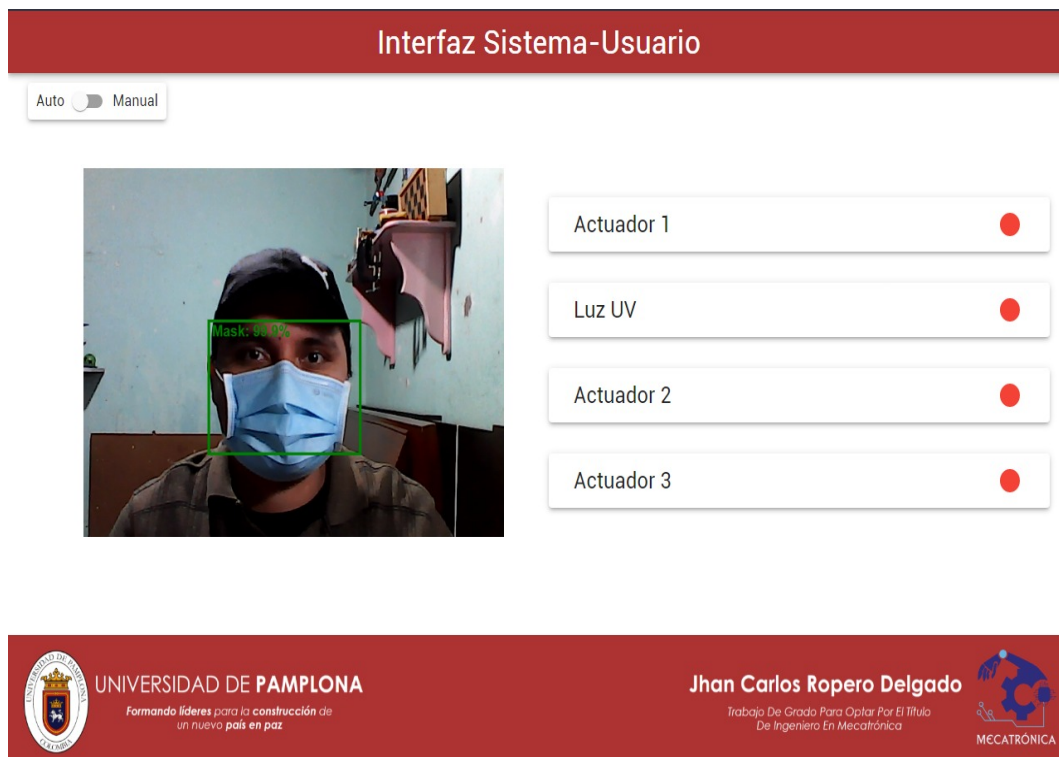


Figura 5.9: Desempeño de la Red Neuronal - Autor con Gorra y Tapabocas.

Fuente: Autor.



Figura 5.10: Desempeño de la Red Neuronal - Autor con Gorra y sin Tapabocas.
Fuente: Autor.



Figura 5.11: Desempeño de la Red Neuronal - Autor con Gorra, Gafas y sin Tapabocas.
Fuente: Autor.

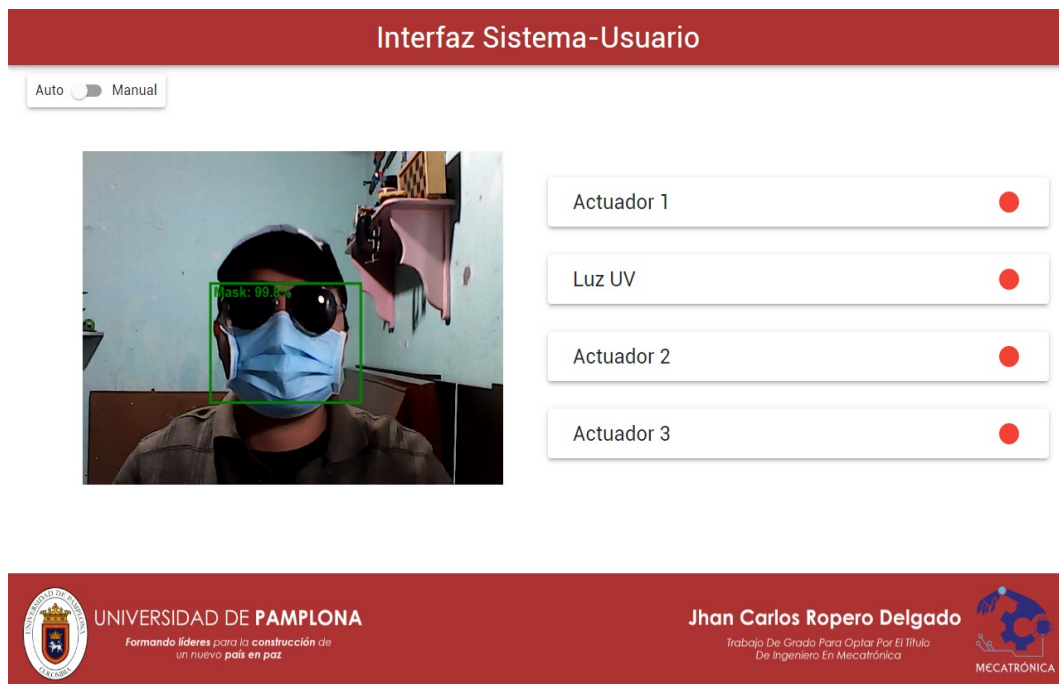


Figura 5.12: Desempeño de la Red Neuronal - Autor con Gorra, Gafas y Tapabocas.
Fuente: Autor.



Figura 5.13: Desempeño de la Red Neuronal - Autor con Bufanda y sin Tapabocas.
Fuente: Autor.



Figura 5.14: Desempeño de la Red Neuronal - Autor con Bufanda y Tapabocas.
Fuente: Autor.

En los casos mostrados desde la figura 5.9 hasta la figura 5.14, se visualia un excelente desempeño de la red neuronal. Se cuenta con accesorios cotidianos como lo son las gorras, gafas y bufandas; aun así, la red es capaz de apreciar el uso o no de tapabocas en las personas.

Mala Iluminación

Ahora, se mostrará el desempeño de la red neuronal dentro de la página web. En estas pruebas se cuenta con los mismos accesorios mencionados anteriormente pero con la diferencia de no tener una fuente de iluminación que ayude a la cámara a tener mejores tomas.



Figura 5.15: Desempeño de la Red Neuronal - Autor sin Tapabocas.
Fuente: Autor.



Figura 5.16: Desempeño de la Red Neuronal - Autor con Tapabocas.
Fuente: Autor.



Figura 5.17: Desempeño de la Red Neuronal - Autor con Gorra y Tapabocas.
Fuente: Autor.



Figura 5.18: Desempeño de la Red Neuronal - Autor con Bufanda y sin Tapabocas.
Fuente: Autor.



Figura 5.19: Desempeño de la Red Neuronal - Autor con Gafas, Bufanda y sin Tapabocas.
Fuente: Autor.



Figura 5.20: Desempeño de la Red Neuronal - Autor Bufanda y Tapabocas.
Fuente: Autor.

Aunque el factor iluminación afecta en gran escala la toma de la fotografía a través de la cámara, se aprecia en los porcentajes de acierto que nos entrega la red neuronal que son un poco bajos, pero, aún así se cumple el objetivo de reconocer el uso o no de tapabocas en las personas.

Casos Hipotéticos

Por último, se muestra un collage de tomas entregadas por la red neuronal, en donde, de forma hipotética hacemos uso inadecuado del tapabocas con el fin de obtener otro ángulo de resultados del desempeño del prototipo.



Figura 5.21: Desempeño de la Red Neuronal - Primer Caso Hipotético.

Fuente: Autor.



Figura 5.22: Desempeño de la Red Neuronal - Segundo Caso Hipotético.
Fuente: Autor.

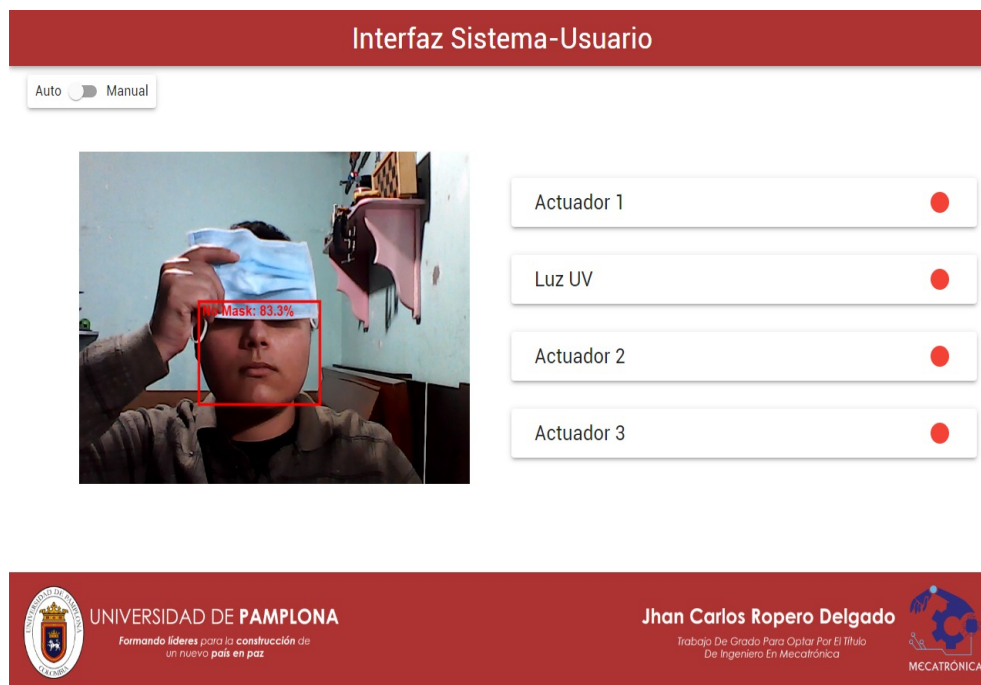


Figura 5.23: Desempeño de la Red Neuronal - Tercer Caso Hipotético.
Fuente: Autor.

Como había mencionado, estos son casos hipotéticos, situaciones abstractas con el objetivo de verificar el funcionamiento del modelo de forma real. En la figura 5.21 el autor hace uso de gafas y de baja iluminación, el porcentaje de acierto de la red es bajo, pero es capaz de identificar que el tapabocas no está en la posición indicada. En los otros dos casos, se tiene una buena iluminación pero se sigue con la premisa de situación hipotética. Una comparación real entre la figura 5.21 y 5.22 es el campo de iluminosidad ante la cámara, siendo así la misma toma a excepción de la variante luminosidad, se aprecia un cambio alto en la respuesta de la red neuronal.

5.6. Recomendaciones

Como recomendaciones para mejorar el prototipo y obtener una implementación más industrial, adaptable, robusto y eficaz, obtenemos:

- **Cámara:** Éste dispositivo es uno de lo más importantes en el proceso, ya que, es el encargado de obtener la imagen de los individuos para un previo reconocimiento facial. Como mejora, se puede cambiar por una cámara de mejor definición y mayor enfoque.
- **Red Neuronal:** El proceso de reconocimiento facial, proviene de una respuesta que nos entrega dicha red ya entrenada. En éste ítem no tengo una mejora, siento que el modelo funciona correctamente, aunque es dependiente de un dataset de imágenes para el entrenamiento y corroboración.
- **Sistema Embebido:** Al trabajar con la Raspberry Pi se obtiene un buen funcionamiento del mismo, pero para obtener un modelo más compacto, sería ideal indagar sobre otros sistemas embebidos que puedan ser cargados con una red neuronal y el proceso previo de control con el tablero eléctrico sin que sea lento el funcionamiento.
- **Componentes:** El presente trabajo de grado tenía como objetivo se desarrollado con componentes de bajo costo. Al obtener el caso contrario, de tener la oportunidad de implementar mejores dispositivos, se obtiene una mejor respuesta del dispositivo.

En general, el prototipo cumple con los objetivos descritos y es capaz de desarrollar la tarea acorde al proceso inmótico.

5.7. Síntesis

Analizando los resultados obtenidos en la implementación del trabajo de grado, se expresan resultados óptimos para una implementación más real.

Las pruebas de funiconamiento realizadas con el prototipo consta de una carga aplicada a las salidas de 110v, en éste caso, se requirió de un bombillo a dicho voltaje para verificar el funcionamiento del mismo.

Debido a la situación del país provocado por la pandemia, no se pudo realizar más pruebas, aumentando la carga para tener otro enfoque del funcionamiento del prototipo. Es decir, tener la posibilidad de conectar un motor eléctrico para ver la evolución que se obtiene con ello. También se hace énfasis que las pruebas tomadas al dispositivo se dificultó con la salida de 12v por no tener un elemento final de control a la mano que funcione con dicho voltaje, aún así, las conexiones se dejaron para esa situación.

Cómo resultado se logra cumplir los objetivos específicos del proyecto, dándole validez al objetivo general.

- **Validar el funcionamiento del sistema de manera experimental.** Al tener la oportunidad de completar el sistema de forma general, obteniendo un prototipo robusto con el plus de un programa informático para la visualización del mismo. Los resultados expuestos, corroboran dicho funiconamiento.
 - **Desarrollar un sistema de control remoto (Aplicación web o App) para la inspección de un proceso inmótico de bajo costo, basado en un código abierto, open-souce, con ayuda de un sistema embebido.** Al terminar todo el proceso comprendido en la implementación del prototipo, se destaca el cumplimiento de éste objetivo, ya que, se logra desarrollar un sistema de control remoto (Internet) para la inspección de un proceso inmótico (Teniendo en cuenta la visión a micro empresas) de bajo costo (Tablero eléctrico con salidas de voltajes estándares para la conexión de elementos finales de control) basado en código abierto, open - souce (JavaScript y Bibliotecas relacionadas) con ayuda de un sistema embebido (Rasberry Pi).
-

6. Conclusiones

Teniendo en cuenta los resultados obtenidos, se determina que el prototipo se desempeña de manera efectiva pese a las limitaciones de hardware que se presentaron en el desarrollo de la propuesta, se logra cumplir con el principal objetivo que es el desarrollo de un sistema de control remoto (Aplicación web o App) para la inspección de un proceso inmótico de bajo costo, basado en un código abierto, open-souce, con ayuda de un sistema embebido.

El modelo de identificación del rostro ofrece un elevado proporción de certeza al instante de ejecutar su labor, y esto se debe a que realiza un buen procesamiento utilizando el dataset de fotografías con el procedimiento de transfer-learning, pero estos valores en el porcentaje de efectividad al período de la identificación varían, pudiendo reducir debido a diversos factores como lo es la luminosidad así como el enfoque a la cara al momento de proceder a la extracción del retrato facial previo el estudio de características de la persona que circula por el sistema de identificación.

Una de las principales limitantes es el uso de una cámara con baja resolución por cuestiones económicas, aun así el sistema embebido implementado es compatible con esta referencia. Teniendo esto en cuenta se puede decir que la red neuronal convolucional MobileNet entrenada con el método de transfer-learning se desempeñó de manera eficaz en el prototipo.

También se evidenció que el dispositivo cumple con los requerimientos para la creación de un dispositivo de inspección inmótica al recibir críticas positivas por ciertos usuarios con pequeños emprendimientos con quienes se dialogó el experimento y se les expuso el prototipo, presentando un buen nivel de adaptabilidad.

Finalmente, el desempeño general del dispositivo puede mejorarse simplemente cambiando los componentes del hardware por unos de mayor calidad, manteniendo la lógica de funcionamiento junto con el proceso de reconocimiento facial. Dicho cambio contribuiría a un mejor desempeño del sistema y a un aumento en la adaptabilidad que brinda al usuario al usarlo.

Bibliografía

- [1] J. Gutiérrez. “la inmótica como garantía de una infraestructura eficiente, dinamó técnica: revista gallega de energía”, 2015.
- [2] M. Zastrow. J. H. Tanne, E. Hayasaki. Covid-19: How doctors and healthcare systems are tackling coronavirus worldwide. marzo, 2020.
- [3] World Health Organization. Who coronavirus disease (covid-19) situation report-114. 2020.
- [4] N. Fernandes. Economic effects of coronavirus outbreak (covid-19) on the world economy nuno fernandes full professor of finance iese business school spain. 2020.
- [5] Somolinos Sanchez. “avances en robótica y visión por computador/coordinador”, 2002, .
- [6] P. Castro. Sistema embebido para la supervisión y adquisición de datos de un control inmótico. 2016.
- [7] Ministerio de Salud. Medida frente a la pandemia covid 19, 2021.
- [8] P. J. Phillips y A. Rosenfeld W. Zhao, R. Chellappa. Face recognition: A literature survey. acm computing surveys, vol. 35, nº 4, pp. 399-458, 2003.
- [9] J. Agudelo. Tecnicas de biometria basadas en patrones faciales del ser humano, 2012.
- [10] B. Pérez. Reconocimiento facial basado en puntos característicos de la cara en entornos no controlados, madrid: Universidad autónoma de madrid, 2013.
- [11] V. Janga. et al. Advanced machine learning-based implementation patterns for computer vision and real-time multimedia applications. 2020., .
- [12] M. Momeny. A noise robust convolutional neural network for image classification, 2021.
- [13] V. Villacampa. Diseño y montaje de un sistema inmótico para una infraestructura laboral. 2007.
- [14] M. Lledó. Diseño de un sistema de control domótico basado en la plataforma arduino. 2012.

- [15] Barrera. Digital life. 2013.
 - [16] Systems International. Análisis y diseño de un prototipo de sistema domótico de bajo costo. 2012.
 - [17] R. Gimeno and J. Ramos. Estudio de técnicas de reconocimiento de patrones, barcelona: Universidad politécnica de catalunya, 2010.
 - [18] K. Delac, M. Grgic, and P. Liatsis. Appearance-based statistical methods for face recognition. 2013.
 - [19] Miranda. Facilidades de los sistemas embebidos. 2014.
 - [20] E. Cerezo. S. Baldassarri. et al. Agentes virtuales 3d para el control de entornos inteligentes domóticos. inxiii congreso internacional de interacción persona-ordenador(pp. 363-372). 2018, .
 - [21] R. Gonzalez; R. Woods. Digital image processing, second edition. editorial prentice-hall, 2002.
 - [22] J. Cancelas. Introducción a la visión por computador, standard, en 100-732, 2010.
 - [23] R. Pinto y J. Sossa. Detección de rostros utilizando descriptores invariantes y algoritmos genéticos, universidad del valle, méxico, pp. 32-41,2013.
 - [24] Etitudela. Componentes de un sistema de visión artificial. 2017.
 - [25] J. Sanchez. Avances en la robotica y vision por computadora, casilla: Universidad casilla de la mancha, 2002., .
 - [26] Y. Molly and William F. An overview of artificial intelligence in education. 2014.
 - [27] G. Tecuci. Artificial intelligence. 2012.
 - [28] S. Shwartz and B. Shai. Understanding machine learning: From theory to algorithms. 2013.
 - [29] C. Villani. What is artificial intelligence. 2018.
 - [30] J. Matich. Redes neuronales: Conceptos básicos y aplicaciones. 2001.
 - [31] B. Caicedo y J. Lopez. Redes neuronales artificiales. 2010.
 - [32] F. Izaurieta. Redes neuronales artificiales. 1999.
 - [33] L. Pigou and D. Sander. Sign language recognition using convolutional neural networks. 2015.
-

-
- [34] E. Varela y E. Campbells. *Redes neuronales artificiales: Una revisión del estado del arte, aplicaciones y tendencias futuras*. 2011.
- [35] M. Campos. *Inspiración biológica de las redes neuronales artificiales*. 2020.
- [36] E. Rivera. *Introducción a redes neuronales artificiales*. 2007.
- [37] S. Albawi. *Understanding of a convolutional neural network*. 2018.
- [38] R. Yamashita. *Convolutional neural networks: an overview and application in radiology*. 2018.
- [39] I. Namatevs. *Deep convolutional neural networks: Structure, feature extraction and training*. 2018.
- [40] S. Sakib2018. *An overview of convolutional neural network: Its architecture and applications*. 2018.
- [41] A. Howard. *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. 2017.
- [42] E. Redmon. *Pinel et esquiro: Quelques commentaires sur les debuts d'une amitie*. 1976.
- [43] J. P. Marques de Sá. *Pattern recognition. concepts, methods and applications, oporto, portugal: Springer*, 2001.
- [44] Universidad Politécnica de Madrid UPM-DISAM. *Reconocimiento de patrones dpto. automática, ingeniería electrónica e informática industrial*. 2019.
- [45] R. C. Gonzalez y R. E. Woods. *Digital image processing, ney jersey: Pearson pretince hall*, 2008.
- [46] P. E. Hart y D. G. Stork R. O. Duda. *Pattern clasificación, a wiley international publication*, 2001.
- [47] Gordon. *Diseño e implementación de un sistema embebido denominado raspberry pi*. 2011.
- [48] I. Goodfellow. *Deep learning, (adaptive computation and machine learning series)*. 2015.
- [49] Dimitry. *Image recognition with k210 boards and arduino ide micropython*. 2015.
- [50] Kaggle. *Challenges in representation learning: Facial expression recognition challenge*. 2019.
- [51] R. López y E. Sotter. *Operación automática eléctrica y de control, laboratorio de robótica y producción automática. universidad del norte*. 2014.
-

A. Apéndice

En éste capítulo, referido como apéndice o anexos del libro sobre la investigación presente, abarca los códigos fuentes y funciones de los mismos para el funcionamiento digital del prototipo.

En primer lugar, se aborda la temática del reconocimiento facial con el uso de las redes neuronales, para ello, se explica de forma general con la siguiente figura.

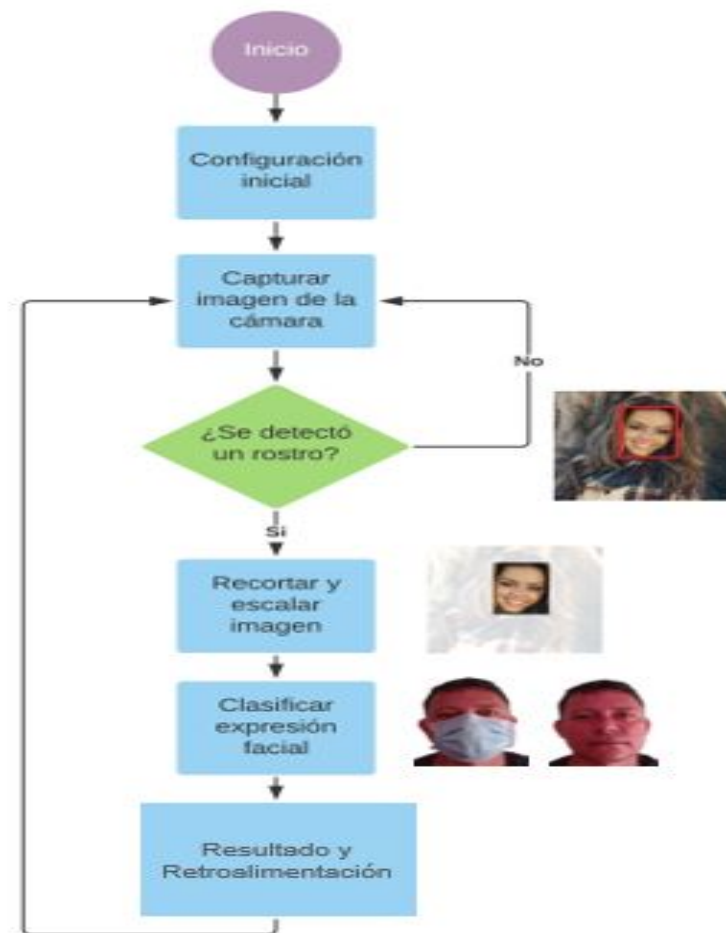


Figura A.1: Diagrama de Flujo - Algoritmo de Reconocimiento Facial.

Fuente: Autor.

Según la Figura A.1, representa un diagrama de flujo, es un esquema que de forma general muestra las fases más importante que tiene el código de programación en éste ítem. Éste contexto se integra o hace parte de la interfaz de forma implícita, ya que, todo éste reconocimiento se realiza a través de la tarjeta de vídeo que se visualiza en dicho sistema.

De forma general, el contexto principal es el relacionado con la interfaz gráfica, ya que, contiene todo el sistema de inspección y de mando sobre el prototipo, para ello se cuenta con el siguiente diagrama de flujo.

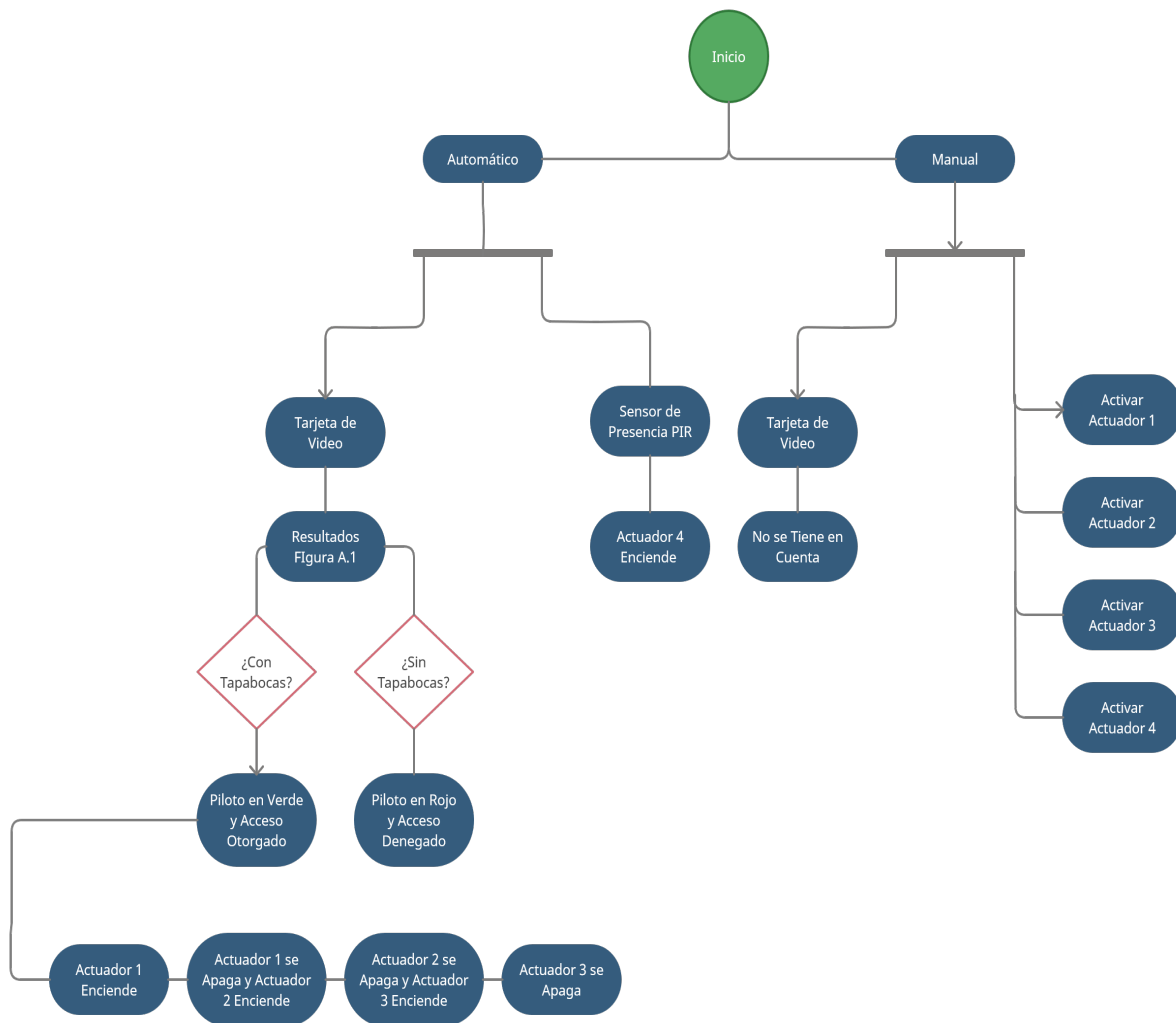


Figura A.2: Diagrama de Flujo - Funcionamiento de la Interfaz Gráfica.

Fuente: Autor.

Teniendo en cuenta el diagrama de flujo de la Figura A.2, se denota el proceso funcional por parte de la interfaz, a través de ella, se puede visualizar el estado del proceso y de los elementos finales de control cómo también, es la etapa de mando para el correcto funcionamiento de los actuadores.

Seguidamente, adjuntaré las líneas de código que se implementaron durante el desarrollo de todo el proceso, líneas de código que colocan en marcha todo el prototipo.

A.1. Programación de la Interfaz Gráfica

Código A.1: Algoritmo Completo Programación Interfaz

```

1 import './App.css';
2 import { useEffect, useState } from 'react'
3 import { Container, Grid, makeStyles } from '@material-ui/core'
4
5 import Navbar from './components/Navbar';
6 import Controls from './components/Controls'
7 import StateConfiguration from './components/StateConfiguration'
8 import VideoCard from './components/VideoCard'
9 import Footer from './components/Footer'
10 import { MOTOR_1, MOTOR_2, MOTOR_3, URL_SERVER } from './constantes';
11
12 const useStyles = makeStyles((theme) => ({
13   marginTop: {
14     marginTop: "30px"
15   }
16 })))
17
18 function App() {
19   const [isAuto, setIsAuto] = useState(false)
20   const [estadoPuertas, setEstadoPuertas] = useState([0, 0, 0])
21   const [estadoLuzUV, setEstadoLuzUV] = useState(0)
22   const classes = useStyles()
23
24   const getEstadoPuerta = (id) =>
25     fetch(`${URL_SERVER}/puertas/GetEstadoPuerta/${id}`).then(res => res.json())
26
27
28   const getEstadoLuzUV = () =>
29     fetch(`${URL_SERVER}/luzuv/GetEstadoLuzUV`).then(res => res.json())
30
31
32   const getEstadoAllPines = () => {
33     fetch(`${URL_SERVER}/pines/GetEstadoAllpines`)
34       .then(res => res.json())
35       .then(({ estadoPuertas, estadoLuzUV }) => {
36         setEstadoPuertas(estadoPuertas)
37         setEstadoLuzUV(estadoLuzUV)
38       })
39   }
40
41   useEffect(getEstadoAllPines, [])
42
43
44   const handleChange = (e) => setIsAuto(e.target.checked)
45
46   return (

```

```

47     <>
48     <Navbar />
49     <Container maxWidth="lg">
50       <StateConfiguration callback={handleChange} />
51       <div className={classes.marginTop}>
52         <Grid container spacing={3} alignItems="center">
53           <Grid item xs={6}>
54             <VideoCard />
55           </Grid>
56           <Grid item xs={6}>
57             <Controls auto={isAuto}
58               estadoPuertas={estadoPuertas}
59               estadoLuzUV={estadoLuzUV}
60               setEstadoPuertas={setEstadoPuertas}
61               setEstadoLuzUV={setEstadoLuzUV} />
62           </Grid>
63         </Grid>
64       </div>
65     </Container>
66     <Footer />
67   </>
68 );
69 }
70
71 export default App;

```

Según el código A.1 - Algoritmo Completo Programación Interfaz, se visualiza la importación de los componentes que hacen parte de cada sección de la interfaz, es decir, el funcionamiento de cada ícono se programa independiente para poder encontrar errores fácilmente y estructurarlo en forma de funciones para un previo uso en el programa principal.

Ejemplo de la Programación de los Componentes para la Interfaz Gráfica

Código A.2: Ejemplo - Algoritmo una Función de la Interfaz Gráfica

```

1 import { Grid, makeStyles } from '@material-ui/core'
2 import { useEffect } from 'react'
3 import Control from './Control'
4 import { URL_SERVER, PUERTA_1, PUERTA_2, PUERTA_3, LUZ_UV } from '../constantes'
5
6 const useStyles = makeStyles((theme) => ({
7   root: {
8   }
9 })))
10
11 export default function Controls({ auto, estadoPuertas, estadoLuzUV, setEstadoPuertas, setEstadoLuzUV }) {
12   const classes = useStyles()
13
14   const setEstadoPuerta = (idPuerta, newEstado) => {
15     fetch(`${URL_SERVER}/puertas/SetEstadoPuerta/${idPuerta}`, {
16       method: 'PUT',
17       headers: {
18         'Content-Type': 'application/json'
19       },
20       body: JSON.stringify({
21         estado: newEstado
22       })

```

```

23   })
24   .then(res => res.json())
25   .then(res => {
26     setEstadoPuertas(prev => {
27       const newState = [...prev]
28       newState[idPuerta] = res.estadoPin
29       return newState
30     })
31   })
32 }
33
34 const setEstadoLuz = (newEstado) => {
35   console.log(newEstado)
36   fetch(`${URL_SERVER}/luzuv/SetEstadoLuzUV`, {
37     method: 'PUT',
38     headers: {
39       'Content-Type': 'application/json'
40     },
41     body: JSON.stringify({
42       estado: newEstado
43     })
44   })
45   .then(res => res.json())
46   .then(res => {
47     setEstadoLuzUV(res.estadoPin)
48   })
49 }
50
51 return (
52   <Grid container direction='column' spacing={4} className={classes.root}>
53     <Grid item>
54       <Control title='Actuador 1' activated={estadoPuertas[PUERTA_1]} auto={auto} onClick={(newEstado) => setEstadoPuertas(newEstado)} />
55     </Grid>
56     <Grid item>
57       <Control title='Luz UV' activated={estadoLuzUV} auto={auto} onClick={(newEstado) => setEstadoLuz(newEstado)} />
58     </Grid>
59     <Grid item>
60       <Control title='Actuador 2' activated={estadoPuertas[PUERTA_2]} auto={auto} onClick={(newEstado) => setEstadoPuertas(newEstado)} />
61     </Grid>
62     <Grid item>
63       <Control title='Actuador 3' activated={estadoPuertas[PUERTA_3]} auto={auto} onClick={(newEstado) => setEstadoPuertas(newEstado)} />
64     </Grid>
65   </Grid>
66 )
67 }

```

Para tener una idea sobre la programación de una de las funciones, se adjuntó el código anterior (Código A.2) el cual, representa la programación de forma automática/manual sobre los actuadores, al ser de forma automática se visualiza el estado de dicho elemento final de control y/o de forma manual por medio de peticiones con el servidor al cual se le otorga la acción a proceder.

A.2. Programación Sistema Embebido

Código A.3: Algoritmo Completo Programación Sistema Embebido

```

1
2 import express from 'express'
3 import cors from 'cors'
4 import lowDb from 'lowdb'
5 import FileSync from 'lowdb/adapters/FileSync.js'
6 import bodyParser from 'body-parser'
7 import { Gpio } from 'onoff'
8
9 import puertRoutes from './routes/puertas.js'
10 import luzUVRoutes from './routes/luzUV.js'
11 import pilotosRoutes from './routes/pilotos.js'
12 import allPinesRoutes from './routes/allPines.js'
13 import hacerSecuenciaRoute from './routes/secuencia.js'
14
15 export const db = lowDb(new FileSync('db.json'))
16
17 db.defaults({
18   puertas: [
19     { id: 0, pin: 4 },
20     { id: 1, pin: 5 },
21     { id: 2, pin: 6 }
22   ],
23   luzuv: { id: 1, pin: 7 }
24 }).write()
25
26 const app = express()
27
28 app.use(bodyParser.json())
29 app.use(cors())
30
31 app.use('/puertas', puertRoutes)
32 app.use('/luzuv', luzUVRoutes)
33 app.use('/pilotos', pilotosRoutes)
34 app.use('/pines', allPinesRoutes)
35 app.use('/secuencia', hacerSecuenciaRoute)
36
37 const PORT = 4000
38
39 // inicializar puertas
40 const infoPuertas = db.get('puertas').value()
41 export const puertas = []
42 for (let i in infoPuertas) {
43   const { id, pin, ledActivo, ledInactivo } = infoPuertas[i]
44   const pines = {
45     id,
46     pin: new Gpio(pin, 'out'),
47     ledActivo: new Gpio(ledActivo, 'out'),
48     ledInactivo: new Gpio(ledInactivo, 'out'),
49   }
50   puertas.push(pines)
51 }
52
53 // inicializar Luz uv
54 const infoLuzUV = db.get('luzuv').value()
55 const { id, pin, ledActivo, ledInactivo } = infoLuzUV
56 export const luzUV = {
57   id,
58   pin: new Gpio(pin, 'out'),

```

```
59 ledActivo: new Gpio(ledActivo, 'out'),
60 ledInactivo: new Gpio(ledInactivo, 'out'),
61 }
62
63 // inicializar pilotos tapabocas
64 const infoPilotos = db.get('pilotos').value()
65 const { conTapabocas, sinTapabocas } = infoPilotos
66 export const pilotos = {
67   conTapabocas: new Gpio(conTapabocas, 'out'),
68   sinTapabocas: new Gpio(sinTapabocas, 'out'),
69 }
70
71 // Configuración inicial pines
72 puertas.forEach(puerta => {
73   const { pin, ledActivo, ledInactivo } = puerta
74   pin.writeSync(0)
75   ledActivo.writeSync(0)
76   ledInactivo.writeSync(1)
77 })
78
79 const { pin: pinluzuv, ledActivo: ledActivoluzuv, ledInactivo: ledInactivoluzuv } = luzUV
80 pinluzuv.writeSync(0)
81 ledActivoluzuv.writeSync(0)
82 ledInactivoluzuv.writeSync(1)
83
84 pilotos.conTapabocas.writeSync(0)
85 pilotos.sinTapabocas.writeSync(0)
86
87 app.listen(PORT, () => console.log('listening on port 4000'))
```

El algoritmo titulado Código A.3, expone la programación cargada en la placa o sistema embebido (Raspberry Pi), a través de éste código se genera la comunicación de forma inalámbrica con la interfaz. Además, se encarga de recibir las señales del sistema de mando para luego dar la orden al tablero eléctrico como elementos finales de control.