



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DE UN SISTEMA
DE CONTROL DE USO DE MASCARILLAS Y TEMPERATURA
CORPORAL DE PERSONAS PARA EL INGRESO A ÁREAS
CERRADAS EN EL CONTEXTO DEL COVID-19”

PROYECTO INTEGRADOR

Previo la obtención del Título de:

INGENIERO EN TELECOMUNICACIONES

Presentado por:

MARÍA DEL SOL PALMA NÚÑEZ

DIEGO ANDRÉS RAMOS BAJAÑA

GUAYAQUIL – ECUADOR

AÑO: 2021

DEDICATORIA

Esta tesis se la dedico a mi querida madre desde lo más profundo de mi corazón por su amor incondicional y paciencia infinita que me forjaron como una profesional íntegra y llena de decisión para cumplir todas las metas que me proponga. En todo momento sostuviste mi mano y no la soltaste, te quiero mucho, este triunfo es tuyo.

A mis hermanos Andrés y Michael por estar siempre ahí y nunca dudar de mí. A Natasha, por cuidarme y acompañarme en todo momento.

María del Sol

Esta tesis se la dedico a mis padres por todo el amor y apoyo que han dado toda mi vida, sin ellos no sería la persona que soy ahora. La dedico en memoria de mi abuelo Ángel quien siempre me alentó y me dio un ejemplo de superación, fuerza y bondad.

A mis hermanos, en especial a Deyaneira por siempre darme consejos y ayudarme cada vez que lo necesité y Sarahi por darme ánimos.

Diego

AGRADECIMIENTOS

Queremos agradecer a nuestros padres que nos apoyaron y guiaron en cada paso de este trabajo de titulación con su sabiduría, amor y mucha paciencia, nada sería lo mismo sin ustedes.

A nuestro profesor de materia integradora, PhD. Washington Medina Moreira, por su dedicación y consejos brindados en la dirección de este trabajo con sus ideas y sugerencias.

A la Escuela Superior Politécnica del Litoral, sus profesores de la carrera de Telecomunicaciones que creyeron en nosotros y fueron parte esencial de nuestra carrera como profesionales.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; María del Sol Palma Núñez y Diego Andrés Ramos Bajaña damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



María del Sol Palma Núñez



Diego Ramos Bajaña

EVALUADORES



**Medina Moreira Washington
Adolfo**
PROFESOR DE LA MATERIA



**Vargas López German
Ricardo**
PROFESOR TUTOR

RESUMEN

Con las nuevas medidas con respecto a la pandemia, cada vez más ciudadanos están regresando a sus actividades presenciales, pero pocos manteniendo los protocolos de salud sugeridos con lo que se tiene el riesgo de un posible rebrote. Es por ello que se plantea el siguiente sistema de control del uso de mascarillas y temperatura para el ingreso a diversos espacios cerrados.

El escenario se basa en que cuando un usuario se acerca a la puerta de entrada de un área concurrida y cerrada del hospital, la cámara del detector de mascarillas y el sensor térmico verifican si el visitante no tiene fiebre y si lleva una mascarilla. Los usuarios pueden ingresar de manera conveniente y rápida sin tomar una temperatura corporal separada o usar una etiqueta de tarjeta de acceso. Esta solución resuelve el inconveniente acceso a áreas cerradas debido a la pandemia y sus mutaciones, además que en hospitales de Ecuador este sistema no se ha implementado.

Este proyecto permite controlar el acceso a áreas cerradas de personas que no toman medidas de seguridad como mascarillas y además que su temperatura corporal sea mayor al límite establecido, si no se cumplen alguna de estas dos condiciones se generara una alerta roja. Además, mediante una pantalla LCD se puede observar a tiempo real la detección facial de la mascarilla.

La característica más importante de este sistema es que puede bloquear la infección secundaria de virus como 'COVID-19' en el hospital, así como aliviar la fatiga del personal de control de acceso. En particular, es posible reducir el tiempo de acceso de los pacientes hospitalizados, por lo que se puede esperar una alta satisfacción, como conveniencia y seguridad, en términos de experiencia del paciente.

ABSTRACT

With the new measures regarding the pandemic, more and more citizens are returning to their face-to-face activities, but few maintaining the suggested health protocols with which there is a risk of a possible re-outbreak. That is why the following system of control of the use of masks and temperature for the entrance to various enclosed spaces is proposed.

The scenario is based on the fact that when a user approaches the front door of a busy and closed area of the hospital, the mask detector camera and thermal sensor check if the visitor does not have a fever and if he is wearing a mask. Users can enter conveniently and quickly without taking a separate body temperature or using an access card tag. This solution solves the inconvenience of access to closed areas due to the pandemic and its mutations, in addition to the fact that in hospitals in Ecuador this system has not been implemented.

This project allows to control the access to closed areas of people who do not take security measures such as masks and also that their body temperature is higher than the established limit, if one of these two conditions is not met, a red alert will be generated. In addition, through an LCD screen you can observe in real time the facial detection of the mask.

The most important feature of this system is that it can block the secondary infection of viruses such as 'COVID-19' in the hospital, as well as relieve the fatigue of access control staff. In particular, it is possible to reduce the access time of hospitalized patients, so high satisfaction, such as convenience and safety, can be expected in terms of patient experience.

ÍNDICE GENERAL

EVALUADORES.....	5
RESUMEN.....	I
ABSTRACT.....	II
ÍNDICE GENERAL	III
ABREVIATURAS.....	VI
SIMBOLOGÍA.....	VII
INDICE DE FIGURAS	VIII
INDICE DE TABLAS	X
CAPÍTULO 1.....	1
1.1 DEFINICIÓN DEL PROBLEMA	1
1.2 OBJETIVOS.....	2
1.2.1 OBJETIVO GENERAL.....	2
1.2.2 OBJETIVOS ESPECÍFICOS.....	2
1.3 ESTADO DEL ARTE.....	2
1.4 PROPUESTA DE SOLUCIÓN.....	7
1.5 JUSTIFICACIÓN.....	9
1.6 ALCANCE.....	10
1.7 METODOLOGÍA	10
1.8 DIAGRAMA DE BLOQUES DE LA SOLUCIÓN PROPUESTA.....	11
CAPÍTULO 2.....	13
2. MARCO TEÓRICO	13
2.1 MACHINE LEARNING (ML)	13
2.2 LENGUAJES DE PROGRAMACIÓN PARA MACHINE LEARNING.....	15
2.3 LIBRERÍAS DE PYTHON.....	16
2.4 REDES NEURONALES CONVOLUCIONALES (CNNs).....	17

2.5	DETECCIÓN FACIAL EN MACHINE LEARNING	17
2.6	IA EN LA SALUD HUMANA.....	17
2.7	TEMPERATURA Y MASCARILLAS COMO PREVENCIÓN DEL COVID-19	18
2.8	MICROORDENADORES.....	19
2.10.1	CARACTERÍSTICAS DE LA MICROCOMPUTADORA RASPBERRY PI	20
2.9	SISTEMAS OPERATIVOS PARA RASPBERRY PI	21
2.10	MÉTODOS DE ENTRENAMIENTO EN DE MACHINE LEARNING	21
2.12.1	APRENDIZAJE SUPERVISADO	21
2.11.2	APRENDIZAJE NO SUPERVISADO	22
2.11.3	APRENDIZAJE SEMI SUPERVISADO	23
2.11	MODELOS PARA DETECCIÓN DE OBJETOS CON MACHINE LEARNING	23
2.13.1	MODELO R-CNN.....	23
2.13.2	MODELO YOLO	24
2.12	SENSOR DE TEMPERATURA INFRARROJO.....	25
2.13	TIPOS DE CÁMARAS WEB.....	26
2.14	SALUD EN HOSPITALES DEL ECUADOR Y EL MUNDO	26
CAPÍTULO 3.....		28
3.	DISEÑO DE LA SOLUCIÓN	28
3.1	DIAGRAMA DE BLOQUES BÁSICO.....	28
3.2	DIAGRAMA DE BLOQUES DETALLADO	29
3.3	PREPARACIÓN DEL MODELO	31
3.4	ENTRENAMIENTO DEL MODELO	33
3.5	DIAGRAMA DE CONEXIONES	35
3.6	IMPLEMENTACIÓN DEL CÓDIGO.....	36
3.7	IMPLEMENTACIÓN EN PROTOTIPO.....	40
CAPÍTULO 4.....		47
4.	ANÁLISIS DE RESULTADOS.....	47
4.1	ESCENARIO.....	47

4.2 GRÁFICAS	53
CONCLUSIONES	57
RECOMENDACIONES	58
BIBLIOGRAFÍA	59

ABREVIATURAS

COVID-19	Coronavirus Disease of 2019
LCD	Liquid Cristal Display
YOLO	You Only Look Once
CNNs	Convolutional Neural Networks
RGB	Red Green Blue
FPS	Frames Per Second
GPU	Graphics Processing Unit
GPIO	General Purpose Input/Output
USB	Universal Serial Bus
HDMI	High-Definition Multimedia Interface
VGA	Video Graphics Adapter
COE	Centro de Operaciones de Emergencia
I2C	Inter-Integrated Circuit
SDA	System Data
SCL	System Clock
LED	Light Emitting Diode
UCI	Unidad de Cuidados Intensivos
IT	Information Technology
IA	Inteligencia Artificial
IR	Infrared
IoT	Internet of Things
ML	Machine Learning

SIMBOLOGÍA

m	Metro
cm	Centímetros
°C	Grados Celsius
%	Probabilidad de Detección
#	Épocas

ÍNDICE DE FIGURAS

Ilustración 1.1 Diagrama de la solución propuesta.....	7
Ilustración 1.2. Diagrama de bloques de la solución.....	11
Ilustración 2.1 Personal de un hospital en Michigan ingresando a urgencias...27	27
Ilustración 3.1. Diagrama de bloques simplificado.....	28
Ilustración 3.2 Diagrama de bloques detallado de la solución.....	29
Ilustración 3.3. Banco de imágenes de personas con mascarilla.....	30
Ilustración 3.4 Banco de imágenes de personas sin mascarilla.....	30
Ilustración 3.5 Gráfico esquemático general de la solución.....	31
Ilustración 3.6 Etapas de un detector de objetos.....	32
Ilustración 3.7. Etiquetado de rostro con mascarilla.....	33
Ilustración 3.8. Etiquetado de rostro sin mascarilla.....	34
Ilustración 3.9 Parámetros para adaptar el modelo a los datos procesados.....	35
Ilustración 3.10 Diagrama de conexiones general.....	36
Ilustración 3.11 Comandos de la librería de adafruit.....	38
Ilustración 3.12 Comando con condición de temperatura.....	38
Ilustración 3.13 Comandos con condiciones de mascarilla y temperatura.....	39
Ilustración 3.14. Visualización del frame cuando no se usa mascarilla.....	39
Ilustración 3.15. Visualización del frame cuando se usa mascarilla.....	40
Ilustración 3.16 Conexiones esenciales de la Raspberry.....	41
Ilustración 3.17 Instalación de recursos, sistema operativo y librerías.....	41
Ilustración 3.18 Diseño de código.....	41
Ilustración 3.19 Configuración de pantalla LCD.....	42
Ilustración 3.20 Implementación en Raspberry.....	42
Ilustración 3.21 Incorporación de alarma al sistema.....	43

Ilustración 3.22 Programación e implementación de alarma al sistema.....	43
Ilustración 3.23 Error del código debido a la calidad de la imagen.....	44
Ilustración 3.24 Pruebas realizadas con el prototipo.....	45
Ilustración 3.25 Pruebas realizadas con el prototipo.....	46
Ilustración 4.1 Prototipo ubicado en el hospital.....	47
Ilustración 4.2 Ubicación de las pruebas hechas junto a enfermería.....	48
Ilustración 4.3 Toma de datos medidos y nombres de los usuarios.....	48
Ilustración 4.4 Desarrollo de las pruebas.....	49
Ilustración 4.5 Sensor digital infrarrojo de temperatura.....	50
Ilustración 4.6 Medición de temperatura.....	50
Ilustración 4.7 Medición de distancia de detección de rostro.....	53
Ilustración 4.8 Grafica de Epoch vs Loss/Accuracy del modelo generado.....	53
Ilustración 4.9 Grafica de comparación de eficiencia de detección.....	55
Ilustración 4.10 Gráfica de error porcentual de temperatura.....	55

ÍNDICE DE TABLAS

Tabla 3.1 Condiciones de prueba del funcionamiento del prototipo.....	45
Tabla 4.1 Porcentaje de exactitud de detección de mascarilla.....	49
Tabla 4.2 Temperatura corporal con medida con el Prototipo.....	51
Tabla 4.3 Temperatura medida con el Termómetro digital Infrarrojo.....	51
Tabla 4.4 Distancia de webcam para detección de mascarillas.....	52

CAPÍTULO 1

1.1 DEFINICIÓN DEL PROBLEMA

Se sabe que varios grupos de ciudadanos y grupos anti-mascarillas en el mundo no desean seguir diversos protocolos de salud en áreas cerradas debido a pensamientos como “a muchos nos molesta que las autoridades nos digan qué hacer”, “falsa pandemia” o “yo sencillamente los odio” y por ende se oponen a medidas que se ha demostrado que salvan vidas [1].

Por estas razones, diversos teóricos de la conspiración, libertarios y antivacunas pasan por alto estas medidas que controlan y prevén una mayor propagación de la pandemia, siendo los protestantes hasta más de 15.000 personas en países como Estados Unidos, Alemania, Canadá, Reino Unido y Francia [2].

Debido a los nuevos protocolos de seguridad con respecto a la pandemia, cada vez más ciudadanos están regresando a sus actividades presenciales, pero pocos lo hacen manteniendo los protocolos de salud sugeridos. Es por eso que se aconseja llevar mascarillas debido a los riesgos de un posible rebrote, especialmente en lugares cerrados, ya que estos dispositivos de seguridad pueden evitar que el 99% de las partículas del virus pasen a través de estos [3].

Internacionalmente, se considera que la mínima manera de contagio y la necesidad de proteger a lo demás son temas de urgencia. Y en ambientes hospitalarios, las mascarillas se utilizan generalmente como parte del equipo de protección de usuarios contra el COVID-19 [4].

Tanto así que en China y Singapur se tienen consecuencias al no usar mascarillas como arrestos, sanciones o multas por \$300 [5]. Además, en Oklahoma se han reportado diversas amenazas de violencia sobre la Ley de Uso de Mascarillas para que esta llegue a ser solo una recomendación. De hecho, un funcionario de seguridad en Michigan fue asesinado por un individuo sin mascarilla al impedir el ingreso de este a un área cerrada de un centro comercial [6].

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

- Diseñar e implementar un prototipo de un sistema de control de uso de mascarillas y temperatura corporal de personas en áreas cerradas una institución que brinda servicios de salud y atención médica.

1.2.2 OBJETIVOS ESPECÍFICOS

- Analizar y estudiar técnicas de machine learning, sistemas embebidos y redes neuronales convolucionales.

- Adaptar el algoritmo YOLO (You Only Look Once) para la detección de mascarillas y reconocimiento facial.

- Entrenar un modelo de machine learning en la nube mediante una base de datos o pool de imágenes que permita detectar mascarillas automáticamente a partir de un video a tiempo real captado por una cámara.

- Correlacionar variables como la temperatura y uso de mascarillas dentro del sistema de control para permitir el acceso a áreas cerradas.

- Analizar la información tanto del modelo de machine learning como de los sensores para mostrarlos en una pantalla y emitir una alerta en caso de que sea necesario e implementar el prototipo.

1.3 ESTADO DEL ARTE

Personal de IT de Taiwán ha creado y utilizado la plataforma de servicios en la nube de Microsoft Azure del país de Taiwán para visualizar de manera constante mediante diversas cámaras instaladas a todos los rostros de las personas que acceden a las instalaciones de un hospital mediante dispositivos de temperatura IR del cuerpo humano mientras que a su vez identifica si el individuo detectado en cuestión está utilizando una mascarilla correctamente generando una alerta hacia el personal encargado sobre seguridad a través de los servicios de Azure Bot [7]. Adicionalmente, se utiliza la tecnología de la Inteligencia artificial, el software de borde

inteligente, cómputo de borde y acceso a la nube para realizar actividades de monitorización todos los rincones del hospital. También se pretende instalar y usar varios prototipos de esta misma Inteligencia Artificial en un mismo servicio de 'IoT Edge' para poder así estudiar las tramas de las imágenes captadas mediante el analizado de datos de Power BI.

En la FIEC, se ha diseñado e implementado un procedimiento que detecta la correcta utilización de mascarillas basado en inteligencia artificial mediante DL, un modelo original TensorFlow y redes neuronales convolucionales de 3 capas y filtros [8].

Existe un prototipo de detección de mascarillas faciales basado en aprendizaje profundo que utiliza la cámara FLIR Firefly DL para verificar el estado de la máscara y reconocer y registrar cualquier violación de su correcto uso de la empresa Teledyne FLIR [9], utiliza dos bibliotecas públicas con más de 1000 imágenes para mostrar ejemplos de inadaptados de máscaras en varios entornos con cámara FLIR Blackfly S GigE.

También se ha implementado un software que permite que solo los empleados con mascarillas pasen por algunas entradas de infraestructura mediante reconocimiento facial con inteligencia artificial de la compañía de servicio IT LG CNS [10], donde también se posee la detección de calor mientras se lee la identidad del empleado y se muestra la frase "use una máscara" junto con el cerramiento de la puerta de ingreso.

El 'robot NetOn autónomo de cuarentena 5G' de la empresa LG Uplus en Corea recorre todas las instalaciones, está equipado con una cámara termográfica que detecta la temperatura corporal y el uso de una mascarilla, y si el robot detecta a una persona que no está usando una máscara mientras se mueve, proporciona una alarma en tiempo real a la sala de control central junto con una guía de voz para usar una máscara. [11]

Un identificador llamado 'FaceMe Security' [12] identifica a personas que no usan una mascarilla y verifica su identidad, así como mide la temperatura corporal basados en el borde, funciona en computadoras con GPU y VPU para la detección e identificación de rostros en tiempo real. Con la función Health Add-on, los oficiales de salud y seguridad pueden verificar la identidad del visitante mediante

la tecnología TrueTheater que realiza un mejoramiento de la cantidad de píxeles de la imagen capturada en el preprocesamiento, mejorando la precisión en un 11,65% con cámaras 3D y 2D.

Adicionalmente, el Hospital H-Yangji posee una técnica para el control de acceso a ese hospital de los visitantes inteligente de la cual se utiliza IA 'Ilsacheonli' [13] que puede monitorear el reconocimiento facial, la verificación de temperatura y el uso de máscaras en una sola parada. Es un algoritmo de aprendizaje profundo basado en inteligencia artificial que identifica más del 99% de los rostros, y el soporte de detección biométrica hace que sea imposible manipular el acceso mediante fotos y videos. Después de que el visitante llena un cuestionario y reconoce el código QR emitido en la puerta, se verifica el reconocimiento facial, la detección de temperatura y el uso de la mascarilla en menos de 1 segundo.

Otra implementación nueva fue realizada por NVIDIA, que ha realizado una aplicación de detector de mascarilla facial en tiempo real con tecnología de inteligencia artificial usando Transfer Learning Toolkit y una implementación eficiente en tiempo real del modelo entrenado usando DeepStream SDK con reconocimiento de cuantificación (QAT) y cuantificación posterior al entrenamiento [14]. Un algoritmo fue desarrollado por la empresa española 'FacePhi' que reconoce como máximo un noventa y nueve por ciento de exactitud el rostro cubierto no de una persona, es un software especialmente diseñado para enfocarse en los rasgos diferenciales y que mediante un entrenamiento del modelo que presenta un rostro guardado en un respectivo formulario de registro de los usuarios, esto se hace al ingresar con un pin o contraseña personal y un sistema de autenticación de voz [15].

El año pasado la Universidad de Chongqing creó un algoritmo de detección de uso de mascarilla basado en YOLO-v4 [16] mejorado, también se considera el reconocimiento de mascarillas faciales y se incluyen tres categorías, face_mask, face y WMI. Además, CSP1_X mejora la capacidad de extracción de características de este documento y CSP2_X impulsa a PANet a acelerar la circulación de características semánticas, mejorando así la solidez del modelo.

De acuerdo a la 'Universidad de Presidencia de la Computación e Ingeniería de la India', ahí se ha diseñado un dispositivo de entrada inteligente que monitorea automáticamente la temperatura del cuerpo humano y detecta una mascarilla en el

sistema de apertura de unas puertas inteligentes habilitadas para IoT, este proyecto aprovecha la inteligencia profunda, TensorFlow, Keras y OpenCV para detectar máscaras faciales, utilizan el algoritmo Haar Cascade para detectar rostros en el Internet de las cosas de bajo costo utilizando el método Raspberry Pi [17].

Otra solución para detección de fiebre relacionada a reconocimiento facial fue diseñada por PixellImpact que utiliza tecnología de termografía y termopila con una tarjeta MIFARE DESFire 13,56Mhz RFID esencial en sistemas de control de acceso de alta seguridad [18]. De la cual se utilizó una comunicación Wiegand mediante una conexión a un PLC de control de acceso para su acceso remoto y reconocimiento facial de luz visible mejorado con aprendizaje profundo integrado.

Con respecto al uso de puertas automáticas de entrada, la universidad en Pakistán propuso una puerta de inspección y desinfección inteligente basado en IoT para la entrada de lugares públicos con una detección y medición de temperatura, utilizando un sensor sin contacto y registro de la persona sospechosa para un mayor control/monitoreo [19]. Además, se implementaron modelos de aprendizaje profundo en tiempo real para la detección y clasificación de máscaras faciales y clasificó a las personas que usan la mascarilla de manera adecuada, inadecuada y sin una mascarilla mediante VGG-16, MobileNetV2, Inception v3 y ResNet-50 usando un enfoque de aprendizaje por transferencia.

Por otra parte, en el mismo país de la India, el Instituto Shatabdi de Ingeniería e Investigación utilizó un método de detección de objetos de aprendizaje profundo para crear una posición de mascarilla y un detector de temperatura utilizando un sensor de temperatura, Arduino, una cámara RGB y un sensor de temperatura para generar imágenes de entrada y capturar la temperatura de una persona [20].

Asimismo, dos empresas están trabajando en la integración de la detección de temperatura y detección facial como lo son "Vsblty" y el desarrollador de tecnología de sensores Photon-X [21]. De la cual, se trata de una aplicación de cámara biométrica multisensorial que detecta a personas con síntomas de COVID-19 como parte de un sistema de control de acceso al edificio, se medirá la temperatura corporal de las personas que ingresan al edificio utilizando la tecnología de reconocimiento facial de Bisbleti y el reconocimiento y análisis de objetos Photon-X. La cual permite la verificación biométrica, la seguridad de los documentos, la captura de imágenes

gráficas en movimiento por computadora y el mapeo fotográfico, mide la frecuencia cardíaca y los niveles de saturación de oxígeno en el cuerpo en comparación con las cámaras infrarrojas estándar que se utilizan actualmente.

De la misma manera, la empresa Suprema de Corea [22] creó una solución de reconocimiento facial que detecta fiebre con el uso de su propio algoritmo de medición de temperatura de precisión y la tecnología de autenticación facial. Puede realizar todo esto con una cámara termográfica de alta resolución y una cámara infrarroja para detectar rostros falsos y verificar la temperatura de la piel medida en la pantalla del terminal, este algoritmo de reconocimiento facial patentado de clase mundial detecta con precisión el rostro con mayor precisión y mide la temperatura dentro de 0,5°C. Al aplicar el método de machine learning, cuanto más autenticación se realiza, mejor se guarda la plantilla y se aplica la tecnología para reducir continuamente la tasa de autenticación falsa. Además, utiliza una pantalla pequeña táctil, la interfaz de Android que será intuitiva y el acceso al menú del administrador a través de la autenticación facial que permiten acceso y configuración del menú.

Por último, se encontró que en Taiwán una universidad realizó la clasificación de mascarillas y detección de temperatura de la cabeza combinados con redes de aprendizaje profundo mediante un método de detección de objetos de aprendizaje profundo para crear una posición de máscara y un detector de temperatura utilizando RetinaNet [23]. Posee dos módulos, tres categorías de posiciones de uso de máscara, una cámara RGB y una cámara térmica para generar imágenes de entrada y capturar la temperatura de una persona.

Finalmente, con la información se desarrolla un mecanismo para recortar las áreas faciales de imágenes en tiempo real que tienen diferencias en el tamaño, orientación y el fondo de la cara [24]. Este paso ayuda a localizar mejor a la persona que está violando las normas de las mascarillas en las áreas públicas / oficinas.

Este proyecto se diferencia de los mencionados anteriormente por 3 principales motivos:

1. Se utilizará un sistema embebido basado en Linux para la implementación en este caso una Raspberry pi.
2. Se conectarán sensores para hacer más eficiente la detección de posibles peligros de bioseguridad.

3. En la detección, se utilizarán técnicas de machine learning con un entrenamiento supervisado del modelo, en lugar de otras alternativas con modelos pre-entrenados.

1.4 PROPUESTA DE SOLUCIÓN

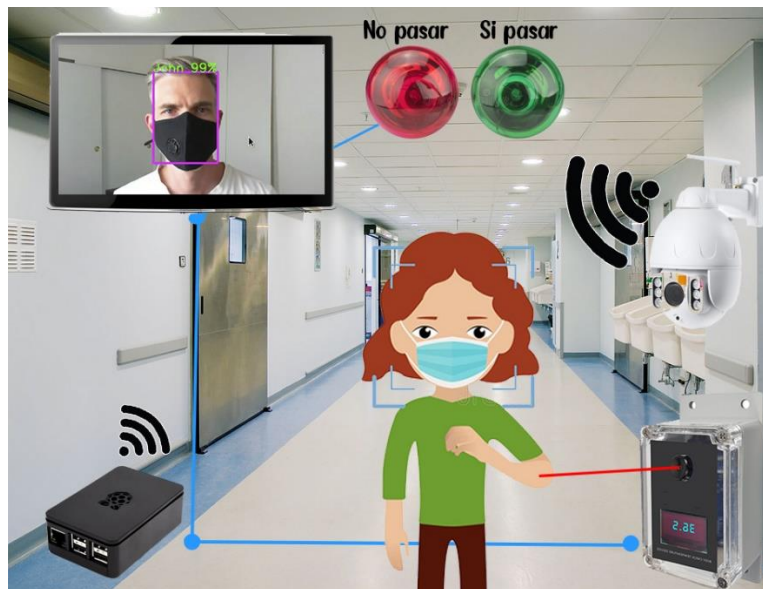


Ilustración 1.1 Diagrama de la solución propuesta. [25]

En la ilustración 1.1, se puede observar un diagrama con la solución a la problemática planteada, ya que se implementará un sistema de detección facial con mascarillas automático a tiempo real utilizando tecnologías de inteligencia artificial conectado a otras variables de control como la temperatura y la mascarilla. Todos estos datos serán analizados por una Raspberry Pi que funcionará como centro de control para todo el sistema.

El escenario se basa en que cuando un visitante se acerca a la puerta de entrada de un área concurrida y cerrada del hospital, la cámara del detector de mascarillas y el sensor térmico se activan automáticamente y confirman si el visitante no tiene fiebre alta y lleva una máscara. Los visitantes pueden ingresar de manera conveniente y rápida sin tomar una temperatura corporal separada o usar una etiqueta de tarjeta de acceso.

Debido a que el cliente de este proyecto es el Hospital León Becerra, el escenario se desarrolla en un área cerrada específica como lo es el área restringida de UCI, el personal médico y familiar autorizado que desea ingresar

al mismo ya ha pasado por la puerta principal de ingreso y la recepción, por lo que se tienen ya los datos personales registrados del mismo junto con su rostro en el sistema de videovigilancia y por ende es un usuario identificado, con autorización.

Luego de este control previo de seguridad, se controlarán variables como temperatura y uso de mascarillas, por lo que al acercarse al prototipo ubicado cerca de la puerta el sistema analizará el rostro para determinar si el usuario cuenta con mascarilla y deberá acercar la mano al sensor infrarrojo y al medir su temperatura deberá cumplir 2 condiciones.

La primera es que use la mascarilla correctamente y la segunda que su temperatura corporal sea inferior a 37° C.

Al cumplir estas condiciones a la vez se encenderá un led de color verde indicando que se permite el paso, caso contrario se encenderá un led rojo y enviará una alerta de seguridad al encargado para notificar esta falla de bioseguridad.

Esta solución resuelve el inconveniente acceso a áreas cerradas debido a la pandemia y sus mutaciones, además que en hospitales de Ecuador este sistema no se ha implementado. De la cual la propiedad más importante es que puede bloquear la posible infección secundaria de virus como 'COVID-19' en el hospital, así como aliviar la fatiga del personal de control de acceso. En particular, se podrá reducir el tiempo para el acceso del personal médico y familiar considerablemente, por lo que se puede esperar una alta satisfacción, como conveniencia y seguridad, en términos de experiencia del paciente.

Este proyecto permitirá controlar el acceso a áreas cerradas de personas que no toman medidas de seguridad como mascarillas y además que su temperatura corporal sea mayor al límite establecido, si no se cumplen alguna de estas dos condiciones se generara una alerta roja. Además, mediante una pantalla se podrá observar a tiempo real de la detección facial de la mascarilla por lo que se diseñará este sistema efectivo para reducir la extensión de la propagación del virus.

Además, este proyecto de detección facial de medición de temperatura que detecta la fiebre alta puede ser usado en instalaciones comunes como lugares deportivos como gimnasios interiores, salas de espectáculos, bibliotecas, etc. que utilizan muchas personas, así como instalaciones de usos múltiples como como

centros para personas mayores, guarderías, instalaciones de bienestar social y destinos turísticos [22].

1.5 JUSTIFICACIÓN

Conociendo la situación actual de la crisis de salud global debido a la pandemia, el peligro consiguiente de las diferentes mutaciones del SARS-CoV-2 y sabiendo que el total de casos de mortalidad en Ecuador son 32,952 fallecidos por COVID-19 hasta agosto del presente año [4], la OMS ha declarado medidas preventivas como tener cuidado al mantener el mínimo distanciamiento entre dos personas, usar mascarilla, si se tiene una habitación cerrada se debe habilitarla con una correcta ventilación y en todo momento prevenir las posibles acumulaciones de personas desde los primeros brotes [26].

Estos protocolos de seguridad hicieron hincapié en el hecho de convertir la utilización correcta de la mascarilla como una práctica común y normal de los posibles acercamientos con otros individuos y también, que estas puedan llegar a ser lo más eficaz posible, es fundamental utilizar las mascarillas correctamente.

Este proyecto pretende que todos los ciudadanos cumplan debidamente las recomendaciones y medidas preventivas del COE Nacional [27] para evitar así nuevos brotes epidemiológicos y poder garantizar una pronta recuperación económica del Ecuador, ya que está demostrado que el uso de mascarilla y medidas básicas de higiene reducen significativamente las posibilidades de contagio.

Además, se garantizarían 2 de los 17 metas y prioridades integrales mundiales declarados por la ONU sobre salud, bienestar y crecimiento económico al fomentar el uso de mascarillas en áreas cerradas en el país, promover el bienestar en todas las edades y mermar el impacto económico del COVID-19 [28].

También, de acuerdo a una organización de Neumología [29], el uso de mascarillas y control de temperatura tiene impacto en el área sanitaria que son externas y superan a los protocolos de salud y prevención anti-Covid como lo son el proteger del aire frío, prevenir fiebres con antelación y evitar enfermedades para resfriados y gripe; prevenir las infecciones por influenza; alergias de enfermedades de los pulmones como la bronquitis y el catarro; y mermar la respiración de aire contaminado.

1.6 ALCANCE

Para la implementación de este proyecto se contempla el diseño de un modelo entrenado de machine learning que permita detectar automáticamente si se está utilizando o no la mascarilla, el algoritmo genera una tasa de éxito que va del 0 al 1 donde el 0 significa que no se está usando mascarilla y 1 que existe un 100% de probabilidad de que la persona está usando mascarilla, estos datos se analizan internamente en un código realizado con python para así mostrar en pantalla el video capturado por la cámara y un cuadro enmarcando los rostros, dicho cuadro será rojo cuando el modelo detecta que es muy poco probable que use mascarilla mientras que el cuadro será verde cuando dicha probabilidad sea alta. Además este proceso se unirá con los datos recopilados por el sensor de temperatura infrarrojo el cual permite detectar la temperatura corporal solo acercando una parte del cuerpo.

Una vez validadas todas las variables se mostrarán los resultados en una pantalla para que se pueda observar el funcionamiento del prototipo en tiempo real.

Existen 2 casos programados

1. La persona no tiene puesta la mascarilla o su temperatura corporal es mayor a 37° C para lo cual se encenderá un led rojo indicando que no cumplió con una de las condiciones, entonces se envía una alerta al correo del encargado de seguridad para notificar esta falta a los protocolos de bioseguridad.
2. La persona tiene mascarilla y su temperatura corporal es menor a 37° C, entonces se encenderá un led verde lo que indica que cumple con las condiciones planteadas.

El prototipo tiene la capacidad de ser autónomo, es decir, que puede funcionar sin conexión a corriente. Esto se hace utilizando una batería y una pantalla de 5 pulgadas diseñada para la Raspberry, esta pantalla tiene poco consumo energético por lo que permite mantener funcionando el proyecto durante 1 hora.

1.7 METODOLOGÍA

De acuerdo al diagrama de bloques de la ilustración 1.2, para realizar este proyecto se debe tener en completo funcionamiento el código de la detección de mascarillas y de la detección de temperatura y luego de esto se procede con la

implementación del sensor físico de temperatura, la cámara para la detección facial y la pantalla con el fin de que la adaptación del sistema completo sea fácil.

El presente proyecto se va a relizar como un prototipo que puede ser escalable y autónomo, debido a que se pretende realizar pruebas en las instalaciones del cliente interesado, el primer paso a realizar esto es inspeccionar las instalaciones del hospital León Becerra de Guayaquil ubicado al sur de la ciudad.

Además, debido a que este sistema funciona en áreas cerradas, se planeará su uso en las mismas. Una vez tenido eso en cuenta, se procede con el diseño del prototipo basado en la solución planteada con la instalación del sistema operativo con el que va a funcionar la raspberry pi, luego del entrenamiento del modelo en base un grupo de imágenes y creación del código del sistema.

En la entrada a dicha área cerrada se debe ubicar de los componentes de telecomunicaciones y electrónicos que utiliza el prototipo como son la camara, pantalla, sensores y Raspberry Pi.

El prototipo se debe colocar a una distancia de 1 metro de la puerta de acceso debido a que tan solo es necesario que la persona acerque su muñeca al sensor y observe la camara.

1.8 DIAGRAMA DE BLOQUES DE LA SOLUCIÓN PROPUESTA

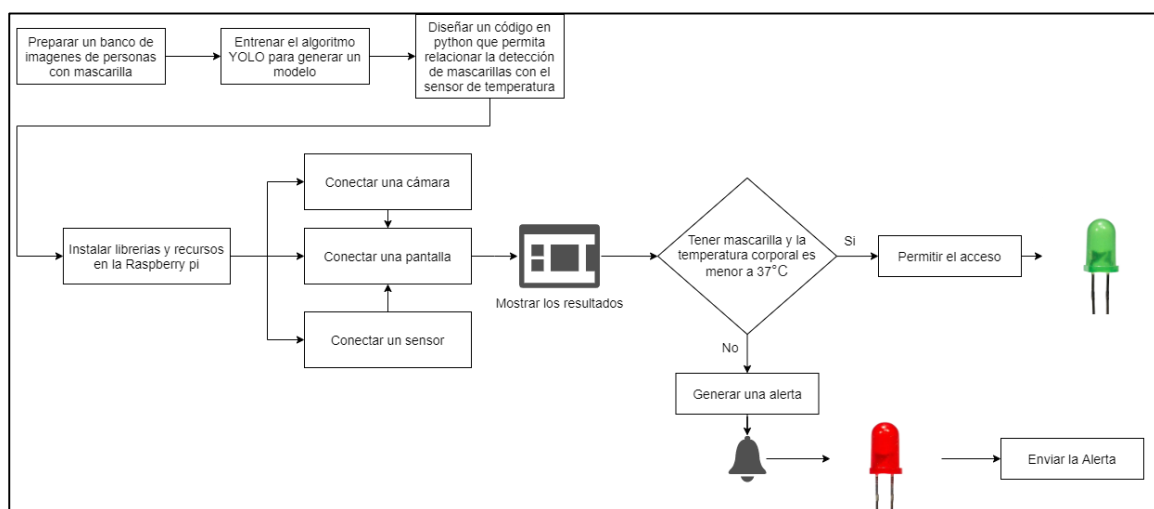


Ilustración 1.2. Diagrama de bloques de la solución.

En la ilustración 1.2 se puede observar el diagrama de bloques de la solución donde se especifica secuencialmente cada uno de los pasos importantes para la correcto diseño e implementación del proyecto. Existen algunos puntos críticos entre ellos está el diseño del código con el lenguaje de programación Python para controlar los pines físicos de la Raspberry y la parte lógica compuesta por el modelo de machine learning. Otro punto importante es la lógica condicional que seguirá este proceso ya que existen dos variables condicionantes: la primera es usar mascarilla y la segunda es que la temperatura corporal sea menor a 37° C.

Al cumplirse las dos condiciones antes mencionadas el prototipo encenderá un led verde indicando que se permite el acceso al área cerrada, caso contrario se generara una alerta indicando que hubo una falta de bioseguridad, esta alerta se enviará a una persona encargada de la seguridad de todo el establecimiento.

CAPÍTULO 2

2. MARCO TEÓRICO

Para el marco teórico se pretende explicar los términos necesarios para el completo entendimiento del proyecto con el fin de poseer una base del diseño de la solución. Al hacerlo, se inicia exponiendo el tema principal del proyecto, el cual es visión por computadora, junto con su concepto, luego continúa con la definición de redes artificiales que utiliza esta técnica de visión por computadora junto con sus características y la definición de machine learning para después abordar temas como el tipo de lenguaje de programación a utilizar. También se indican algunos de los avances sobre machine learning en el área de la salud.

Finalmente, se tratan temas relacionados al hardware a utilizar como el tipo de microordenador, tipos de sensores de temperatura, métodos de visión por computadora y alarmas de bioseguridad en caso de incumplimiento de protocolos de salud para la prevención del COVID-19.

2.1 MACHINE LEARNING (ML)

Es un subtema de la inteligencia que simula procesos realizados por el humano o también llamada inteligencia artificial (IA). Esto consiste en el estudio de cómo hacer que las máquinas sean más parecidas a las humanas tanto en su comportamiento como en sus decisiones, dándoles así la capacidad de aprender y desarrollar sus propios programas [30]. Esto se hace con la menor manipulación de un ser humano posible, es decir, sin programación explícita.

Además, este proceso de ML se automatiza y mejora en función de las experiencias de las máquinas a lo largo del proceso, se envían datos de buena calidad a las máquinas y se utilizan diferentes algoritmos para construir modelos ML para entrenar a las máquinas con estos datos [31]. El algoritmo a utilizar depende del tipo de datos disponibles y del tipo de actividad que se vaya a implementar y depende del funcionamiento de la red de ML planeada.

En comparación con la programación tradicional, en la típica manera de programación se alimentaría al sistema con los datos de entrada para generar resultados [32]. Pero cuando se trata de un aprendizaje que se realizará de manera automática, los datos de entrada junto con la salida se introducen en el sistema durante la fase de aprendizaje y se elabora un programa por sí mismo tendiendo así una retroalimentación de manera continua.

En grandes industrias, se manipulan varias fuentes de información importante para optimizar sus operaciones y poder generar así decisiones inteligentes. El aprendizaje automático ayuda a crear modelos que pueden procesar y analizar diversas cantidades de datos ya sea de clientes o de empresas para ofrecer resultados precisos [33]. Al construir modelos de aprendizaje automático tan precisos, las empresas pueden aprovechar oportunidades rentables y evitar riesgos que pueden llegar a ser predecibles.

Para la creación de modelos, como el "YOLOv4" que suelen ser generados en base al aprendizaje automático, se deben pasar por diversos pasos para creación correcta del mismo [34]. Primero se debe reunir todos los datos de información necesarios como imágenes o datos numéricos, luego se debe preparar estos datos, elegir un modelo base, capacitación, evaluación, ajuste de hiperparámetros y finalmente, predicción.

Esta reunión de datos encierra procesos internos como el entrenamiento y fase de pruebas. Ahí, el algoritmo toma como valores iniciales un conjunto de datos llamado "datos de entrenamiento" y recorre todos los espacios para hallar patrones en los datos que se ingresaron, se ordena al modelo qué decisiones tomar para obtener los resultados esperados, teniendo así al resultado del proceso de formación [35]. Y en la parte final, con el modelo ya creado se puede alimentar con datos del input para generar la predicción correcta.

El aprendizaje de un algoritmo que incluye ML se monitorea o supervisa en el sentido de que ya se conoce la salida y el algoritmo se corrige cada vez para optimizar sus resultados. El algoritmo se entrena sobre el grupo de datos dados y se modifica hasta que alcanza un nivel aceptable de rendimiento con valores arrojados de probabilidad entre 0 y 1.

2.2 LENGUAJES DE PROGRAMACIÓN PARA MACHINE LEARNING

En comparación a otros lenguajes usados para programar que podrían usarse para usos de aprendizaje automático como:

- R: Dado que este lenguaje fue inventado para las tareas de los científicos, los autores no intentaron hacerlo intuitivo. Partieron del hecho de que este lenguaje será utilizado por personas que estén familiarizadas con el análisis matemático, los métodos estadísticos y las desviaciones probabilísticas. Es por eso que R puede parecer un lenguaje muy complejo.
- C ++: Es indispensable para aplicaciones de alta carga, se utiliza para desarrollar motores de búsqueda, robots de intercambio comercial, sistemas de administración de bases de datos y enrutamiento de tráfico de red.
- JavaScript: Se usa comúnmente como una herramienta integrada para acceder mediante programación a varios objetos de aplicación, es imposible crear sitios interactivos modernos sin el conocimiento de esta tecnología. Se utiliza en navegadores web, se puede escribir juegos en él y desarrollar varias aplicaciones para computadoras y dispositivos móviles.
- Julia: Ofrece algunas de las excelentes funciones para diseñadores gráficos, así como varios códigos y secuencias de comandos que se pueden usar para mejorar las imágenes y los gráficos. Por lo tanto, esta función se puede utilizar para mejorar los gráficos y la representación de sitios web o imágenes convertidas.

Algunos lenguajes como C, C ++ y Java se utilizan para el desarrollo de software, aplicaciones y modelos, mientras que existen otros como .NET y HTML que se utilizan para desarrollar aplicaciones web. Por lo tanto, todo tipo de lenguajes de programación tienen sus propios usos, ya que son los más adecuados para una tarea específica. Python ha sido uno de los mejores lenguajes de programación en los últimos años, ya que ofrece varios beneficios a los usuarios.

Los demás lenguajes de programación no son tan populares como Python con respecto a su fácil lectura y complejidad relativamente menor en comparación con otros lenguajes [36]. Python ayuda a reducir la carga de conceptos complejos como cálculo y álgebra lineal que requieren mucho esfuerzo y tiempo para implementar.

Python posee amplia variedad de bibliotecas que admite y se puede utilizar tanto en el desarrollo de aplicaciones como en el aprendizaje de manera automática. Elimina la necesidad de codificar algoritmos básicos desde cero. El aprendizaje automático se basa en una variedad de algoritmos conocidos para funcionar como los bosques aleatorios, k- medias y muchos otros de los cuales Python posee librerías especializadas [37]. Las librerías de las bibliotecas de aprendizaje de manera automática en Python, como scikit-learn y TensorFlow, contienen estos algoritmos como funciones listas para usar.

2.3 LIBRERÍAS DE PYTHON

Otro beneficio de usar Python en ML son las bibliotecas preconstruidas con diferentes paquetes para sus varias aplicaciones que comúnmente usadas en ML [38]:

- Numpy: Es una biblioteca de Python ampliamente utilizada para manejar matrices con matrices. Numpy puede manejar matrices y matrices multidimensionales de gran tamaño, junto con una gran colección de operaciones matemáticas para operar en estas matrices. Por lo tanto, permite realizar operaciones con una gran cantidad de funciones matemáticas, algebraicas y de transformación [39].
- OpenCV: Es una biblioteca propia de Python que suele ser agregada en softwares con visión por computadora en inteligencia artificial, aprendizaje automático, reconocimiento facial, etc.
- TensorFlow: Se utiliza comúnmente para crear aplicaciones de ML. Permite a los programadores construir aplicaciones de aprendizaje automático mediante el uso de una variedad de herramientas, marcos y recursos comunitarios.
- Keras, es una biblioteca de red neuronal que se ejecuta sobre Theano o TensorFlow y es perteneciente a Python.

Todas las mencionadas ofrecen facilidades para manipular las imágenes arrojadas por los algoritmos más populares [38], algunos de estos paquetes para diversas reglas matemáticas con diversos métodos de operación de matrices multidimensionales con algoritmos de tipos de aprendizaje de manera automática y visión por computadora.

2.4 REDES NEURONALES CONVOLUCIONALES (CNNs)

Las CNNs son un tipo específico de red neuronal que ha revolucionado la visión por computadora y el reconocimiento de patrones. Unas de sus aplicaciones incluyen el reconocimiento de voz, el procesamiento de señales de audio, el procesamiento de series de tiempo, análisis del significado de textos y, más recientemente, incluso para jugar juegos de tablero y superar a los maestros de este juego [40].

2.5 DETECCIÓN FACIAL EN MACHINE LEARNING

Un proceso de detección facial es una técnica en la que una imagen digital es analizada por una red neuronal, esta extraerá las características de la imagen y definirá si se trata de un rostro humano. Por lo general, los servicios de verificación de identificación de usuarios a través de controles de asistencia funcionan midiendo y cuantificando los rasgos faciales de una imagen dada [41].

Un conjunto de datos consta principalmente de imágenes o videos para tareas como detección de objetos y clasificación de múltiples etiquetas. En la visión por ordenador, diversas imágenes del rostro del ser humano se han utilizado ampliamente para desarrollar sistemas de detección y poder reconocer rostros, y demás proyectos que utilizan imágenes de rostros como una gran base de datos iniciales [42].

Dentro de las aplicaciones del aprendizaje automático están las relacionadas con el reconocimiento de imágenes, hay muchos casos de uso de reconocimiento de rostros de individuos, principalmente con fines de seguridad, como identificar delincuentes, buscar personas desaparecidas, ayuda forense. investigaciones, diagnóstico de enfermedades, seguimiento de la asistencia en las escuelas, etc. [43].

2.6 IA EN LA SALUD HUMANA

Para el cuidado correcto y bienestar humano, una aplicación de ML se basa en el diagnóstico de enfermedades y dolencias ya que ayudan a que este proceso sea más factible de diagnosticar. El hallazgo de nuevas medicinas en una etapa temprana de una enfermedad es otra aplicación crucial que involucra tecnologías como la medicina de precisión y la secuenciación de próxima generación [44]. Las pruebas realizadas en ambientes clínicos cuestan tiempo y dinero para completar y entregar resultados. Las aplicaciones de análisis predictivos basados en ML mejoran estos factores y dan mejores resultados.

Las tecnologías de ML también son fundamentales para realizar predicciones de brotes. Los científicos de todo el mundo están utilizando estas tecnologías para predecir brotes epidémicos [45] .

Con lo que se está revolucionando los diagnósticos y desempeñando un papel en el ciclo de descubrimiento de fármacos, así como en el análisis de datos genéticos y la medicina personalizada. Investigadores de una universidad de Estados Unidos publicaron un artículo [46] en la “Revista Nature” que describe cómo el aprendizaje profundo puede clasificar imágenes de lesiones cutáneas con la misma precisión que los dermatólogos verificados por la junta.

En cualquier hospital, las evaluaciones de movilidad pueden decir mucho sobre la salud de las personas mayores. La detección de caídas es una función importante de la salud relacionada con la inteligencia ambiental. En los ancianos, que generalmente son más frágiles y menos móviles, las caídas están relacionadas con un aumento de la mortalidad y el consiguiente miedo a caminar (que, a su vez, perjudica aún más la movilidad) [47]. Los sistemas de detección de caídas (por ejemplo, una combinación de sensores térmicos y de movimiento) permiten una asistencia más rápida y una mejor sensación de seguridad e incertidumbre.

El control de las infecciones es otra área en la que la inteligencia ambiental puede resultar útil. Debido a la pandemia, los seres humanos han creado más conciencia de todas las maneras de propagación de la enfermedad y con esto la búsqueda de recientes contactos humanos realizados ha sido esencial [48] . Sin embargo, el rastreo de contactos es difícil en la vida cotidiana, pero en un entorno hospitalario más controlado con mayor vigilancia, esto puede llegar a ser una ventaja importante para conocer si se tiene una enfermedad a priori.

Por último, la salud mental suele evaluarse mediante cuestionarios y valoración clínica. Sin embargo, esto solo proporciona una instantánea incompleta de la situación. Los sensores ambientales pueden monitorizar, por ejemplo, la frecuencia y duración de episodios depresivos o esquizofrénicos, así como la eficacia de diferentes regímenes terapéuticos [49].

2.7 TEMPERATURA Y MASCARILLAS COMO PREVENCIÓN DEL COVID-19

Las mascarillas también se conocen como cubrebocas para el rostro con un filtro, estas pueden prevenir al menos el 95% de las partículas en el aire [50]. Diversos trabajadores de la salud las usan a menudo para prevenir la transmisión de enfermedades infecciosas, patógenos y otras partículas peligrosas.

Las mascarillas N95, que se consideran el estándar de oro de las mascarillas faciales en algunas industrias [50], se ajustan cómodamente para formar un sello en la cara. Protegen contra enfermedades y bacterias manteniendo sus gotas en sus fibras para que las partículas no puedan entrar en la nariz o la boca del usuario de la máscara. Cuando una mascarilla se usa correctamente en forma ajustada, no puede entrar aire exterior por los lados de la mascarilla o de la persona que lleva la mascarilla. Esto ahorra costos de tiempo para prever la propagación de COVID-19 entre las personas que usan mascarillas de la manera correcta.

El Hospital de Investigación Infantil St. Jude [51] afirma que “Muchas personas con COVID-19 tienen fiebre en algún momento durante su enfermedad. Es una buena idea seguir controlando su temperatura en todo momento cuando tenga síntomas de fiebre como escalofríos, dolores corporales y sensación de calor o enrojecimiento.” De donde se puede inferir que la variable de la temperatura debe ser controlada en todo momento con el fin de prevenir el contagio temprano.

2.8 MICROORDENADORES

Entre los diversos sistemas informáticos creados sobre la base de micro-procesadores, las micro-computadoras se distinguieron de inmediato por sus peculiaridades: tiene mayor productividad que los ordenadores voluminosos de las primeras generaciones [52].

Las microcomputadoras son incomparablemente más preferidas en términos de dimensiones, costo, consumo de energía, confiabilidad. Si a esto le sumamos la flexibilidad y adaptabilidad de las microcomputadoras a las más variadas tareas de la ciencia [53], la tecnología y la producción, entonces uno puede imaginar no solo el más amplio campo de aplicación de la tecnología de microprocesadores, sino también el mayor nivel de uso resultante de la tecnología informática moderna en general.

Algunas micro-computadoras son:

- Microbit: esta placa en sí es un dispositivo completo con un conjunto de funciones básicas con botones programables, una pantalla en forma de matriz de 25 leds rojos, sensores de aceleración y temperatura. Esto permite que un desarrollador novato implemente rápidamente un dispositivo simple o comience un curso de capacitación [54].
- Raspberry Pi: Es una combinación entre una computadora (como una computadora portátil) y un panel de microcontrolador, como se puede encontrar en muchos dispositivos inteligentes de la cual uno de sus usos más comunes es como televisor o centro multimedia [55].
- Arduino: es un pequeño dispositivo electrónico que consta de una sola placa de circuito impreso y puede controlar varios sensores, motores, luces, transmitir y recibir datos. Tiene la presencia de muchas bibliotecas para módulos, sensores, pantallas, etc., disponibles para uso gratuito y posee la posibilidad de funcionamiento autónomo, es decir, sin uso de pilas [56].

A pesar de existir diversos tipos de microcomputadoras, la Raspberry Pi 4 es compatible con la metodología de ML y posee una mayor rapidez a comparación con versiones anteriores [57].

2.10.1 CARACTERÍSTICAS DE LA MICROCOMPUTADORA RASPBERRY PI

Esta micro-computadora es una tarjeta madre con procesador, RAM, puertos USB, un puerto HDMI y un conector Ethernet. Entonces, si conecta un teclado, mouse y monitor, puede usarlo como una PC de escritorio [55]. Al mismo tiempo, Open webOS tiene una interfaz de usuario optimizada para pantallas táctiles, por lo que la cuestión de su uso en una computadora permanece abierta.

Raspberry Pi 4 usa un Broadcom BCM2711 a 1.5 GHz para el procesador, que es equivalente a los equipos de escritorio modernos ahora. Puede ejecutar funciones típicas igual que las computadoras de escritorio y portátiles de ahora [58]. Tiene una variación diferente desde 2 GB hasta 4 GB tiene una función inalámbrica IEEE 802.11ac de 2.4 y 5.0 GHz incorporada mientras que Bluetooth 5.0. Además, tiene 4 puertos USB para ser revisados de forma remota manualmente usando un teclado y un mouse inalámbricos. También tiene una toma GPIO completa de 40 pines. Es más barato que otros modelos y es la placa más utilizada por los programadores y es fácil de usar.

2.9 SISTEMAS OPERATIVOS PARA RASPBERRY PI

La Raspberry Pi no posee con un sistema operativo innato y necesitará un administrador de sistema operativo que le facilite la descarga, instalación y configuración de su Raspberry Pi. Este administrador tendrá varios sistemas operativos para elegir de los cuales los sistemas operativos más comunes disponibles en los últimos modelos de Raspberry Pi son “Raspbian”, “OSMC”, “OpenELEC”, “Windows IoT Core” y “RISC OS” [59].

Una versión de Linux oficial para esta micro-computadora es el Raspbian que está construida específicamente para Raspberry Pi, este sistema operativo es gratuito y ofrece más ventajas que uno típico: posee más de 30,000 paquetes precompilados [60] están disponibles en un formato amigable para una rápida instalación de Raspberry Pi. Estos se instalaron inicialmente en junio de 2012 para obtener los resultados más altos de Raspberry Pi. Raspbian sigue desarrollándose activamente con tantos paquetes “Debian” como sea posible basándose en la mejora de la estabilidad y el rendimiento.

2.10 MÉTODOS DE ENTRENAMIENTO EN DE MACHINE LEARNING

2.12.1 APRENDIZAJE SUPERVISADO

Este tipo de algoritmo aprende de los datos de entrenamiento etiquetados, lo que le ayuda a predecir el resultado de datos inesperados. Crear, escalar e implementar con éxito un aprendizaje automático supervisado y preciso. El modelo de ciencia de datos requiere tiempo y experiencia técnica de un equipo de científicos de datos altamente capacitados. Además, el procesador de datos debe reconstruir los modelos para asegurarse de que la información proporcionada sea correcta hasta que cambien los datos [61].

En el aprendizaje supervisado, un ejemplo de la vida cotidiana sucede si se desea determinar los tiempos de viaje de un usuario, primero se debe crear es un conjunto de datos de entrenamiento con el tiempo total de viaje y factores relacionados como el clima, el tiempo, etc. Según este conjunto de entrenamiento supervisado, el algoritmo puede ver una relación directa entre la cantidad de lluvia y el tiempo que le toma llegar a casa. También puede haber una conexión entre el tiempo de salida del trabajo y el tiempo en la carretera. Ese algoritmo creado con un

aprendizaje supervisado puede encontrar algunas conexiones con los datos ya etiquetados.

2.11.2 APRENDIZAJE NO SUPERVISADO

El aprendizaje no supervisado consiste en el hecho de que proporcionar una gran cantidad de datos, pero estos datos no tienen etiquetas adecuadas, por ejemplo: en un conjunto de entrenamiento, el algoritmo extrae ciertas estructuras para su clasificación sin alguna agrupación previa.

Este tipo de aprendizaje utiliza algoritmos para analizar y agrupar conjuntos de datos no etiquetados. Estos algoritmos descubren patrones ocultos o agrupaciones de datos sin necesidad de intervención humana. Su capacidad para descubrir similitudes y diferencias en la información lo convierte en la solución ideal para el análisis exploratorio de datos, estrategias de venta cruzada, segmentación de clientes y reconocimiento de imágenes. [62].

Esta manera de aprendizaje se clasifica en 5 categorías diferentes de algoritmos, que agrupan los datos en función de similitudes o relaciones entre variables:

- **Agrupación:** Se trata principalmente de encontrar una estructura o patrón en una colección de datos no categorizados. Los algoritmos de agrupamiento de aprendizaje no supervisado procesarán sus datos y encontrarán agrupaciones naturales si existen en los datos. [63].
- **Particionamiento:** En este método de agrupamiento, los datos se agrupan de tal manera que un dato puede pertenecer a un solo grupo [63].
- **Aglomeración:** En esta técnica de agrupación, cada dato es un grupo. Las uniones iterativas entre los dos clústeres más cercanos reducen el número de clústeres [63].
- **Superposición:** En esta técnica, se utilizan conjuntos borrosos para agrupar datos. Cada punto puede pertenecer a dos o más grupos con distintos grados de pertenencia [63].
- **Probabilístico:** Esta técnica utiliza la distribución de probabilidad para crear los grupos [63].

2.11.3 APRENDIZAJE SEMI SUPERVISADO

Es una rama del aprendizaje automático que intenta resolver problemas que requieren o incluyen datos etiquetados y no etiquetados. El aprendizaje semi-supervisado emplea conceptos matemáticos como las características de los métodos de agrupación y clasificación [64].

El aprendizaje semi supervisado es una combinación de aprendizaje supervisado y no supervisado. Utiliza una pequeña cantidad de datos etiquetados y una gran cantidad de datos no etiquetados, lo que proporciona los beneficios del aprendizaje tanto no supervisado como supervisado y evita los desafíos de encontrar una gran cantidad de datos etiquetados. Eso significa que puede entrenar un modelo para etiquetar datos sin tener que usar tantos datos de entrenamiento etiquetados. [64].

Por lo tanto, el aprendizaje semi supervisado permite que el algoritmo aprenda de una pequeña cantidad de documentos de texto etiquetados mientras sigue clasificando una gran cantidad de documentos de texto sin etiquetar en los datos de capacitación.

2.11 MODELOS PARA DETECCIÓN DE OBJETOS CON MACHINE LEARNING

Algunos de los modelos más importantes para detección de objetos son los siguientes:

2.13.1 MODELO R-CNN

La familia de métodos R-CNN se refiere a una "Red neuronal convolucional basada en regiones", fue desarrollada por Ross Girshick. Incluye las técnicas R-CNN, Fast R-CNN y Faster-RCNN diseñadas y demostradas para la localización y el reconocimiento de objetos [65].

El modelo utiliza una técnica de visión por computadora para proponer regiones candidatas o cuadros delimitadores de objetos potenciales en la imagen denominada "búsqueda selectiva", aunque la flexibilidad del diseño permite utilizar otros algoritmos de propuesta de región [65].

La arquitectura del modelo toma la fotografía de un conjunto de propuestas de regiones como entrada que se pasan a través de una red neuronal convolucional profunda. Se utiliza una CNN previamente entrenada, como un VGG-16, para la extracción de características. El final de la CNN profunda es una capa personalizada llamada Capa de agrupación de región de interés, que extrae características específicas para una región candidata de entrada determinada [65].

Luego, la salida de la CNN es interpretada por una capa completamente conectada, luego el modelo se bifurca en dos salidas, una para la predicción de clase a través de una capa softmax y otra con una salida lineal para el cuadro delimitador. Luego, este proceso se repite varias veces para cada región de interés en una imagen determinada [65].

2.13.2 MODELO YOLO

Es una familia popular entre los modelos de reconocimiento de objetos, YOLO o "You Only Look Once" fue desarrollado por Joseph Redmon [65].

Los modelos R-CNN pueden ser generalmente más precisos [65], sin embargo, la familia de modelos YOLO son rápidos, mucho más rápidos que R-CNN, logrando la detección de objetos en tiempo real.

El enfoque implica una red neuronal única entrenada de extremo a extremo que toma una fotografía como entrada y predice cuadros delimitadores y etiquetas de clase para cada cuadro delimitador directamente [65]. La técnica ofrece una menor precisión predictiva (por ejemplo, más errores de localización), aunque opera a 45 cuadros por segundo y hasta 155 cuadros por segundo para una versión del modelo con velocidad optimizada.

El modelo funciona dividiendo primero la imagen de entrada en una cuadrícula de celdas, donde cada celda es responsable de predecir un cuadro delimitador si el centro de un cuadro delimitador cae dentro de la celda [65]. Cada celda de la cuadrícula predice un cuadro delimitador que involucra las coordenadas x, y, el ancho y alto y la confianza. Una predicción de clase también se basa en cada celda.

Por ejemplo, una imagen se puede dividir en una cuadrícula de 7×7 y cada celda de la cuadrícula puede predecir 2 cuadros delimitadores, lo que da como resultado 94 predicciones de cuadros delimitadores propuestas. El mapa de

probabilidades de clase y los cuadros delimitadores con confiancias se combinan luego en un conjunto final de cuadros delimitadores y etiquetas de clase. La imagen tomada del documento a continuación resume los dos resultados del modelo. [65]

2.12 SENSOR DE TEMPERATURA INFRARROJO

Los sensores de temperatura infrarrojos detectan ondas electromagnéticas en el rango de 700 nm a 14000 nm. Mientras que el espectro infrarrojo se extiende hasta 1.000.000 nm, los sensores de temperatura IR no miden por encima de 14000 nm [66]. Debido a que la energía infrarroja emitida por cualquier objeto es proporcional a su temperatura, la señal eléctrica proporciona una lectura precisa de la temperatura del objeto al que apunta.

Las señales infrarrojas pasan al sensor a través de una ventana hecha de plástico especial. Si bien el plástico normalmente no permite que las frecuencias infrarrojas lo atraviesen, los sensores usan una forma que es transparente a frecuencias particulares. Este plástico filtra las frecuencias no deseadas y protege los componentes electrónicos dentro del sensor del polvo, la suciedad y otros objetos extraños [66].

Existen diversas formas de conducir el calor como, conducción, convección o radiación. La radiación es un proceso en el que la energía térmica en forma de ondas electromagnéticas es emitida por un objeto caliente y absorbida por un objeto más frío. Gran parte de la radiación emitida se encuentra en la región infrarroja (IR) del espectro electromagnético, pero algunas ondas también se propagan a la banda de luz visible [67].

La banda de longitud de onda de IR se extiende de 0,7 a 1000 micrones; sin embargo, los sistemas prácticos de medición de IR utilizan solo ciertas bandas de longitud de onda de entre 0,7 y 14 micrones porque la radiación es la más fuerte en este rango [67].

Si un objeto está expuesto a la energía IR irradiada por una fuente de calor, como un calentador eléctrico, una bombilla, el sol u otras fuentes, la energía que llega al objeto se llama energía incidente que tiende a reflejarse en la superficie del objetivo en cuestión. Teóricamente, el coeficiente de reflectividad del objeto puede variar de 0 a 1 [67].

Según el material, el grosor y demás características del objeto, la radiación generada puede atravesar el objeto o transmitirse. El coeficiente de transmisión puede variar de 0 (sin energía transmitida a través del objeto) a 1.0 (100% de energía transmitida a través del objeto). Los ejemplos de alta transmitancia incluyen vidrio, cuarzo, película plástica y varios gases. Los materiales opacos en el espectro IR tienen coeficientes de transmisión cercanos a cero [67].

La mayoría de los detectores de infrarrojos son del tipo de longitud de onda única o de longitud de onda dual, los detectores de longitud de onda única miden la energía IR de una banda y el instrumento calcula la temperatura del objeto en función de la salida del detector y la emisividad preestablecida [67].

2.13 TIPOS DE CÁMARAS WEB

Al momento de seleccionar el tipo de cámara web para una tecnología que usa ML, se debe considerar entre las alternativas a una altamente compatible con el micro-ordenador a utilizar, las necesidades del código y el hardware necesario para conectarlo físicamente a su micro-ordenador y su versión. También se debe tener en cuenta la resolución de imagen y su adaptación a la librería escogida para tener un funcionamiento correcto. Además, el módulo de la cámara debe ser capaz de tomar videos de alta definición, así como fotografías, lo que permite que la cámara capture imágenes y detecte las etiquetas dadas como, por ejemplo: rostros con y sin mascarilla [68].

2.14 SALUD EN HOSPITALES DEL ECUADOR Y EL MUNDO

La UNICEF [69] ha informado que la amenaza de lo que podría ser una variante de Covid-19 es de fácil contagio entre los ecuatorianos y que requerirá una acción urgente en todo el mundo. Esta nueva variante que fue identificada por primera vez en Sudáfrica, se ha extendido a al menos otros 13 países según la ONU [70], lo que ha llevado a varias naciones a intensificar las medidas de contención a pesar de que poder identificar todas sus características completamente podría tomar varios meses y que esta nueva amenaza ya se encuentra en diversos países de Europa y Asia.



Ilustración 2.1 Personal de un hospital en Michigan ingresando a urgencias [71].

Esta nueva ola de contagios está ejerciendo una fuerte presión sobre los hospitales como se observa en la Ilustración 2.1 con personal de un hospital atendiendo a un contagiado en un estacionamiento, lo que obliga a los médicos a tratar a los pacientes en lugares exteriores de las instituciones médicas y a rechazar el tratamiento de los pacientes gravemente enfermos, incluidos los pacientes con cáncer [66].

En Quito, a pesar de que se posee una gran variedad de sitios clínicos y servicios que incluyen hospitales gubernamentales que brindan atención de emergencia, especializada y de diagnóstico gratuita y de nivel comunitario para personas de bajos recursos, desde el mes de junio hasta octubre del presente año, 4146 residentes locales [72] han muerto por complicaciones causadas por COVID-19. Además, según voluntarios médicos, algunas personas han muerto en sus hogares debido a que les negó la hospitalización debido a que no existen suficientes camas de hospital [66].

CAPÍTULO 3

3. DISEÑO DE LA SOLUCIÓN

3.1 DIAGRAMA DE BLOQUES BÁSICO

El diseño de la simulación constará de 5 etapas según se observa en la ilustración 3.1 con una versión simplificada de los pasos a seguir para la realización de este proyecto y da una idea de la secuencia necesaria para la ejecución del mismo.

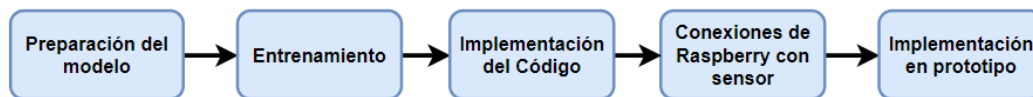


Ilustración 3.1. Diagrama de bloques simplificado.

Estas etapas tratan de la preparación del modelo y su entrenamiento, la implementación del modelo en un código para que este pueda leerlo y ejecutarlo, la realización de las conexiones y preparación del hardware a utilizar junto a sus periféricos de entrada y salida, y finalmente la unión del sistema en un prototipo funcional.

A continuación, se indica el diagrama de bloques detallado en la ilustración 3.2 con un explaye de cada etapa mencionada en la ilustración 3.1 en tres fases separadas en 3 cuadros como la primera fase que sería la fase de programación de variables como son la temperatura y la detección del uso de mascarillas, la segunda fase que sería la implementación de la fase 2 en los componentes físicos.

Y por último, la fase 3 que sería la programación de la última variable que sería la de la alarma dependiendo de la condición creada con las 2 variables mencionadas a la vez.

3.2 DIAGRAMA DE BLOQUES DETALLADO

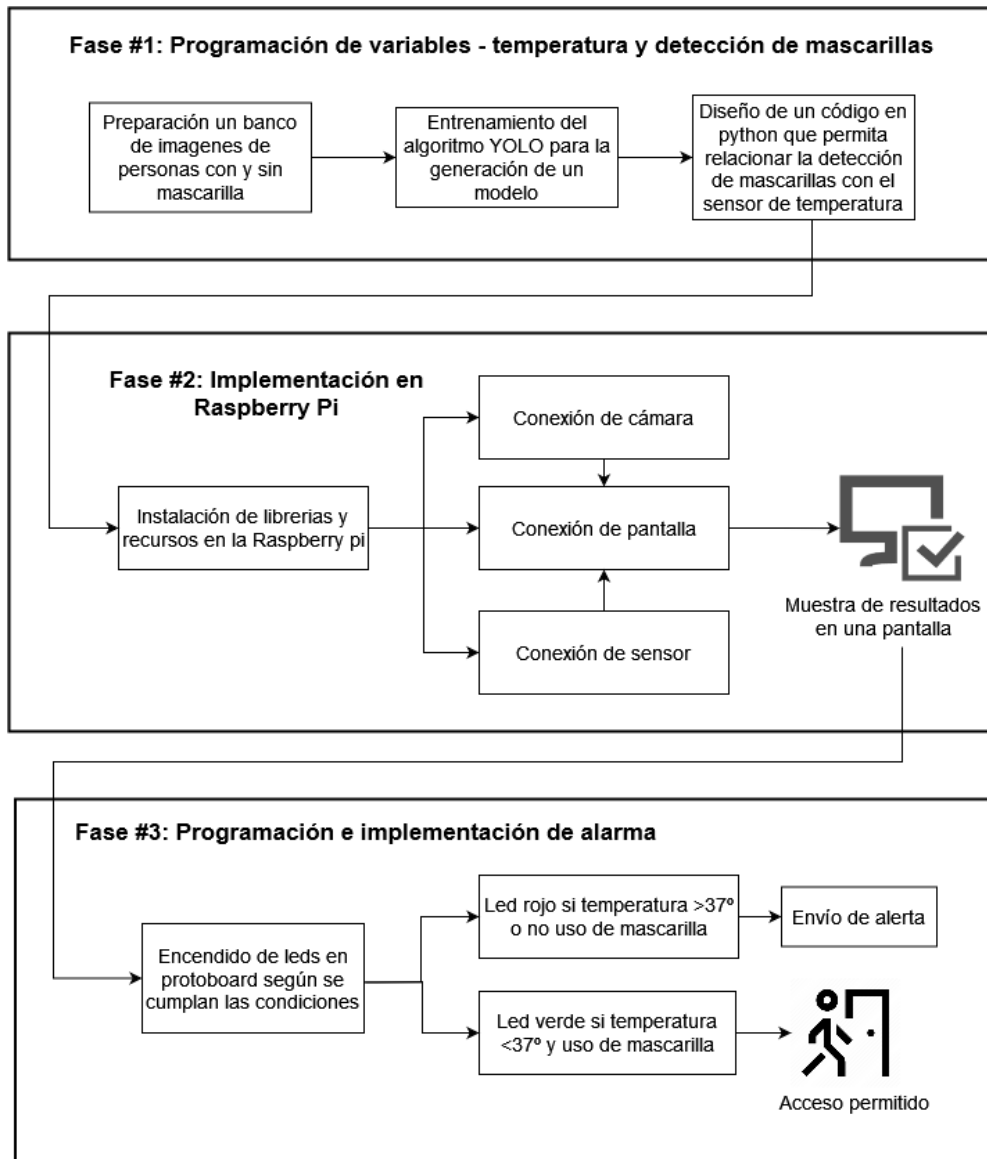


Ilustración 3.2 Diagrama de bloques detallado de la solución.

De la cual la primera fase trata sobre la preparación de un banco de imágenes de personas con y sin mascarilla, como se observa en la ilustración 3.3 y 3.4, y el entrenamiento del algoritmo que se puede observar más a detalle en la sección 3.3 y 3.4. Que comprenden procesos como conversión de datos de entrenamiento con matrices, colocación de etiquetas a algunas imágenes del dataset y partición de tanto del conjunto de imágenes compiladas del entrenamiento y del conjunto de imágenes de prueba usando librerías propias de Python ideales para procesamiento de imágenes.

En esta misma fase, el siguiente paso se trata del diseño e implementación de un código con el lenguaje de programación Python que permita integrar tanto el modelo que detecta mascarillas con la detección de temperatura corporal humana. Este código lee el modelo y lo aplica al video obtenido de una webcam de marca Rippa 1080p con conector USB y tiene las instrucciones para interactuar con el sensor de temperatura infrarrojo y los leds por medio de varias librerías, también debe controlar los pines físicos de la micro-computadora y la parte lógica compuesta por el modelo de ML.



Ilustración 3.3. Banco de imágenes de personas con mascarilla.

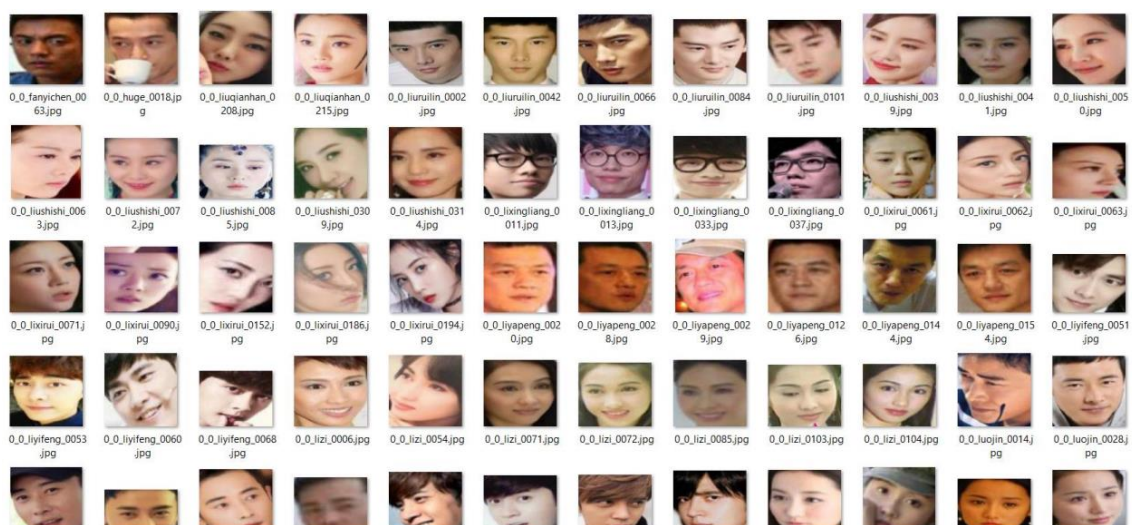


Ilustración 3.4 Banco de imágenes de personas sin mascarilla.

La segunda fase se trata de la integración de los archivos arrojados del modelo y su código en una Raspberry Pi junto con demás herramientas necesarias para su funcionamiento como un sensor de temperatura, webcam, una pantalla para observar los resultados, entre otras. Se instalan primero las librerías y sistema operativo dentro de la Raspberry, se implementa el código y se muestran los resultados como se observa en la siguiente ilustración 3.5.



Ilustración 3.5 Gráfico esquemático general de la solución [73] [74] [75].

La tercera fase trata de la lógica condicional que seguirá este proceso ya que existen dos variables condicionantes, la primera es usar mascarilla y la segunda es que la temperatura corporal sea menor a 37° C. Al cumplirse las dos condiciones antes mencionadas el prototipo encenderá un led verde indicando que se permite el acceso al área cerrada, caso contrario se generara una alerta indicando que hubo una falta de bioseguridad, esta alerta se enviará a una persona encargada de la seguridad de todo el establecimiento.

A continuación, se explicará cada etapa del diagrama de bloques básico de la ilustración 3.1 con respecto al diseño del modelo a utilizar.

3.3 PREPARACIÓN DEL MODELO

Para el diseño del modelo personalizado se toma como base el modelo de computer vision YOLO v4 [76]. Dado que ofrece una mayor velocidad con respecto a su versión anterior YOLO v3 y a otros modelos como R-CNN, fast R-CNN y faster R-CNN. En específico los modelos R-CNN brindar mayor precisión, pero con una velocidad notablemente inferior y con un consumo de procesamiento mayor a comparación de YOLO. Esta velocidad en la detección permite obtener mayores

fotogramas por segundo (fps) en la detección de objetos en cámaras a tiempo real, lo que da ventajas en dispositivos con poca capacidad de procesamiento como el Raspberry pi.

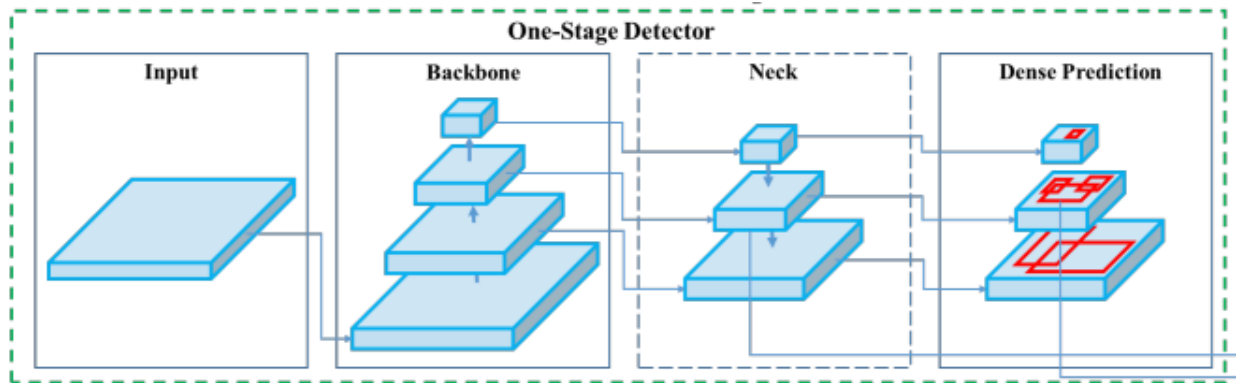


Ilustración 3.6 Etapas de un detector de objetos [77].

En la ilustración 3.6 podemos observar las diferentes etapas que tiene el detector YOLO para realizar la predicción de una imagen o video. La primera etapa es el input o dato de entrada a analizar, en este proyecto se toma como datos de entrada una secuencia de imágenes en forma de video con un frame rate de 30 fotogramas por segundo. El backbone o columna vertebral se utilizan como extractores de características, están entrenados previamente en conjuntos de datos de clasificación de imágenes y luego se ajustan en el conjunto de datos de detección.

Mientras más capas existan en esta etapa mayor será la eficacia al momento de realizar la detección, pero puede acarrear problemas de rendimiento por ello existen variaciones del modelo. El Neck o cuello contiene capas adicionales que van entre el backbone y el dense prediction. Se utilizan para extraer diferentes mapas de características de diferentes etapas de la red. El Dense Prediction es la etapa de predicción y se trata de una red encargada de realizar la clasificación y regresión de los cuadros delimitadores con la finalidad de realizar la detección. Su output o salida son 4 valores que describen el cuadro delimitador predicho, este cuadro contiene las coordenadas (x, y, h, w) donde 3 son de espacio y una extra del fondo, además otra salida es la probabilidad de aparición en k clases.

Una vez determinado el modelo a utilizar se realizan algunas variaciones en el código fuente alojado en un repositorio abierto de github [78], estas modificaciones se

realizan con la finalidad de adaptar el modelo al conjunto de datos de entrenamiento para generar el modelo personalizado.

Se diseñará el modelo YOLOv4 de ML y se lo entrenará con un dataset de 20000 imágenes de personas dividido en dos grupos: con mascarilla y sin ella, estas imágenes fueron obtenidas de un repositorio abierto de rostros asiáticos [78].

3.4 ENTRENAMIENTO DEL MODELO

Para realizar el entrenamiento del modelo primeramente se debe preparar el conjunto de datos. Las imágenes fueron obtenidas del siguiente repositorio [79], el conjunto de imágenes contiene 1000 imágenes de personas con mascarilla y 1000 imágenes de personas sin mascarilla con diferentes planos fotográficos y de diferentes edades, estas imágenes deben ser procesadas para que el modelo las pueda interpretar.

Para el procesamiento se utiliza un software de código abierto llamado labellmg, este programa permite etiquetar las imágenes y adaptarlas al modelo YOLO, una vez se etiqueta el área de interés de la imagen se genera un archivo txt con las coordenadas de los 4 ejes del cuadro identificador.

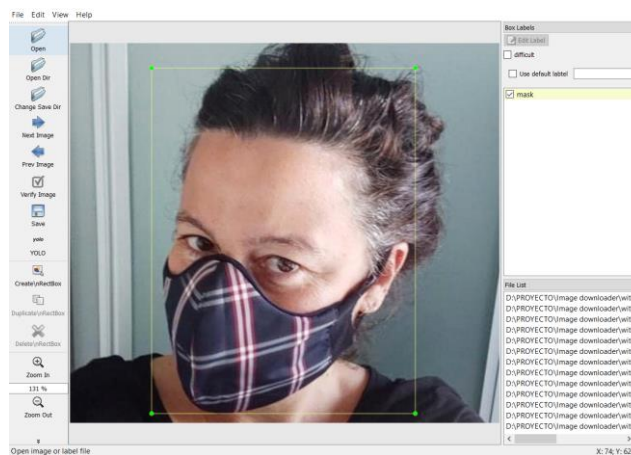


Ilustración 3.7. Etiquetado de rostro con mascarilla.

En la ilustración 3.7 se puede observar el área que se etiqueta para entrenar al algoritmo YOLO v4, en este caso se etiqueta el rostro completo el cual posee una mascarilla colocada correctamente, el proceso se repite para todo el conjunto de imágenes de personas con mascarillas.

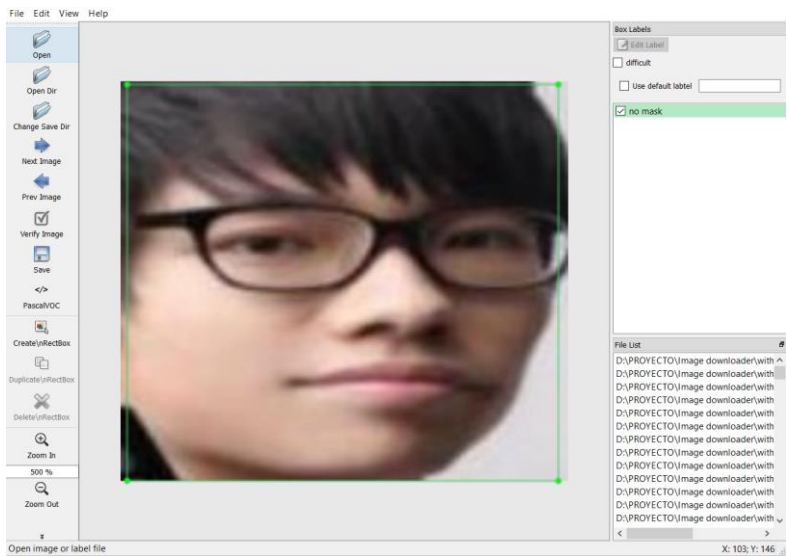


Ilustración 3.8. Etiquetado de rostro sin mascarilla.

La ilustración 3.8 muestra el etiquetado para el rostro de una persona sin mascarilla, el proceso es el mismo realizado desde la ilustración 3.2.

Además, debido a las ventajas mencionadas en la sección 2.2 sobre los lenguajes de programación, el lenguaje utilizado será Python. Este modelo detecta la mascarilla y las posiciones de los puntos clave de la cara, y devuelve un valor de probabilidad con respecto al uso de la mascarilla en base a 2 etiquetas: mask o no mask en un recuadro de color verde o rojo respectivamente.

Se debe representar este modelo como una secuencia de bytes con información esencial guardándolo en un archivo dentro del disco para poder leer sus atributos correctamente en su interfaz. Después se carga esta información guardada y se la ejecuta.

Al concluir con el proceso de etiquetado se procede a configurar parámetros del modelo para el entrenamiento como el número de clases que hace referencia al número de casos con los cuales se va a entrenar. Para este proyecto se utiliza 2 clases llamadas mask y no mask las cuales hacen referencia a si tiene puesta o no mascarilla.

Se configuran los siguientes parámetros en el archivo yolov4-custom.cfg para adaptar el modelo a los datos procesados de acuerdo con el ancho y altura de la resolución de las imágenes ya vistas, la tasa de aprendizaje, los pasos de línea, la cantidad de clases y sus propiedades:

```
width,height = 416x416
max_batches = (2000 * #classes) = 4000
steps = (0.8 * max_batches, 0.9 * max_batches)
classes = 2
filters = (classes + 5) * 3
```

Ilustración 3.9 Parámetros para adaptar el modelo a los datos procesados.

Al completar la configuración se procede a ejecutar el entrenamiento personalizado en una máquina virtual que se ejecuta en la nube de Google llamada Google Colab, esto permite que el entrenamiento sea más rápido dado que para este volumen de datos se necesita una maquina con una GPU y CPU de alta gama.

Google Colab permite ejecutar código y entrenar modelos utilizando servidores de Google con la capacidad suficiente para ejecutar el entrenamiento del modelo de este proyecto, el entrenamiento tuvo una duración de 18 horas.

Ejecutar un script en Google Colab no tiene costo, pero posee una limitación por tiempo de inactividad por ello se ejecutó un comando en la consola del navegador con la finalidad de dar clic en la pantalla repetidamente en un lapso de tiempo para que no se interrumpa el entrenamiento, sin embargo, se puede reanudar el proceso cargando los mismos datos y el ultimo check point generado por el algoritmo.

Una vez completado en entrenamiento se obtiene un archivo .weights que contiene la información recopilada de todo el entrenamiento de datos, este archivo se convierte a un modelo de TensorFlow para poder utilizar la librería del mismo nombre en la implementación del código en Python y lograr incorporar las funcionalidades establecidas al inicio de este proyecto.

3.5 DIAGRAMA DE CONEXIONES

A pesar de existir diversos tipos de microcomputadoras, en este proyecto se utilizará la Raspberry Pi 4 B+, es una minicomputadora que se conecta a televisores, monitores de computadora, etc. y es compatible con la metodología de ML en donde la versión 4 se usará debido a su rapidez a comparación con versiones anteriores.

Raspbian, con la versión de Debian 11 [80], es el sistema operativo oficial de la Raspberry Pi y es una versión de Linux construida específicamente para Raspberry Pi y se adapta mejor a los objetivos planteados específicos para este proyecto.

Luego de desarrollar el código se lo implementa en la Raspberry pi 4 B+, se realiza la conexión con la webcam y tanto como el sensor de temperatura infrarrojo MLX90614 como los leds se conectan a los pines GPIO de la Raspberry siguiendo el diagrama de conexiones siguiente en la ilustración 3.10 con los componentes físicos a utilizar en el presente proyecto de tesis.

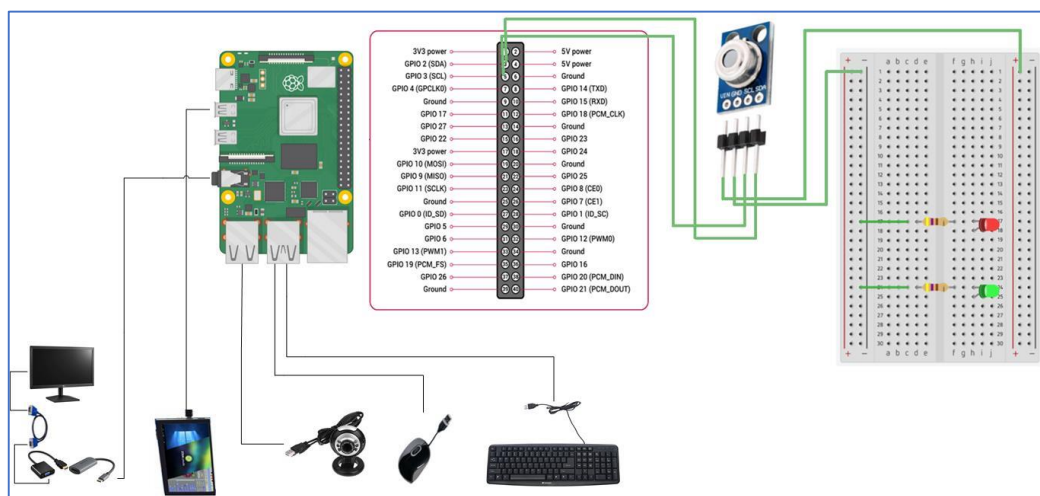


Ilustración 3.10 Diagrama de conexiones general [81].

En la ilustración 3.6 se visualiza el diagrama de conexiones con respecto a la Raspberry y al Protoboard, como periféricos de entrada se tiene a la webcam de marca Rippa 1080p con conector USB, un mouse, el sensor infrarrojo MLX90614 de termopila de 4 pines y el teclado los cuales estarán conectados a 3 de los 4 puertos USB 2.0 respectivos de la Raspberry. La pantalla LCD se conectó al puerto DSI y el monitor con los adaptadores de cable VGA a HDMI y su cable fuente de poder. También se observa la conexión del sensor desde el Protoboard y de los 2 leds rojo y verde con sus respectivas resistencias de 68 Ohms y jumpers a los pines físicos de la Raspberry.

3.6 IMPLEMENTACIÓN DEL CÓDIGO

Una vez conectados los periféricos se procede a programar en Python un programa que lee el modelo generado y conecta lógicamente todos los dispositivos para que se comuniquen entre sí y se pueda obtener los resultados planteados.

El código inicia importando las librerías necesarias para ejecutar el proyecto, dichas librerías son:

- Tensorflow
- Keras
- Imutils
- Numpy
- OpenCv
- RPi.GPIO
- Busio
- adafruit_mlx90614

Estas librerías brindan funciones y métodos capaces de permitir la comunicación y control de tanto software como hardware, el ejemplo más claro de esto es la librería RPi.GPIO la cual permite controlar los pines analógicos de la Raspberry del cual se puede conectar sensores, leds, pantallas, motores y definir las salidas de voltaje para cada pin.

Estas librerías ya mencionadas en la sección 2.3 ofrecen facilidades para manipular las imágenes arrojadas por el algoritmo del modelo YOLOv4, algunos de estos paquetes ofrecen manipulación de operación con matrices multidimensionales con algoritmos de tipos de aprendizaje de manera automática y visión por computadora.

Posteriormente se procede a definir los pines analógicos que utilizaran el sensor infrarrojo y los leds, para así definir una función que se encargara de leer el modelo y aplicarlo al video capturado de una webcam. Este video se captura por medio de la librería OpenCv, se coloca dentro de un bucle While para que cada frame sea analizado mediante Keras y TensorFlow para aplicar el modelo entrenado, obteniendo un recuadro alrededor del rostro de una persona que determina la posición del rostro, similar el etiquetado que se realizó en el procesamiento de las imágenes.

En la parte superior del recuadro que encierra el rostro y se muestra la probabilidad de las 2 clases del entrenamiento mask y no mask. Estas clases determinan la probabilidad de que una persona este utilizando o no mascarilla, el criterio de elección entre una clase u otra es la probabilidad ya que cuando una probabilidad sea mayor esta se mostrara en la frame generado un texto con dicha información.

Para la detección de temperatura se usa el sensor de temperatura infrarrojo MLX90614 conectado a los pines de la Raspberry, estos pines se definen digitalmente con la librería `adafruit_mlx90614` mediante los siguientes comandos de la ilustración 3.11 que se observa a continuación:

```
i2c = io.I2C(board.SCL, board.SDA, frequency = 100000)
    mlx = adafruit_mlx90614.MLX90614(i2c)
ambientTemp = "{:.2f}".format(mlx.ambient_temperature)
targetTemp = "{:.2f}".format(mlx.object_temperature)
```

Ilustración 3.11 Comandos de la librería de adafruit.

En la primera línea de código se puede observar cómo se define el objeto I2C el cual es un protocolo para comunicarse con periféricos de baja velocidad, ya que la Raspberry tiene dos buses I2C.

Cada bus tiene una Central I2C conectada a dos líneas bidireccionales, una línea de datos en serie (SDA) y un reloj en serie (SCL).

Entonces se definen los pines en la tarjeta y la frecuencia, en la segunda línea se inicializa el objeto con la librería antes mencionada. Las dos últimas líneas son las que muestran la temperatura capturada del ambiente y del objetivo con un formato de 2 decimales.

Al contar con todas las variables definidas se plantean las condiciones que se deben cumplir para permitir el acceso al área cerrada, se toman 2 variables de interés una es la `mask_accuracy` que da la probabilidad de que una persona tenga mascarilla que va de 0 a 1 y el `targetTemp` que da la temperatura captada por el sensor.

Se define la condición mediante un `if` como se muestra a continuación:

```
if mask_accuracy > 0.7 and float(targetTemp) < 37.0:
```

Ilustración 3.12 Comando con condición de temperatura.

Mediante el operador lógico AND se asegura que se tengan que cumplir 2 condiciones para entrar al caso positivo las cuales son:

- *mask_accuracy* > 0.7, que la probabilidad de que tenga mascarilla sea mayor al 70%.
- *targetTemp* < 37.0, que la temperatura medida de la persona sea inferior a 37 grados Celsius.

Ilustración 3.13 Comandos con 2 condiciones de uso de mascarilla y temperatura.

Una vez encendida la cámara web, el modelo detecta los rostros mostrados de la secuencia de video de imagen y se enfoca en el área de interés de cada rostro que es el área inferior.

Después de que se haya extraído la región dicha en un rectángulo, se aplica la detección en esa área y se obtiene con qué tipo de etiqueta se está trabajando ya sea con o sin mascarilla.

Una vez se cumplan estas condiciones se procede a encender un led indicador de color verde que significa que puede acceder al área, caso contrario se encenderá un led rojo indicando que se incumplen una de las dos condiciones.

Cuando entra al caso negativo (else) además de encender el led rojo se envía una alerta de seguridad por correo electrónico indicando que una persona no está usando mascarilla o su temperatura corporal es alta.

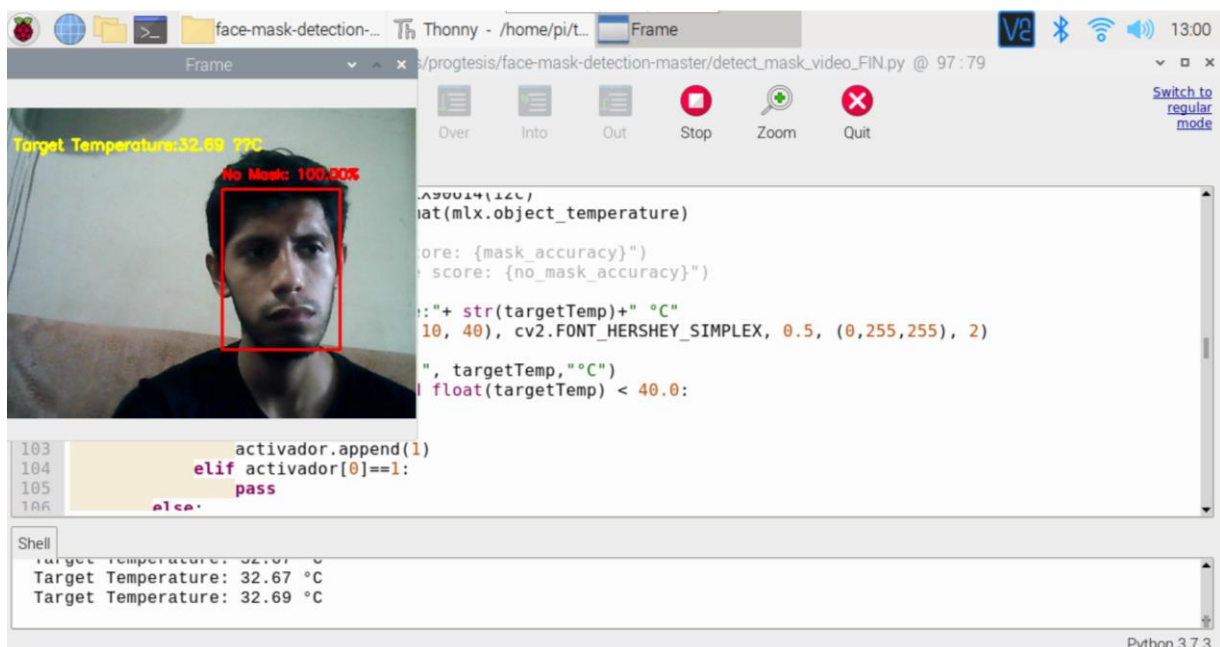


Ilustración 3.14. Visualización del frame generado por el código cuando no se usa mascarilla

En la ilustración 3.14 se puede observar el video capturado por la cámara web, el reconocimiento que hace del rostro de la persona y el análisis a tiempo real que muestra un recuadro similar al realizado durante el procesamiento de las imágenes de entrenamiento. En la parte superior muestra la probabilidad que no use mascarilla la cual es 100%, mostrando la efectividad del modelo generado. A una esquina del frame se muestra la temperatura medida a tiempo real por lo que el usuario podrá ver toda esta información durante la ejecución del proyecto.

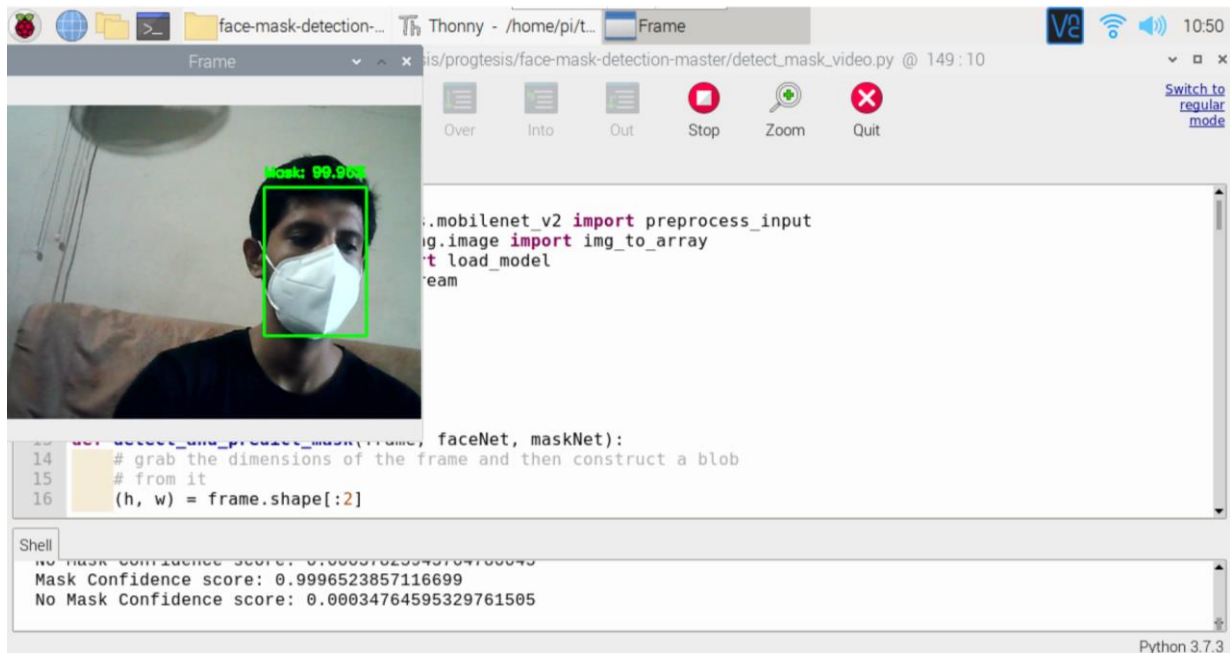


Ilustración 3.15. Visualización del frame generado por el código cuando se usa mascarilla

La ilustración 3.15 se muestra el caso para cuando se usa la mascarilla correctamente, se puede notar como se genera un recuadro que cubre todo el rostro correctamente y muestra la probabilidad que tenga mascarilla la que para este caso es 99.96%. Se reafirma la eficiencia del modelo para detectar rostros con y sin mascarilla con un alto grado de fiabilidad. Si se desean tasas más altas de probabilidad para casos especiales o complicados es necesario aumentar el volumen de imágenes de entrenamiento en los dos casos analizados.

3.7 IMPLEMENTACIÓN EN PROTOTIPO



Ilustración 3.16 Conexiones esenciales de la Raspberry.

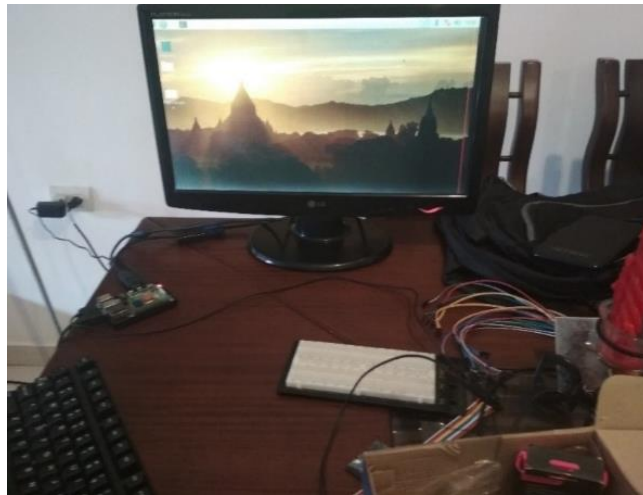


Ilustración 3.17 Instalación de recursos, sistema operativo y librerías.

```
New Load Save Run Debug Tools Help
legoTemp.py *
1 #VIn = pin 1
2 #SCL = pin 5
3 #SDA = pin 3
4
5 import board
6 import busio as io
7 import adafruit_mlx90614
8 from time import sleep
9
10 i2c = io.I2C(board.SCL, board.SDA, frequency=100000)
11 mlx = adafruit_mlx90614.MLX90614(i2c)
12
13 ambientTemp = "{:.2f}".format(mlx.ambient_temperature)
14 targetTemp = "{:.2f}".format(mlx.object_temperature)
15
16 sleep(1)
17 print(type(ambientTemp))
18 print("Ambient Temperature:", ambientTemp, "°C")
19 print("Target Temperature:", targetTemp, "°C")
20
21
22 <class 'str'>
23 Ambient Temperature: 27.91 °C
24 Target Temperature: 24.42 °C
```

Ilustración 3.18 Diseño de código.

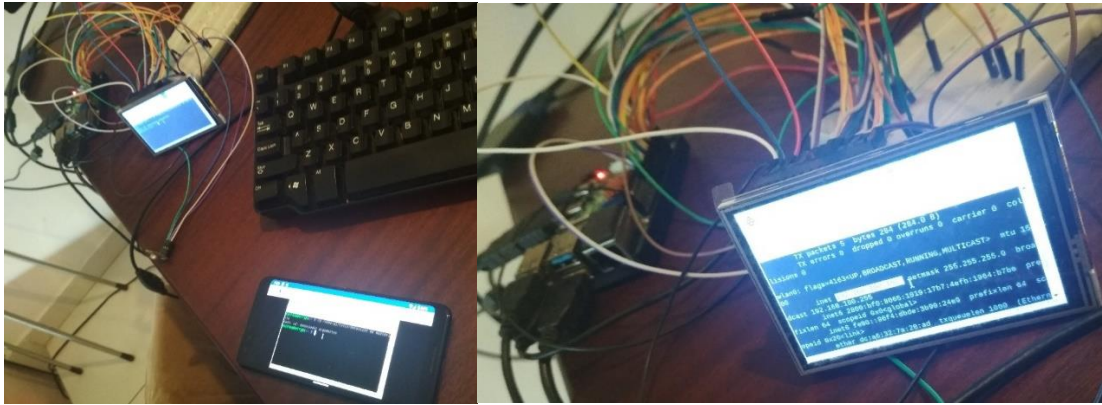


Ilustración 3.19 Configuración de pantalla LCD.



Ilustración 3.20 Implementación en Raspberry.

Se realizaron las conexiones del diagrama de conexiones de la sección 3.5 en la Raspberry, se instaló Raspbian y se la tuvo conectada a una red inalámbricamente. Se utilizó el código ya diseñado y se lo implementó junto con sus librerías respectivas ya dichas y se realizaron las conexiones de la pantalla LCD tanto en los pines de la Raspberry como en el Protoboard para que con la programación creada se pueda

observar su interfaz en esta pantalla como se observa en la última imagen la ilustración 3.20.

Se visualiza una pantalla LCD encendida y correctamente programada, la implementación del sensor dentro del sistema y sus conexiones en el Protoboard y Raspberry. Se realizaron varias pruebas del código y su detección de mascarillas con un valor de probabilidad de uso o no de mascarilla. Se utilizó una webcam de marca INS modelo IS-PC28.

```
def detect_mask_video.py K
1
2 from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
3 from tensorflow.keras.preprocessing.image import img_to_array
4 from imutils.video.models import load_model
5 import numpy as np
6 import argparse
7 import imutils
8 import time
9 import cv2
10 import os
11
12
13 def detect_and_predict_mask(frame, faceNet, MaskNet):
14
```

Ilustración 3.21 Incorporación de alarma al sistema.

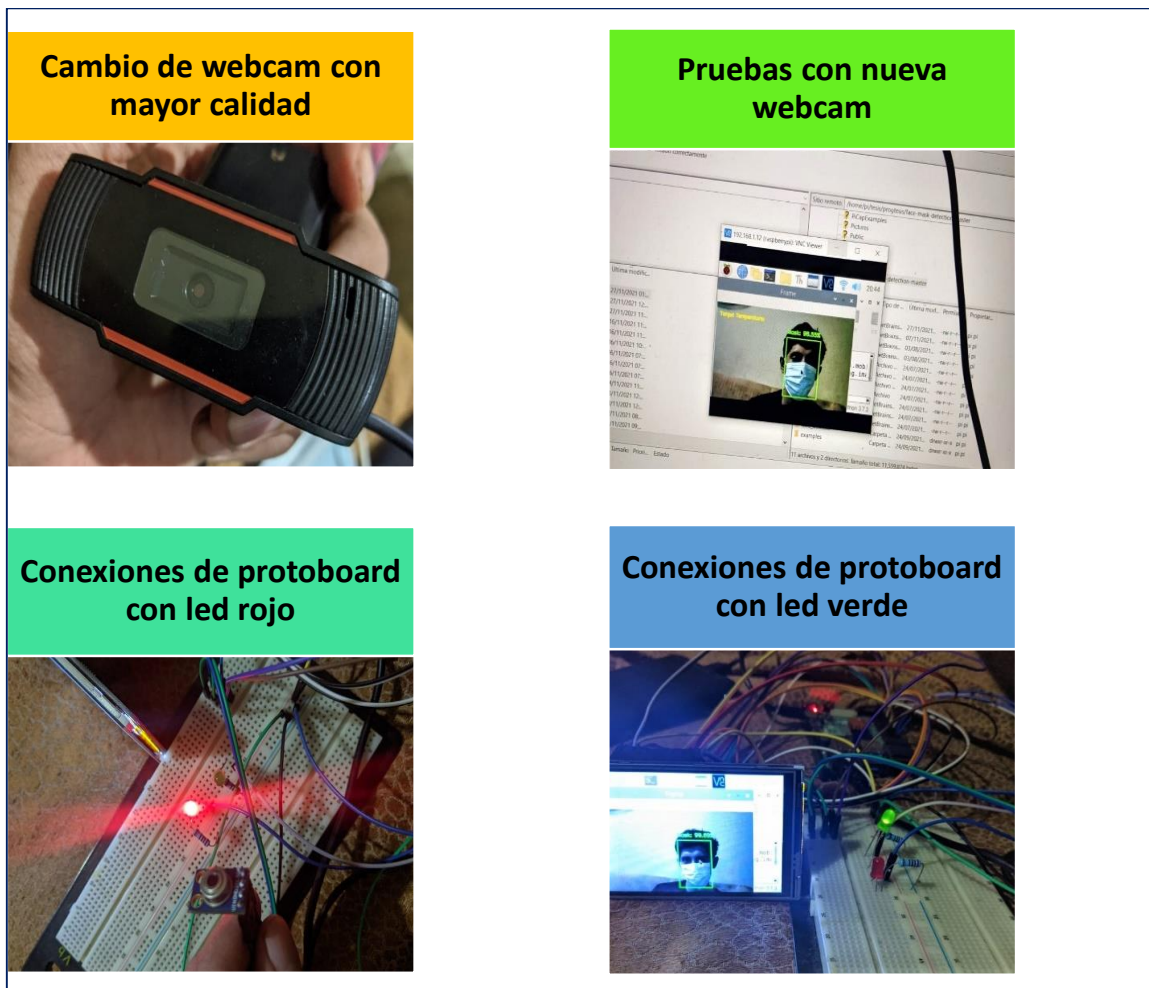
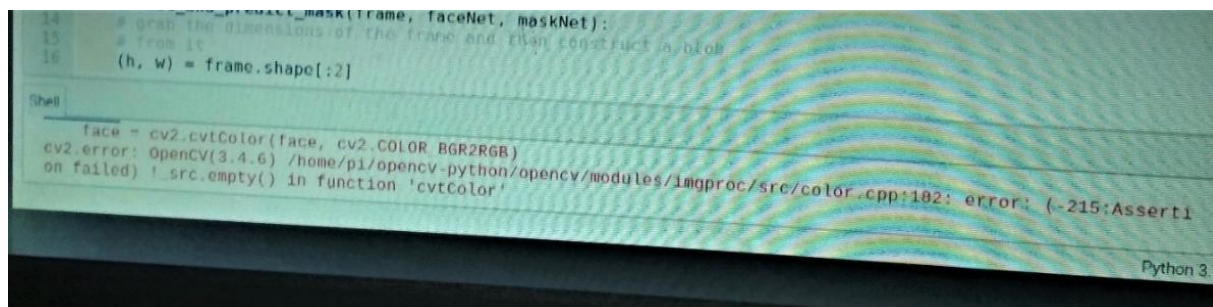


Ilustración 3.22 Programación e implementación de alarma al sistema.

Con la cámara que se observa en la ilustración 3.22 se observaron problemas con el tiempo de ejecución del código y problemas de resolución de imagen por lo que se optó por cambiar de marca y de resolución. Este error se puede observar en la siguiente ilustración 3.23 sobre librerías utilizadas.



```
14 # grab the dimensions of the frame and then construct a blob
15 # from it
16 (h, w) = frame.shape[:2]

Shell
face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
cv2.error: OpenCV(3.4.6) /home/pi/opencv-python/opencv/modules/imgproc/src/color.cpp:102: error: (-215:Assert1
on failed) ! src.empty() in function 'cvtColor'

Python 3.7
```

Ilustración 3.23 Error del código debido a la calidad de la imagen

Además, se le añadió la ubicación del display de la temperatura y se la ubicó a lado del valor arrojado de probabilidad de uso de mascarilla en el stream de video como se puede visualizar en la ilustración 3.18 en las pruebas con la nueva webcam.

El modelo seleccionado de la cámara web para el proyecto es altamente compatible con Raspberry Pi y a las necesidades del código con respecto a su resolución de imagen y su adaptación a la librería de “opencv” con respecto a funcionamiento correcto, también el módulo de la cámara se puede usar para tomar videos de alta definición, así como fotografías, lo que permite que la cámara capture imágenes y detecte rostros con y sin mascarilla.

Este modelo es de la marca “Rippa” con una resolución de 1280*720 pixeles y con entrada USB que se adapta a la versión del hardware de la Raspberry a utilizar en este proyecto.

Finalmente se programó el sistema de alarma y se lo agregó al sistema de acuerdo al código diseñado anteriormente y conexiones realizadas según el diagrama de conexiones. Teniendo así al sistema completo mostrado en las siguientes ilustraciones 3.24 y 3.25.

El sistema se compone principalmente de un algoritmo de detección de mascarillas y un algoritmo de medición de temperatura. Las funciones principales del

sistema se pueden dividir en tres partes: detección de mascarillas, medición de temperatura y envío de alarma.

Luego se realizaron pruebas con las siguientes condiciones:

Tabla 3.1 Condiciones de prueba del funcionamiento del prototipo	
Tiempo	1 hora
Distancia desde webcam	1 metro
Distancia desde sensor	5 centímetros

Para estas pruebas se ejecutó el programa con una distancia de 1 metro desde la webcam y 5 cm desde el sensor infrarrojo en un total de 1 hora de donde se capturaron 2 fotos con un porcentaje de uso de mascarilla y del sensor de temperatura.

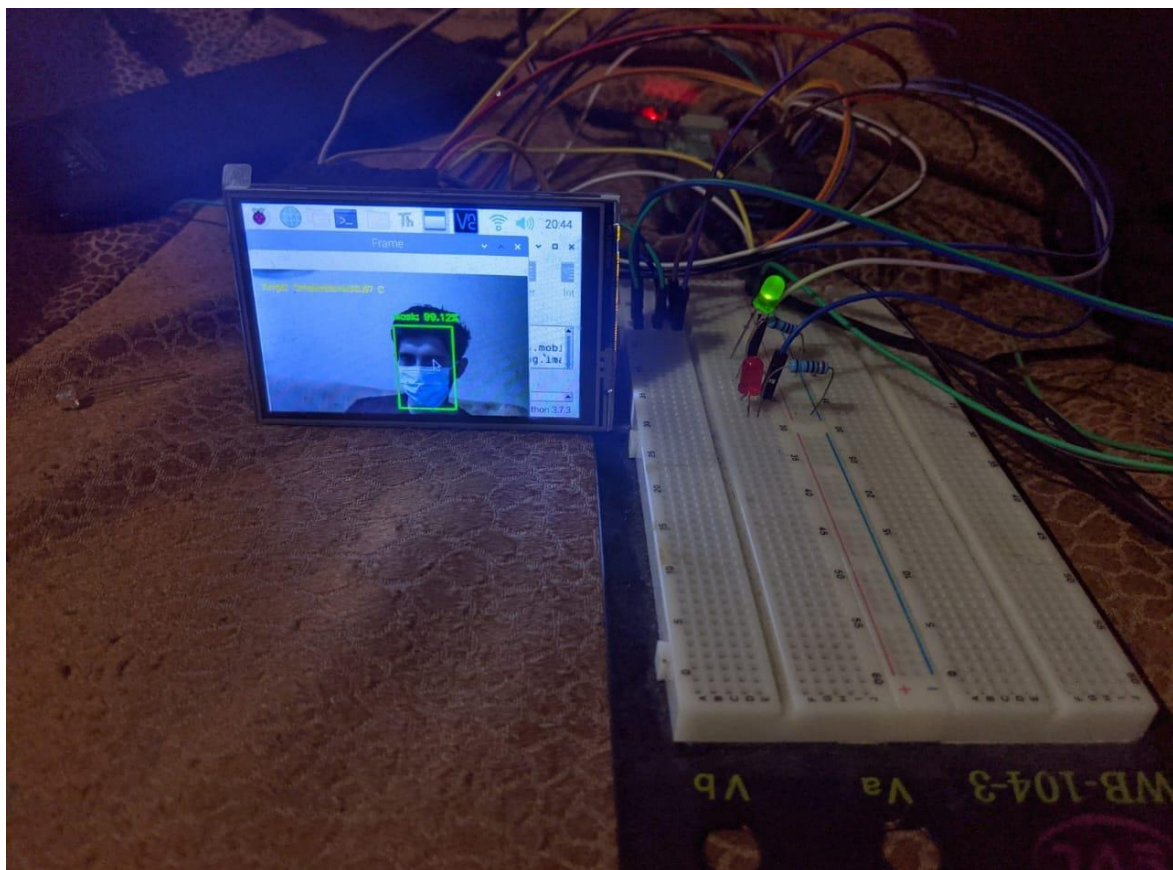


Ilustración 3.24 Pruebas realizadas con el prototipo

Se observa en la ilustración 3.24 la detección de mascarilla y su uso con un valor de probabilidad de uso de 99,69% con la etiqueta de “mask” y temperatura de 36.57°C, y que debido a que estas cumplen las condiciones ya dichas se observa un led verde encendido dando así un ingreso permitido.

Este ingreso permitido no se refleja en una alarma ya que no representa un riesgo biológico siendo solamente utilizada en caso de que exista un riesgo de contagio indicado en una temperatura corporal alta y falta de uso de mascarilla.

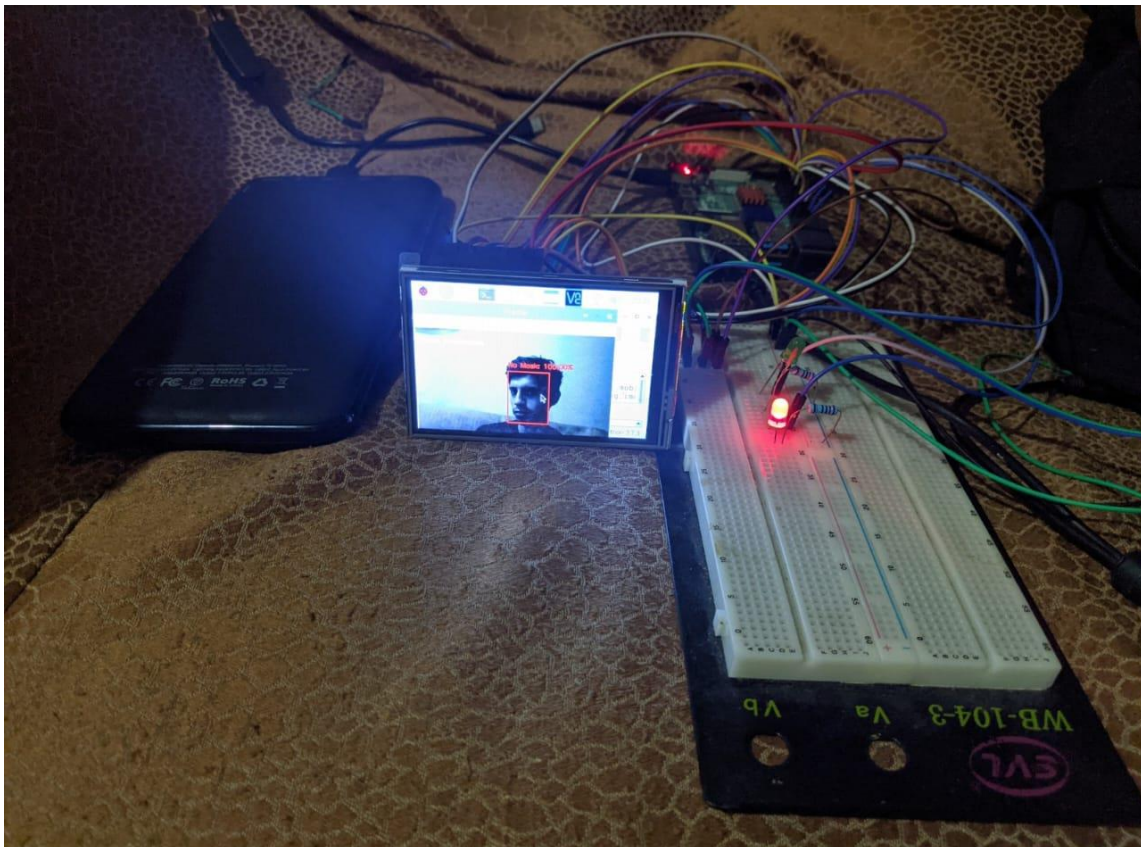


Ilustración 3.25 Pruebas realizadas con el prototipo

Se observa en la ilustración 3.25 la detección del no uso de mascarilla y con un valor de probabilidad de 98,99% con la etiqueta de “no mask” y temperatura de 37.14°C, y que debido a que no se cumple la condición del uso de mascarilla, se observa un led rojo encendido dando así un ingreso denegado y generando una alarma.

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

En este capítulo se presentará el escenario donde se realizaron las pruebas con el prototipo del proyecto, cómo estas se realizaron, los resultados hallados en tablas correspondientes y el análisis de estos con el fin de conocer la efectividad del prototipo realizado.

4.1 ESCENARIO

Una vez ya construido el prototipo a su cabalidad y verificado su correcto funcionamiento, se lo ubicó en un área cerrada del Hospital León Becerra como lo es el área de cuidados intensivos o UCI como se observa en la siguiente ilustración.



Ilustración 4.1 Prototipo ubicado en el hospital.

En esta área de UCI, las pruebas se realizaron con una duración de 4 horas con un total de 20 personas con un promedio de 1 persona cada 12 minutos desde las 08:00am a 12:00am, siendo este horario uno de los más concurridos en el día.



Ilustración 4.2 Ubicación de las pruebas a realizar junto al área de enfermería.

Para tener un mejor control, se optó por anotar los primeros nombres de las personas que ingresaban para poder identificarlos en las siguientes tablas de resultados.

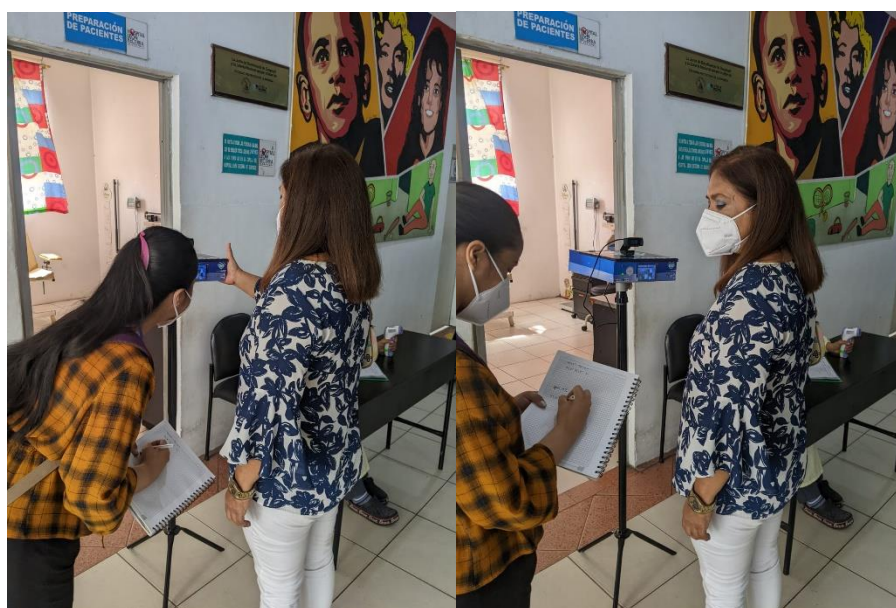


Ilustración 4.3 Toma de datos medidos y nombres de los usuarios.

Una vez ya planteado el escenario, se encendió el prototipo con su batería y se lo ubicó en la puerta de ingreso de la sala Santa María, se tuvieron como variables de control a la temperatura medida, la probabilidad resultante en la pantalla LCD de uso o no de mascarilla dependiendo de los usuarios, la distancia a la cual se realizaron las mediciones por la webcam de detección de mascarilla, y la respuesta de las luces LED como decisión final.



Ilustración 4.4 Desarrollo de las pruebas.

Los resultados con respecto a la **variable de detección de mascarilla** se observan a continuación.

Tabla 4.1 Porcentaje de exactitud de detección de mascarilla			
#1: Andrés	With mask: 99.73%	#11: José	With mask: 80.50%
#2: Mariana	With mask: 99.23%	#12: David	With mask: 99.99%
#3: Sebastián	With mask: 99.94%	#13: Gloria	With mask: 84.44%
#4: Julio	With mask: 98.63%	#14: Diana	With mask: 94.50%
#5: Azucena	With mask: 99.95%	#15: Manuel	With mask: 95.64%
#6: Moisés	With Mask: 99.90%	#16: Lola	With mask: 86.50%
#7: Víctor	With mask: 97.30%	#17: Michael	With mask: 95.20%
#8: Jimena	With mask: 87.99%	#18: Giuliano	With mask: 99.20%
#9: Mario	With mask: 99.73%	#19: Elizabeth	With mask: 99.40%
#10: Jacinto	With mask: 94.96%	#20: Danny	With mask: 91.30%

De acuerdo al total de usuarios de la muestra, se observó que el modelo utilizado logró identificar los rostros que utilizaban correctamente la mascarilla y los que no lo hacían. Se observó que el prototipo etiquetó como “with mask” a 20 de 20 personas que sí utilizaban mascarillas correctamente, es decir, un 100% del total de personas pasaron ese filtro ya que tenían un porcentaje de exactitud en la pantalla de 80.50% a 99.99% y que efectivamente, al observarlas detalladamente todos usuarios estaban usando su mascarilla debidamente.

Luego se realizó el análisis con respecto a la **variable de temperatura**.

Se tuvo la oportunidad de traer al escenario un termómetro digital infrarrojo modelo SK-T008 como se observa en la siguiente figura.



Ilustración 4.5 Sensor digital infrarrojo de temperatura [82].

Se lo utilizó para tener valores reales y poder comparar con los resultantes del prototipo, a continuación, se observa los valores reales de las 20 personas monitoreadas.

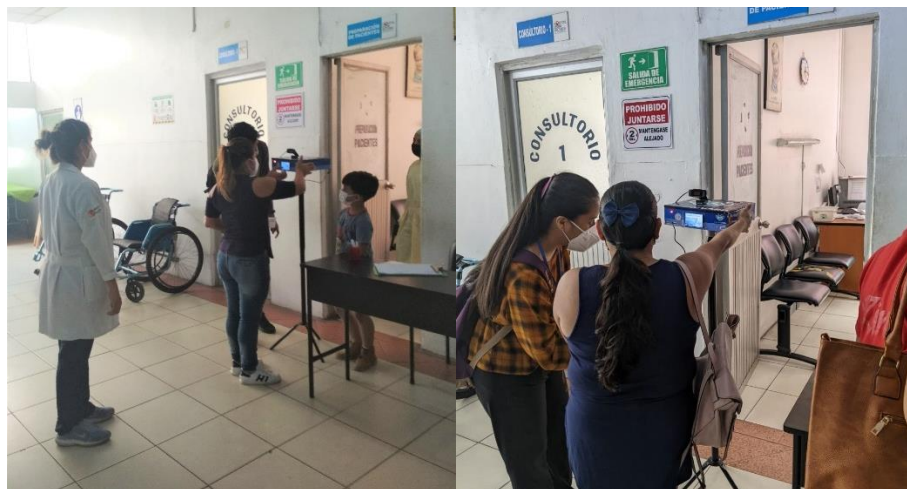


Ilustración 4.6 Medición de temperatura.

Tabla 4.2 Temperatura corporal con medida con el Prototipo			
#1: Andrés	34.65°C	#11: José	36.13°C
#2: Mariana	36.70°C	#12: David	36.1°C
#3: Sebastián	36.9°C	#13: Gloria	35.9°C
#4: Julio	36.55°C	#14: Diana	36.5°C
#5: Azucena	35.63°C	#15: Manuel	36.17°C
#6: Moisés	36.25°C	#16: Lola	36.49°C
#7: Víctor	35.55°C	#17: Michael	36.03°C
#8: Jimena	35.29°C	#18: Giuliano	35.8°C
#9: Mario	35.13°C	#19: Elizabeth	36.5°C
#10: Jacinto	36.39°C	#20: Danny	36.79°C

También se observó con respecto a la variable de la temperatura, que las 20 personas que utilizaron el sensor, el valor arrojado por la pantalla LCD de todas no fue mayor a 37°C con lo que se afirma que estas no presentaban fiebre con lo que esta variable de temperatura funcionó debidamente ya que se identificó a un posible portador de COVID-19 con uno de sus síntomas más comunes [83].

Tabla 4.3 Temperatura corporal con medida con el Termómetro digital Infrarrojo			
#1: Andrés	35.65°C	#11: José	36.17°C
#2: Mariana	36.80°C	#12: David	36.2°C
#3: Sebastián	36.7°C	#13: Gloria	36.0°C
#4: Julio	36.53°C	#14: Diana	36.9°C
#5: Azucena	35.93°C	#15: Manuel	36.27°C
#6: Moisés	36.85°C	#16: Lola	36.48°C
#7: Víctor	35.75°C	#17: Michael	36.04°C
#8: Jimena	35.79°C	#18: Giuliano	35.9°C
#9: Mario	35.83°C	#19: Elizabeth	36.6°C
#10: Jacinto	36.69°C	#20: Danny	36.78°C

Si se comparan los valores del termómetro digital infrarrojo con el del prototipo, se pudo observar valores de temperatura muy cercanos con diferencias de 0.02°C y con valores de error porcentuales máximos de 3.10% y 2.01% con lo que se verifica que la variable de temperatura del prototipo funciona correctamente. Y que 1 usuario

tuvo la temperatura de 34.65°C debido a que la persona ubicó su mano lejos del sensor y la retiró rápidamente, pero al compararla con el valor medido de la tabla de temperatura del sensor digital, este se encuentra dentro del error porcentual indicando que el sensor funciona correctamente.

En total de las 20 pruebas realizadas, sí se les permitió el ingreso a todas ya que todas cumplieron con las 2 variables de control y estas fueron debidamente verificadas por el ojo humano con respecto al uso de mascarilla y mediante la comparación con un sensor digital con respecto a la temperatura. En todos los casos se encendió el led verde ya que ambas variables arrojaron valores booleanos de 'True', es decir la variable de detección de mascarillas señaló mediante un cuadro verde la etiqueta de "With mask" con una probabilidad de entre 80.50% y 99.90% de exactitud y se encendió el led verde indicando su acceso permitido al área cerrada en cuestión.

En total fueron 20 de 20 personas permitidas exitosamente en el ingreso al área cerrada debido a que estas no presentaron posibles síntomas comprobados de la COVID-19 [84].

También se analizó la **variable de distancia efectiva** a la cual la webcam detecta nítidamente un rostro con y sin mascarilla, las 20 personas se acercaron a la webcam del prototipo y se midió su distancia con respecto a la misma de donde se obtuvo el siguiente resultado.

Tabla 4.4 Distancia de webcam para detección de mascarillas			
#1: Andrés	1 m	#11: José	1 m
#2: Mariana	1.10 m	#12: David	3 m
#3: Sebastián	80 cm	#13: Gloria	2.50 m
#4: Julio	1.30 m	#14: Diana	4 m
#5: Azucena	1.30 m	#15: Manuel	3 m
#6: Moisés	2 m	#16: Lola	1.80 m
#7: Víctor	2.5 m	#17: Michael	3.90 m
#8: Jimena	3 m	#18: Giuliano	2.55 m
#9: Mario	1.80m	#19: Elizabeth	1.10 m
#10: Jacinto	2.10 m	#20: Danny	1 m



Ilustración 4.7 Medición de distancia de detección de rostro.

De estos resultados se observa que todas las mediciones son menores a 4 metros con respecto a la webcam del prototipo y que la distancia máxima para que la webcam detecte un rostro con mascarilla es de 5 metros, a partir de este valor la webcam comienza a tener fallas de resolución y el algoritmo pierde resolución en las imágenes recibidas con lo que se pierde una menor precisión a partir de esta distancia ya que no logra detectar y etiquetar rostros correctamente.

4.2 GRÁFICAS

A continuación, se presenta un análisis con respecto a la exactitud del modelo utilizado antes y después de su entrenamiento.

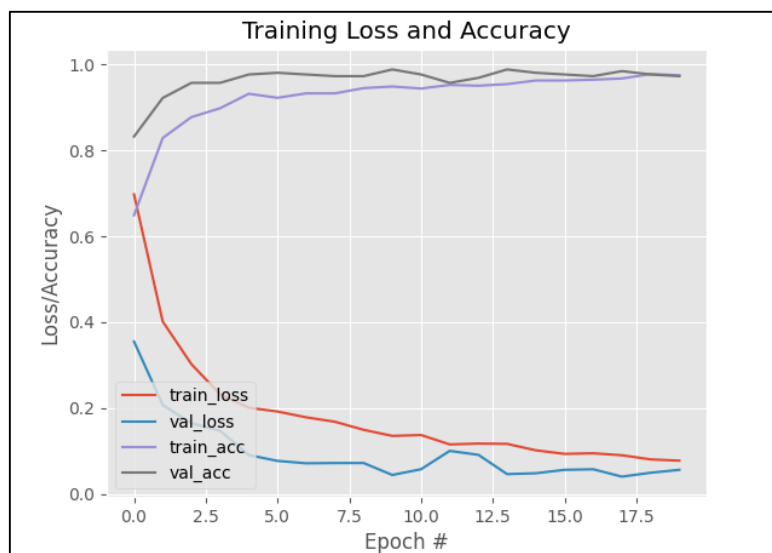


Ilustración 4.8 Grafica de Epoch vs Loss/Accuracy del modelo generado en el entrenamiento.

En la ilustración 4.7 se puede observar una gráfica de Epoch vs Loss/Accuracy, donde el número de Epoch o épocas representa el número de veces que el conjunto de datos atraviesa la Red Neuronal.

Si se utiliza un número de épocas muy alto se puede caer en overfitting o sobreajuste, es decir el modelo está entrenado para unos datos muy específicos y no es capaz de generalizar por lo que ante datos similares no podrá identificar correctamente los patrones.

Para este proyecto se utilizó un Epoch de 18, este número va a depender de la cantidad de datos y su diversidad, como se contemplaron solo 2 casos los cuales son con y sin mascarilla, entonces el número de épocas no debe ser muy alto.

En el eje x se encuentra la probabilidad de Loss/Perdida y Accuracy/Exactitud los valores van desde 0 a 1 que significa 100%.

Se pueden observar 4 curvas en la gráfica, las cuales son las siguientes:

La curva de `train_loss` representa la pérdida de información en el entrenamiento mientras un lote de datos pasa a través de la red neuronal, es decir se mide mientras ocurre un epoch. En las primeras épocas la pérdida de entrenamiento se encontraba entre el 20% al 60%, pero una vez se alcanza el final de las épocas este porcentaje baja hasta 9%.

La curva de `train_accuracy` representa la información que fue exitosa durante las épocas, se puede notar como esta curva alcanza su máximo en la última época llegando a un 98% de éxito.

La curva de `val_loss` representa la pérdida después de validación y se mide después que pasa una época, debido a esto la curva de pérdida de validación muestra un porcentaje menor en la última etapa al compararse con la curva de pérdida de entrenamiento.

La curva de `val_accuracy` representa el éxito de validación una vez pasa por una época, se puede observar que esta gráfica alcanza picos más altos antes de llegar a la última época, pero en dichos picos los valores de pérdida se encuentran altos por lo que según las pruebas los mejores resultados se obtienen con el número de épocas definido.

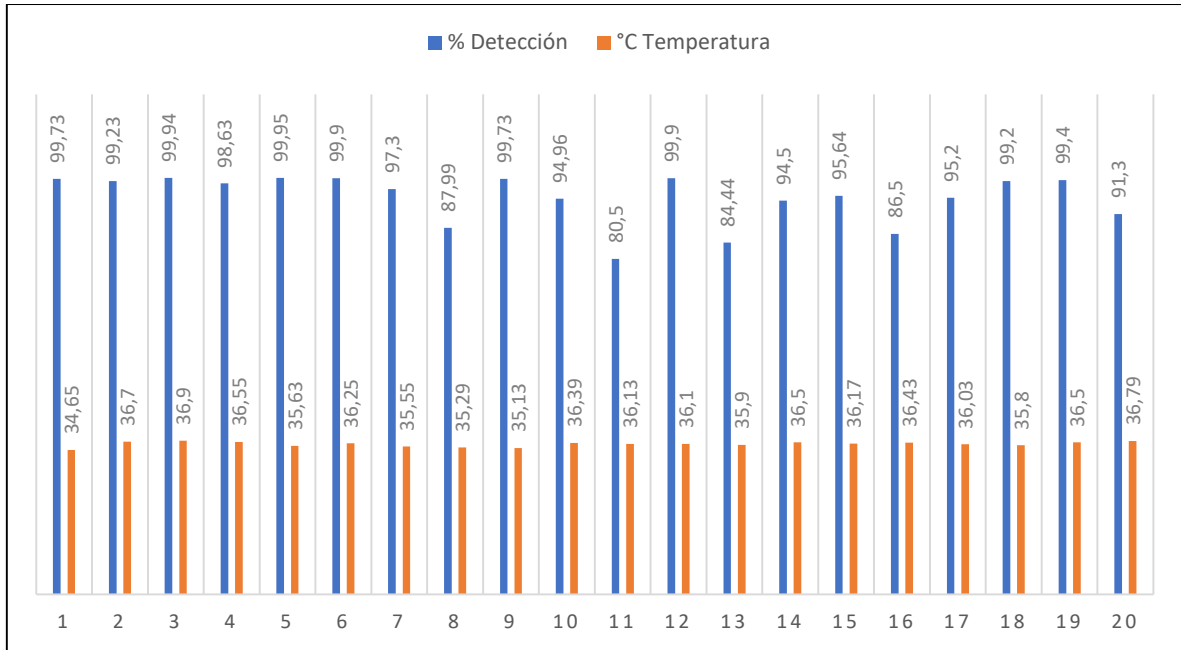


Ilustración 4.9 Grafica de comparación de eficiencia de detección en 20 personas.

En todas las mediciones el porcentaje está en valores altos entre 80.50% y 99.9%, sin embargo, estos valores de temperatura y porcentaje de uso de mascarilla suelen bajar debido a factores externos como la distancia a la que la persona se acerca a la cámara y la altura de la misma.

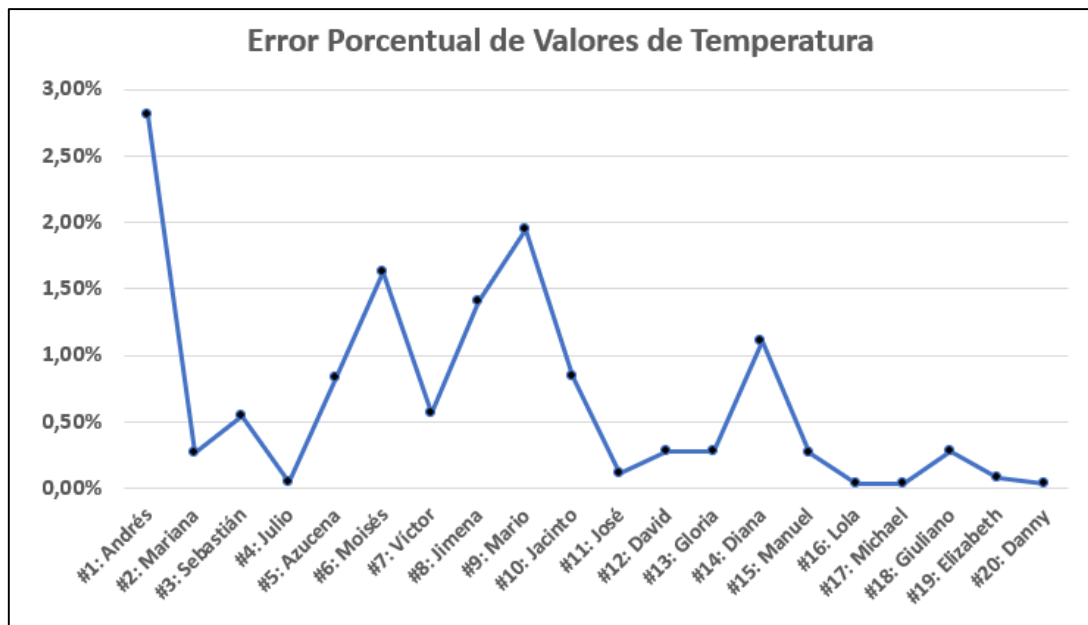


Ilustración 4.10 Gráfica de error porcentual de temperatura.

A continuación, se presenta una gráfica con el error porcentual de los valores del sensor del prototipo con el sensor digital infrarrojo.

Para todas las pruebas, el máximo error porcentual hallado fue de 2.80% y el menor fue de 0.03%.

Si se comparan los valores del termómetro digital infrarrojo con el del prototipo, se pudo observar valores de temperatura muy cercanos con diferencias de 0.02°C con lo que se verifica que la variable de temperatura del prototipo funciona correctamente. Y que 1 usuario tuvo la temperatura de 34.65°C debido a que la persona ubicó su mano lejos del sensor y la retiró rápidamente, pero al compararla con el valor medido de la tabla de temperatura del sensor digital, este se encuentra dentro del error porcentual indicando que el sensor funciona correctamente.

CONCLUSIONES

1. Se logró diseñar e implementar un prototipo de un sistema de control de uso de mascarillas y temperatura corporal de personas en áreas cerradas de una institución prestadora de servicios de salud.
2. Debido a que el algoritmo YOLO (You Only Look Once) se entrenó en la fase de entrenamiento en la nube con un banco de 2000 imágenes de personas con y sin mascarilla, se observó que esta cantidad aumentó la relación directa entre el número de imágenes en el entrenamiento y la efectividad de la detección, sin embargo, esto a su vez hizo que el entrenamiento sea más lento ya que este tomó más de 18 horas.
3. Se correlacionaron variables como la temperatura y uso de mascarillas dentro del sistema de control para permitir el acceso a áreas cerradas cuando se cumplan ambas condiciones sobre el valor de probabilidad de uso de mascarillas mayor a 70% y temperatura medida menor a 37°C.
4. Al analizar los resultados obtenidos del sistema, se comprobó que la precisión del mismo para detectar rostros con mascarillas es alta, ya que se obtuvo probabilidades de entre el 80.50% a 99.99% a distancias entre la persona y el prototipo, menores de 5 metros.
5. Al comparar los valores del termómetro digital infrarrojo con el del prototipo, se pudo observar valores de temperatura muy cercanos con diferencias de 0.02°C y con un valor de error porcentual máximo de 2.80% con lo que se verificó que la variable de temperatura del prototipo funcionó correctamente.
6. Se logró controlar el ingreso a áreas cerradas mediante alertas visuales establecidas como leds indicativos de color verde y rojo, donde el rojo señaló que la persona no puede ingresar dado que posee una temperatura corporal mayor a 37°C o no utiliza mascarilla, y el led verde fue un indicativo de que se cumplieron con las medidas de bioseguridad por lo cual se le permitió el acceso.

RECOMENDACIONES

1. Aumentar el número y variedad de imágenes en el entrenamiento para así obtener mejores resultados de predicción con diferentes tipos de mascarillas.
2. Se recomienda verificar la exactitud y calidad de los dispositivos a utilizar con antelación con el fin de evitar reemplazos, daños y datos aberrantes que retrasen el tiempo planificado para el proyecto.
3. Utilizar cámaras con alta resolución para que así el sistema pueda diferenciar de mejor manera las características que detecta el modelo.
4. En lo posible conseguir un sensor con alta exactitud para que no existan así errores de calibración y precisión en la medición de temperatura.

BIBLIOGRAFÍA

- [1] W. Park, «Prevención del covid-19: por qué hay gente que se opone a medidas que han demostrado salvar vidas,» *BBC Future*, 04 septiembre 2020.
- [2] BBC News Mundo, «Mascarillas contra el coronavirus: cómo el rechazo al tapabocas une a la extrema derecha y la extrema izquierda,» 18 agosto 2020.
- [3] K. Kevin, «Mascarillas contra el coronavirus: ¿a quién protegen de la covid-19: a ti o a los demás?,» *The Conversation*, 16 julio 2020.
- [4] M. d. S. P. MSP, «INSPI,» 23 agosto 2021. [En línea]. Available: <https://www.salud.gob.ec/actualizacion-de-casos-de-coronavirus-en-ecuador/>. [Último acceso: 25 octubre 2021].
- [5] T. Wong, «Prevención del coronavirus: por qué en algunos países la gente usa mascarillas en público y en otros no,» *BBC News, Singapur*, 18 mayo 2020.
- [6] El Comercio, «Guardia fue asesinado por exigir el uso de la mascarilla en Michigan, EE.UU.,» *El Comercio*, 04 mayo 2020.
- [7] CASADOMO, «Sistema de detección de temperatura y mascarilla en un hospital de Taipei,» 27 abril 2020.
- [8] C. O. B. Manzano, «Luz verde: implementación de un sistema de detección del uso de mascarillas por parte de la ciudad ecuatoriana,» [En línea]. Available: <https://www.fiec.espol.edu.ec/es/galeriaposter/luz-verde-implementaci%C3%B3n-de-un-sistema-de-detecci%C3%B3n-del-uso-de-mascarillas-por-parte>. [Último acceso: 10 octubre 2021].
- [9] C. Jeong-hoon, «FLIR desarrolla un prototipo de detección de mascarilla facial basado en aprendizaje profundo,» *FAJournal*, 04 diciembre 2020.

- [10] Dong-A Revista de educación científica de Corea, «LG CNS, detección automática de temperatura corporal y si se lleva mascarilla en los documentos de acceso,» 26 marzo 2020.
- [11] LG, «LG U+ LG U+ lanzará un 'robot de cuarentena 5G' que detecta la temperatura corporal mientras usa una máscara,» 03 septiembre 2020. [En línea]. Available: <https://www.lg.co.kr/media/release/22369>. [Último acceso: 10 octubre 2021].
- [12] K. Kyung-han, «CyberLink Face Me, 'detección de máscara' y 'medición de temperatura corporal' al mismo tiempo,» *TECHWORLD*, 08 octubre 2020.
- [13] C. Myung-hoon, «Reconocimiento facial, control de temperatura, uso de mascarilla... Monitoreo de acceso al hospital IA a la vez,» *Prensa DongA Media Group*, 03 agosto 2020.
- [14] NVIDIA, «Implementación de una aplicación de detección de máscaras faciales basada en IA en tiempo real para COVID-19 por NVIDIA,» 12 abril 2021. [En línea]. Available: <http://cresprit.com/ko/2021/04/12/20210412/>. [Último acceso: 10 octubre 2021].
- [15] FacePhi, «FacePhi puede verificar la identidad incluso con mascarillas,» 13 julio 2020. [En línea]. Available: <https://facephi.com/noticias/facephi-firma-un-acuerdo-con-santander-merchant-platform-solutions>. [Último acceso: 10 octubre 2021].
- [16] W. Z. J. Yu, «Algoritmo de detección de uso de mascarillas basado en YOLO-v4 mejorado,» *MDPI*, 08 mayo 2021. [En línea]. Available: <https://www.mdpi.com/1424-8220/21/9/3263/htm>. [Último acceso: 10 octubre 2021].
- [17] H. Y. S. D. P. B Varshini, «Puertas inteligentes habilitadas para IoT para monitorear la temperatura corporal y la detección de máscaras faciales,» *Global Transitions Proceedings*, vol. 2, nº 2, pp. 246-254, 07 agosto 2021.
- [18] PixellImpact, «Borne / totem avec solutions de dépistage de fièvre Terminal de reconnaissance faciale pour dépistage de fièvre et de port du masque,» 20

abril 2020. [En línea]. Available: <https://www.pixel-impact.fr/bornes-totem-depistage-fievre/>. [Último acceso: 03 noviembre 2021].

- [19] S. Hussain, «IoT and Deep Learning Based Approach for Rapid Screening,» 05 marzo 2021. [En línea]. Available: <https://www.mdpi.com/2076-3417/11/8/3495/pdf>. [Último acceso: 10 octubre 2021].
- [20] R. W. K. J. M. Warungase, «IJARIE Revista Internacional de Investigación Avanzada e Ideas Innovadoras en Educación,» 15 02 2021. [En línea]. Available: https://ijarie.com/AdminUploadPdf/Face_Mask_and_Body_Temperature_Detection_System_to_Prevent_COVID_for_Work_Environment_ijarie14217.pdf. [Último acceso: 10 octubre 2021].
- [21] NaverCorporation, «Vsbly,integración de inspección de temperatura de reconocimiento facial y prueba de ciudad inteligente,» 27 abril 2020.
- [22] Corporation, Korea Electric Power, «Imprescindible para la cuarentena en la era del COVID-19, una solución de reconocimiento facial que detecta fiebre,» 24 agosto 2020.
- [23] A. R. K. P. I. Farady, «Mask Classification and Head Temperature,» [En línea]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9249454>. [Último acceso: 10 octubre 2021].
- [24] S. Sethi, «Detección de mascarillas mediante aprendizaje profundo: un enfoque para reducir el riesgo de propagación del coronavirus,» *sciencedirect*, 08 2021.
- [25] Glasstech, «Puertas de Hospital no Herméticas | Glasstech,» *Glasstech*, 01 octubre 2020.
- [26] World Health Organization, «Coronavirus disease (COVID-19) advice for the public: When and how to use masks,» 01 diciembre 2020. [En línea]. Available: <https://www.who.int/en/emergencies/diseases/novel-coronavirus->

2019/advice-for-public/when-and-how-to-use-masks. [Último acceso: 26 octubre 2021].

- [27] Gestion de Riesgos del Centro de Operaciones de Emergencia COE, «Resoluciones COE Nacional 03 de enero de 2021,» 03 enero 2021. [En línea]. Available: <https://www.gestionderiesgos.gob.ec/resoluciones-coe-nacional-03-de-enero-de-2021/>. [Último acceso: 27 octubre 2021].
- [28] ONU Naciones Unidas, «Objetivos de Desarrollo Sostenible,» 15 noviembre 2020. [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. [Último acceso: 03 noviembre 2021].
- [29] C. Hernández, «Los cinco beneficios de llevar mascarilla más allá de protegerte del Covid,» 29 noviembre 2020.
- [30] T. d'Arc, «Que es la inteligencia artificial: 8 ejemplos en tu vida diaria,» 21 agosto 2020. [En línea]. Available: https://www.smarthint.co/es/o-que-e-inteligencia-artificial-exemplos/?utm_source=blog&utm_medium=post&utm_campaign=buscainteligen. [Último acceso: 30 noviembre 2021].
- [31] ManagementSolutions, «Auto Machine Learning, hacia la automatización de los modelos,» [En línea]. Available: <https://www.managementsolutions.com/sites/default/files/publicaciones/esp/auto-machine-learning-sp.pdf>. [Último acceso: 05 febrero 2022].
- [32] J. D. Meza, «Curso De Python Aprende PYTHON de una buena vez,» 2021. [En línea]. Available: <https://www.programarya.com/Cursos/Python/entrada-y-lectura-de-datos>. [Último acceso: 05 febrero 2022].
- [33] Banco Bilbao Vizcaya Argentaria, «'Machine learning': ¿qué es y cómo funciona?,» 2019. [En línea]. Available: <https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/>. [Último acceso: 05 febrero 2022].
- [34] D. S. R. Raneros, «UVaDOC Repositorio Documental de la Universidad de Valladolid,» enero 2021. [En línea]. Available:

<https://uvadoc.uva.es/bitstream/handle/10324/45359/TFM-G1316.pdf;jsessionid=8F1E32E3F475FBAFB859E74E7E757D03?sequence=1>. [Último acceso: 05 febrero 2022].

- [35] AprendeMachineLearning, «7 pasos del Machine Learning para construir tu máquina,» 11 septiembre 2018. [En línea]. Available: <https://www.aprendemachinelearning.com/7-pasos-machine-learning-construir-maquina/>. [Último acceso: 05 febrero 2022].
- [36] Harshit, «Comparison of Python with Other Programming Languages,» 10 mayo 2020. [En línea]. Available: <https://www.geeksforgeeks.org/comparison-of-python-with-other-programming-languages/>. [Último acceso: 05 febrero 2022].
- [37] J. M. Heras, «15 Librerías de Python para Machine Learning,» 20 octubre 2020. [En línea]. Available: <https://www.iartificial.net/librerias-de-python-para-machine-learning/>. [Último acceso: 30 noviembre 2021].
- [38] N. I. Mokhtari, «Which is Better For Your Machine Learning Task, OpenCV or TensorFlow?,» 01 agosto 2021. [En línea]. Available: <https://towardsdatascience.com/which-is-better-for-your-machine-learning-task-opencv-or-tensorflow-ed16403c5799>. [Último acceso: 05 febrero 2022].
- [39] P. Mahto, «NumPy For Machine Learning,» 06 septiembre 2020. [En línea]. Available: <https://medium.com/mlpoint/numpy-for-machine-learning-211a3e58b574#:~:text=NumPy%20stands%20for%20'Numerical%20Python,on%20these%20arrays%20and%20matrices..> [Último acceso: 05 febrero 2022].
- [40] P. F. T. A. R. N. K. A.-S. Ahmed S. Shamsaldin, «The Study of The Convolutional Neural Networks Applications,» diciembre 2019. [En línea]. Available: https://www.researchgate.net/publication/338206036_The_Study_of_The_Convolutional_Neural_Networks_Applications. [Último acceso: 05 febrero 2022].

- [41] L. Institute, «Reconocimiento facial: Descubre cómo funciona y quién (y para qué) lo utiliza,» 15 octubre 2019. [En línea]. Available: <https://www.lisainstitute.com/blogs/blog/reconocimiento-facial-como-funciona-quien-utiliza>. [Último acceso: 30 noviembre 2021].
- [42] TIBC, «¿Qué es el aprendizaje supervisado?,» [En línea]. Available: <https://www.tibco.com/es/reference-center/what-is-supervised-learning>. [Último acceso: 05 febrero 2022].
- [43] S. Garg, «RECONOCIMIENTO FACIAL MEDIANTE INTELIGENCIA ARTIFICIAL,» Acervo Lima, [En línea]. Available: <https://es.acervolima.com/reconocimiento-facial-mediante-inteligencia-artificial/>. [Último acceso: 05 febrero 2022].
- [44] R. R. B. J. C. de Jesus Montero Rodriguez, «Algoritmos de aprendizaje automático de vanguardia para el diagnóstico de enfermedades,» Instituto Tecnológico Superior de Misantla, [En línea]. Available: https://rcs.cic.ipn.mx/2019_148_7/Algoritmos%20de%20aprendizaje%20automatico%20de%20vanguardia%20para%20el%20diagnostico%20de%20enfermedades.pdf. [Último acceso: 12 abril 2019].
- [45] D. H. RAMÍREZ, «Repositorio Institucional Unilibre, RIU,» 2018. [En línea]. Available: <https://repository.unilibre.edu.co/bitstream/handle/10901/17289/EL%20MAC%20HINE%20LEARNING.pdf?sequence=1&isAllowed=y>. [Último acceso: 05 febrero 2022].
- [46] J. K. & G. H. Kumar Abhishek, «Predicting the clinical management of skin lesions using deep learning,» 08 abril 2021.
- [47] D. A. C. Briceño, «EVALUACIÓN DE MODELOS DE VISIÓN POR COMPUTADOR EN VIDEO PARA LA DETECCIÓN DE LA POSE HUMANA Y CAÍDAS,» 2020. [En línea]. Available: <https://repository.urosario.edu.co/bitstream/handle/10336/24437/EVALUACION%20DE%20MODELOS%20DE%20VISI%20POR%20COMPUTADOR%20EN%20VIDEO%20PARA%20LA%20DETECCION%20DE%20>

LA%20POSE%20HUMANA%20Y%20CA%20CDDAS.pdf?sequence=1. [Último acceso: 05 febrero 2022].

- [48] L. M. C. M. P. J. R. N. F. A. Aira Céspedes, «Revistas de investigación Universidad Nacional Mayor de San Marcos,» 28 diciembre 2021. [En línea]. Available: <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/download/21862/17543/75813>. [Último acceso: 05 febrero 2022].
- [49] M. R. T. N. P. J. E. Garcia-Ceja, «ScienceDirect,» 2018. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S1574119217305692>. [Último acceso: 05 febrero 2022].
- [50] Y. Y. Lee, «Equipo de protección personal (EPP) para trabajadores de la salud,» 14 octubre 2020. [En línea]. Available: https://www.cdc.gov/spanish/niosh/docs/2013-138_sp/default.html. [Último acceso: 30 noviembre 2021].
- [51] St. Jude Children's Research Hospital, «COVID-19 y fiebre,» marzo 2021. [En línea]. Available: <https://together.stjude.org/es-us/atenci%C3%B3n-apoyo/covid-19-resources/fever-and-covid-19.html>. [Último acceso: 05 febrero 2022].
- [52] C. A. G. López, «Historia de la computación,» 15 octubre 2020. [En línea]. Available: http://biblioteca.usac.edu.gt/tesis/07/07_2010.pdf. [Último acceso: 30 noviembre 2021].
- [53] C. W. Johnson, «Microcomputers In Educational Research,» enero 1982. [En línea]. Available: https://www.researchgate.net/publication/234743987_Microcomputers_In_Educational_Research. [Último acceso: 05 febrero 2022].
- [54] J. F. Muñoz, «Manual de Programación MicroBit,» enero 2019. [En línea]. Available: <http://microes.org/descargas/manual-de-programacion-microbit.pdf>. [Último acceso: 05 febrero 2022].

- [55] N. Heath, «What is the Raspberry Pi 4? Everything you need to know about the tiny, low-cost computer,» 02 julio 2019.
- [56] DescubreArduino, «ARDUINO, ¿QUÉ ES Y PARA QUE SIRVE? APRENDE CON TUTORIALES Y PROYECTOS,» [En línea]. Available: <https://descubrearduino.com/>. [Último acceso: 05 febrero 2022].
- [57] A. Allan, «Benchmarking Machine Learning on the New Raspberry Pi 4, Model B,» *Hackster*, 2019.
- [58] E. Upton, «Raspberry Pi 4 on sale now from \$35,» *Raspberry Pi* , 24 junio 2019.
- [59] C. CAWLEY, «How to Install an Operating System on a Raspberry Pi,» 08 julio 2021. [En línea]. Available: <https://www.makeuseof.com/tag/install-operating-system-raspberry-pi/#:~:text=Your%20Raspberry%20Pi%20doesn't,your%20Raspberry%20Pi's%20SD%20card..> [Último acceso: 05 febrero 2022].
- [60] Raspberry Pi, «Raspberry Pi Documentation,» [En línea]. Available: <https://www.raspberrypi.com/documentation/computers/os.html>. [Último acceso: 05 febrero 2022].
- [61] D. Petersson, «techtarget,» 03 2021. [En línea]. Available: <https://www.techtarget.com/searchenterpriseai/definition/supervised-learning>. [Último acceso: 05 02 2022].
- [62] IBM Cloud Education, «What Is Semi-Supervised Learning,» 21 09 2021. [En línea]. Available: <https://www.ibm.com/cloud/learn/unsupervised-learning>. [Último acceso: 05 febrero 2022].
- [63] D. Johnson, «Unsupervised Machine Learning: Algorithms, Types with Example,» 2022 01 2021. [En línea]. Available: <https://www.guru99.com/unsupervised-machine-learning.html>. [Último acceso: 05 02 2022].

- [64] Algorithmia, «What Is Semi-Supervised Learning,» 11 08 2021. [En línea]. Available: <https://algorithmia.com/blog/semi-supervised-learning>. [Último acceso: 28 11 2021].
- [65] J. Brownlee, «A Gentle Introduction to Object Recognition With Deep Learning,» 22 05 2019. [En línea]. Available: <https://machinelearningmastery.com/object-recognition-with-deep-learning/>. [Último acceso: 30 noviembre 2021].
- [66] surecontrols, «Infrared Temperature Sensors,» surecontrols, 02 01 2013. [En línea]. Available: <https://www.surecontrols.com/infrared-temperature-sensors/>. [Último acceso: 29 11 2021].
- [67] J. R. GYORKI, «Understanding the Infrared Temperature Sensor,» 14 07 2009. [En línea]. Available: <https://www.sensortips.com/temperature/infrared-temperature-sensor/>. [Último acceso: 29 11 2021].
- [68] Laia Espuny Planelles, «Deep Learning: consideraciones al elegir un sistema de vídeo analítica inteligente,» [En línea]. Available: <https://www.lanaccess.com/deep-learning-consideraciones-al-elegir-un-sistema-de-video-analitica-inteligente/>. [Último acceso: 05 febrero 2022].
- [69] UNICEF, «Todo lo que sabemos sobre la variante ómicron,» 2022.
- [70] ONU, «Noticias ONU,» 28 enero 2021. [En línea]. Available: <https://news.un.org/es/story/2021/01/1487282>. [Último acceso: 05 febrero 2022].
- [71] M. R. K. Wells, «ERs Are Swamped With Seriously Ill Patients, Although Many Don't Have Covid,» *KAISER FAMILY FOUNDATION NEWS*, 29 octubre 2021.
- [72] Na8, «Detección de Objetos con Python,» 24 06 2020. [En línea]. Available: <https://www.aprendemachinelearning.com/deteccion-de-objetos-con-python-yolo-keras-tutorial/>. [Último acceso: 28 11 2021].

- [73] Tartila, «consultorio médico, rooms., hospital, edificio, interior, emergencia, clínica, médico, sala de espera, y, cirugía, médicos, caricatura, vector, ilustración».
- [74] B. Srinivasan, «Face Mask Detection using Python, Keras, OpenCV and MobileNet | Detect masks real-time video streams».
- [75] FindGlocal, «Kokenos Cyber,» [En línea]. Available: <https://www.findglocal.com/MX/Tecate/637458932932000/Kokenos-Cyber>. [Último acceso: 05 febrero 2022].
- [76] A. Eyab, «YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet),» 14 octubre 2021. [En línea]. Available: <https://github.com/AlexeyAB/darknet>. [Último acceso: 30 enero 2022].
- [77] J. Hui, «YOLOv4-Neck,» 2020.
- [78] J. Redmon, «Darknet: Open Source Neural Networks in C,» pjreddie, 2016. [En línea]. Available: <https://pjreddie.com/darknet/>. [Último acceso: 11 12 2021].
- [79] C. Deb, «Face-Mask-Detection,» github, 07 06 2021. [En línea]. Available: <https://github.com/chandrikadeb7/Face-Mask-Detection/tree/master/dataset>. [Último acceso: 11 12 2021].
- [80] LinuxAdictos, «Welcome to Raspbian,» 15 julio 2021. [En línea]. Available: <https://www.raspbian.org/FrontPage>. [Último acceso: 05 febrero 2022].
- [81] Raspberry Pi Foundation, «Getting started with Raspberry Pi,» 14 octubre 2021. [En línea]. Available: <https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started>. [Último acceso: 13 diciembre 2021].
- [82] Omega-SpecirisCompany, «Introducción a los termómetros infrarrojos,» 15 04 2021. [En línea]. Available: <https://mx.omega.com/prodinfo/termometro-infrarrojo.html>. [Último acceso: 10 enero 2022].

- [83] Centro para el Control y la Prevención de Enfermedades, «Síntomas,» 13 09 2020. [En línea]. Available: <https://espanol.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html>. [Último acceso: 10 enero 2022].
- [84] Mayo Clinic, «Enfermedad del coronavirus 2019 (COVID-19),» 01 febrero 2022. [En línea]. Available: <https://www.mayoclinic.org/es-es/diseases-conditions/coronavirus/symptoms-causes/syc-20479963>. [Último acceso: 05 febrero 2022].