

**UNIVERSIDAD RICARDO PALMA**  
**FACULTAD DE INGENIERÍA**  
**PROGRAMA DE TITULACIÓN POR TESIS**  
**ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**



**DETECCIÓN DEL USO CORRECTO DE MASCARILLAS  
UTILIZANDO UNA RED NEURONAL CONVOLUCIONAL PARA  
EL INGRESO DE PERSONAS A UN LABORATORIO DE UNA  
UNIVERSIDAD**

**TESIS**  
**PARA OPTAR EL TÍTULO PROFESIONAL DE  
INGENIERO ELECTRÓNICO**

**PRESENTADA POR**

**Bach. XAVIER ALEXANDER CHIRINOS CARRANZA**

**Bach. PAÚL MANUEL CALERO SEGURA**

**ASESOR: DR. ING. HUAMANÍ NAVARRETE, PEDRO FREDDY**

**LIMA-PERÚ**

**2021**

## **DEDICATORIA**

A mis padres Hilda y Javier por darme su apoyo en toda mi etapa profesional, a mis hermanos Bryam y Sebastian por darme su apoyo emocional, y a toda mi familia por ayudarme a salir adelante.

Xavier Alexander Chirinos Carranza

A mis padres Isabel y Héctor por su apoyo incondicional en todas las etapas de mi formación. A mi pareja Brenda Barturen por la motivación y confianza para poder culminar este proyecto.

Paúl Manuel Calero Segura.

## **AGRADECIMIENTO**

Agradecemos a la Universidad Ricardo Palma por ayudarnos en nuestra formación profesional, por brindarnos los mejores profesores. A nuestro asesor Pedro Huamaní por el apoyo para terminar este proyecto de tesis, a nuestros compañeros de clase, con quienes compartimos buenos momentos.

Xavier Chirinos y Paúl Calero

# ÍNDICE GENERAL

RESUMEN .....	ix
ABSTRACT.....	x
INTRODUCCIÓN.....	1
<b>CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA .....</b>	<b>3</b>
1.1. Formulación del problema .....	3
1.1.1.Problema General .....	3
1.1.2.Problemas Específicos .....	3
1.2. Objetivos.....	4
1.2.1. Objetivo General.....	4
1.2.2. Objetivos Específicos .....	4
1.3. Importancia y justificación .....	4
1.3.1. Importancia .....	4
1.3.2. Justificación .....	4
1.4. Limitaciones.....	5
1.4.1. Limitaciones .....	5
<b>CAPÍTULO II: MARCO TEÓRICO .....</b>	<b>6</b>
2.1.Marco Histórico .....	6
2.2.Antecedentes de la investigación.....	6
2.2.1.Antecedentes nacionales .....	6
2.2.2.Antecedentes internacionales.....	9
2.3.Bases Teóricas relacionadas con el tema.....	10
2.3.1.MATLAB – Deep learning (Toolbox del MATLAB).....	10
2.3.2.Capas de la Red Neuronal Convolucional .....	11
2.3.3 Métricas de Validación .....	13
2.4.Definición de términos básicos.....	13
2.5.Diseño de la Investigación.....	14
2.5.1.Variables de investigación.....	14
2.5.2.Tipo y Método de investigación .....	14
2.5.3.Técnicas e Instrumentos de recolección de datos .....	15
2.5.4.Procedimiento para la recolección de datos.....	15

<b>CAPÍTULO III: DISEÑO DE LA INVESTIGACIÓN.....</b>	<b>16</b>
3.1.Diagrama de bloques .....	16
3.2.Procedimiento de adquisición de imágenes .....	17
3.3.Procesamiento de las imágenes para la base de datos .....	20
3.4.Entrenamiento y Ejecución de las redes neuronales convolucionales en Matlab.....	23
3.4.1.Modelo de red neuronal convolucional N°1 .....	23
3.4.2.Modelo de red neuronal convolucional N°2.....	26
3.4.3.Modelo de red neuronal convolucional N°3.....	29
3.5 Implementación de la interfaz con App Designer .....	31
<b>CAPÍTULO IV: PRUEBAS Y RESULTADOS.....</b>	<b>36</b>
4.1.Validación de la Red Neuronal Convolucional Particular 1 .....	36
4.1.1 Evaluación de la Matriz de Confusión.....	36
4.1.2 Evaluación de Accuracy y Precision .....	37
4.2.Validación de la Red Neuronal Convolucional Particular 2.....	38
4.2.1 Evaluación de la Matriz de Confusión.....	38
4.2.2 Evaluación de Accuracy y Precision .....	40
4.3.Validación de la Red Neuronal Convolucional Particular 3.....	40
4.3.1 Evaluación de la Matriz de Confusión.....	40
4.3.2 Evaluación de Accuracy y Presicion .....	42
4.4.Comparación de los Modelos de Redes Neuronales Convolucionales.....	42
4.5.Prueba Final .....	45
4.6 Costo del Proyecto .....	54
<b>CONCLUSIONES .....</b>	<b>56</b>
<b>RECOMENDACIONES .....</b>	<b>58</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>59</b>
<b>ANEXOS.....</b>	<b>62</b>

## ÍNDICE DE FIGURAS

Figura N°1: Capa Pooling. ....	11
Figura N°2: Gráfico de la capa de salida y la función SoftMax. ....	12
Figura N°3: Matriz de confusión.....	13
Figura N°4: Diagrama de Bloques. ....	1711
Figura N°5: Dispositivo Móvil.....	18
Figura N°6: Creación de las carpetas. ....	18
Figura N°7: Características del video. ....	19
Figura N°8: Algoritmo para convertir video en frames. ....	19
Figura N°9: Resultado en frames. ....	19
Figura N°10: Imagen.....	20
Figura N°11: Algoritmo para el procesamiento de imágenes. ....	21
Figura N°12: Procesamiento de imágenes para la carpeta “Con Mascarilla”.....	21
Figura N°13: Procesamiento de imágenes para la carpeta “Mal Uso”.....	22
Figura N°14: Procesamiento de imágenes para la carpeta “Sin Mascarilla”. ....	22
Figura N°15: Nueva Base de datos. ....	23
Figura N°16: Entrenamiento de red neuronal convolucional particular 1. ....	23
Figura N°17: Programación de capas convolucionales.....	24
Figura N°18: Opciones de entrenamiento. ....	25
Figura N°19: Diagrama de capas convolucionales de CNN particular 1.....	25
Figura N°20: Inicio de entrenamiento de CNN particular 1. ....	25
Figura N°21: Resultados del entrenamiento y validación de CNN particular 1. ....	26
Figura N°22: Entrenamiento de red neuronal convolucional particular 2. ....	27
Figura N°23: Programación de capas convolucionales.....	27
Figura N°24: Opciones de entrenamiento .....	28
Figura N°25: Diagrama de capas convolucionales CNN particular 2.....	28
Figura N°26: Inicio de entrenamiento de la CNN particular 2. ....	29
Figura N°27: Resultado del entrenamiento y validación de la CNN particular 2. ....	29
Figura N°28: Diagrama de capas convolucionales de la CNN particular 3.....	30
Figura N°29: Inicio de entrenamiento de la CNN particular 3. ....	30
Figura N°30: Resultado del entrenamiento y validación de la CNN particular 3.....	31
Figura N°31: Explorador Add-On de MATLAB.....	32
Figura N°32: Classify Webcam Images.....	32

Figura N°33: Guardado de la red neuronal convolucional en la interfaz App Designer.	33
Figura N°34: Programación completa en el App Designer.....	34
Figura N°35: Interfaz mediante App Designer. ....	35
Figura N°36: Matriz de confusión para la red neuronal convolucional particular 1.....	37
Figura N°37: Matriz de confusión para la red neuronal convolucional particular 2.....	39
Figura N°38: Matriz de confusión para la red neuronal convolucional particular 3.....	41
Figura N°39: Instalación del trípode con la cámara.....	45
Figura N°40: Conexión de la cámara con el App Designer. ....	46
Figura N°41: App Designer sin detectar nada.....	46
Figura N°42: Prueba 1.....	47
Figura N°43: Resultado 1.....	47
Figura N°44: Prueba 2.....	48
Figura N°45: Resultado 2.....	48
Figura N°46: Prueba 3.....	49
Figura N°47: Resultado 3.....	49
Figura N°48: Prueba 4.....	50
Figura N°49: Resultado 4.....	50
Figura N°50: Prueba 5.....	51
Figura N°51: Resultado 5.....	51
Figura N°52: Prueba 6.....	52
Figura N°53: Resultado 6.....	52
Figura N°54: Pruebas 7.....	53
Figura N°55: Pruebas 8.....	53
Figura N°56: Pruebas 9.....	54
Figura N°57: Pruebas 10.....	54
Figura N°58: Trípode.....	62
Figura N°59: Cámara Web.....	62

## ÍNDICE DE TABLAS

Tabla N°1: Valores según la matriz de confusión del modelo de red particular 1 .....	37
Tabla N°2: Valores según la matriz de confusión del modelo de red particular 2 .....	39
Tabla N°3: Valores según la matriz de confusión del modelo de red particular 3 .....	41
Tabla N°4: Comparación de los valores obtenidos en la matriz de confusión entre los modelos de redes neuronales convolucionales particulares.....	43
Tabla N°5: Comparación de los valores obtenidos en la métrica Precision entre los modelos de redes neuronales convolucionales particulares.....	43
Tabla N°6: Comparación de los valores obtenidos en la métrica Accuracy entre los modelos de redes neuronales convolucionales particulares.....	44
Tabla N°7: Costos de los materiales del proyecto de tesis .....	55

## RESUMEN

En la presente investigación se optó por trabajar con tres modelos de redes neuronales convolucionales particulares para la detección de mascarillas, y se evaluó la capacidad de reconocimiento de una persona usando mascarilla, sin mascarilla y haciendo mal uso de esta, justificando con valores numéricos y ecuaciones la que posee mayor efectividad al momento de la detección, utilizando para este cometido tres parámetros significativos como lo son: la matriz de confusión, la métrica Precision y la métrica Accuracy. En primer lugar, se realizó la adquisición de imágenes y seguidamente su procesamiento para conformar la base de datos, la cual se organizó en tres grupos que son: Con Mascarilla, conformada por 968 imágenes, Sin Mascarilla, conformada por 988 imágenes, y Mal Uso conformada por 967 imágenes, dándonos un total de 2923 imágenes. Teniendo la base de datos se procedió al entrenamiento de los tres modelos de redes neuronales convolucionales, la primera red utilizó 3 capas convolucionales, la segunda red 5 capas convolucionales y la tercera red 7 capas convolucionales, a través del Toolbox Deep Learning de Matlab, seguidamente se realizó una comparación entre las redes neuronales convolucionales para saber que red será la más indicada y llevarla a la simulación respectiva.

En cuanto a los resultados obtenidos, se evidenciaron que la red neuronal convolucional que posee más capas tiene un mejor desempeño en el reconocimiento de la mascarilla, sin mascarilla y haciendo mal uso de esta, logrando clasificar 668 imágenes de 673 para la validación, con una precision y accuracy del 99%, demostrado a través de la matriz de confusión y cálculo respectivo. Con este resultado se eligió la tercera red neuronal convolucional particular para la detección de mascarilla. Seguidamente se procedió con la implementación de la interfaz del proyecto a través del App Designer del Matlab para su visualización en tiempo real. Finalmente, se recreó un ambiente parecido a la entrada de un laboratorio de una universidad, para su simulación, logrando el correcto funcionamiento en el reconocimiento del uso correcto de la mascarilla.

**Palabras claves:** Red neuronal convolucional, Toolbox Deep Learning, Matlab, App Designer.

## ABSTRACT

In this research we chose to work with three particular convolutional neural network models for the detection of masks, and we evaluated the ability to recognize a person wearing a mask, without a mask and using it incorrectly, justifying with numerical values and equations the most effective at the time of detection, using for this purpose three significant parameters such as: the confusion matrix, the Precision metric and the Accuracy metric. First, the images were acquired and then processed to form the database, which was organized into three groups: With Mask, consisting of 968 images, Without Mask, consisting of 988 images, and Misuse, consisting of 967 images, giving us a total of 2923 images. Having the database, we proceeded to the training of the three convolutional neural network models, the first network was used 3 convolutional layers, the second network 5 convolutional layers and the third network 7 convolutional layers, through Matlab's Deep Learning Toolbox, then a comparison was made between the convolutional neural networks to know which network will be the most suitable and take it to the respective simulation.

As for the results obtained, it was evidenced that the convolutional neural network that has more layers has a better performance in mask recognition, without mask and making bad use of it, managing to classify 668 images out of 673 for validation, with a precision and accuracy of 99%, demonstrated through the confusion matrix and respective calculation. With this result, the third particular convolutional neural network was chosen for mask detection. Then we proceeded with the implementation of the project interface through the Matlab App Designer for its real-time visualization. Finally, an environment similar to the entrance of a university laboratory was recreated for its simulation, achieving the correct functioning in the recognition of the correct use of the mask.

**Keywords:** Convolutional Neural Network, Toolbox Deep Learning, Matlab, App Designer.

## INTRODUCCIÓN

Las redes neuronales artificiales vienen investigándose desde mediados del siglo XX, actualmente se ha visto una alta tasa de crecimiento, debido a su gran desempeño en ingeniería, pues se aprovechan los conceptos en la parte académica como objeto de investigación. Una de las principales características de las redes neuronales es que son sistemas fáciles de usar, simplemente conociendo su programación, algoritmos o librerías, podemos usar esta herramienta para nuestros propósitos. Ahora, debido a la pandemia que se vive a nivel mundial, causada por el SARS-COV 2, se han hecho más investigaciones con respecto a las redes neuronales, buscando alternativas o soluciones para ayudar a combatir este virus que tanto daño le hizo al mundo.

En este proyecto de tesis, propone usar redes neuronales convolucionales para la detección del uso correcto de mascarillas para el ingreso a personas, recreando una simulación de un laboratorio de una universidad; la causa principal por el cual se hace esta investigación es que las personas usan incorrectamente la mascarilla, o en algunos casos se olvidan de llevarla puesta. Según la OMS, la mascarilla ayuda a evitar que el virus se propague, y hay que ser más cuidadosos en ambientes cerrados como lo es un laboratorio de una universidad.

El interés de realizar este proyecto de tesis es ayudar a las personas a tomar conciencia en usar correctamente la mascarilla y llevar un mejor control antes de ingresar a un ambiente cerrado como es un laboratorio de una universidad con peligro de contagiarse, gracias a las redes neuronales convolucionales y al software Matlab con versión R2021a que incluye el App Designer en el Deep Learning Toolbox, será posible desarrollar este proyecto.

Para el entrenamiento y validación se usarán tres modelos de redes neuronales convolucionales particulares, usando una base de datos compuesta de imágenes de personas con mascarilla, sin mascarilla y haciendo mal uso de estas, con la finalidad de obtener la red neuronal convolucional más efectiva para la detección de la mascarilla.

El presente proyecto de tesis se divide en cuatro capítulos importantes, en el capítulo I se realizó el planteamiento y delimitación del problema, donde se observa la problemática, los objetivos, la importancia y justificación del presente proyecto de tesis y las limitaciones. En el capítulo II se describe el marco teórico, los antecedentes nacionales e internacionales y la base teórica relacionada al presente proyecto de tesis.

En el capítulo III se describe todo el diseño de ingeniería para este proyecto de tesis, se especifica como se realizó el procedimiento para la adquisición de imágenes y su procesamiento para la base de datos, luego como se llevó a cabo el entrenamiento y ejecución de los tres modelos de redes neuronales particulares.

Finalmente, en el capítulo IV se presentan las pruebas y resultados obtenidos de la matriz de confusión y sus respectivas métricas para cada modelo de red neuronal convolucional, para así escoger la mejor red neuronal convolucional y hacer su simulación respectiva.

Se concluye que, teniendo más cantidad de capas convolucionales en el entrenamiento de la red neuronal convolucional, se obtiene mejores resultados en la validación y detección de mascarillas, se recomienda tener en la base de datos imágenes con un solo fondo para así facilitar el entrenamiento y se sugiere tener una mayor cantidad de imágenes, a fin de que la red neuronal convolucional tenga mejores resultados.

# CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA

## 1.1. Formulación del problema

En el último año las Universidades fueron afectadas por la pandemia ocasionada por el SARS-CoV-2. Este virus se transmite a través de partículas que expulsa una persona enferma al hablar, toser o estornudar. Las partículas pueden quedarse en el aire por un tiempo prolongado, y más aún en lugares con poca ventilación, también al ser inhaladas por las personas que están cerca al enfermo, lo cual nos llevó a un cambio de pensamiento sobre el uso de protección para evitar el contagio de este virus, por esa razón las mascarillas ahora forman parte de nuestra rutina diaria.

En el Perú la mayoría de las personas no toma conciencia sobre este medio de protección, tanto que no lo utilizan o se lo ponen incorrectamente, lo cual es perjudicial y más aún dentro de un espacio cerrado como es el laboratorio de una Universidad. Por ende, sería necesario contar con un sistema de reconocimiento que detecte el uso correcto de la mascarilla, para tener un mejor control en el ingreso de personas a un laboratorio en el horario de clases dentro de una universidad.

### 1.1.1. Problema General

¿Cómo detectar el uso correcto de mascarillas utilizando una red neuronal convolucional en una simulación de ingreso de personas a un laboratorio de una Universidad?

### 1.1.2. Problemas Específicos

- a) ¿Cómo establecer una base de datos que contenga imágenes de personas con mascarillas, sin mascarillas y haciendo mal uso, para el ingreso a un laboratorio de una Universidad?
- b) ¿Cómo implementar y entrenar una red neuronal convolucional, para la identificación del uso correcto de la mascarilla antes de ingresar a un laboratorio de una universidad durante el horario de clase?
- c) ¿Cómo comparar y validar los tres modelos de redes neuronales convolucionales a través de las métricas de medición: Accuracy, Precision y

Matriz de confusión, con el propósito de identificar aquella con mejor grado de acierto?

## 1.2. Objetivos

### 1.2.1. Objetivo General

Detectar el uso correcto de mascarillas utilizando una red neuronal convolucional para el ingreso de personas a un laboratorio de una Universidad.

### 1.2.2. Objetivos Específicos

- a) Establecer una base de datos que contenga imágenes de personas con mascarillas, sin mascarillas y haciendo mal uso de estas, de tal forma que simule el ingreso a un laboratorio de una Universidad.
- b) Implementar y entrenar tres modelos de redes neuronales convolucionales, con la ayuda de Matlab, para la identificación del uso correcto de la mascarilla en una simulación de ingreso a un laboratorio de una Universidad durante el horario de clase.
- c) Comparar y validar los tres modelos de redes neuronales convolucionales a través de las métricas de medición: Accuracy, Precision y Matriz de confusión, con el propósito de identificar aquella con mejor grado de acierto.

## 1.3. Importancia y justificación

### 1.3.1. Importancia

Desde la aparición del COVID-19 en China y luego de ser declarada como una pandemia por parte de la OMS, se dispusieron diferentes medidas de bioseguridad como: el distanciamiento social, restricción de viajes, aislamiento, uso de alcohol en gel y uso de mascarillas para evitar propagar el virus a más personas y así reducir el número de contagios.

Por lo tanto, este proyecto de tesis se enfoca en la detección del uso correcto de mascarillas en las personas para ingresar a un laboratorio de una universidad, como medida de bioseguridad para así tratar de controlar y reducir el número de contagios.

### 1.3.2. Justificación

El desarrollo de este proyecto se justifica porque es posible realizar las capturas necesarias de imágenes requeridas para establecer la base de datos simulando el

ingreso a un laboratorio de una Universidad, además de disponer del software Matlab con su librería Deep Learning Toolbox para poder implementar, entrenar y validar la red neuronal convolucional, así como también contar con la información bibliográfica referente a los modelos de redes neuronales convolucionales a usar.

#### 1.4. Limitaciones

##### 1.4.1. Limitaciones

Esta investigación posee las siguientes limitaciones:

- En este trabajo de tesis la captura de las fotografías para nuestra base de datos se limita al periodo de Junio de 2021 a Octubre de 2021.
- El desarrollo de este Proyecto de tesis está enfocado en el ingreso al laboratorio de una Universidad, pero por motivo de pandemia se limita a la recreación de dicho ingreso en un ambiente de nuestro domicilio.
- La detección del uso de la mascarilla está limitado a contar con buena iluminación para el buen funcionamiento.
- La detección del uso de la mascarilla en tiempo real se limita a detectar solo a una persona a la vez, en el instante que hace el ingreso al ambiente de recreación.
- El desarrollo de este Proyecto de tesis se limita a la detección de las mascarillas KN95 y quirúrgicas, de colores blanco, verde, celeste y negro.
- El entrenamiento de las redes neuronales convolucionales se realizará en una laptop ACER con procesador Ryzen 5 a 2.10 Ghz, Ram 12 Gb con Windows 10 de 64 Bits.

## CAPÍTULO II: MARCO TEÓRICO

### 2.1. Marco Histórico

En el pasado para poder tener un adecuado control del uso correcto de mascarilla en personas, solo se contaba con un individuo que supervisaba el uso correcto de este, dicha persona se situaba al ingreso de cualquier espacio cerrado ya sea un aula, oficina o al ingreso de un centro comercial donde es necesario usar de manera adecuada la mascarilla. Por otro lado, con el paso de los meses la tecnología, en especial la inteligencia artificial se ha convertido en una herramienta fundamental para automatizar estos eventos, dado que proporciona una ayuda en estos tipos de casos. Así como, por ejemplo, tenemos el sistema de Aplicación de Yolo en la tarea de detección de máscara, el cual de acuerdo con (Ren & Ren, 2021), realiza un modelo utilizando "YOLO" el cual es un sistema de detección de objetos que funciona sobre Darknet, una red neuronal escrita en C que funciona como Framework para aumentar la velocidad de procesamiento de la detección de máscaras en tiempo real y no comprometer la precisión. Además, dado el pequeño volumen y el desequilibrio extremo de los conjuntos de datos de detección de máscaras, adoptan un último progreso realizado en la clasificación visual de pocos disparos, CNAP simples, para mejorar el rendimiento de la clasificación. Esto demuestra lo importante que tiene y tendrá la inteligencia artificial en nuestra vida, buscando como objetivo conjunto automatizar tareas diarias, reduciendo en el proceso los posibles errores humanos.

### 2.2. Antecedentes de la investigación

#### 2.2.1. Antecedentes nacionales

(Tito Apaza & Ramirez Mejía, 2020) en la tesis: Reconocimiento automático de placas de rodaje utilizando una red neuronal convolucional para el ingreso de vehículos en la Universidad Ricardo Palma

Presentan la simulación de tres modelos de CNN con tres, cinco y ocho capas intermedias que fueron entrenados y evaluados utilizando el Toolbox Deep Learning de Matlab para el reconocimiento de placas de rodaje en vehículos. Capturó imágenes de veintidós vehículos para la creación de su base de datos utilizando una cámara web, luego estableció una carpeta de

entrenamiento con seiscientas imágenes de placas vehiculares de las cuales cuatrocientas cuarenta fueron para entrenamiento y ciento sesenta para validación.

Los tres modelos de red neuronal convolucional contaban con una capa de entrada, una capa de salida y diferentes capas intermedias. El primer modelo de CNN contaba con tres capas intermedias, el segundo modelo de CNN contaba con cinco capas intermedias y el tercer modelo de CNN contaba con ocho capas intermedias.

Comprobó la eficiencia de las tres CNN, logrando un porcentaje de efectividad del 40%, 75% y 95% respectivamente. Con esa respuesta, eligió el último modelo de CNN para implementar el reconocimiento de placas vehiculares ya que al contar con más capas intermedias la CNN tuvo mayor efectividad para reconocer las características de las placas vehiculares obteniendo mayor índice de aciertos. Luego, creó una interfaz utilizando la aplicación App Designer para visualizar con la cámara en tiempo real, después simuló el ingreso vehicular al estacionamiento de la Universidad Ricardo Palma. Validó el funcionamiento de la tercera CNN obteniendo 95% de efectividad en la correcta detección de placas vehiculares.

(Caya Perez, 2020) en la tesis titulada Evaluación de modelos de redes neuronales convolucionales aplicado a radiografías de tórax, para apoyar al proceso de diagnóstico de neumonía asociada al covid-19.

Presenta la comparación de tres modelos de redes neuronales convolucionales para la aplicación hacia las radiografías de tórax, para así contribuir en el proceso de diagnóstico de neumonía ocasionado por el COVID-19, por medio de la clasificación de imágenes. Gracias a la CNN se logró una reducción del tiempo que toma la labor de detección de la enfermedad en radiografías de tórax, además de determinar cuál de los tres modelos de red neuronal es el más apto para la detección de la enfermedad ocasionado por el COVID-19. Los modelos que se utilizaron para este proyecto de tesis son un modelo de implementación particular, ResNet50 e InceptionV3. Para la implementación de ResNet50 y InceptionV3 se necesitó aplicar transfer learning. Además, se aplicó data augmentation para

así determinar la utilidad como también la influencia en el proceso de entrenamiento de cada uno de los tres modelos. Utilizó un base de datos conformada por imágenes que contenían radiografías de tórax de casos positivos a COVID-19 y también casos normales para el entrenamiento y validación de los tres modelos. Finalmente, basado en los entrenamientos y resultados obtenidos, el modelo que más obtuvo efectividad de los tres evaluados fue InceptionV3 con un 0.9886 de exactitud cuando se entrenó con data augmentation y 0.9848 sin data augmentation

(Cayllahua Aquino & Suárez Macedo, 2019) en la tesis titulada Redes neuronales de aprendizaje profundo para el reconocimiento facial y control de acceso de estudiantes a un laboratorio.

Presenta propone en este proyecto de tesis como objetivo principal la implementación de una red neuronal convolucional de aprendizaje profundo, para el reconocimiento facial y además para tener un control de acceso de estudiantes que pertenezcan a la carrera de Ingeniería Mecatrónica; para ello, utilizó una metodología que consistía en el entrenamiento de la red neuronal convolucional, con la finalidad de extraer los datos relevantes que se encuentran en los rasgos faciales de las fotografías tomadas a un grupo de estudiantes, y cuando se comprobó con un nuevo grupo de fotografías de las mismas personas, se logró el reconocimiento de las personas. La muestra que se utilizó fue de 426 fotografías que pertenecían a un total de 14 alumnos que utilizaron el Laboratorio de Control el cual estaban matriculados en el semestre 2019-I, para este proyecto se utilizó un software empleado para el entrenamiento de la red, el cual fue el MATLAB y su librería Toolbox Deep Learning. Así como también, se realizaron diferentes pruebas y validaciones para la selección de la red neuronal convolucional, en la etapa de pruebas se inició con una capa de convolución, después en la siguiente prueba fueron dos, y finalmente tres capas, las cuales otorgaron los siguientes porcentajes de exactitud, con la primera prueba se logró un 15.63%, con la segunda prueba un 94.00% y con la tercera prueba un 67.13%, respectivamente. De esta manera, se decidió por elegir la red neuronal convolucional con dos capas de convolución, de 16 y 32 filtros, para realizar el reconocimiento facial.

### 2.2.2. Antecedentes internacionales

(Wei, Jiangjian , Chuanhong , Minmin, & Chengbin , 2017) en el artículo titulado A Cascade Framework for Masked Face Detection realizado en la escuela de Ingeniería electrónica e Información Shanghai University.

Propone desarrollar un detector facial con CNN y que tenga una estructura en cascada. Son tres redes neuronales convolucionales llamadas “Mask-12”, “Mask-24-1” y “Mask-24-2”, cada una de las CNN es un clasificador binario. Mask-12 es la CNN de primera etapa, su estructura está conformada por 5 capas intermedias y está diseñada para imágenes de entrada de cualquier tamaño luego se aplicó supresión no máxima (NMS) para poder fusionar las imágenes superpuestas de salida. Mask-24-1 es la CNN de segunda etapa, su estructura está conformada por 6 capas intermedias, la capa de entrada está dimensionada a 24x24 luego se aplicó (NMS) para fusionar las imágenes superpuestas de salida. Mask-24-2 es la CNN de tercera etapa, su estructura es la misma que Mask-12, la capa de entrada esta dimensionada a 24x 24. Para la base de datos de preentrenamiento usaron el conjunto de datos de WIDER FACE, luego eligieron 200 imágenes recopiladas de internet, de las cuales 160 son para el entrenamiento y 40 para la validación. Finalmente realizaron las pruebas donde se logró un rendimiento satisfactorio de 86.6% de precisión.

(Sahand , Haniyeh , & Ali Ahmadi, 2021) en el artículo A Face-Mask Detection Approach based on YOLO Applied for a New Collected Dataset

Propone dos métodos para detectar rostros con mascarillas y rostros sin mascarilla en tiempo real. Para la base de datos selecciona un total de 9459 imágenes obtenidas de internet, de los cuales 7320 son para entrenamiento y 2139 para validación. Los rostros de este conjunto de imágenes tienen diferentes grados de orientación y también tienen diferentes tipos de entorno, brillo y ángulos variables. Estas imágenes son etiquetadas dibujando un cuadro delimitador alrededor del rostro. Además, estas imágenes fueron aumentadas utilizando Data Augmentation manipulando el brillo y adición de ruido gaussiano que aumento el conjunto de datos alrededor de 25000 imágenes. El primer método utiliza el detector facial YOLO para poder buscar

y clasificar rostros (con mascarilla y sin mascarilla), el tamaño de entrada es de 416 x 416 y las salidas son vectores de 7 dimensiones. En el segundo método utiliza el detector facial YOLO para detectar los rostros y adiciona dos CNN cada una con una arquitectura diferente, para clasificar los rostros (con mascarilla y sin mascarilla). El primer CNN cuenta con tres capas de convolución de 3x3 de 8, 16 y 32 neuronas respectivamente, seguidas de capas de activación RELU y un Maxpolling de 2x2. Finalmente, una capa densa con 2 neuronas y activación SoftMax. La segunda arquitectura es similar a la primera con la diferencia de tener otro bloque conv-MaxPool y un adicional de capa densa con 128 neuronas y 256 neuronas antes de SoftMax. Para la validación se utilizó cuatro métricas que son: Precision, Accuracy, Recall y Map. En los métodos propuestos se logra una precisión del 99.5%.

## 2.3. Bases Teóricas relacionadas con el tema

### 2.3.1. MATLAB – Deep learning (Toolbox del MATLAB)

Según (Casado Fernández), MATLAB es el nombre abreviado de “MATriz LABoratory”. Es un programa para realizar cálculos numéricos con vectores y matrices. Uno de sus puntos fuertes es que permite construir nuestras propias herramientas reutilizables. Podemos crear fácilmente nuestras propias funciones y programas especiales (conocidos como M-archivos) en código Matlab, los podemos agrupar en Toolbox (también llamadas librerías).

Según, (MathWorks, 2021), Deep learning es un tipo de aprendizaje automático en el que un modelo aprende a realizar tareas de clasificación directamente a partir de imágenes, texto o sonido. El aprendizaje profundo generalmente se implementa mediante una arquitectura de red neuronal. El término "profundo" se refiere al número de capas de la red: cuantas más capas, más profunda es la red. Las redes neuronales tradicionales contienen solo 2 o 3 capas, mientras que las redes profundas pueden tener cientos.

### 2.3.2. Capas de la Red Neuronal Convolutiva

- Capa de entrada

Según (Rivas & Bertha , 2018), la capa de entrada se encarga únicamente de recibir las señales o patrones del exterior y propagar dichas señales a todas las neuronas de la siguiente capa.

- Capa de convolución

Según (Jalomo, Preciado, & Gudiño, 2019) ,es la capa más importante de una RNC, las neuronas de la primera capa convolutiva no están conectadas a cada uno de los píxeles de la imagen de entrada, sino solamente a aquellos que se encuentran dentro de su campo receptivo. En la segunda capa convolutiva cada neurona está conectada solamente a las neuronas localizadas en un pequeño rectángulo en la primera capa. Esta arquitectura permite que la red se concentre en características de bajo nivel en la primera capa oculta, luego las ensambla en entidades de nivel superior en la siguiente capa oculta, y así sucesivamente.

- Capa Pooling

Según (Jalomo, Preciado, & Gudiño, 2019), el objetivo de esta capa es submuestrear la imagen de entrada en orden para reducir la carga computacional, el uso de memoria y el número de parámetros Al igual que en las capas convolucionales, en la capa Pooling cada neurona está conectada a la salida de un limitado número de neuronas en las capas previas, localizadas dentro de un pequeño campo receptivo rectangular. Esta capa no tiene pesos, todo lo que hace es obtener el valor máximo (max-pooling) o el valor medio de toda esa ventana (avg-pooling). En la siguiente figura se muestra la capa Pooling.

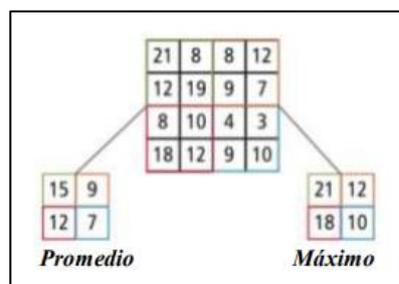


Figura N°1: Capa Pooling.

Fuente: (Jalomo, Preciado, & Gudiño, 2019).

- Full Conected

Según (Mathivet, 2019), están compuestas de una o varias capas ocultas de neuronas, vinculadas a las entradas o a las capas anteriores, y de una capa de salida, vinculada a las neuronas ocultas. Se las denomina feed-forward pues la información no puede ir más que desde las entradas hasta las salidas, sin volver atrás.

- Capa de Salida y función SoftMAX

Según (Torres, 2016), la capa de salida es una función SoftMAX en la que la salida de cada neurona corresponde a la probabilidad estimada de la clase correspondiente. Visualmente lo podríamos representar de la siguiente forma:

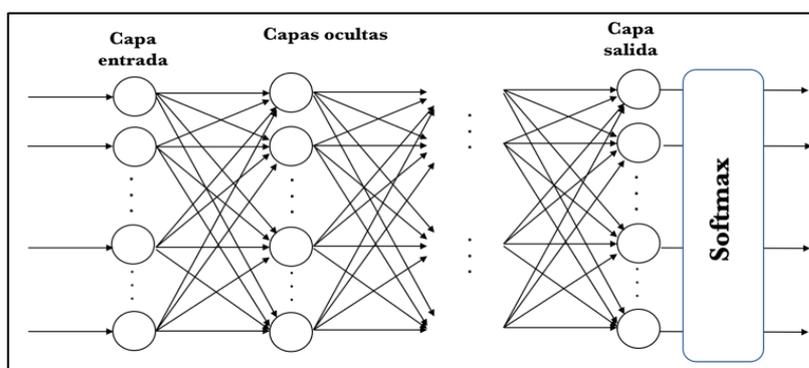


Figura N°2: Gráfico de la capa de salida y la función SoftMax.

Fuente: (Torres, 2016).

Según (Torres, 2016), la función SoftMAX se basa en calcular “las evidencias” de que una determinada imagen pertenece a una clase en particular y luego se convierten estas evidencias en probabilidades de que pertenezca a cada una de las posibles clases. Para medir la evidencia de que una determinada imagen pertenece a una clase en particular, una aproximación consiste en realizar una suma ponderada de la evidencia de pertenencia de cada uno de sus píxeles a esa clase. En la expresión matemática (1) se representa la ecuación de SoftMax:

$$Softmax_i = \frac{e^{evidencia_i}}{\sum_j e^{evidencia_j}} \quad (1)$$

### 2.3.3 Métricas de Validación

- Matriz de confusión

Según (Benítez, Escudero, & Kanaan, 2013), las medidas de evaluación se pueden expresar en función de la matriz de confusión o tabla de contingencia. La matriz de confusión contiene una partición de los ejemplos en función de su clase y predicción. La siguiente figura muestra el contenido de la matriz de confusión.

		clase real	
		positiva	negativa
predicción	positiva	verdadero positivo (tp)	falso positivo (fp)
	negativa	falso negativo (fn)	verdadero negativo (tn)

Figura N°3: Matriz de confusión.

Fuente: (Benítez, Escudero, & Kanaan, 2013).

- Accuracy

Según (Benítez, Escudero, & Kanaan, 2013), también llamada en español exactitud, corresponde a los ejemplos que se han clasificado correctamente sobre el total de ejemplos. En la siguiente expresión matemática (2) se representa la ecuación de Accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

- Precision

Según (Benítez, Escudero, & Kanaan, 2013), también llamado valor predictivo positivo, corresponde a los ejemplos positivos bien clasificados sobre el total de ejemplos con predicción positiva. En la siguiente expresión matemática (3) se representa la ecuación de Precision.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

## 2.4. Definición de términos básicos

- Mascarilla: Máscara que cubre la boca y la nariz de su portador para protegerlo de la inhalación y evitar la exhalación de posibles agentes patógenos, tóxicos o nocivos. (Real Academia Española, 2001)

- Red Neuronal Convolutacional: Es un caso concreto de redes neuronales Deep Learning, un rasgo diferencial de las CNN es que hacen la suposición explícita de que las entradas son imágenes, cosa que nos permite codificar ciertas propiedades en la arquitectura para reconocer elementos concretos en las imágenes. (Torres, 2016)
- COVID 19: Es la enfermedad causada por el nuevo coronavirus conocido como SARS-CoV-2. La OMS tuvo noticia por primera vez de la existencia de este nuevo virus el 31 de diciembre de 2019, al ser informada de un grupo de casos de «neumonía vírica» que se habían declarado en Wuhan (República Popular China). (Organización Mundial de la Salud, 2019)
- Aprendizaje Supervisado: Es la ciencia de enseñar a las computadoras a hacer predicciones basadas en datos. Implica dar a una computadora un conjunto de datos y pedirle que haga una predicción. al principio tendrá muchas predicciones incorrectas, pero en el transcurso de miles de predicciones, la computadora actualizará su algoritmo para hacer mejores predicciones. (T. Norman, 2017)
- App Designer: Es un entorno de desarrollo interactivo para diseñar una aplicación y programar su comportamiento. Proporciona una versión totalmente integrada del editor de MATLAB y un gran conjunto de componentes interactivos de la IU. También ofrece un administrador de diseño de cuadrículas para organizar la interfaz de usuario y opciones de reordenación automáticas para hacer que su app detecte y responda a los cambios del tamaño de pantalla. (MathWorks, 2021)

## 2.5. Diseño de la Investigación

### 2.5.1. Variables de investigación

Variable independiente: Red neuronal convolutacional

Variable dependiente: Detección del uso correcto de la mascarilla.

### 2.5.2. Tipo y Método de investigación

El tipo de investigación que se utiliza es aplicada. El método de investigación utilizado es el experimental, ya que se implementaron tres tipos de redes neuronales convolucionales para luego entrenarlas y en base

a su validación se estableció la red neuronal convolucional más apropiada para lograr así la detección del uso correcto de la mascarilla, en tiempo real.

#### 2.5.3. Técnicas e Instrumentos de recolección de datos

Para la recolección de datos se usó la técnica de observación experimental, ya que la toma de información para la base de datos se realizó con la ayuda de una cámara de celular, el cual tiene una resolución de 48 MP permitiendo la grabación de los videos de las personas con mascarilla, sin mascarilla y haciendo mal uso de estas. El instrumento empleado para la detección en tiempo real es una cámara web HD 720P integrada a una laptop, con la cual se verifico como se estará usando la mascarilla y con ello validar la información a través de la red neuronal convolucional.

#### 2.5.4. Procedimiento para la recolección de datos

En primer lugar, se grabó videos de aproximadamente 10 segundos por cada persona haciendo uso correcto de la mascarilla, sin la mascarilla y haciendo mal uso de estas. Luego se extrajo los fotogramas de cada video, con las imágenes obtenidas se clasificó en tres carpetas diferentes, la cual cada carpeta está conformada de 900 imágenes. Teniendo la base de datos será utilizada para el entrenamiento y validación de los tres modelos de redes neuronales convolucionales, y así proceder a la simulación.

## CAPÍTULO III: DISEÑO DE LA INVESTIGACIÓN

En este capítulo se muestra el diseño del proyecto de tesis, para validar los objetivos, tanto específicos como el general, definidos en el capítulo 1. Primero se mostrará el diagrama de bloques del desarrollo del proyecto de tesis, luego el procedimiento para la adquisición de las imágenes, las cuales conformarán la base de datos. Después se trabajó en el procesamiento de imágenes para entrenar los tres modelos de redes neuronales convolucionales particulares propuestos, seguidamente se validó los tres modelos para definir el más apropiado en la detección de mascarillas. Por último, se implementará la interfaz con ayuda de la aplicación App Designer para la simulación respectiva.

### 3.1. Diagrama de bloques

El diagrama que se muestra a continuación, detalla las etapas que comprende el proyecto de tesis planteado.

En primer lugar, se observa la etapa de adquisición de datos, la cual está conformada de imágenes recolectadas a través de una cámara de celular.

Luego la siguiente etapa representa la creación de la base de datos, la cual fue utilizada para los entrenamientos de las redes neuronales convolucionales particulares.

Después de tener lista la base de datos viene la etapa de entrenamiento, donde se usó la herramienta Deep Learning Toolbox de MATLAB, seguidamente se evaluó las tres redes neuronales convolucionales particulares usando la matriz de confusión y las métricas Precision y Accuracy, propuestas en este proyecto de tesis.

Por último, se determinó la red neuronal convolucional definida y se implementó la interfaz con ayuda de la aplicación App Designer en MATLAB para su respectiva simulación. Ver la figura N°4.

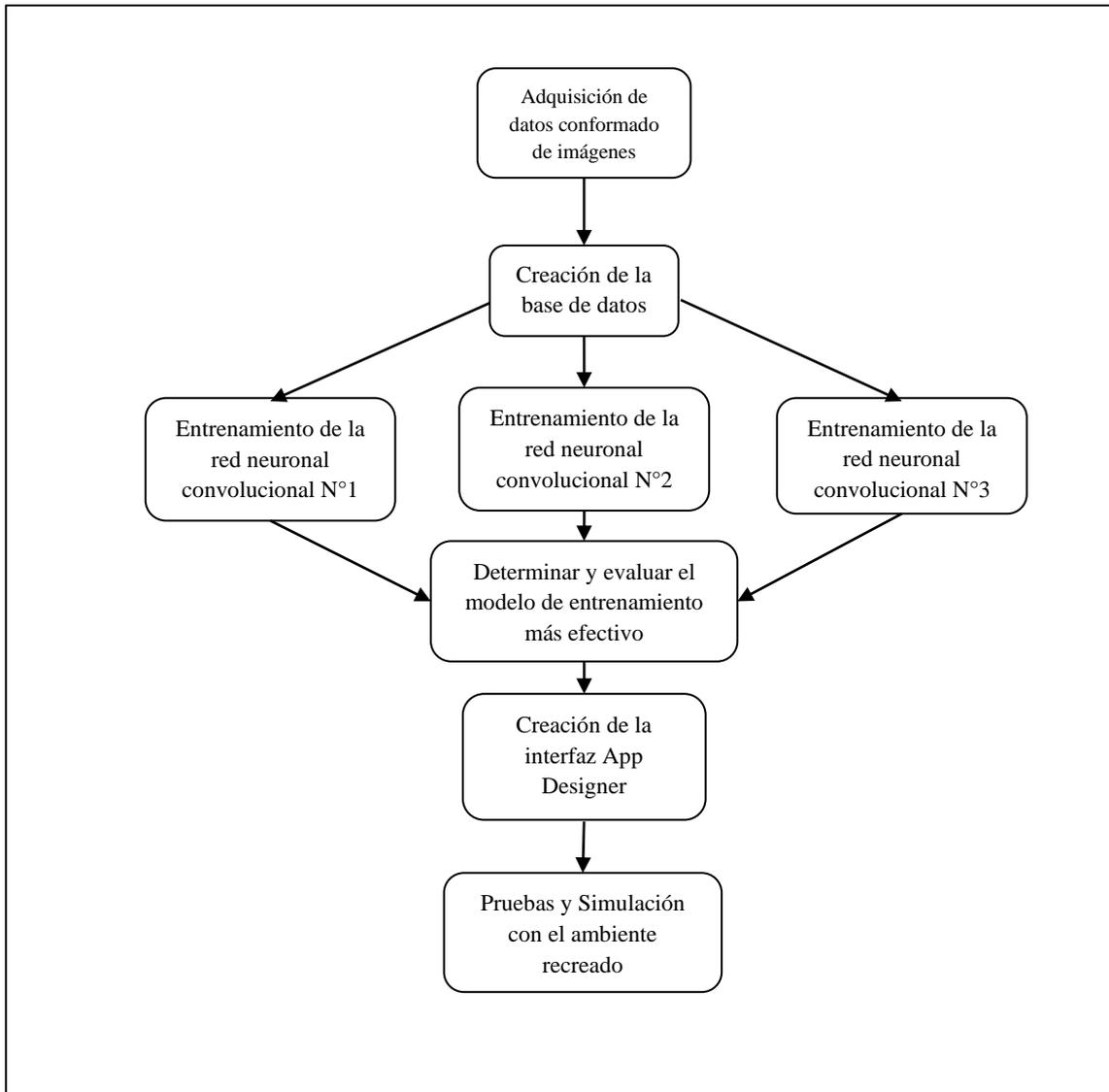


Figura N°4: Diagrama de Bloques.

Fuente: Elaboración propia.

### 3.2. Procedimiento de adquisición de imágenes

En este proceso se optó en primer lugar, grabar el rostro de 30 personas para que conformarán la base de datos, antes de empezar con la grabación, se decidió tener un fondo blanco, para que así al momento de entrenar la red neuronal convolucional no tenga mucha carga computacional y pueda facilitar el reconocimiento de las imágenes.

En el momento de la grabación, se le pidió a cada persona mostrar su rostro con mascarilla, sin mascarilla y haciendo mal uso de estas (mascarilla tapando la boca y nariz descubierta), cada proceso de grabación tuvo un aproximado de 10 a 11

segundos, y en ese tiempo se pidió a cada persona mover su rostro en ángulos distintos.

Para el proceso de grabación se utilizó un dispositivo móvil, Huawei Y8P, este dispositivo cuenta con una pantalla de 6,3", de 1080x2400 pixeles, 4 GB de RAM, y lo más importante con una cámara de 48 MP. El dispositivo móvil se visualiza en la siguiente figura.



Figura N°5: Dispositivo Móvil.

Fuente: Elaboración propia.

Luego de terminar con el proceso de grabación, se separó en tres carpetas los videos recopilados, Con Mascarilla, Sin Mascarilla y Mal Uso. La siguiente figura muestra las carpetas, las cuales contiene los videos recopilados.



Figura N°6: Creación de las carpetas.

Fuente: Elaboración propia.

Seguidamente con la ayuda del MATLAB, se tomaron los frames de cada video, en esta etapa cada video arrojó un aproximado de 315 imágenes, la siguiente figura nos muestra las características de un solo video, se puede ver el número de frames y la duración, como también el formato del video.

```
General Properties:
  Name: '1_1.mp4'
  Path: 'C:\Users\XAVIER\Downloads\New Folder\ESCALA DE GRISES'
  Duration: 10.7400
  CurrentTime: 0
  NumFrames: 315

Video Properties:
  Width: 720
  Height: 1280
  FrameRate: 29.3306
  BitsPerPixel: 24
  VideoFormat: 'RGB24'
```

Figura N°7: Características del video.

Fuente: Captura de pantalla del Matlab

Después de ver las características del video, se visualizó que había un gran número de imágenes que parecían repetidas, por eso se cambió el algoritmo para obtener un aproximado de 30 frames por cada video, y así cada imagen sea distinta. Las figuras N°8 y N°9 muestran el algoritmo en MATLAB y el resultado al ejecutarlo.

```
1 - a=VideoReader('1_1.mp4');
2 - get(a)
3 - img=1;
4 - for i = 1:6:a.NumberOfFrames;
5 -     filename=strcat('frame',num2str(img),'.jpg');
6 -     aa = read(a, i);
7 -     imwrite(aa,filename);
8 -     img=img+1;
9 - end
```

Figura N°8: Algoritmo para convertir video en frames.

Fuente: Captura de pantalla del Matlab.



Figura N°9: Resultado en frames.

Fuente: Captura de pantalla del Matlab

Por último, después de solucionar el tema de los frames, se prosiguió a tomar los frames de cada video, obteniendo un total de 2923 imágenes, seguidamente se ordenaron las imágenes en la carpeta que le correspondía, obteniendo un total de 968 imágenes en la carpeta de Con Mascarilla, 988 imágenes en la carpeta Mal Uso y 967 imágenes en la carpeta Sin Mascarilla. De los cuales 2250 imágenes fueron utilizados para el proceso de entrenamiento y 673 para la validación.

### 3.3. Procesamiento de las imágenes para la base de datos

Para este proceso se decidió cambiar el formato de las imágenes que lleva la base de datos, en primer lugar, se optó por cambiar la resolución a 100x100, ya que, al cambiarlo a este formato, el entrenamiento de la red neuronal convolucional no sea tan exhausta.

Luego de cambiar las imágenes de la base de datos a una resolución de 100x100, se cambió a una escala de grises, para que cada imagen que conforma la base de datos tenga un dimensionamiento igual. La finalidad de llevarlo a escala de grises es para evitar un mayor procesamiento y así el entrenamiento de la red neuronal convolucional sea más sencillo.

En la figura N°11 se detalla el algoritmo en MATLAB para el procesamiento de las imágenes que conforma la base de datos. lo primero que hace este algoritmo es llamar el directorio donde se encuentra la base de datos usando el comando de MATLAB “imread”.

Antes de continuar con el proceso, se aplicó un bucle “for”, ya que este proceso se tenía que hacer con cada imagen de la base de datos, cabe recordar que cada video nos arrojó un aproximado de 30 frames, cada imagen tiene un numero propio, como se observa en la figura N°10.



Figura N°10: Imagen.

Fuente: Captura de pantalla del Matlab

Este “for” nos permitió hacer un bucle para la conversión de cada imagen. Luego de hacer un bucle, se utilizó “imresize”, este comando nos brinda la capacidad de cambiar el tamaño de la imagen, y en este caso lo cambiamos a una resolución de 100x100.

Seguidamente se utilizó el comando “rgb2gray”, el cual convierte la imagen que estaba a color en una de escala de grises. Este proceso que se realizó en MATLAB se calculó a través de una operación matemática basada en una ponderación de los canales de color.

Por último, con el comando “imwrite” se vuelve a escribir en una nueva carpeta que contenga las nuevas imágenes procesadas, la cual estará en resolución 100x100 y en escala de grises.

```
1 i=imread('C:\Users\XAVIER\Downloads\New Folder\ESCALA DE GRISES\frame1.jpg')
2 for i=1:53
3     nom=num2str(i);
4     arch=strcat('C:\Users\XAVIER\Downloads\New Folder\ESCALA DE GRISES','\frame',nom, '.jpg');
5     I=imread(arch);
6     try
7         I=imresize(I, [100 100], 'nearest');
8         I=rgb2gray(I);
9     catch exception
10    end
11    imwrite(I, strcat('C:\Users\XAVIER\Downloads\New Folder\Grises', '\frame', nom, '.jpg'));
12 end
```

Figura N°11: Algoritmo para el procesamiento de imágenes.

Fuente: Captura de pantalla del Matlab.

Después de hacer el procesamiento para cada imagen, se ordenó en cada carpeta respectiva, la figura N°12 muestra el resultado del procesamiento de algunas imágenes que conformo la carpeta “Con Mascarilla”.

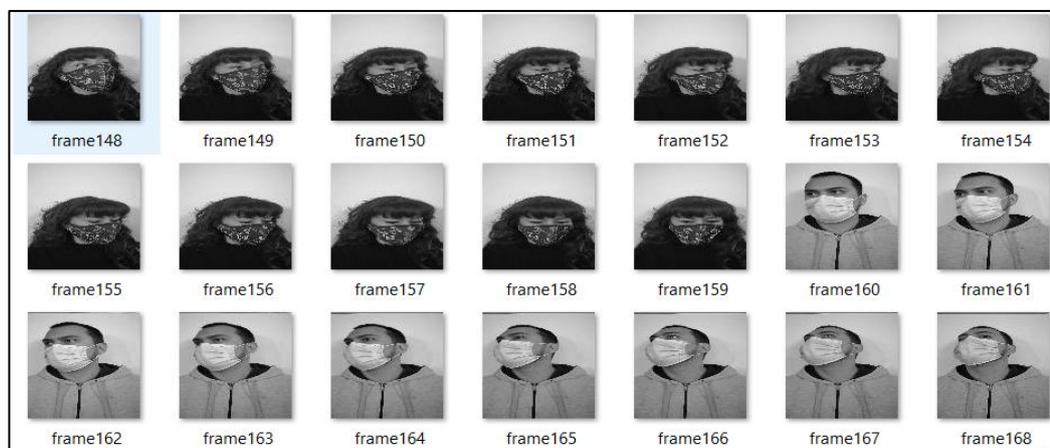


Figura N°12: Procesamiento de imágenes para la carpeta “Con Mascarilla”.

Fuente: Captura de pantalla del Matlab.

La figura N°13 muestra el resultado del procesamiento de algunas imágenes que conformo la carpeta “Mal Uso”.

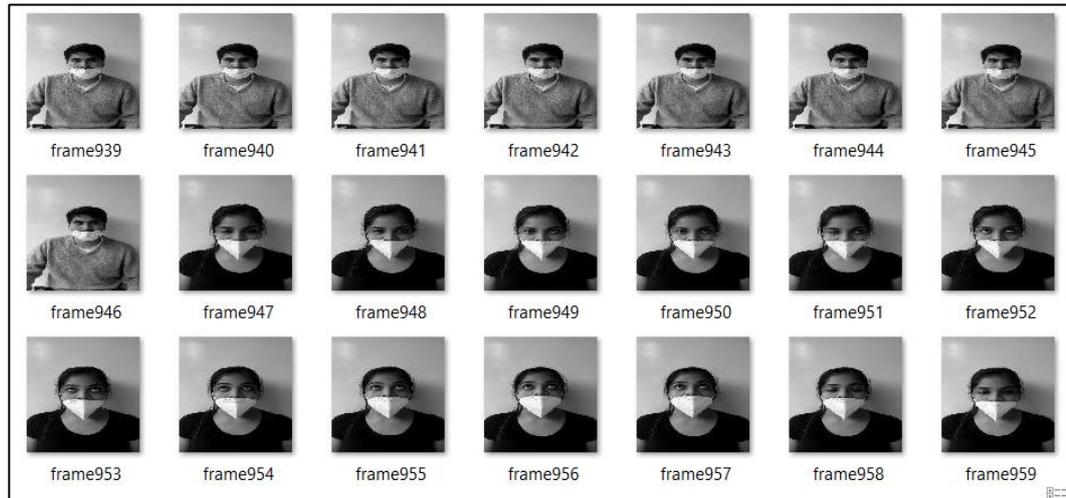


Figura N°13: Procesamiento de imágenes para la carpeta “Mal Uso”.

Fuente: Captura de pantalla del Matlab.

Por último, la figura N°14 muestra el resultado del procesamiento de algunas imágenes que conformo la carpeta “Sin Mascarilla”



Figura N°14: Procesamiento de imágenes para la carpeta “Sin Mascarilla”.

Fuente: Captura de pantalla del Matlab.

En la figura N°15 se muestra la base de datos ordenada, la cual se llevó al entrenamiento para las tres redes neuronales convolucionales.

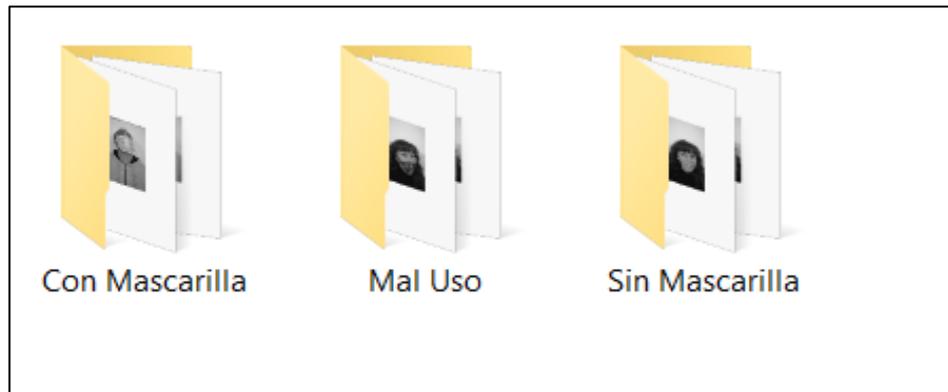


Figura N°15: Nueva Base de datos.

Fuente: Elaboración propia.

### 3.4. Entrenamiento y Ejecución de las redes neuronales convolucionales en Matlab

Teniendo la base de datos establecida, se procedió a crear y entrenar las tres redes neuronales convolucionales utilizando el Toolbox Deep Learning de Matlab. Se designaron tres capas de convolución para el primer modelo de red. El tamaño del filtro de la capa de convolución es de 3x3 y comienza con 8 neuronas. (Tito Apaza & Ramirez Mejía, 2020)

Luego, se procedió a agregar las demás capas intermedias Maxpooling, batchnormalizationlayer y relulayer según indica (Wei, Jiangjian , Chuanhong , Minmin, & Chengbin , 2017)

#### 3.4.1. Modelo de red neuronal convolucional N°1

Para la creación de la Red Neuronal Convolucional Particular 1 se definió utilizar 750 imágenes por cada grupo para el entrenamiento, y 210 para la validación tal como se muestra en la figura 16.

```
clear all;
clc;
close all;

cat = {'Con Mascarilla', 'Sin Mascarilla', 'Mal Uso'};

rootFolder = 'C:\BASE DE DATOS';

imds = imageDatastore(fullfile(rootFolder, cat), ...
    'LabelSource', 'foldernames');

$labelCount = countEachLabel(imds);

img = readimage(imds,1);
$size(img);

numTrainFiles = 750;
[imdsTrain, imdsValidation] = splitEachLabel(imds,
numTrainFiles, 'randomize');
```

Figura N°16: Entrenamiento de red neuronal convolucional particular 1.

Fuente: Captura de pantalla del Matlab

Luego se configuraron las capas de la red neuronal convolucional a ejecutar, la capa de entrada (imageInputLayer), está dimensionada a una resolución de 100x100 píxeles en escala de grises.

La red neuronal convolucional está integrada por un grupo de capas las cuales se denominan capas intermedias, estas capas intermedias son conformadas a su vez por cuatro capas: Capa convolucional, capa de normalización por lotes, capa de función de activación (ReLU) y capa de agrupación máxima, las cuales fueron ejecutadas tres veces.

Particularmente la capa de agrupación máxima solo fue aplicada una vez en este modelo de red. Después se adiciona la capa de salida, que está compuesta por tres capas: Capa fullyConnected, capa de activación Softmax y la capa de clasificación. La siguiente figura N°17 muestra la arquitectura del primer modelo de red neuronal convolucional propuesto.

```
%Capa de entrada
layers = [
    imageInputLayer([100 100 1])

%Capas intermedias

    convolution2dLayer(3,8, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2, 'Stride', 2)

    convolution2dLayer(3,16, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(3,32, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

%Capa de salida

    fullyConnectedLayer(3)
    softmaxLayer
    classificationLayer
];
```

Figura N°17: Programación de capas convolucionales.

Fuente: Captura de pantalla del Matlab.

Después de definir la arquitectura de la Red Neuronal Convolucional Particular 1: “capa de entrada”, “las capas intermedias” y “capa de salida” se realizó la programación para el entrenamiento y clasificación, tal como se muestra en la figura N°18.

```

options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.01, ...
    'MaxEpochs',10, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(imdsTrain, layers, options);

YPred = classify(net, imdsValidation);
YValidation = imdsValidation.Labels;

accuracy = sum(YPred ==
    YValidation)/numel(YValidation);

```

Figura N°18: Opciones de entrenamiento.

Fuente: Captura de pantalla del Matlab.

En la figura N°19 se detalla el diagrama de la Red Neuronal Convolutional Particular 1.

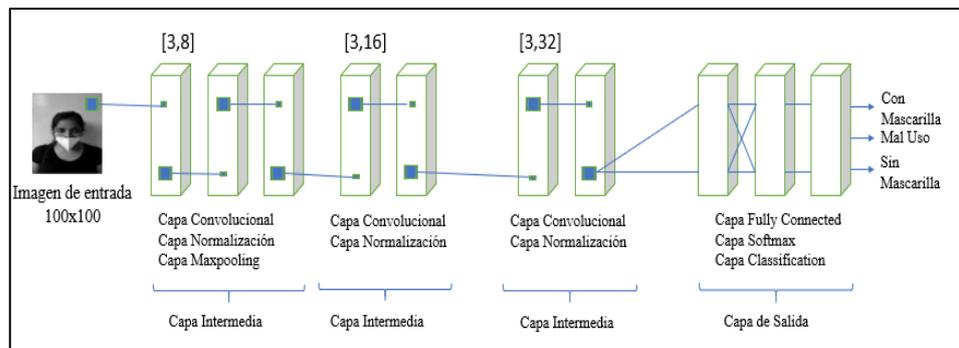


Figura N°19: Diagrama de capas convolucionales de CNN particular 1.

Fuente: Elaboración propia.

Con la programación realizada, y con el apoyo del Deep Learning Toolbox de MATLAB, se logró una figura que muestra el entrenamiento de la Red Neuronal Convolutional Particular 1, como se aprecia en la figura N°20.

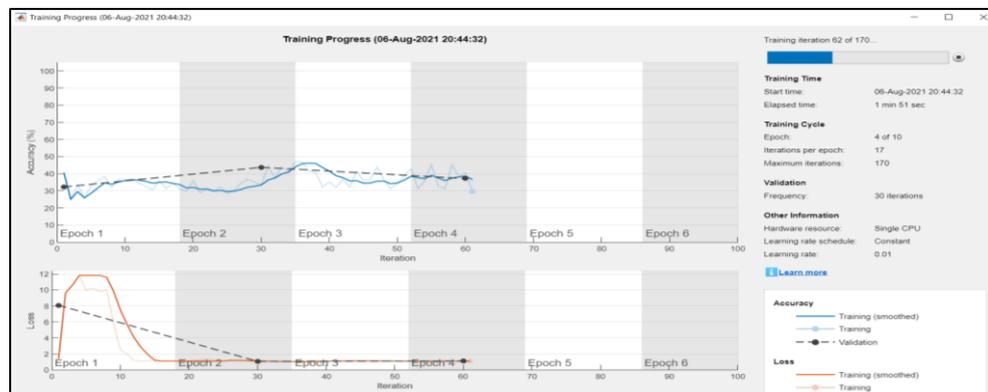


Figura N°20: Inicio de entrenamiento de CNN particular 1.

Fuente: Captura de pantalla del Matlab.

El tiempo de entrenamiento duró cuatro minutos y treinta y ocho segundos, el cual conforma 30 iteraciones como se visualiza en la figura N°21.

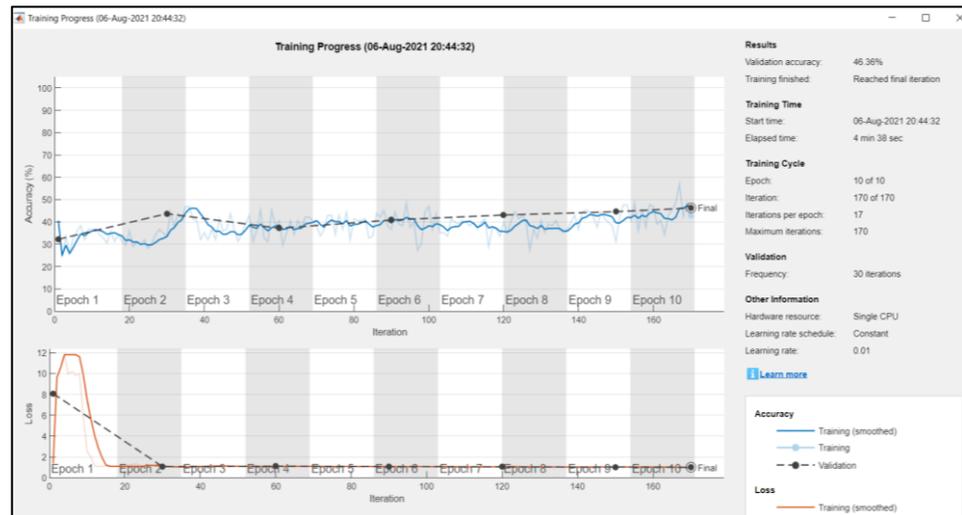


Figura N°21: Resultados del entrenamiento y validación de CNN particular 1.

Fuente: Captura de pantalla del Matlab.

Finalmente, después de entrenada la Red Neuronal Convolutiva Particular 1 se visualiza que tiene un Validation accuracy de 46.36%.

### 3.4.2. Modelo de red neuronal convolutiva N°2

La Red Neuronal Convolutiva Particular 2 está diseñado con el objetivo de minimizar errores obtenidos por parte de la Red Neuronal Convolutiva Particular 1, para este segundo modelo el entrenamiento y validación se realizó de igual manera con 750 imágenes para entrenamiento y 260 para la validación.

En la siguiente figura N°22 y figura N°23 se visualiza la programación, en este modelo se utilizó cinco capas intermedias, cada una de ella con su función de activación RELU, con la finalidad de obtener un resultado mejor al anterior.

```

clear all;
clc;
close all;

cat = {'Con Mascarilla', 'Sin Mascarilla', 'Mal Uso'};

rootFolder = 'C:\BASE DE DATOS';

imds = imageDatastore(fullfile(rootFolder, cat), ...
    'LabelSource', 'foldernames');

%labelCount = countEachLabel(imds);

img = readimage(imds, 1);
%size(img);

numTrainFiles = 750;
[imdsTrain, imdsValidation] = splitEachLabel(imds,
numTrainFiles, 'randomize');

```

Figura N°22: Entrenamiento de red neuronal convolucional particular 2.

Fuente: Captura de pantalla del Matlab.

```

layers = [
    imageInputLayer([100 100 1])

    convolution2dLayer(3,8, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2, 'Stride', 2)

    convolution2dLayer(3,16, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2, 'Stride', 2)

    convolution2dLayer(3,32, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(3,64, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(3,128, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(3)
    softmaxLayer
    classificationLayer
];

```

Figura N°23: Programación de capas convolucionales.

Fuente: Captura de pantalla del Matlab.

Para la programación del entrenamiento y clasificación, se cambió el parámetro MaxEpochs a 15 para aumentar el rendimiento de la red neuronal convolucional particular 2, tal como se muestra en la figura N°24.

```

options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.01, ...
    'MaxEpochs',15, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(imdsTrain, layers, options);

YPred = classify(net, imdsValidation);
YValidation = imdsValidation.Labels;

accuracy = sum(YPred ==
    YValidation)/numel(YValidation);

```

Figura N°24: Opciones de entrenamiento.

Fuente: Captura de pantalla del Matlab.

En la figura N°25 se detalla el diagrama de la Red Neuronal Convolutional Particular 2.

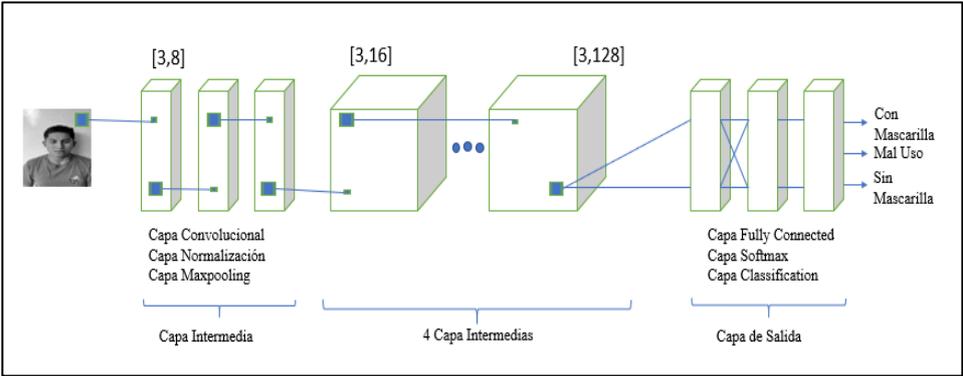


Figura N°25: Diagrama de capas convolucionales CNN particular 2.

Fuente: Elaboración propia.

Con la programación realizada, y con el apoyo del Deep Learning Toolbox de MATLAB, se logró una figura que muestra el entrenamiento de la Red Neuronal Convolutional Particular 2, como se aprecia en la figura N°26.

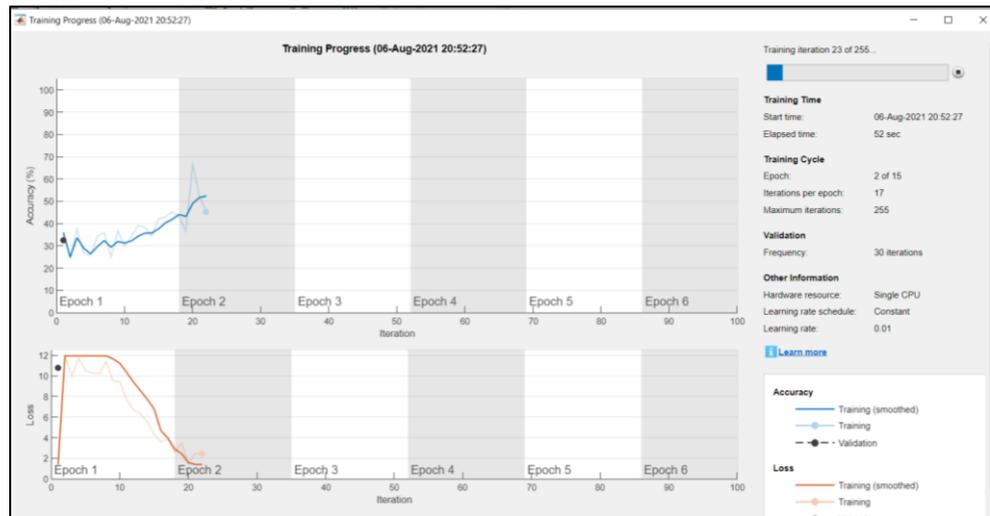


Figura N°26: Inicio de entrenamiento de la CNN particular 2.

Fuente: Captura de pantalla del Matlab.

El tiempo de entrenamiento duró nueve minutos y veinte segundos, el cual conforma 30 iteraciones como se visualiza en la figura N°27.

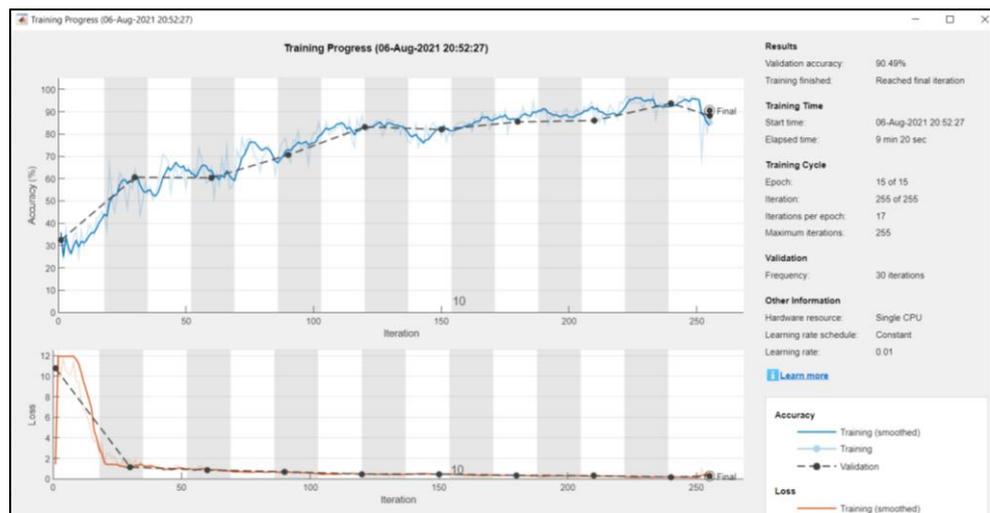


Figura N°27: Resultado del entrenamiento y validación de la CNN particular 2.

Fuente: Captura de pantalla del Matlab

Finalmente, después de entrenada la Red Neuronal Convolutiva Particular 2 se visualiza que tiene un Validation accuracy de 90.49%.

### 3.4.3. Modelo de red neuronal convolutiva N°3

La Red Neuronal Convolutiva Particular 3 tiene como base al modelo anterior, se varió la cantidad de capas intermedias, este último modelo tiene siete capas intermedias cada una de ellas con su función de activación RELU. En la programación del entrenamiento se volvió a cambiar el parámetro

MaxEpoch a 20, todo esto con la finalidad de hacer más eficiente a la Red Neuronal Convolutacional Particular 3.

En la figura N°28 se detalla el diagrama de la Red Neuronal Convolutacional Particular 3.

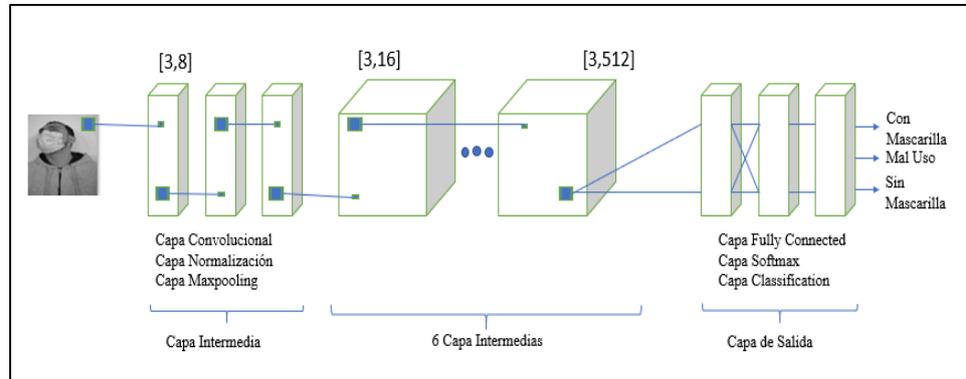


Figura N°28: Diagrama de capas convolucionales de la CNN particular 3.

Fuente: Elaboración propia.

Con la programación realizada, y con el apoyo del Deep Learning Toolbox de MATLAB, se logró el entrenamiento de la Red Neuronal Convolutacional Particular 3 como se aprecia en las figuras N°29 y N°30.

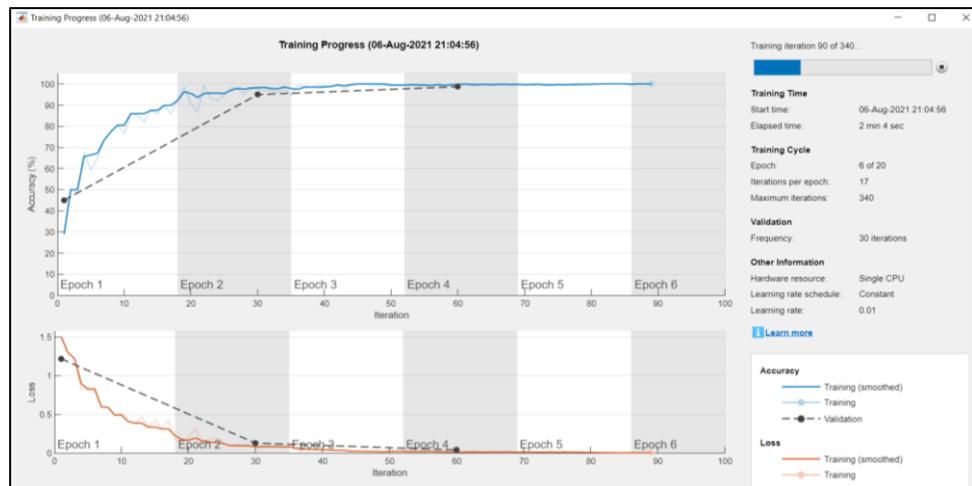


Figura N°29: Inicio de entrenamiento de la CNN particular 3.

Fuente: Captura de pantalla del Matlab.

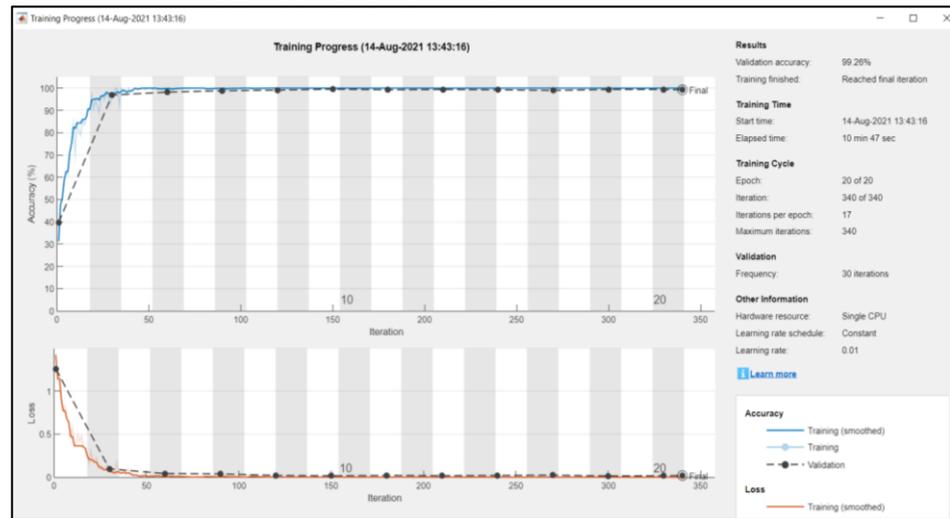


Figura N°30: Resultado del entrenamiento y validación de la CNN particular 3.

Fuente: Captura de pantalla del Matlab.

Finalmente, después de entrenada la Red Neuronal Convolutiva Particular 3 se visualiza que tiene un Validation accuracy de 99.26%. Este último modelo obtuvo mejores resultados con relación a los dos modelos anteriores.

### 3.5 Implementación de la interfaz con App Designer

Luego de entrenar las tres redes neuronales convolucionales particulares, se procedió a crear la interfaz en App Designer usando el software MATLAB, para así poder tener una visualización y captura, tanto de la persona si está o no usando su mascarilla, como también si la está usando mal; Todo esto en tiempo real.

La primera etapa de este proceso es tener el complemento en el software MATLAB, para la compatibilidad con la cámara y así no tener problema en enlazar la interfaz con la cámara que tomará la captura en tiempo real. El MATLAB ofrece descargar complementos a través del explorador Add-On, como se muestra en la figura N°31.

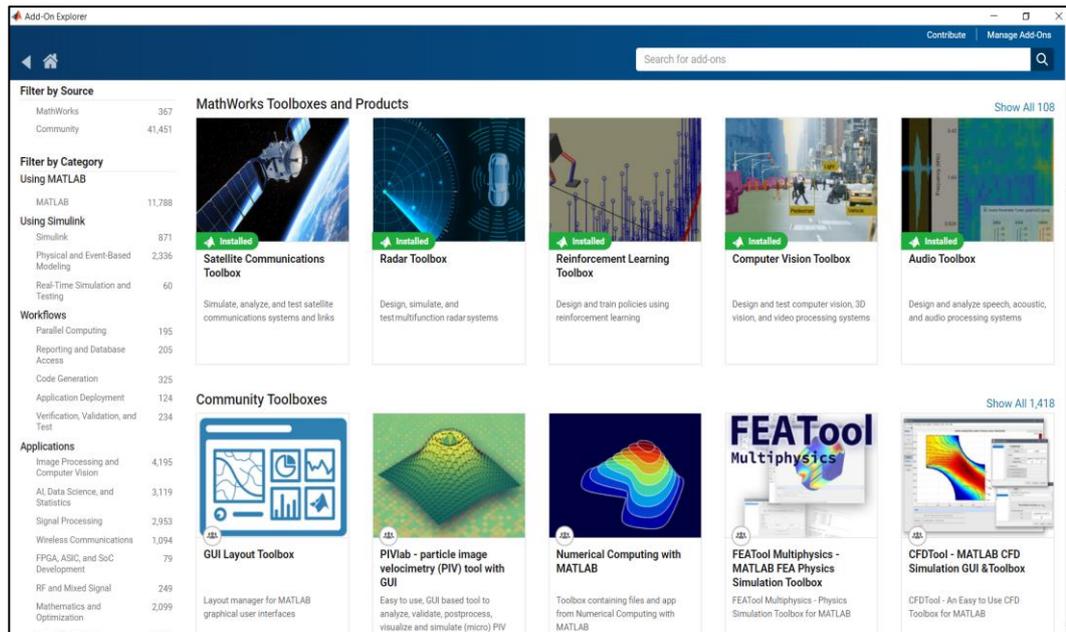


Figura N°31: Explorador Add-On de MATLAB.

Fuente: Captura de pantalla del Matlab.

El complemento que se descargó fue Classify Webcam Images Using Deep Learning, este complemento permitió interactuar con la cámara y la aplicación App Designer que contendrá la red neuronal convolucional particular. El complemento se muestra en la siguiente figura.

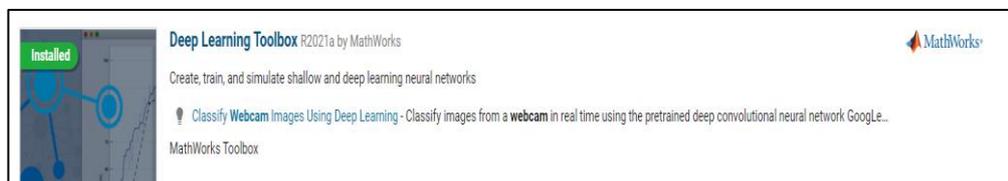


Figura N°32: Classify Webcam Images.

Fuente: Captura de pantalla del Matlab.

La segunda etapa en el proceso de creación de la interfaz fue configurar la programación para que la interfaz interactúe con la cámara y la red neuronal convolucional particular. Primero se guardó la red neuronal convolucional entrenada, en la siguiente figura se muestra que la red neuronal convolucional se guardó en la interfaz App Designer. El parámetro “net” hace referencia a la red neuronal convolucional entrenada, al almacenar la red neuronal convolucional ya no será necesario entrenar nuevamente, solamente se deberá acceder a la aplicación App Designer.

```
properties (Access = public)
    Property % Description
    net
end
```

Figura N°33: Guardado de la red neuronal convolucional en la interfaz App Designer.

Fuente: Captura de pantalla del Matlab.

Seguidamente se procedió a configurar la programación con respecto al proceso de captura de imagen, procesamiento de imagen y el resultado final, el cual informará si está usando la mascarilla correcta o incorrecta, como también si está usando la mascarilla mal puesta.

Primero se cargó la red neuronal convolucional almacenada, luego se hizo un bucle con el comando “while” y dentro de este se especifica la cámara que se estará usando para la toma de captura, luego con la imagen capturada a color se redimensiona a una resolución 100x100 y escala de grises, como se hizo con la base de datos para el entrenamiento.

La red clasificó la imagen comparando con sus tres salidas, luego mostró en la interfaz con un texto, todo este proceso tuvo una pausa de tres segundos, en la figura N°34 se muestra la programación.

Después de procesar la imagen capturada, se hace una predicción con el comando “predict”, y da como resultado un vector del tamaño de la salida. Este comando “predict” vendría a dar la respuesta de la capa SoftMAX, en resumen, da a entender que tan probable es que sea uno o el otro.

Después se utiliza el comando “if” y “else”, para comparar y saber quién tiene el mayor valor de las tres salidas, al tener el mayor valor realiza una última comparación, el cual tiene que superar el 0.9 del umbral de predicción dicho valor fue definido en base a las pruebas realizadas en la simulación, para así la red neuronal convolucional pueda clasificarlo de manera correcta.

Por último, se agregó a la interfaz un componente tipo lámpara, primero con el comando “strcmp”, compara dos textos, su función es darle un color al resultado final, se le dio un color verde si la salida es con mascarilla, y le dará un color rojo si no está usando la mascarilla o está haciendo mal uso de la mascarilla. Si no detecta nada la lámpara es de color gris a la espera de una nueva captura. Todo este proceso tiene una pausa de 3 segundos, después de transcurrido estos segundos,

podrá tomar otra captura y así sucesivamente. En la figura N°34 se muestra la programación completa en el App Designer.

```
load('red.mat','net');
while true
    cam=webcam('VGA WebCam');
    img=snapshot(cam);
    img2 = imresize(img, [100 100]);
    img2 = rgb2gray(img2);
    app.Image.ImageSource = img;
    out = predict(net,img2);
    if out(1)>out(2)
        max=out(1);
    else
        max=out(2);
    end
    if max<out(3)
        max=out(3) ;
    end
    if max>=0.9
        salida=classify(net,img2);
        app.lblEstado.Text = salida;
        if strcmp(string(salida),"Con Mascarilla")
            app.RegistroLamp.Color=[0 1 0];
        else
            app.RegistroLamp.Color=[1 0 0];
        end
        pause (3);
    end
    app.lblEstado.Text = ' ';
    app.RegistroLamp.Color=[0.65 0.65 0.65];
end
```

Figura N°34: Programación completa en el App Designer.

Fuente: Captura de pantalla del Matlab.

En la figura N°35 se muestra en el lado izquierdo los comandos usados en la programación, las cuales reflejan los estados, y al lado derecho se muestra la ventana donde se visualizó el título de la interfaz, la imagen capturada, la lampara y el estado en texto, todo esto en tiempo real

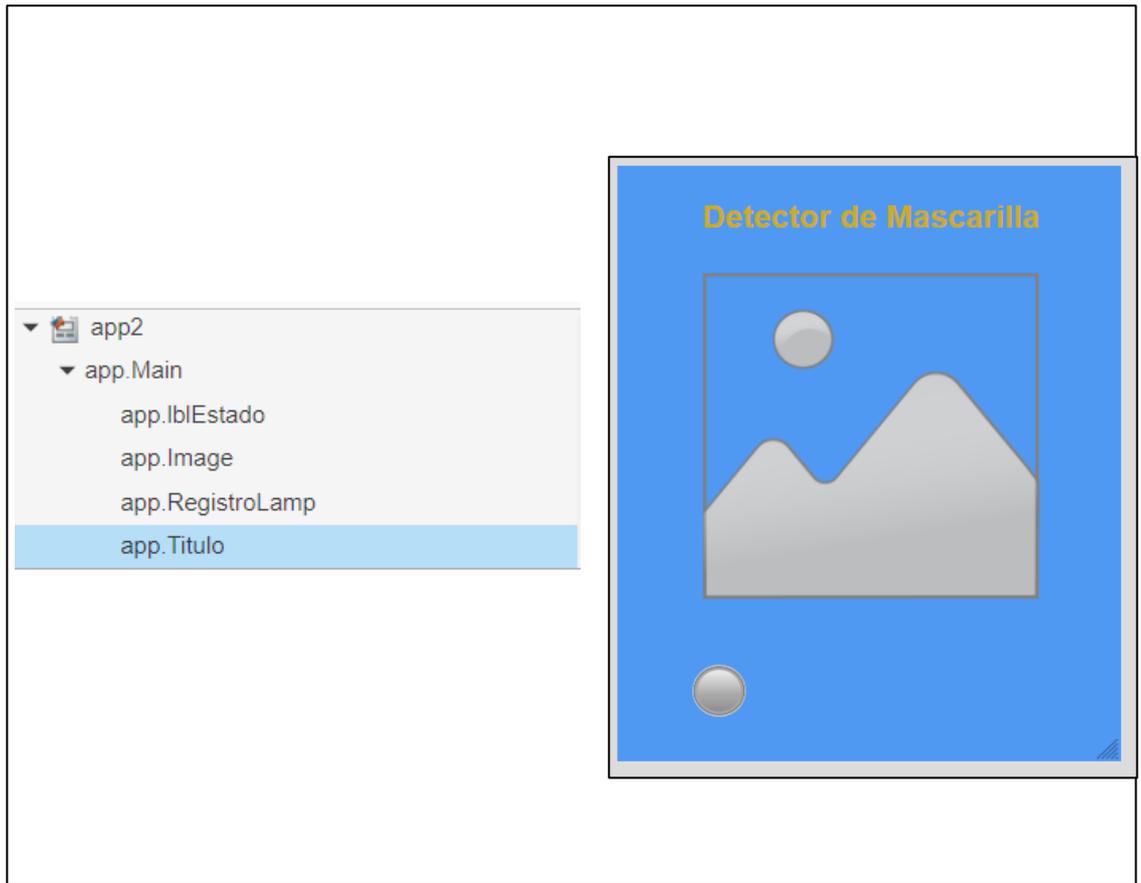


Figura N°35: Interfaz mediante App Designer.

Fuente: Captura de pantalla del Matlab.

## CAPÍTULO IV: PRUEBAS Y RESULTADOS

En el presente capítulo, se muestran las pruebas y resultados obtenidos en las validaciones realizadas con los tres modelos de redes neuronales convolucionales implementadas en el capítulo anterior.

Para cada modelo de red neuronal convolucional particular se realizó la matriz de confusión con ayuda del Deep Learning Toolbox en MATLAB utilizando el siguiente parámetro:

`“plotconfusion(YValidation,YPred)”;`

Donde YValidation es el parámetro de entrada, conocido como “target” (definida por el usuario) y YPred es la salida, conocido como “resultado” (resultado final de la red neuronal convolucional particular).

Posteriormente, se procedió a realizar los cálculos de las métricas Accuracy y Precision para cada red neuronal convolucional particular.

### 4.1. Validación de la Red Neuronal Convolucional Particular 1

#### 4.1.1 Evaluación de la Matriz de Confusión

En la siguiente figura N°36 se presenta la matriz de confusión de la red neuronal convolucional particular 1, para el valor “Con Mascarilla” se obtuvo 175 imágenes clasificadas de manera correcta teniendo un porcentaje de 40.2% de precisión. En el valor “Mal Uso” se obtuvo 41 imágenes clasificadas de forma correcta teniendo un porcentaje de 6.1% de precisión. En el valor “Sin Mascarilla” se obtuvo 96 imágenes clasificadas de manera correcta teniendo un porcentaje de 14.3% de precisión.

La red logra clasificar de manera correcta 312 imágenes, de un total de 673 imágenes de validación. Obteniendo un porcentaje total de clasificación correcta de 46.4% y un porcentaje de 53.6% que comprenden los falsos negativos, falsos positivos y verdaderos negativos para cada valor establecido.

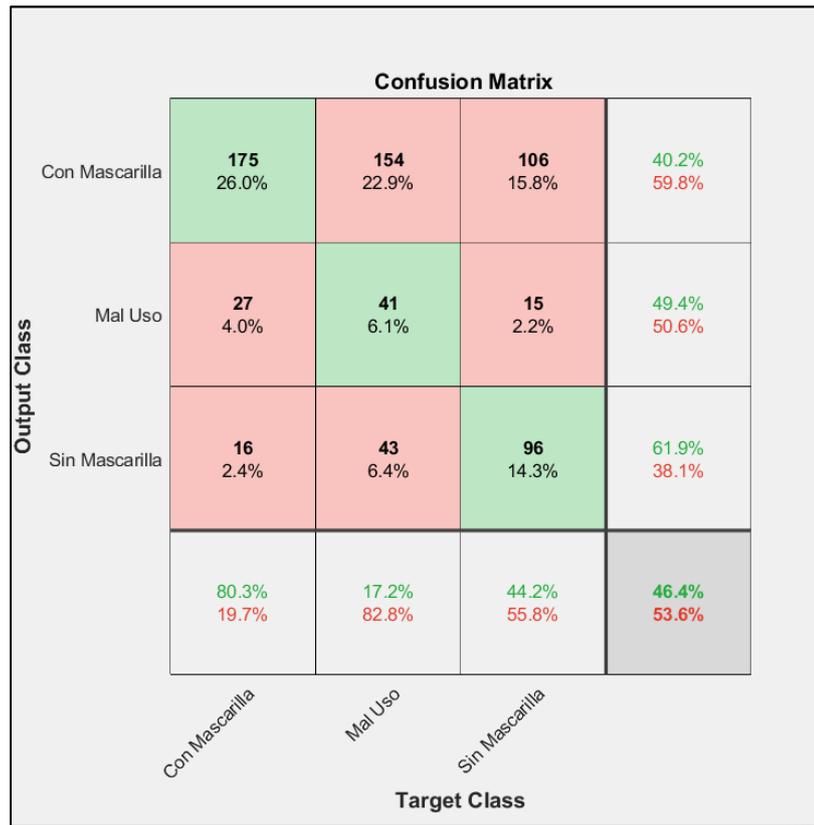


Figura N°36: Matriz de confusión para la red neuronal convolucional particular 1.

Fuente: Captura de pantalla del Matlab.

La siguiente tabla muestra los valores de los verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN).

Tabla N°1: Valores según la matriz de confusión del modelo de red particular 1.

Valor	TP	FP	TN	FN
<b>Con Mascarilla</b>	175	260	195	43
<b>Mal Uso</b>	41	42	393	197
<b>Sin Mascarilla</b>	96	59	397	121

Fuente propia.

#### 4.1.2 Evaluación de Accuracy y Precision

Aplicando las ecuaciones de las métricas mencionadas vistas en el capítulo II, y con los datos obtenidos anteriormente en la matriz de confusión donde se aprecian los valores verdadero positivo (TP), falso positivo (FP), verdadero negativo (TN) y falso negativo (FN).

Se validó la red neuronal convolucional a través de las métricas Accuracy y Precision, para cada valor de entrada y son mostradas en las siguientes expresiones matemáticas.

Valor “Con Mascarilla”

$$Accuracy = \frac{175 + 195}{175 + 195 + 260 + 43} = 0.55$$

$$Precision = \frac{175}{175 + 260} = 0.40$$

Valor “Mal Uso”

$$Accuracy = \frac{41 + 393}{41 + 393 + 42 + 197} = 0.65$$

$$Precision = \frac{41}{41 + 42} = 0.50$$

Valor “Sin Mascarilla”

$$Accuracy = \frac{96 + 397}{96 + 397 + 59 + 121} = 0.73$$

$$Precision = \frac{96}{96 + 59} = 0.62$$

## 4.2. Validación de la Red Neuronal Convolutacional Particular 2

### 4.2.1 Evaluación de la Matriz de Confusión

En la siguiente figura N°37 se presenta la matriz de confusión de la red neuronal convolucional particular 2, similar al caso anterior se evaluó cada valor para obtener los parámetros “FP,FT,TN,TP”. Para el valor “Con Mascarilla” se obtuvo 209 imágenes clasificadas de manera correcta teniendo un porcentaje de 31.1% de precisión. En el valor “Mal Uso” se obtuvo 206 imágenes clasificadas de forma correcta teniendo un porcentaje de 30.6% de precisión. En el valor “Sin Mascarilla” se obtuvo 194 imágenes clasificadas de manera correcta teniendo un porcentaje de 28.8% de precisión.

La red logra clasificar de manera correcta 609 imágenes, de un total de 673 imágenes de validación. Obteniendo un porcentaje total de clasificación correcta de 90.5% y un porcentaje de 9.5% que comprenden los falsos negativos, falsos positivos y verdaderos negativos para cada valor establecido.

		Confusion Matrix			
Output Class	Con Mascarilla	209 31.1%	28 4.2%	16 2.4%	82.6% 17.4%
	Mal Uso	7 1.0%	206 30.6%	7 1.0%	93.6% 6.4%
	Sin Mascarilla	2 0.3%	4 0.6%	194 28.8%	97.0% 3.0%
		95.9% 4.1%	86.6% 13.4%	89.4% 10.6%	90.5% 9.5%
		Con Mascarilla	Mal Uso	Sin Mascarilla	Target Class

Figura N°37: Matriz de confusión para la red neuronal convolucional particular 2.

Fuente: Captura de pantalla del Matlab.

La siguiente tabla muestra los valores de los verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN).

Tabla N°2: Valores según la matriz de confusión del modelo de red particular 2.

Valor	TP	FP	TN	FN
<b>Con Mascarilla</b>	209	44	411	9
<b>Mal Uso</b>	206	14	421	32
<b>Sin Mascarilla</b>	194	6	450	23

Fuente propia.

#### 4.2.2 Evaluación de Accuracy y Precision

Aplicando las ecuaciones de las métricas mencionada y vistas en el capítulo II, y con los datos obtenidos anteriormente en la matriz de confusión donde podemos apreciar los valores verdadero positivo (TP), falso positivo (FP), verdadero negativo (TN) y falso negativo (FN). Se validó la red neuronal convolucional a través de las métricas Accuracy y Precision, para cada valor de entrada y son mostradas en las siguientes expresiones matemáticas.

Valor “Con Mascarilla”

$$Accuracy = \frac{209 + 411}{209 + 411 + 44 + 9} = 0.92$$

$$Precision = \frac{209}{209 + 44} = 0.83$$

Valor “Mal Uso”

$$Accuracy = \frac{206 + 421}{206 + 421 + 14 + 32} = 0.93$$

$$Precision = \frac{206}{206 + 14} = 0.94$$

Valor “Sin Mascarilla”

$$Accuracy = \frac{194 + 450}{194 + 450 + 6 + 23} = 0.97$$

$$Precision = \frac{194}{194 + 6} = 0.97$$

### 4.3. Validación de la Red Neuronal Convolutiva Particular 3

#### 4.3.1 Evaluación de la Matriz de Confusión

En la siguiente figura N°38 se presenta la matriz de confusión de la red neuronal convolucional particular 3, al igual que los dos casos anteriores se evaluó cada valor. Para el valor “Con Mascarilla” se obtuvo 217 imágenes clasificadas de manera correcta teniendo un porcentaje de 32.2%

de precisión. En el valor “Mal Uso” se obtuvo 234 imágenes clasificadas de forma correcta teniendo un porcentaje de 34.8% de precisión. En el valor “Sin Mascarilla” se obtuvo 217 imágenes clasificadas de manera correcta teniendo un porcentaje de 32.2% de precisión.

La red logra clasificar de manera correcta 668 imágenes, de un total de 673 imágenes de validación. Obteniendo un porcentaje total de clasificación correcta de 99.3% y un porcentaje de 0.7% que comprenden los falsos negativos, falsos positivos y verdaderos negativos para cada valor establecido.

		Confusion Matrix			
Output Class	Con Mascarilla	217 32.2%	4 0.6%	0 0.0%	98.2% 1.8%
	Mal Uso	1 0.1%	234 34.8%	0 0.0%	99.6% 0.4%
	Sin Mascarilla	0 0.0%	0 0.0%	217 32.2%	100% 0.0%
		99.5% 0.5%	98.3% 1.7%	100% 0.0%	99.3% 0.7%
		Target Class			
		Con Mascarilla	Mal Uso	Sin Mascarilla	

Figura N°38: Matriz de confusión para la red neuronal convolucional particular 3.

Fuente: Captura de pantalla del Matlab.

La siguiente tabla muestra los valores de los verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN).

Tabla N°3: Valores según la matriz de confusión del modelo de red particular 1.

Valor	TP	FP	TN	FN
Con Mascarilla	217	4	451	1
Mal Uso	234	1	434	4
Sin Mascarilla	217	0	456	0

Fuente propia.

#### 4.3.2 Evaluación de Accuracy y Presicion

Aplicando las ecuaciones de las métricas mencionada y vistas en el capítulo II, y con los datos obtenidos anteriormente en la matriz de confusión donde podemos apreciar los valores verdadero positivo (TP), falso positivo (FP), verdadero negativo (TN) y falso negativo (FN). Se validó la red neuronal convolucional a través de las métricas Accuracy y Precision, para cada valor de entrada y son mostradas en las siguientes expresiones matemáticas.

Valor “Con Mascarilla”

$$Accuracy = \frac{217 + 451}{217 + 451 + 4 + 1} = 0.99$$

$$Precision = \frac{217}{217 + 4} = 0.98$$

Valor “Mal Uso”

$$Accuracy = \frac{234 + 434}{234 + 434 + 1 + 4} = 0.99$$

$$Precision = \frac{234}{234 + 1} = 0.99$$

Valor “Sin Mascarilla”

$$Accuracy = \frac{217 + 456}{217 + 456 + 0 + 0} = 1$$

$$Precision = \frac{217}{217 + 0} = 1$$

#### 4.4. Comparación de los Modelos de Redes Neuronales Convolucionales

En la tabla N°4 se presenta la comparación de los valores obtenidos en la matriz de confusión para los tres modelos de redes neuronales convolucionales particulares. Se puede apreciar que el modelo particular 3 tiene mayor cantidad de aciertos TP “Verdaderos Positivos”, eso indica que tiene una mejor clasificación de imágenes con relación a los dos modelos restantes.

Tabla N°4: Comparación de los valores obtenidos en la matriz de confusión entre los modelos de redes neuronales convolucionales particulares.

Redes Neuronales Convolucionales	Valores	Parámetros de la Matriz de confusión			
		TP	TN	FP	FN
Modelo Particular 1	Con Mascarilla	175	195	260	43
	Mal Uso	41	393	42	197
	Sin Mascarilla	96	397	59	121
Modelo Particular 2	Con Mascarilla	209	411	44	9
	Mal Uso	206	421	14	32
	Sin Mascarilla	194	450	6	23
Modelo Particular 3	Con Mascarilla	217	451	4	1
	Mal Uso	234	434	1	4
	Sin Mascarilla	217	456	0	0

Fuente propia

En la tabla N°5 se presenta la comparación de los valores obtenidos en la métrica Precisión para los tres modelos de redes neuronales particulares. Se aprecia que el modelo particular 3 obtuvo el mayor promedio de Precisión con relación a los dos modelos restantes

Tabla N°5: Comparación de los valores obtenidos en la métrica Precisión entre los modelos de redes neuronales convolucionales particulares.

Redes Neuronales Convolucionales	Valores	Validación a través de la métrica Precisión	
		Precision	Promedio Total (%)
Modelo Particular 1	Con Mascarilla	0.55	64%

	Mal Uso	0.65	
	Sin Mascarilla	0.73	
Modelo Particular 2	Con Mascarilla	0.92	94%
	Mal Uso	0.93	
	Sin Mascarilla	0.97	
Modelo Particular 3	Con Mascarilla	0.99	99%
	Mal Uso	0.99	
	Sin Mascarilla	1	

Fuente propia.

En la tabla N°6 se presenta la comparación de los valores obtenidos en la métrica Accuracy para los tres modelos de redes neuronales particulares. Se aprecia que el modelo particular 3 obtuvo el mayor promedio de Accuracy con relación a los dos modelos restantes.

Tabla N°6: Comparación de los valores obtenidos en la métrica Accuracy entre los modelos de redes neuronales convolucionales particulares.

Redes Neuronales Convolucionales	Valores	Validación a través de la métrica Accuracy	
		Accuracy	Promedio Total (%)
Modelo Particular 1	Con Mascarilla	0.40	50%
	Mal Uso	0.50	
	Sin Mascarilla	0.62	
Modelo Particular 2	Con Mascarilla	0.83	91%
	Mal Uso	0.94	
	Sin Mascarilla	0.97	

Modelo Particular 3	Con Mascarilla	0.98	99%
	Mal Uso	0.99	
	Sin Mascarilla	1	

Fuente propia.

Estas tablas presentadas permiten comparar y evaluar el desempeño de los tres modelos particulares y así poder conocer cuál es el más apto para detectar mascarillas en tiempo real.

#### 4.5. Prueba Final

Debido a la pandemia ocasionado por el SARS-COV-2(Covid 19), no fue posible realizar la simulación dentro de una Universidad para recrear el ingreso a un laboratorio, por ello se realizó la recreación de un ambiente que simule el ingreso a un laboratorio de una Universidad.

Primero se realizó la instalación del módulo de detección, con la ayuda de un trípode y una Cámara IP, el cual tiene la función de detectar si la persona lleva la mascarilla puesta correctamente, seguidamente se conectó a la laptop para que se enlace con la aplicación App Designer en MATLAB. En las siguientes figuras se muestra la instalación del trípode con la Cámara y la laptop, también se muestra la imagen del App Designer cuando no detecta ninguna persona.



Figura N°39: Instalación del trípode con la cámara.

Fuente: Elaboración propia.



Figura N°40: Conexión de la cámara con el App Designer.  
Fuente: Elaboración propia.



Figura N°41: App Designer sin detectar nada.  
Fuente: Captura de pantalla del Matlab.

Luego de implementar el ambiente para la simulación, se procedió a realizar las pruebas finales para el correcto uso de la mascarilla. Primero se realizó las pruebas cuando una persona usa correctamente la mascarilla, en las siguientes figuras se

muestra como la persona se acerca a la cámara y el resultado se visualizará en la interfaz App Designer.



Figura N°42: Prueba 1.  
Fuente: Elaboración propia.



Figura N°43: Resultado 1.  
Fuente: Captura de pantalla del Matlab.



Figura N°44: Prueba 2.  
Fuente: Elaboración propia.



Figura N°45: Resultado 2.  
Fuente: Captura de pantalla del Matlab.

Después de realizar las pruebas con mascarilla, se realizaron las pruebas con la mascarilla mal puesta, las siguientes figuras muestran el resultado desde que la persona se acerca usando la mascarilla mal puesta y como se visualiza en el App Designer.



Figura N°46: Prueba 3.  
Fuente: Elaboración propia.



Figura N°47: Resultado 3.  
Fuente: Captura de pantalla del Matlab



Figura N°48: Prueba 4.  
Fuente: Elaboración propia.



Figura N°49: Resultado 4.  
Fuente: Captura de pantalla del Matlab.

Finalmente se realizaron las pruebas cuando la persona no tiene puesta la mascarilla, las siguientes imágenes muestran como la persona camina hacia la cámara sin mascarilla y como se visualiza en el App Designer.



Figura N°50: Prueba 5.  
Fuente: Elaboración propia.

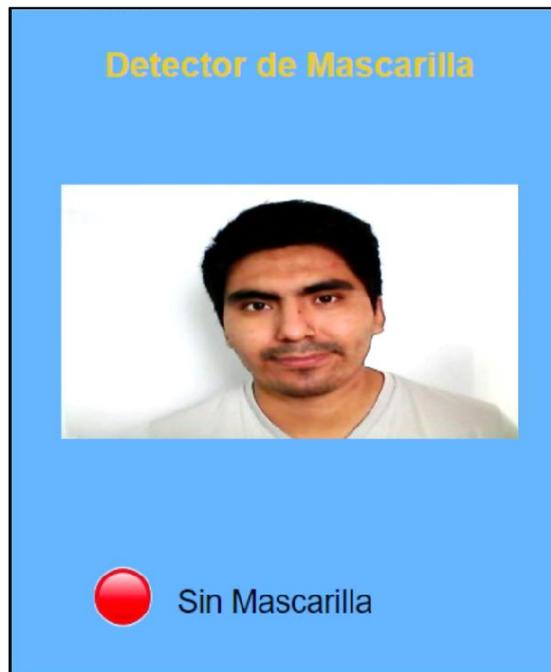


Figura N°51: Resultado 5.  
Fuente: Captura de pantalla del Matlab.



Figura N°52: Prueba 6.  
Fuente: Captura de pantalla del Matlab.



Figura N°53: Resultado 6.  
Fuente: Captura de pantalla del Matlab.

Después de terminar las pruebas correspondientes, se realizaron algunas pruebas adicionales. Las siguientes figuras muestran a la persona con diferentes mascarillas, como también portando un accesorio como es el caso del protector facial y una gorra, para ver cómo responde la red neuronal convolucional.



Figura N°54: Pruebas 7.  
Fuente: Captura de pantalla del Matlab.



Figura N°55: Pruebas 8.  
Fuente: Captura de pantalla del Matlab.

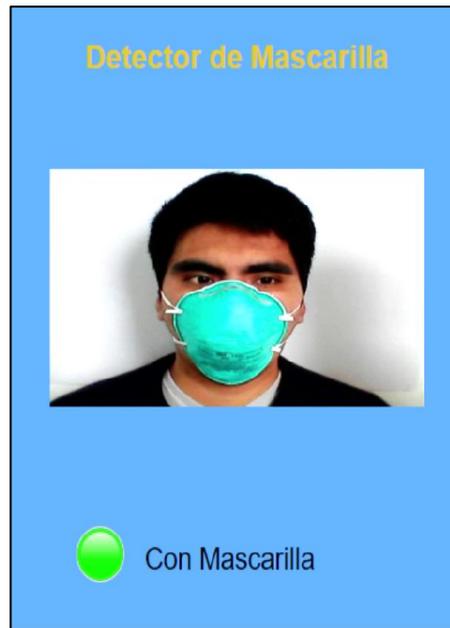
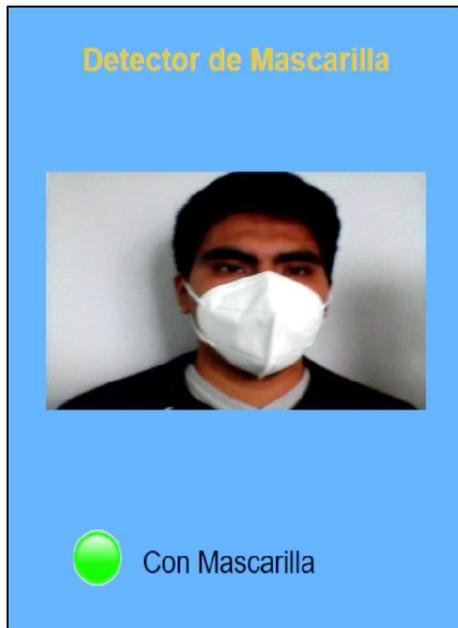


Figura N°56: Pruebas 9.  
Fuente: Captura de pantalla del Matlab.



Figura N°57: Pruebas 10.  
Fuente: Captura de pantalla del Matlab.

#### 4.6 Costo del Proyecto

A continuación, se muestra en la tabla N°7 los costos de los materiales que se utilizaron en la elaboración del proyecto de tesis titulado: “Detección del uso correcto de mascarillas utilizando una red neuronal convolucional para el ingreso de personas a un laboratorio de una Universidad”

Tabla N°7: Costos de materiales del proyecto de tesis.

COSTO DE MATERIALES				
ITEM	DESCRIPCIÓN	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
1	DISPOSITIVO CELULAR	1	S/.0.00	S/.0.00
2	CAMARA IP	1	S/.250.00	S/.250.00
3	SOFTWARE MATLAB	1	S/.0.00	S/.0.00
4	LAPTOP ACER	1	S/.0.00	S/.0.00
5	TRÍPODE PARA LA CÁMARA	1	S/.70.00	S/.70.00
6	COSTO PARA LA SIMULACIÓN	1	S/.50.00	S/.50.00
7	COSTO TOTAL			S/.370.00

Fuente propia.

## CONCLUSIONES

1. Se estableció una base de datos utilizando un dispositivo móvil para la grabación de video, luego se sometió a un procesamiento con la ayuda del software Matlab para así extraer los fotogramas que contienen las imágenes de las personas con mascarillas, sin mascarillas y haciendo mal uso de estas. Seguidamente se estableció una carpeta de nombre “BASE DE DATOS” que contiene tres subcarpetas de nombres “Con Mascarilla”, “Sin Mascarilla”, y “Mal Uso” como se detalla en la figura N°15, teniendo en cuenta que participaron como muestra treinta personas, dando origen a dos mil novecientos veintitrés imágenes; Es decir, treinta y dos imágenes por persona en cada carpeta. Finalmente, se utilizaron dos mil doscientos cincuenta imágenes para el entrenamiento y seiscientos setenta y tres imágenes para la validación.
2. Se implementaron y entrenaron tres modelos de redes neuronales convolucionales particulares con la ayuda de Matlab a través de la herramienta Toolbox Deep Learning como se especifica en la sección 3.4, tomando como referencia el proyecto de tesis de (Tito Apaza & Ramirez Mejía, 2020) y el artículo de (Wei, Jiangjian , Chuanhong , Minmin, & Chengbin , 2017). Los tres modelos de redes neuronales convolucionales, están conformados por la capa de entrada, capas intermedias y la capa de salida. La capa intermedia del modelo de red neuronal convolucional particular 1 está conformado por tres capas especificado en la sección 3.4.1, la capa intermedia del segundo modelo de red neuronal convolucional particular 2 está conformado por cinco capas intermedias como se especifica en la sección 3.4.2. y finalmente la capa intermedia de la red neuronal convolucional particular 3 está conformada por siete capas como se especifica en la sección 3.4.
3. Se compararon y validaron los tres modelos de redes neuronales convolucionales a través de las métricas de medición: Accuracy, Precision y Matriz de confusión. Los tres modelos de redes neuronales convolucionales particulares fueron validados a través de la matriz de confusión obteniendo los parámetros true positive “TP”, true negative “TN”, false positive “FP” y false negative “FN”, estos

valores están especificados en la sección 3.4. Luego dichos parámetros fueron reemplazados en las ecuaciones de Accuracy y Precision especificadas en las expresiones matemáticas (2) y (3). Seguidamente se compararon los resultados especificados en las tablas N°05 y N°06, donde se concluye que el modelo de red neuronal particular 3 es el más efectivo ya que cuenta con un mayor número de capas intermedias teniendo un índice de Precision y Accuracy de 99%, por lo tanto, la red neuronal convolucional particular 3 fue usada para la prueba final especificada en la sección 4.5. Finalmente se implementó una interfaz con la ayuda de la aplicación App Designer en Matlab donde se visualizó el reconocimiento en tiempo real.

Se simuló el ingreso a un laboratorio de una Universidad y se observó el correcto funcionamiento de la red neuronal convolucional particular 3 como se detalla en la sección 4.5, ya que en la interfaz se logró observar la clasificación de forma acertada en el uso correcto de la mascarilla. Por lo tanto, se concluye que se logró comparar y validar el funcionamiento de las redes neuronales convolucionales.

## RECOMENDACIONES

1. Se recomienda que las imágenes elegidas para la base de datos tengan el mismo fondo para cuando se realice la clasificación en tiempo real, ya que si tienen un fondo diferente que involucre caracteres diferentes, no habrá una adecuada clasificación.
2. Se recomienda para la implementación de las redes neuronales convolucionales, tener de entrada imágenes en escala de grises con el fin de minimizar el tiempo de entrenamiento.
3. Para el entrenamiento se recomienda tener grandes cantidades de imágenes posibles, de personas con mascarillas, sin mascarillas y haciendo mal uso de estas, así la red neuronal convolucional podrá clasificar de forma correcta.

## REFERENCIAS BIBLIOGRÁFICAS

- Araujo, A., Pérez, J., & Rodriguez, W. (2018). Aplicacion de una Red Neuronal Convolutacional para el Reconocimiento de Personas a Través de la Voz. (U. d. Andes, Ed.) *In Sexta Conferencia Nacional de Computación, Informática y Sistemas (Vol. 11)*, 77-81.
- Arias Páez, A. S., & Rubiano Venegas, D. A. (2018). Método automático de reconocimiento de voz para la clasificación de vocales al lenguaje de señas colombiano. (*Tesis de licenciatura*). Universidad Católica de Colombia, Bogotá, Colombia.
- Bae, H.-S., Lee, H.-J., & Lee, S.-G. (2016). Voice recognition based on adaptive MFCC and deep learning. (IEEE, Ed.) *IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, 1542-1546.
- Benítez, R., Escudero, G., & Kanaan, S. (2013). *Inteligencia artificial avanzada*. España: UOC.
- Casado Fernández, C. (s.f.). *MANUAL BÁSICO DE MATLAB*. Servicios Informáticos U.C.M.
- Caya Perez, J. C. (2020). *Evaluación de modelos de redes neuronales convolucionales aplicado a radiografías de tórax, para apoyar al proceso de diagnóstico de neumonía asociada al COVID-19*. Lima-Perú.
- Cayllahua Aquino, N. A., & Suárez Macedo, J. C. (2019). *Redes neuronales de aprendizaje profundo para el reconocimiento facial y control de acceso de estudiantes a un laboratorio*. Lima-Perú.
- Cayllahua Aquino, N., & Suárez Macedo, J. (2019). *Redes neuronales de aprendizaje profundo para el reconocimiento facial y control de acceso de estudiantes a un laboratorio*. Lima, Perú.
- Jalomo, J., Preciado, E., & Gudiño, J. (2019). *Diseño de una red neuronal convolutacional para la clasificación de señales*. Mexico,: ECORFAN.

- Mascorro, G. A., & Torres, G. A. (2013). Reconocimiento de voz basado en MFCC, SBC y Espectrogramas. *Ingenius*, (10), 15.
- Mathivet, V. (2019). *Inteligencia Artificial para desarrolladores*. EDICIONES ENI.
- MathWorks. (2021). *Introducing Deep Learning with MATLAB*.
- Nanotec. (27 de Junio de 2016). *La evolución del control de accesos*. Obtenido de Nanotec: <http://www.nanotec.es/la-evolucion-del-control-accesos/>
- Ocas Quiroz, C. E., & Sánchez Pedro, E. J. (2019). Sistema de seguridad para el control de acceso a una vivienda mediante el reconocimiento de voz utilizando coeficientes cepstrum MFCC Y DTW. (*Tesis de licenciatura*). Universidad Nacional de Trujillo, Trujillo, La Libertad, Perú.
- Organización Mundial de la Salud. (2019). *¿Cuál es la diferencia entre aislamiento, cuarentena y distanciamiento?* Obtenido de Organización Mundial de la Salud: <https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses>
- Paul, A. K., Das, D., & Kamal, M. M. (2009). Bangla speech recognition system using LPC and ANN. (IEEE, Ed.) *Seventh International Conference on Advances in Pattern Recognition*, 171-174.
- Real Academia Española. (2001). *Diccionario de la lengua española*. Madrid, España: Academias de la Lengua Española.
- Ren, L., & Ren, Z. (2021). Application of Yolo on Mask Detection Task. *IEEE 13th International Conference on Computer Research and Development (ICCRD)*, (págs. 130-136).
- Rivas, W., & Bertha, M. (2018). *Redes neuronales artificiales aplicadas al reconocimiento de patrones*. Ecuador: UTMACH.
- Rodriguez Castello, D. (2017). Extracción de cráneo en imágenes de resonancia magnética del cerebro utilizando una red neuronal convolucional 3D. (*Tesis de licenciatura*). Universitat Politècnica de Catalunya, Barcelona, España.

- Sahand , A., Haniyeh , A., & Ali Ahmadi, A. (2021). *A Face-Mask Detection Approach based on YOLO*. Tehran-Iran: 26th International Computer Conference.
- sitiobigdata. (19 de Enero de 2019). Obtenido de <https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/>
- T. Norman, A. (2017). *Aprendizaje Automatico en Accion*. Estados Unidos: TEKTIME.
- Tito Apaza, M. R., & Ramirez Mejía, B. (2020). *Reconocimiento automático de placas de rodaje utilizando una red neuronal convolucional para el ingreso de vehículos en la Universidad Ricardo Palma*. Lima, Perú.
- Torres, J. (2016). *Deep learning introducción práctica con Keras*. Barcelona: WATCH THIS SPACE.
- Wei, B., Jiangjian , X., Chuanhong , Z., Minmin, Y., & Chengbin , P. (2017). *A Cascade Framework for Masked Face Detection*. Ningbo, China: IEEE.
- Zambrano Mora, C. A. (2016). MÓDULO DE CONTROL DE PACIENTE, CAPTURANDO DATOS CON EL GLUCÓMETRO UTILIZANDO RASPBERRY. *(Tesis de licenciatura)*. Universidad de Guayaquil Facultad de Ciencias Matemáticas y Físicas Carrera de Ingeniería en Sistemas Computacionales, Guayaquil.

## ANEXOS

Anexo 1: Trípode usado en las pruebas finales, se compró por medio de la página web de Lumingo.



Figura N°58: Trípode.

Fuente: Dispositivo Móvil personal.

Anexo 2: Cámara web usado en las pruebas finales, se compró por medio de la página web de Lumingo.



Figura N°59: Cámara Web.

Fuente: Dispositivo Móvil personal.