ARTÍCULO ORIGINAL INGnosis. 2022;8(1): 01-09

Detección del uso de mascarillas mediante Visión Artificial y Redes Neuronales frente al Covid-19

Detection of facial masks using Artificial Vision and Neural Networks to avoid contagion of Covid-19

Detecção do uso de máscaras por meio de Visão Artificial e Redes Neurais contra Covid-19

Geldres Marchena Teodoro Alberto[1], Córdova Alva Fatima del Rocio[2], Izaga Ruiz Jhoselem Danuska[3], Layza Escobedo Jorge Luis[4], Rodriguez Salirrosas Juleisy Arlette[5], Gavelán Terry Piero Jesús[6],

Resumen

En el presente artículo se planteó como objetivo buscar la detección del uso de la mascarilla en las personas mediante un programa de visión artificial, aplicando el modelo de redes neuronales, donde se identifica si la persona lleva la mascarilla, así buscando un aporte para la mitigación y reducción de casos de contagio del Covid-19. Además, se utiliza el lenguaje Python junto con Frameworks como TensorFlow para la ejecución de las redes neuronales entrenadas y librerías como Keras y Sklearn, utilizadas principalmente en el proceso de aprendizaje. Parte de la metodología se enfocó en la descarga y clonación de materiales, creación y entrenamiento de la red neuronal, la preparación de Anaconda junto con Jupyter Notebook para la verificación del sistema. En los resultados encontramos que la programación detectó correctamente mediante un medio local, usando el Google Colab, para lo cual se tomó como referencia un banco de imágenes preestablecido y se realizó el reconocimiento. Por otra parte, se utiliza Jupyter Notebook para la detección mediante video en tiempo real. Por último, se concluye que se logró detectar dos tipos de imágenes de personas que están usando o no mascarilla con las variables "mask" y "no mask", mediante el entrenamiento de las redes neurales con un batch size de 8, steps de 50 y epochs de 25, con un resultado de classification loss de 0.2120.

Palabras clave: *Mascarilla, Programación, Redes Neuronales, Visión Artificial.*

Abstract

The objective of this article was to seek the detection of the use of the mask in people through an artificial vision program, applying the neural network model, where it is identified if the person wears the mask, thus seeking a contribution to mitigation and reduction of cases of contagion of Covid-19. In addition, the Python language is used together with Frameworks such as TensorFlow for the execution of trained neural networks and libraries such as Keras and Sklearn, used mainly in the learning process. Part of the methodology focused on the download and cloning of materials, creation and training of the neural network, the preparation of Anaconda together with Jupyter Notebook for the verification of the system. In the results we found that the programming correctly detected through a local medium, using Google Colab, for which a pre-established image bank was taken as a reference and the recognition was carried out. On the other hand, Jupyter Notebook is used for real-time video detection. Finally, it is concluded that it was possible to detect two types of images of people who are wearing or not wearing a mask with the variables "mask" and "no_mask", by training neural networks with a batch size of 8, steps of 50 and epochs of 25, with a classification loss result of 0.2120.

Keywords*: Mask, Programming, Neural Networks, Computer Vision.*

Resumo

O objetivo deste artigo foi buscar a detecção do uso da máscara em pessoas através de um programa de visão artificial, aplicando o modelo de rede neural, onde é identificado se a pessoa usa a máscara, buscando assim uma

[1] *Escuela de Ingeniería Industrial. Magister. Universidad Privada del Norte. Trujillo. Perú. ryan.leon@upn.edu.pe.* https://orcid.org/0000-0002-0599-0141
[2] *Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo. Perú. fatima.cordovalva@gmail.com.* https://orcid.org/0000-0001-6727-3976
[3] *Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo. Perú. jhoselemizaga@gmail.com.* https://orcid.org/0000-0003-0990-4316
[4] *Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo. Perú. jorgeluislaes01@gmail.com.* https://orcid.org/0000-0002-3501-6745
[5] *Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo. Perú. arlette.200320@gmail.com.* https://orcid.org/0000-0001-5642-4185
[6] *Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo. Perú. pierojgavelan@gmail.com.* https://orcid.org/0000-0002-8014-0905
Recibido: 20/02/2022 Aceptado: 16/05/2022

contribuição para mitigação e redução da casos de contágio de Covid-19. Além disso, a linguagem Python é usada junto com Frameworks como TensorFlow para a execução de redes neurais treinadas e bibliotecas como Keras e Sklearn, usadas principalmente no processo de aprendizagem. Parte da metodologia se concentrou em baixar e clonar materiais, criar e treinar a rede neural, preparar o Anaconda junto com o Jupyter Notebook para verificação do sistema. Nos resultados constatamos que a programação foi detectada corretamente em meio local, utilizando o Google Colab, para o qual foi tomado como referência um banco de imagens pré-estabelecido e realizado o reconhecimento. Por outro lado, o Jupyter Notebook é usado para detecção de vídeo em tempo real. Por fim, conclui-se que foi possível detectar dois tipos de imagens de pessoas que usam ou não máscara com as variáveis "máscara" e "sem_mascara", por meio do treinamento de redes neurais com tamanho de lote 8, passos de 50 e épocas de 25, com resultado de perda de classificação de 0,2120.

Palavras-chave: *Máscara, Programação, Redes Neurais, Visão Computacional.*

Introducción

En marzo del 2020 la llegada de la pandemia del coronavirus sacudió Latinoamérica, trayendo consigo una profunda crisis económico-social que resultó de la cuarentena obligatoria que se impuso a los pobladores para evitar su propagación, (Pradilla & Márquez, 2021). Y aunque se evidenció el cumplimiento de los protocolos de seguridad por algunos pobladores, existían otros sectores donde ni los casos de contagio en ascenso, ni las muertes que se hacían cada vez mayores lograban hacer comprender las consecuencias del Covid-19. Según el BID (2021), en el mes de noviembre, los contagiados a nivel mundial llegaban a 252,989,299 personas, y en Latinoamérica y el Caribe se tuvo un reporte de 45,093,578 contagiados; mientras que, en Perú el nivel de contagiados llegó a 2,211,366. Esto hace notable que, aunque existan normas que hacen vigentes los protocolos de seguridad, hay personas que las evaden.

Según Gutiérrez, Trujillo, Bedoya & Botero (2021), el coronavirus causante de la infección COVID-19, se denomina coronavirus del síndrome respiratorio agudo severo 2. El virus SARS-Cov-2 es muy contagioso y se transmite rápidamente de persona a persona a través de la tos o secreciones respiratorias, y por contactos cercanos (Maguiña, Gastelo & Tequen, 2020). Asimismo, debido a la evidencia sobre la transmisión por vía aérea, lleva a la política de salud pública a recomendar el uso universal de mascarillas (Ramírez, J. 2020). Es por ello que, ante esta necesidad, ha ocasionado que exista una demanda masiva de mascarillas, teniendo diferentes tipos de ellas, por eso, es importante conocer los aspectos básicos como su correcta identificación, las presentaciones que brinda, ya sea para su uso eficiente, como garantizar un correcto nivel de protección (Márquez, Gaspar, García & Achau, 2020).

Además, es necesario implementar herramientas que nos ayuden a enfrentar la pandemia, siendo una de ellas el uso de la tecnología de Inteligencia Artificial, la cual antes de la propagación de Covid-19 era comúnmente subestimada. La IA sirve como la herramienta más apropiada para predecir y controlar la propagación de la pandemia. El uso de tecnologías de IA como el aprendizaje automático, las aplicaciones de visión por computadoras y el procesamiento del lenguaje natural, que utiliza aplicaciones de big data, son fundamentales bajo la propagación de COVID-19. (AL-Hashimi & Hamdan, 2021). Igualmente, la visión artificial es una alternativa para los sistemas de localización de personas y objetos, así también como en la extracción de características referentes a las actividades humanas. (Leo, Medioni, Trivedi, Kanade, & Farinella, 2017)

En relación a estudios anteriores encontramos a Pereira A., Donadon T. & Oliveira F. (2021) que en su artículo de investigación denominado "Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección" en consideración de la pandemia por Covid 19, desarrolló una aplicación web para el monitoreo de la utilización de mascarillas faciales mediante la técnica de visión por computadora con el uso de framework Flask, en el lenguaje de programación Python, donde se clasifican en rostros con y sin mascarillas. Por lo tanto, como resultado obtienen un nivel de precisión del 63%, un nivel de revocación y f-score del 93% y 75% respectivamente; además, se concluye que es viable la implementación de la aplicación en dispositivos de bajo costo.

Chóez M. & Palma C. (2020) en su investigación "Prototipo Web Detector de Mascarilla mediante RTMP y Visión Artificial para el control dentro del transporte público del Norte de Guayaquil en el tiempo de Pandemia" se desarrolla un prototipo de sistema web "TPMaskDetector" para la detección del uso de mascarillas en los pasajeros dentro de los transportes públicos en la parroquia Tarqui de la ciudad de Guayaquil mediante el reconocimiento de rostro con video transmitido a través de un servicio

RTMP, el cual permite la transmisión de video, además, se utilizó Vue JS, PostgreSQL y RESTFULL para la manipulación de información, base de datos y comunicación entre aplicaciones respectivamente. Por último, como resultado se obtuvo, mediante validaciones de usuarios y juicios de expertos, un nivel de cumplimiento respecto a usuarios finales del 99.88%.

Andrade H., Hidalgo P. & Sinche S. (2021) en su artículo de investigación "Modelo para detectar el uso correcto de mascarillas en tiempo real utilizando redes neuronales convolucionales" desarrolla un modelo de red neuronal convolucional utilizando Tensorflow basado en MobileNetV2, buscando con ello detectar el uso correcto de la mascarilla en tiempo real, dividiendo en 3 categorías "with mask", "without mask" y "bad mask"; por último, obtienen como resultados una exactitud 95% para 10 y 20 interacciones, en tanto a precisión se le atribuye el menor valor a "bad mask" y por parte de exhaustividad, se obtiene el valor más bajo para "without mask" y el mayor valor para "bad mask", siendo lo mismo para el valor de F1.

La importancia del presente proyecto se basa que en la actualidad la pandemia ocasionada por el Covid-19 es uno de los primordiales problemas de salud a nivel mundial y dentro de las principales herramientas que se utilizan para afrontarlo son los elementos de bioseguridad como el uso de las mascarillas en espacios públicos; es por ello, que se busca la detección del uso de la mascarilla en las personas mediante un programa de visión artificial usando redes neuronales, identificando si la persona lleva o no puesta la mascarilla, con lo cual se busca aportar en la mitigación y reducción de casos de contagio del Covid-19.

Material y métodos

En la creación de un sistema de reconocimiento en tiempo real referente a la detección del uso de mascarillas en tiempos de COVID 19, es fundamental tener a disposición un ambiente de trabajo, por lo cual se utiliza Anaconda, que permite ejecutar los comandos que en él se detallan y a su vez que la información y datos a usar en el entrenamiento de redes neuronales tenga un espacio dentro del directorio. Asimismo, es necesario generar la instalación en el computador de ciertos Frameworks compatibles con el lenguaje de programación Python como lo es TensorFlow, el cual proviene de la estructura basada en diagramas de flujos, siendo los nodos, operaciones y los bordes representan tensores, por eso sirve para la ejecución de las redes neuronales entrenadas a base de tensores (Andrade, H. 2021). Además de tener a disposición algunas librerías de programación como Keras y Sklearn, utilizadas principalmente en el proceso de aprendizaje. También, de tener instalado el servidor de Jupyter Notebook y, por último, la programación se realizará mediante Google Colab. Además, se utiliza una red neuronal, parte de la Inteligencia Artificial, para facilitar el aprendizaje del sistema y de esta forma se pueda optimizar la revisión de mascarillas mediante video en tiempo real. Puesto que, la Inteligencia Artificial continuamente tendrá la capacidad de identificar patrones mediante simulaciones con el objetivo de acelerar la obtención de resultados e incrementar la precisión en cada uno de los pronósticos. (Chang, A. 2020)

Clonación y Descarga de Materiales:

La primera parte del entrenamiento abarca la recolección de las imágenes. Una vez recolectadas se crea una carpeta con el nombre de dataset (Montesdeoca, E. 2020). Para ello, es necesario descargar y clonar una carpeta completa de archivos, en la que se encontraban diversos documentos que facilitaban el entrenamiento y construcción de nuestro sistema de detección. En ella, se puede resaltar al dataset de las 877 imágenes que habían sido etiquetadas con los criterios de mask (personas con mascarilla) y No_mask (personas sin mascarilla).

Además, se descargan archivos como Keras, para el aprendizaje autónomo de las redes neuronales; Snapshots como una carpeta para guardar el último archivo de entrenamiento de la red neuronal, classes y annotations_test en donde se guardaban las matrices de Labeling que habían pasado anteriormente las imágenes de la dataset.

Luego, toda esa carpeta de archivos fue cargada a un espacio en la red en un entorno de drive, para su posterior utilización e importación de forma virtual en los demás procedimientos.

Creación y Entrenamiento de Redes Neuronales en Google Colab:

Se inició con la programación de la red neuronal, para lo cual es esencial la instalación de Keras en un entorno virtual, después se procede a importar la carpeta que anteriormente se había cargado, para ello fue necesario permitir todos los accesos requeridos por la plataforma de Google Colab. Cabe resaltar que el lenguaje de programación que se utilizará en el sistema es Python, dado que posee un alto nivel de interpretación, se ejecuta directamente mediante líneas de código y soporta diferentes tipos de programación (Chuquimarca, Pinzón & Rosales, 2021), por lo cual es necesario la instalación de sus módulos y paquetes en la plataforma.

Posteriormente, Python trabaja con un conjunto de librerías esenciales tanto para el procesamiento de imágenes y la clasificación (Ale, N. y Fabián, J. 2019), por ello, se instalaron las librerías de códigos establecidas en forma general en el lenguaje, como nunpy, panda, seaborn, pylot, maltlob y sklearn. orientado al trabajo con imágenes y entrenamiento de modelos. Por último, con la programación necesaria se empezó el entrenamiento de las redes neuronales. Esta empieza con la extracción de las etiquetas de las imágenes alojadas en el dataset para empezar a cargarse al modelo. Para ello, anteriormente es necesario realizar un preprocesamiento de cada una de las imágenes, de las cuales, el tamaño en cada una de las fotografías es de 256 x 256 píxeles. Con esta base de datos, la red neuronal empieza a realizar las iteraciones y aprender a identificar de manera autónoma los dos tipos de etiquetas seleccionadas de “No_Mask” y “Mask”, según sea el caso de la imagen.

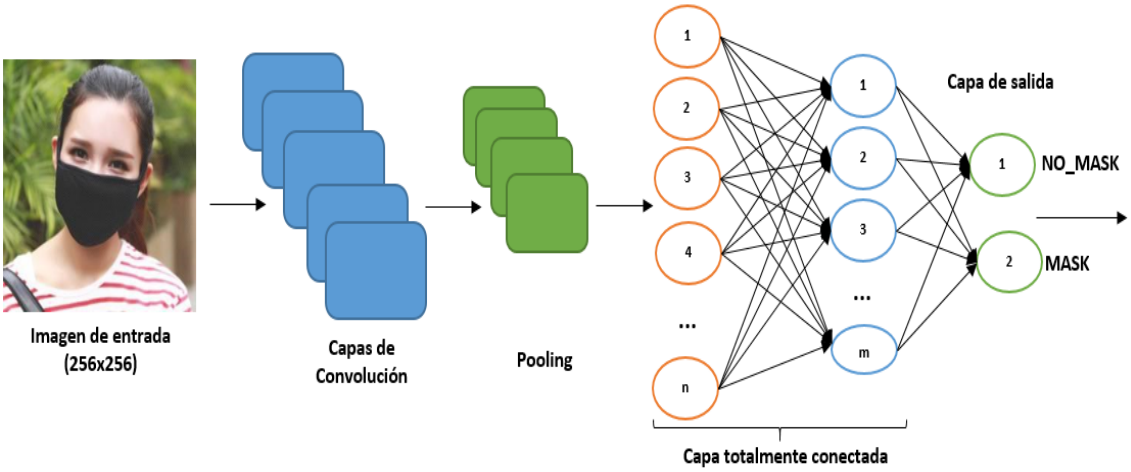

Figura 1. Representación del entrenamiento de la red neuronal
Nota: Elaboración propia.

Además, en esta parte es importante tener en cuenta la cantidad de épocas que se realizan en el entrenamiento, pasos y el tamaño de lote, pues es importante encontrar el número adecuado para que la red neuronal tenga una mejor eficiencia. En esta oportunidad se determinó como batch size de 8, steps de 50 y se analizó con 3 distintas cantidades para los epochs, es así que se consideraron utilizar 60, 30 y 25 para iniciar con el proceso de Deep Learning. Luego, una vez realizadas las pruebas respectivas, se utilizó para el sistema final el dato de 25 epochs, debido a que se obtiene un menor resultado en la función loss.

A su vez, se tuvo en consideración el tiempo de entrenamiento, puesto que, dependiendo la cantidad de épocas utilizadas, esto puede tardar varias horas en entrenar toda la red neuronal y en el transcurso de todo ese tiempo, el computador no puede apagarse por ningún motivo porque todo el entrenamiento volvería desde un inicio. Asimismo, es fundamental mencionar que se debe disponer de una cantidad similar de imágenes para cada una de las clases, que en esta oportunidad fueron un total de 877 fotografías. De las cuales, la mitad de ellas están etiquetadas con la clasificación de “MASK” y la otra mitad pertenecen a “NO_MASK”, de esa forma el entrenamiento de las redes neuronales es equitativo.

Finalmente, después de haber entrenado las redes neuronales, se prueba el sistema de reconocimiento en otro modelo del Google Colab mediante programación y para ello, se tiene que descargar la última época de entrenamiento, en este caso localizado con el nombre de resnet50_cvs_25.h5. Luego, esta se debe cargar a la carpeta drive, específicamente en Snapshots. Seguido, se importa Tensorflow para la

programación, se utiliza igualmente los modelos de Python, y se instala y llama al Keras, así como la importación de librerías y la determinación de los códigos de 0 para "Mask" y 1 para "No_mask". Además, se programa la detección mediante un cuadro de enfoque con color rojo y código RGB de [255,0,0] si el rostro no tiene mascarilla y con color verde, con del código RGB de [0,255,0] si es que el rostro tiene mascarilla. Es así que se ejecutaron los códigos de programación de manera exitosa, y se logró reconocer las imágenes específicas que estaban en la dataset.

Descarga de Carpetas Preparadas e Instalación de Anaconda:

Por otro lado, para el reconocimiento del video en tiempo real, se pasaron a descargar todas las carpetas que antes habían estado siendo modificadas para así empezar a programarlas en un entorno local. Para ello, fue necesario instalar Anaconda, el cual es un software de ejecución de sistemas de reconocimiento, y a su vez se crea un nuevo entorno para el proyecto, el cual se denominó "Automata".

Luego se empezó a trabajar desde la consola de Anaconda, primero con la activación del entorno que se había creado anteriormente, después con la ubicación de la carpeta en donde estaban almacenados todos los archivos y finalmente con la ejecución de Jupyter Notebook para el funcionamiento del sistema.

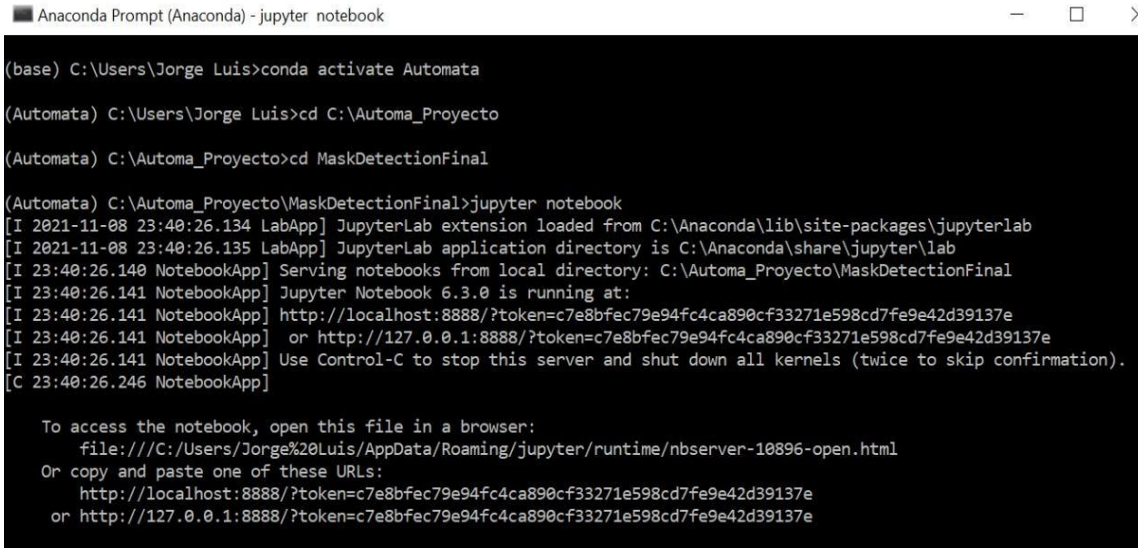

Figura 2. Representación del espacio de trabajo en Anaconda.
Nota: Elaboración propia.

Estando en la interfaz de Jupyter Notebook, se apreciaron los documentos propios de la carpeta del computador y a su vez el archivo denominado "mask_detection_final.ipynb". En este último se encuentra toda la programación hecha para el funcionamiento del sistema de reconocimiento, tanto la instalación de librerías de codificación como para la ejecución del sistema. Es importante ejecutar este archivo primero, pues en él están las librerías que son instaladas por única vez. Y luego, se procede a ejecutar el archivo de "mask_detection_final_Ejecución.ipynb" en donde se encuentra la programación hecha para el funcionamiento del sistema mediante la cámara del computador.

Diferenciación de los Tres tipos de Reconocimiento en el Sistema:

Por último, en la parte final del archivo "mask_detection_final_Ejecución.ipynb" se va a visualizar tres tipos de reconocimiento para el sistema, cada uno con su propia programación. El primero es para identificar imágenes de forma específica, es decir por imágenes de manera individual y solamente se realiza el cambio del nombre de la imagen y nos manda el resultado de esa imagen en específico. El segundo tipo es un reconocimiento completo de todas las imágenes, es decir hace un recorrido total de todas las imágenes del dataset y nos muestra los resultados por cada una de ellas. El tercer tipo es para el reconocimiento en tiempo real a través de la cámara del computador mediante Júpyter Notebook donde se visualiza el enfoque del rostro de la persona y detección de la mascarilla. Todos aquellos caracterizados por contener una función de identificación o reconocimiento y otra de resultados, cuando ya se muestra el recuadro final.

No obstante, el funcionamiento del algoritmo trabaja detectando un blob dentro de cada cuadro del video, que sirve como entradas para la red neuronal de detección de rostros con o sin mascarillas y a la salida se obtiene la ubicación y respuesta según sea el caso señalado con un cuadro de color rojo o verde, que se programó anteriormente y muestra con un porcentaje de probabilidad de la detección.

Figura 3. Distribución de carpetas en Jupyter Notebook.
Nota: Elaboración propia.

Resultados

Luego de realizar la programación, podemos observar los resultados de manera local que nos brinda la programación mediante Google Colab y Jupyter Notebook. Primero observamos que la programación detecta correctamente el uso de las mascarillas en las imágenes.

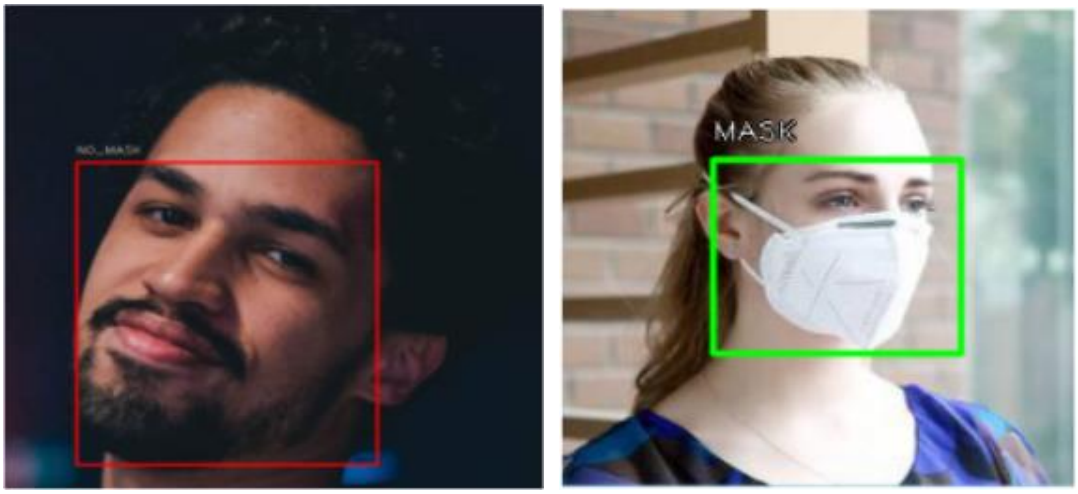

Figura 4. Representación del resultado de “NO_MASK” (izquierda) y “MASK” (derecha)

Las redes neuronales, luego de las iteraciones y del aprendizaje han logrado aprender a diferenciar a personas con mascarilla y sin mascarilla. Para esto se tomó como referencia a las imágenes del banco de las mismas que teníamos previamente en un drive y se pasó a dar inicio al ciclo de reconocimiento. Al hacer esto, como se puede evidenciar en las imágenes, las personas que tienen mascarilla son señaladas con un cuadro verde, que es la confirmación del uso correcto de mascarilla y de cuadro rojo vemos a las personas sin mascarillas.

Por otra parte, se verifica la correcta utilización del sistema mediante video en tiempo real, usando la cámara del computador en la plataforma de Jupyter Notebook. Asimismo, para el rendimiento de las técnicas de visión por computadora es importante obtener una buena calidad de imagen del video proporcionada por el sistema, tomando en cuenta la iluminación, sombra y ausencia de luz. (Niño C., Castro S., & Medina, B. 2020)

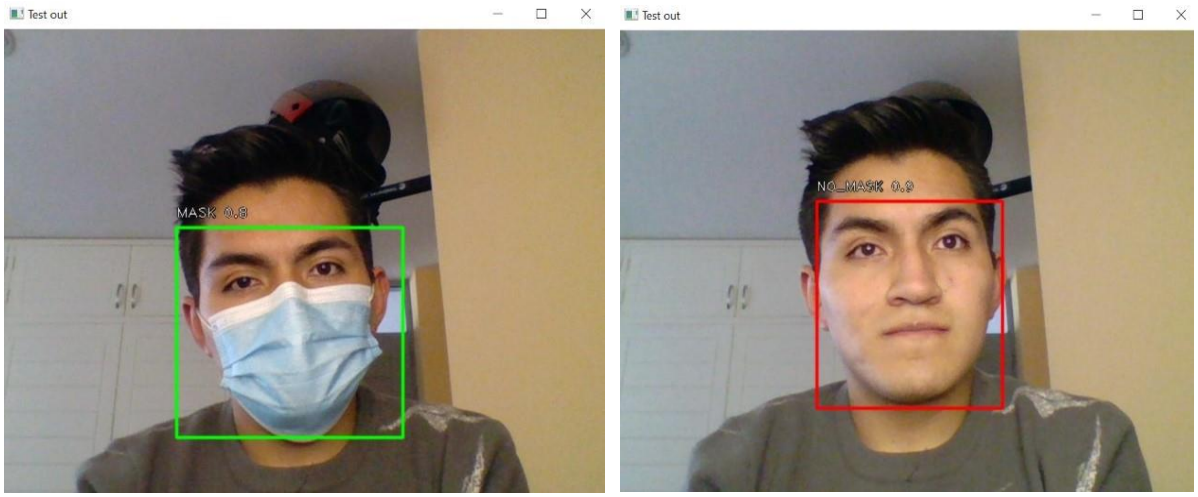

Figura 5. Representación en tiempo real de "MASK" (izquierda) y "NO_MASK" (derecha)

Los resultados obtenidos comprueban que el modelo de la red neuronal es capaz de diferenciar y clasificar según los dos tipos de resultados de rostros con mascarilla y sin mascarilla mediante video. Además, cabe resaltar que el reconocimiento en tiempo real no genera un retraso de tiempo perceptible. Asimismo, se ha tenido presente el criterio de calificación loss como una función que analiza la pérdida del nivel de reconocimiento en las redes neuronales. La que, a su vez indica el margen de error del resultado obtenido y el resultado real que se esperaba obtener, para lo cual, se decidió comparar diversos entrenamientos de redes neuronales variando la cantidad de epochs en cada una (25, 30 y 60) y manteniendo valores fijos de batch size (8) y steps (50). Pues según la teoría, se menciona que mientras más, el criterio loss se acerque a cero, la eficiencia de la red neuronal aumentará. Para luego, aquel entrenamiento ser utilizado en el sistema de reconocimiento.

Obteniendo así, para la época de 60 los resultados finales de la función loss de 1.2402, regression loss de 1.024 y classification loss de 0.2162. En segundo lugar, para el análisis en la iteración de 30 épocas, analizando la última época, resultaron datos de la función loss de 1.3909, regression loss de 1.1187 y classification loss de 0.2719. Mientras que, para el entrenamiento de la red con 25 épocas, se obtuvo como resultados de función loss de 1.2326, regression loss de 1.0205 y classification loss 0.2120. De esta forma, evidenciando según el comportamiento de la red neuronal, se decide utilizar este último entrenamiento para el funcionamiento de nuestro sistema, ya que era más uniforme y sus valores eran más cercanos a cero.

No obstante, para un mayor conocimiento acerca de la eficiencia del sistema de reconocimiento ya creado, se hizo un recorrido de una muestra de 268 imágenes, en la cual, las redes neuronales identificaban si las personas usaban o no mascarilla. Aquella cantidad fue prudente para analizar del total de 877 fotografías que había en la base de datos descargada en la computadora, teniendo en consideración un nivel de confianza de 95% con un margen de error del 5%. Es así que, mediante el interfaz de Jupyter Notebook, se logró determinar una eficiencia del 71.64% sobre el funcionamiento en el sistema, tal como se muestra en la figura 6.

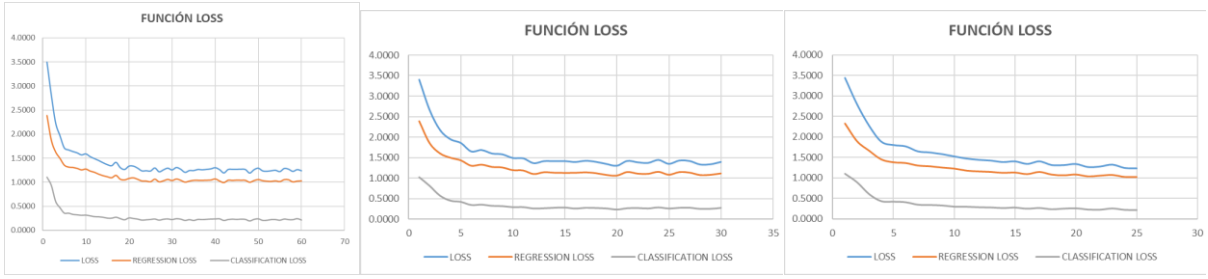

Figura 6. Gráfico función loss para 60 épocas (izquierda), 30 épocas (centro) y 25 épocas (derecha).

Discusión

En consideración con los resultados obtenidos por el sistema y referente al antecedente de Pereira A., Donadon T. & Oliveira F. (2021) con la "Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección" que desarrollaron una aplicación web para el monitoreo de la utilización de mascarillas faciales mediante la técnica de visión por computadora con el uso de framework Flask, en el lenguaje de programación Python, obtuvieron como resultado obtuvieron un nivel de precisión del 63%. Mientras que, en nuestro trabajo se realizó un sistema con red neuronal con reconocimiento en tiempo real y con una eficiencia elaborada a partir de una muestra del total de imágenes de un 71.64% y un valor de classification loss final de 0.2120. Lo que hace que sea bastante aceptable en la implementación para cada una de las empresas a nivel nacional.

Además, sobre el antecedente de Andrade H., Hidalgo P. & Sinche S. (2021) de "Modelo para detectar el uso correcto de mascarillas en tiempo real utilizando redes neuronales convolucionales", utilizan el Tensorflow y MobileNetV2 como sistemas outputs en la presentación de respuestas de las redes neuronales. Por otro lado, en el presente sistema trabajado comprobamos su correcta utilización y ejecución de la red neuronal con Google Colab, Keras, Tensorflow Anaconda y Jupyter Notebook, pues eran sistemas más accesibles que permitían una vinculación adecuada.

Por otra parte, de acuerdo con Chóez M. & Palma C. (2020) en su "Prototipo Web Detector de Mascarilla mediante RTMP y Visión Artificial para el control dentro del transporte público del Norte de Guayaquil en el tiempo de Pandemia" que desarrolla un prototipo de sistema web "TPMaskDetector", podemos mencionar que el proyecto presentado actualmente también es viable para ser utilizado en las diversas industrias como del transporte y empresas comerciales, por ser un sistema de código abierto.

Por último, la peculiaridad de este proyecto en comparación con otras investigaciones anteriores es que se obtuvo un sistema eficiente en más de su 50% en identificación de mascarillas y a su vez fácil de manejar y ser instalado en cada una de las empresas. Además, de que se obtiene la programación lista en interfaces y esta puede ser modificada según las necesidades de otras investigaciones futuras como por ejemplo en la identificación de equipos de seguridad en los trabajadores y en tiempo real a través de una cámara.

Conclusiones

Se comprueba el uso correcto de la programación y una detección del programa utilizando una red neuronal y mediante un video en tiempo real de las personas que usan o no la mascarilla quirúrgica, usando deep learning.

Además, se logró evidenciar en imágenes si las personas están usando o no mascarilla luego del entrenamiento de las redes neurales mediante la programación con batch size de 8, steps de 50 y las distintas iteraciones con epochs de 60, 30 y 25, concluyendo que el menor valor de loss lo brinda el epochs 25. Asimismo, el entrenamiento de la red nos da un resultado de la función loss de 1.2326, regression loss de 1.0205 y classification loss 0.2120, tomando en mayor consideración el valor de la pérdida por clasificación, debido que se utiliza en el presente trabajo la función de clasificación puesto que se predice una etiqueta como "Mask" y "No_Mask".

Entonces, se concluye al respecto de las redes neuronales, que se logró un aprendizaje amplio mediante las distintas iteraciones que se tuvieron a lo largo del proyecto, buscando la mejor precisión al momento de detectar las imágenes en tiempo real o de manera local. También, se obtiene una eficiencia a partir de la muestra de 268 imágenes de 71.64%.

Asimismo, se pudo realizar la diferenciación para las clasificaciones de rostros con y sin mascarilla mediante 3 tipos de reconocimientos en el sistema. Primero, se identificó imágenes de forma específica; segundo, reconocimiento total de todas las imágenes, ambos tipos mediante la plataforma de Google Colab y por último el reconocimiento en tiempo real de manera local mediante Anaconda y Jupyter. Cabe recalcar que cada uno de los reconocimientos cuenta con su propia programación.

Por último, se logró comprender de una manera más amplia el lenguaje de programación Phyton y la plataforma de Google Colab que fue utilizada para la programación y donde se importaron distintas librerías y Frameworks que fueron utilizadas.

Referencias

Chang, A. C. (2020). Artificial intelligence and COVID-19: Present state and future vision. Intelligence-Based Medicine: Reported de una investigation, 3, 100012.

Ramírez-Guerrero, J. (2021). The importance of nonmedical face masks in the general population during the COVID-19 pandemic. A narrative review. Revista Mexicana de Anestesiología, 44(2), 130–138.

Ale, NA y Fabián, J. (2019). Detección del estado fisiológico de los ojos en conductores mediante técnicas de visión artificial: Técnicas de visión por computadora para la detección del estado fisiológico de los ojos conductores. INGENIARE - Revista Chilena de Ingeniería, 27 (4), 564–572.

Oliveira-Teixeira, F., Donadon-Homem, T. P., & Pereira-Junior, A. (2021). Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección. Revista Científica General José María Córdova, 19(33), 205-222.

BID-Banco Interamericano de Desarrollo (2021). COVID-19: Reporte Situacional. (Citado el 03 de noviembre del 2021) ElBID. 4-48. Recuperado de: https://www.iadb.org/es/coronavirus/current-situation-pandemic

Andrade Carrera, H., Sinche Maita, S., e Hidalgo Lascano, P. (2021). Modelo para detectar el uso correcto de mascarillas en tiempo real utilizando redes neuronales convolucionales. Revista de Investigación en Tecnologías de La Información, 9 (17), 111–120.

Chóez Vera, MN y Palma Toledo, CM (2021). Prototipo Web detector de mascarilla mediante RTMP y visión artificial para el control dentro del transporte público del norte de Guayaquil en tiempo de pandemia. Tesis de Licenciatura. Universidad de Guayaquil, Guayaquil, Ecuador.

Montesdeoca Ordoñez, ED (2020). Implementación de un sistema de reconocimiento del uso de mascarillas como medida de precaución contra el covid19 usando deep learning. Trabajo de Titulación de Ingeniería de Sistemas. Universidad Técnica de Machala, Machala, Ecuador.

Niño Rondón, C. V., Castro Casadiego, S. A., & Medina Delgado, B. (2020). CARACTERIZACIÓN PARA LA UBICACIÓN EN LA CAPTURA DE VIDEO APLICADO A TÉCNICAS DE VISIÓN ARTIFICIAL EN LA DETECCIÓN DE PERSONAS. Revista Colombiana de Tecnologías de Avanzada, 2(36), 83-88.

Maguiña Vargas, C., Gastelo Acosta, R., & Tequen Bernilla, A. (2020). El nuevo Coronavirus y la pandemia del Covid-19. Revista Médica Herediana, 31(2), 125-131.

Márquez Peiró, J., Gaspar Carreño, M., García Cases, S., & Achau Muñoz, R. (2020). "Mascarillas: producto imprescindible en la pandemia COVID-19". Revista OFIL – ILAPHAR 2020, 30(3); 189-191.

Pradilla Cobos, E., & Márquez López, L. (2021). "Las ciudades latinoamericanas y el coronavirus". Sao Paulo. Revista Cadernos Metrópole 23 (52)

AL-Hashimi, M. y Hamdan, A. (2021). Las aplicaciones de la inteligencia artificial para controlar COVID-19: Estudios en sistemas, decisión y control. SpringerLink, 55–75

Chuquimarca Jiménez, L.; Pinzón Tituana, S.; Rosales Pincay, A. (2021). Detección de Mascarilla para COVID-19 a través de Aprendizaje Profundo usando OpenCV y Cascade Trainer GUI. Revista Científica y Tecnológica UPSE, 8 (1) pág. 68-73.

Andrade Carrera, H. E. (2021). Desarrollo de un sistema de control de acceso en base a detección de temperatura corporal y al correcto uso de mascarillas en tiempo real. Tesis para obtención de Título. Universidad Politécnica Nacional, Quito, Ecuador.

Gutiérrez J., Trujillo D., Bedoya A. & Botero A (2021) Infección por COVID-19 previo al inicio de la epidemia en Colombia. Revista Infectio 2021; 25(4): 296-299.

Leo, M., Medioni, G., Trivedi, M., Kanade, T., & Farinella, G. M. (2017). Computer vision for assistive technologies: Computer Vision and Image Understanding. ElSevier, 154, 1–15.