Development of a face detection system with or without a mask using software

León León, Ryan Abraham[1]; Arroyo Anticona, Danner Enrique[2]; Carranza Albites, Rebecca[3]; Lozano Castañeda, Roger[4]; Terrones Pinedo, Claudia[5]

[1.] Universidad Privada del Norte (UPN), Trujillo, Perú, ryan.leon@upn.edu.pe
[2.] Universidad Privada del Norte (UPN), Trujillo, Perú, arroyoenrique99@gmail.com
[3.] Universidad Privada del Norte (UPN), Trujillo, Perú, april.carranza92@gmail.com
[4.] Universidad Privada del Norte (UPN), Trujillo, Perú, RogerL_04@Hotmail.com
[5.] Universidad Privada del Norte (UPN), Trujillo, Perú, *claudia_2000beatriz@hotmail.com*

Abstract*– This article presents a software with artificial vision using the Python language in Visual Studio Code which is capable of detecting if a person is wearing a mask or not, this project was made in order to be able to identify people who are without masks or misplaced either at work or at the entrance of a premises, for example this would be applied in companies that work with toxic waste, gases, dust and any agent that is transmitted by air, thanks to this it will be possible to identify certain people and give them a feedback on the correct use of the mask. For the development of this software, the main modules such as opencv-contrib-python, numpy and mediapipe were used, in addition to having their dependencies which are installed automatically, as well as a camera with which to capture the images; The first step was to search for a data set and resize it, followed by training it using opencv and finally using mediapipe for face detection and implementing the model in programming. Finally, after the tests carried out, a result of 83.13% validation was obtained, however, its percentage would increase or decrease depending on the quality of the camera, since the errors would decrease if you have a camera with a good quality implemented.

Keywords: Software; computer vision; Python; Visual Studio Code; modules

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2022.1.1.753
ISBN: 978-628-95207-0-5 **ISSN:** 2414-6390

Desarrollo de un sistema de detección de rostro con o sin mascarilla mediante el uso de software
Development of a face detection system with or without a mask using software

León León, Ryan Abraham[1]; Arroyo Anticona, Danner Enrique[2]; Carranza Albites, Rebecca[3]; Lozano Castañeda, Roger[4]; Terrones Pinedo, Claudia[5]

[1.] Universidad Privada del Norte (UPN), Trujillo, Perú, ryan.leon@upn.edu.pe
[2.] Universidad Privada del Norte (UPN), Trujillo, Perú, arroyoenrique99@gmail.com
[3.] Universidad Privada del Norte (UPN), Trujillo, Perú, april.carranza92@gmail.com
[4.] Universidad Privada del Norte (UPN), Trujillo, Perú, RogerL_04@Hotmail.com
[5.] Universidad Privada del Norte (UPN), Trujillo, Perú, *claudia_2000beatriz@hotmail.com*

Resumen– En este artículo se presenta un software con visión artificial usando el lenguaje Python en Visual Studio Code el cual es capaz de detectar si una persona lleva mascarilla o no, este proyectose hizo con el fin de poder identificar a las personas que estén sin mascarillas o mal puestas ya sea en el trabajo o entrada de un local, por ejemplo esta se aplicaría en empresas que trabajen con residuos tóxicos, gases, polvo y toda agente que se transmita por aire, graciasa esto se podrá identificar a ciertas personas y darles una retroalimentación sobre el uso correcto de la mascarilla. Para el desarrollo de este software, se usaron los principales módulos como opencv-contrib-python, numpy y mediapipe, además de contar con sus dependencias las cuales se instalan automáticamente, además de una cámara con la cual captar las imágenes; el primer paso fue el de buscar un data set y redimensionarlo, seguido a esto entrenarlo mediante opencv y por último usar mediapipe para la detección del rostro e implementar el modelo en la programación. Finalmente, después de las pruebas realizadas se obtuvo un resultado del 83.13% de validación, sin embargo, su porcentaje aumentaría o disminuiría dependiendo de la calidad de la cámara, ya que los errores disminuirían si se tiene una cámara con una buena calidad implementada.

Palabras clave: Software; visión artificial; Python; Visual Studio Code; módulos.

I. INTRODUCCIÓN

A finales del año 2019 en la ciudad de Wuhan, China, se registraron los primeros casos de una enfermedad aún no conocida en el mundo, comenzó a propagarse rápidamente desconcertando a los expertos en salud. Recibió el nombre de SARS-CoV-2. Identificado como parte de la familia de los coronavirus, causante de la nueva enfermedad llamada COVID-19, cuya propagación por el mundo ha provocado un estado de emergencia de salud pública de importancia internacional, caracterizada por la OMS como una pandemia. Las personas afectadas presentaban tos, estornudos, fiebre y en el peor de los casos, una neumonía grave que provocaba insuficiencia respiratoria aguda [1]. La transmisión del virus es muy alta; ocurre de una persona a otra a través del tacto, saliva, estornudos, tos, flemas, objetos o superficies contaminadas. Según la Organización Mundial de la Salud (OMS), los cuidados básicos de higiene personal pueden reducir la tasa de trasmisión del virus, como la higiene de manos con alcohol y gel, además del uso de equipos de protección personal, como mascarillas quirúrgicas (N95) o mascarillas hechas a mano. Los estudios demuestran que las mascarillas tienen la capacidad y el efecto de bloquear entre un 95,15 % y un 99,98 % de aerosoles [2-5]. Por lo que las empresas deben implementar y brindar los EPP's necesarios al personal de trabajo y según las labores a realizar, el entorno de trabajo y el grado de exposición, se debe adecuar el EPP adecuado para cada situación [6-7]. La inteligencia artificial ha permitido que el ser humano sea capaz de crear y entrenar redes neuronales capaces de pensar por sí misma y realizar la toma de decisiones, en el caso de este proyecto se utilizó un algoritmo que se basa en un modelo LBPH que realizan la función de detectar qué persona está utilizando mascarilla y quien no. Con los algoritmos disponibles e infinidad librerías que nos ofrece Python y opencv [8-10]. La inteligencia artificial es "simplemente" la transferencia de la inteligencia a las máquinas, es decir que el computador haga

cosas inteligentes. Las técnicas de inteligencia artificial son capaces tanto de sustituir elementos tradicionales como de realizar tareas que hasta el momento era imposible abordar sin la presencia del operador humano [11-13]. Una herramienta que utiliza Big Data y se basa en redes neuronales, para la detección e identificar enfermedades graves como el cáncer de mama. El algoritmo solo realiza un diagnóstico correcto, sino que, además, valora la gravedad de cada caso y recomienda el tratamiento más adecuado [14-16]. La inteligencia artificial es y será una herramienta económica, eficaz y eficiente que aportará soluciones para problemas de ingeniería, así como pueden ser aplicados en otros campos debido a su avance tecnológico de las últimas décadas el cual permite tomar mejores decisiones con respecto a los métodos ya tradicionales y obtener excelentes resultados [17-20].

La presente investigación se enfocará en realizar un sistema de verificación de uso de mascarillas mediante la inteligencia artificial, la cual, permitirá detectar el uso apropiado de mascarillas en las personas. Esto se debe a la situación actual que está atravesando el Perú frente al COVID –19, dicho sistema será implementado mediante un software en tiendas o industrias del país, esto permitirá reducir los costos en empresas que requieran de su uso, así mismo reducirán el contagio y se generara un mejor control.

II. MARCO TEORICO

Esta investigación se fundamenta en el entrenamiento de clasificación con el método cascada, teniendo como modulo principal el OPEN CV y su software MEDIAPIPE. El uso del clasificador en cascada incluye dos etapas principales: etapa de entrenamiento y etapa de detección. Se ha utilizado el modelo basado en LBPH (Local Binary Patterns Histograms) en la fase de detección. Esto permite la detección de los rostros con mascarilla y sin mascarilla. La detección de objetos usando un clasificador en cascada basado en características de Haar es un método efectivo de detección de objetos propuesto en 2001 por Paul Viola y Michael Jones en su artículo "Detección rápida de objetos usando cascada mejorada de características simples"[21]. Este es un método basado en el aprendizaje automático en el que se entrena una función en cascada a partir de muchas imágenes positivas y negativas. Luego se usa para detectar objetos en otras imágenes.

III. MATERIALES Y METODOS

3.1. Recursos Utilizados

Los recursos para la elaboración del software es una laptop o un pc, con un procesador Intel i3 o Ryzen 3 y como sistema principal se usa Windows desde la versión 7 hasta la actual, la cual es la versión 11.

Para la fase de detección se utilizó el modelo LBPH, el cual afecta en la tasa de acierto, positiva o negativa, según el tamaño de la imagen a mostrar, realiza comparaciones pixeles a pixeles, por ello si la imagen es pequeña, la identificación es menor y no obtiene mucha información a comparación de una imagen grande. El modelo LBPH utilizado, viene del modelo LBP (Local Binary Patterns) [22]. Una descripción más formal del operador LBP se puede dar como:

$$LBP(x_C, y_C) = \sum_{p=0}^{P-1} 2^p \, s(i_p - i_C)$$

Con (x_C, y_C) como el píxel central con intensidad i_C; e i_{n} la intensidad del píxel vecino. S es la función signo definida como:

$$S(x) = \begin{cases} 1 \; si \; x \geq 0 \\ 0 \quad e.o.c. \end{cases}.$$

Esta descripción permite capturar detalles con mucha precisión en las imágenes. Para un punto (x_C, y_C), la posición de su vecino (x_p, y_p), $p \in P$, puede ser calculado como:

$$x_p = x_C + R\cos(\frac{2\pi p}{P})$$

$$y_p = y_C - R\sin(\frac{2\pi p}{P})$$

Donde R es el radio del círculo que forman los vecinos y P es el número de puntos de muestra.

El operador es una extensión de la codificación original LBP, por eso a veces es llamado LBP extendido. Si los puntos coordenados en el círculo no corresponden a las coordenadas de la imagen, el punto debe ser interpolado. OpenCV implementa una interpolación bilineal:

$$f(x,y) \approx [1 - xx] \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

Por definición el operador LBP es robusto frente a transformaciones en escala de grises monótonas.

Finalmente se añade la información espacial en el modelo de identificación facial. Por ende, se dividir la imagen LBP en m regiones locales y extraer la información del histograma de cada una. Estos histogramas son llamados como Local Binary Pattern Histogram (LBPH).

El software que se utilizó fue software Microsoft Visual Studio Code 1.62.3, el cual es un editor de código que ofrece un sistema intuitivo y fácil de usar, donde se puede utilizar diferentes tipos de lenguaje [23]. Se instalaron las extensiones de Python el cual es un lenguaje de programación fácil de aprender que es utilizado en diversas plataformas y sistemas operativos [24] y Pylance el cual proporciona experiencia de edición más eficaz y fácil para el uso de Python [25]. Estas dos extensiones son de ayuda para la edición de código.

El lenguaje que se utilizó es Python 3 el cual es una versión actualizada nos ofrece un soporte en la orientación a objetos, programación interpretativa y la programación funcional [26].

Los módulos principales utilizados son: opencv-contrib-python, es un software libre y de código abierto enfocado en visión artificial [27]; numpy, donde este es un software de análisis numérico, permitiendo realizar vectores y matrices multidimensionales [28]; y mediapipe, es un software de

código abierto que permite analizar los componentes de archivos de video y pueden ser en vivo o grabados [29]. Además, automáticamente se instalaron los siguientes módulos para un óptimo funcionamiento: absl-py, attrs, cycler, kiwisolver, matplotlib, Pillow, protobuf, pyparsing, python-dateutil y six.

3.2. Metodología Instalación de módulos

Instalación de módulos

La instalación de estos módulos se realiza desde cmd, donde se usa el comando "pip install (nombre del módulo)" para instalarlos uno a uno, en nuestro caso se usó un archivo de texto en el cual están escritos los nombres y versiones de los módulos a utilizar, sin embargo, el comando cambia a "pip install -r (nombre del archivo de texte.txt)". Para esto se usan los comandos para ir a la carpeta como por ejemplo "d:" y "cd (nombre de la carpeta)" para navegar por las carpetas.

En la Fig. 1 se puede observar la serie de comandos para la instalación de las librerías.

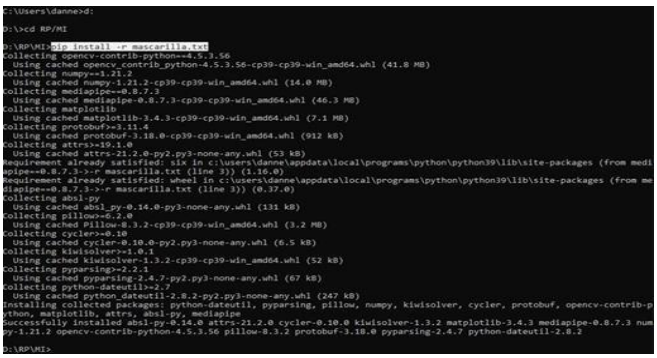

Fig. 1 Instalación de módulos en CMD.

DATASET

Primero se utilizó un dataset el cual se usó para hacer un entrenamiento haciendo uso de un algoritmo, el cual se basa en un modelo LBPH. Para realizar esta prueba se usó un recopilatorio de KAGGLE, del cual se usó 385 imágenes con personas con mascarillas y 385 de personas sin mascarillas, con un tamaño de 72 pixeles de alto y 72 pixeles de ancho.

El proceso para la redimensión de las imágenes, para el desarrollo de este software se usan las librerías de "os" el cual nos permitirá navegar por nuestro sistema operativo y establecer rutas de guardado y "open cv" el cual nos permitirá modificar una gran cantidad de imágenes en poco tiempo. (Ver Fig. 2).

REDIMENSIÓN DE LAS IMAGENES

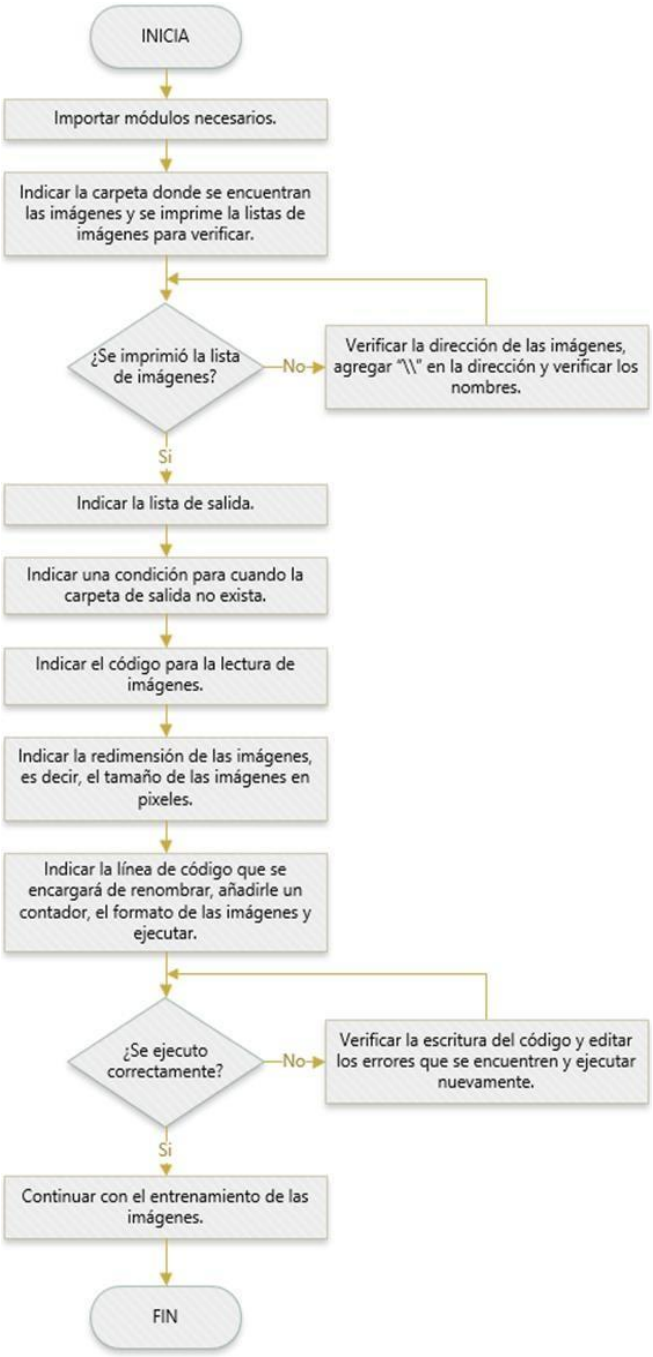

Fig. 2 Flujograma de la programación para redimensionar imágenes.

El entrenamiento se desarrolla con la importación de la librería "os" para establecer la ruta en la cual se encuentren las imágenes redimensionadas, seguido a esto se usó "opencv" para la modificación de las imágenes a escala de grises y el entrenamiento en cascada el cual se realiza automáticamente usando muestras positivas y negativas usando el modelo LBPH, también se usa "numpy" el cual se encarga de clasificar las imágenes en etiquetas y realizar los cálculos del modelo, y finalmente se importa time para indicar el tiempo restante para la finalización del entrenamiento. (Ver Fig. 3).

ENTRENAMIENTO DEL MODELO

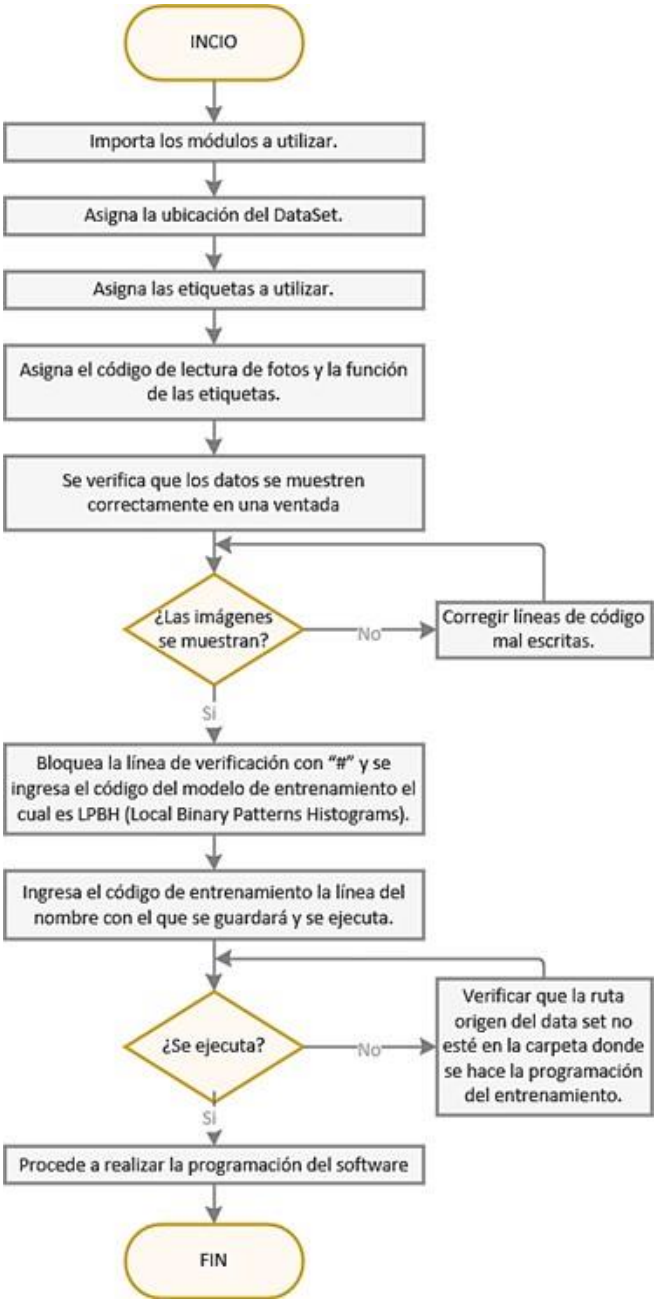

Fig. 3 Flujograma de la programación del entrenamiento del modelo.

El desarrollo del software se realiza importando la librería "opencv" el cual nos permite modificar la cámara de video y establecerlo en escala de grises y por último se importa la librería "mediapipe" el cual facilita la detección de rostros y aplicar el modelo. (Ver Fig. 4).

DESARROLLO DEL SOFTWARE

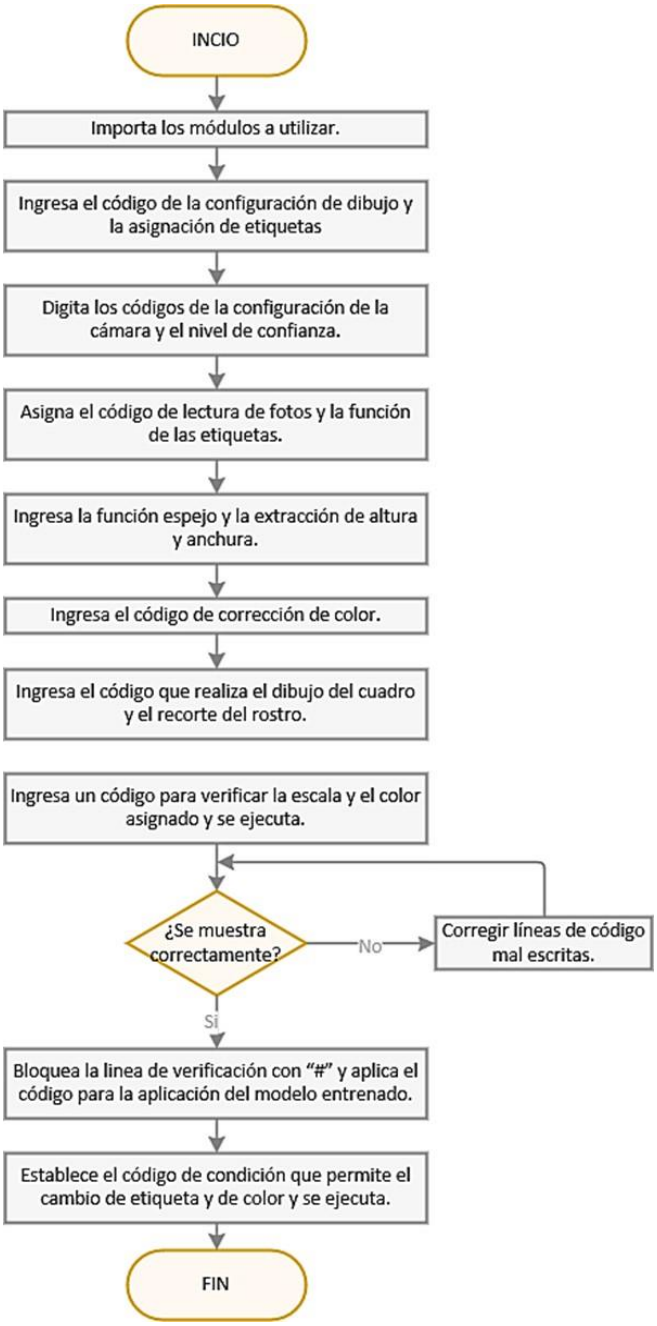

Fig. 4 Flujograma de la programación del software de detección de mascarilla.

TOMA DE RESULTADOS

En la Fig. 5 se puede observar el procedimiento que se siguepara registrar los resultados obtenidos de las pruebas

Fig. 5 Procedimiento de los resultados.

Finalmente se observa el funcionamiento del software reconociendo si una persona lleva mascarilla o no. (Ver fig.6).

Fig. 6 Pruebas realizadas a los integrantes de grupo.

IV. RESULTADOS

Se realizó una serie de 40 pruebas por cada integrante del equipo, 20 con mascarilla y 20 mal o sin mascarilla, se obtuvieron como resultados una cantidad de veces que el sistema ha dado un resultado correcto e incorrecto, los resultados obtenidos varían debido a los diferentes factores que cada integrante presentaba, como calidad de cámara para realizar las pruebas e iluminación o ambiente en el cual no se podía enfocar correctamente el rostro de la persona. Por ello el sujeto 2 presente una mayor cantidad de pruebas incorrectas de "mal o sin mascarilla" debido a que en su momento no presentaba una buena iluminación en el ambiente que realizo las pruebas, pese al resultado se obtuvo un alto porcentaje de validación que es un 83.13%. (Ver tabla 1).

TABLA 1

PUNTAJE DE PRUEBAS Y PORCENTAJE DE VALIDACIÓN

SUJETOS	Con Mascarilla (Correcto)	Con mascarilla (Incorrecto)	Mal o sin mascarilla (Correcto)	Mal o sin mascarilla (Incorrecto)
Sujeto 1	13	7	18	2
Sujeto 2	17	3	13	7
Sujeto 3	19	1	19	1
Sujeto 4	15	5	19	1
Total	64	16	69	11
%	80.00%		86.25%	
% Total	83.13%			

Sumado a lo ya mencionado, se observa su porcentaje de validación que obtuvo cada sujeto con las 40 pruebas realizadas, 20 con mascarilla y 20 mal o sin mascarilla. (Ver Fig. 7)

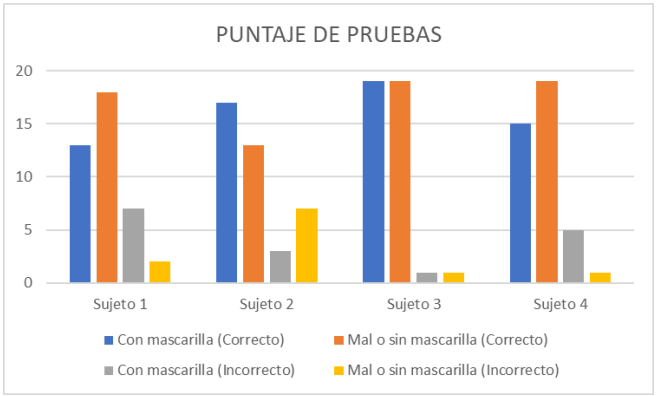

Fig. 7 Puntaje de pruebas.

V. DISCUSIÓN

En el artículo de investigación "Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección" [31]. En el cual implementaron un sistema de detección de mascarillas, en el cual utilizaron una base total de 194 imágenes con el cual obtuvieron un 63% de precisión. Mientras que nuestro porcentaje total de validación es del 83.13%, el cual fue obtenido de un total de 160 pruebas de 4 personas.

En el trabajo de investigación "Implementación De Un Sistema De Reconocimiento Del Uso De Mascarillas Como Medida De Precaución Contra El Covid19 Usando Deep Learning" [8]. Realizo 4 modelos de pruebas de las cuales una de ellas dio resultados muy bajos mostrando una gran diferencia con sus otros modelos. Los resultados en su precisión con mascarilla promedio de las pruebas realizadas muestran un 87% y sin mascarilla un75%, mientras que en nuestras pruebas realizadas de todos los sujetos los resultados no muestran mucha diferencia obtenemos en promedio resultados del 80% con mascarilla y 86.25% sin mascarilla o mal puesta, dando así una precisión general del 81% y en nuestra investigación el 83.13%. Es por ello que, así como se afirma en el trabajo anterior en sus conclusiones y el nuestro, damos a conocer que mucho influye los diferentes factores en cuanto a la toma de detección de rostros como es la iluminación del entorno, la calidad de las cámaras y el tipo de procesamiento de imágenes en la programación.

TABLA 2

TABLA COMPARATIVA DE ESTRATEGIAS EMPLEADAS

Desarrollo de un sistema de detección de rostro con o sin mascarilla mediante el uso de software	**Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección**	**Implementación De Un Sistema De Reconocimiento Del Uso De Mascarillas Como Medida De Precaución Contra El Covid19 Usando Deep Learning**
Se utilizó un algoritmo LBPH	No aplicaron una red neuronal	Crearon una red neuronal para su utilidad
Se utilizó un dataset prediseñado.	Utilizaron un modelo previamente entrenado de algoritmo Haar Cascade, aplicando 194 imágenes.	Creación de un dataset.
Se implementó un algoritmo clasificador Haar Cascade, con 770 fotografías.		Utilizaron un algoritmo Deep Learning, con 1500 fotografías.

VI. CONCLUSIÓN

Se logro realizar la investigación necesaria para realizar un sistema de verificación del uso de mascarillas mediante la inteligencia artificial.

Así mismo se logró crear un prototipo del programa de detección de mascarillas, el cual permitirá detectar el correcto uso de las mascarillas en las personas.

El prototipo de detección de mascarillas funciono de manera correcta al detectar si la persona cuenta o no con mascarillas, con ellos nos permite reducir los casos de contagios en empresas al tener un constante monitoreo de las personas que ingresas a dicha empresa.

El proyecto es de mucha utilidad para la situación actual que se afronta a nivel mundial, ya que sirve para prevenir contagios al detectar las mascarillas, además que este tiene un porcentaje de validación por encima del 80% con la mascarilla puesta y de un 86% con la mascarilla mal puesta o sin usar, además de que su porcentaje total de validación para ambos casos es del 83.13%.

VII. REFERENCIAS

[1] PEREIRA JÚNIOR, Alexandre; DONADON HOMEM, Thiago Pedro; OLIVEIRA TEIXEIRA, Fabio. Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección. Revista Científica General José María Córdova, 2021, vol. 19, no 33, p. 205-222. Disponible: https://www.redalyc.org/journal/4762/476268269010/

[2] GUEVARA, Marta Lucía; ECHEVERRY, Julian David; URUEÑA, William Ardila. Detección de rostros en imágenes digitales usando clasificadores en cascada. Scientia et technica, 2008, vol. 1, no 38. Disponible: https://www.redalyc.org/pdf/849/84903801.pdf

[3] BRAVO, Cristián J.; RAMÍREZ, Patricio E.; ARENAS, Jorge. Aceptación del reconocimiento facial como medida de vigilancia y seguridad: Un estudio empírico en Chile. Información tecnológica, 2018, vol. 29, no 2, p. 115-122., doi: http://dx.doi.org/10.4067/S0718- 07642018000200115

[4] MÁRQUEZ DÍAZ, Jairo. Inteligencia artificial y Big Data como soluciones frente a la COVID-19. Revista de bioética y derecho, 2020,no 50, p. 315-331. Disponible: https://scielo.isciii.es/scielo.php?pid=S1886-58872020000300019&script=sci_arttext&tlng=en

[5] ADE, Carl J., et al. Does wearing a facemask decrease arterial blood oxygenation and impair exercise tolerance? Respiratory Physiology & Neurobiology, 2021, vol. 294, p. 103765, doi: https://doi.org/10.1016/j.resp.2021.103765.

[6] RARAZ-VIDAL, Jarvis Giusseppe, et al. Condiciones laborales y equipos de protección personal contra el Covid-19 en personal de salud,Lima-Perú. Revista de la Facultad de Medicina Humana, 2021, vol. 21, no 2, p. 335-345, doi: https://doi.org/10.25176/RFMH.v21i2.3608

[7] ANDRÉS, Jesús M. Aranaz, et al. Mascarillas como equipo de protección individual durante la pandemia de COVID-19: cómo, cuándo y cuáles deben utilizarse. Journal of Healthcare Quality Research, 2020, vol. 35, no 4, p. 245-252, doi: https://doi.org/10.1016/j.jhqr.2020.06.001

[8] MONTESDEOCA ORDOÑEZ, Erik Daniel. Implementación de un sistema de reconocimiento del uso de mascarillas como medida de precaución contra el covid19 usando deep learning. 2020. Tesis Doctoral. Tesis de Licenciatura. Machala: Universidad Técnica de

Machala. Disponible: http://repositorio.utmachala.edu.ec/handle/48000/15890

[9] CAICOYA, M. El papel de las mascarillas en el control de la epidemia COVID-19. Journal of Healthcare Quality Research, 2020, vol. 35, no 4, p. 203, doi: 10.1016/j.jhqr.2020.05.001

[10] Mujica Rodríguez, Iván E.; Toribio Salazar, Luz M.; Cóndor Cámara, Daniel F. Inteligencia artificial como apoyo a intervenciones no farmacológicas para combatir la COVID-19. Revista Peruana de Medicina Experimental y Salud Pública, 2020, vol. 37, p. 582-584, doi: https://doi.org/10.17843/rpmesp.2020.373.5704.

[11] ZAPATA HIDALGO, Andrés Miguel; SILVA TAPIA, Rodrigo. "Desarrollo de un prototipo para detección de mascarillas y control de aforo de personas en, 2022. Disponible: http://repositorio.espe.edu.ec/bitstream/21000/28495/1/T-ESPE-019847.pdf.

[12] GUALDRÓN, Oscar Eduardo; SUÁREZ, Oscar Manuel Duque. Diseño de un sistema de reconocimiento de rostros aplicando inteligencia y visión artificial. REVISTA COLOMBIANA DE TECNOLOGIAS DE AVANZADA (RCTA), 2014, vol. 2, no 24, p. 117-126. Disponible: https://ojs.unipamplona.edu.co/ojsviceinves/index.php/rcta/article/view/ 1222.

[13] LLATA, J. R., et al. Aplicación de inteligencia artificial en sistemas automatizados de producción. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 2000, vol. 4, no 10, p. 100-110.Disponible: https://www.redalyc.org/pdf/925/92541011.pdf

[14] GUERRA, Ángel; MAYOR, RICARDO GARCÍA. Retos éticos que plantea el uso de la inteligencia artificial en el diagnóstico y tratamiento clínico. Cuadernos de Bioética, 2018, vol. 29, no 97, p. 303-304. Disponible: https://www.redalyc.org/journal/875/87557374009/87557374009.pdf

[15] CORTES, Luini Leonardo Hurtado; VILLARREAL-LÓPEZ, Edwin; VILLARREAL-LÓPEZ, Luís. Detección y diagnóstico de fallas mediante técnicas de inteligencia artificial, un estado del arte. Dyna, 2016, vol. 83, no 199, p. 19-28, doi: https://doi.org/10.15446/dyna.v83n199.55612.

[16] GAMERO, Aldo Medina; CHAMORRO, Mónica Regalado. La Inteligencia Artificial en el control de la COVID-19. Atencion Primaria,2021, doi: 10.1016/j.aprim.2021.102099.

[17] GONZÁLEZ-PAYARES, María; USTARIS-SIERRA, Alan; CADAVID-PEÑA, Julián. Uso de mascarillas en tiempos de COVID- 19: Algunas manifestaciones en la piel del personal de la salud. IPSA Scientia, revista científica multidisciplinaria, 2020, vol. 5, no 1, p. 152- 158, doi: https://doi.org/10.25214/27114406.1028.

[18] OSPINO CASTRO, Adalberto; ROBLES ALGARÍN, Carlos; DURAN PABÓN, Alejandro. Modelado y simulación de un panel fotovoltaico empleando técnicas de inteligencia artificial. Ingeniería Energética, 2014, vol. 35, no 3, p. 225-233. Disponible: http://www.redalyc.org/articulo.oa?id=329132445007

[19] REYES-ORTIZ, Oscar Javier; MEJIA, Marcela; USECHE-CASTELBLANCO, Juan Sebastián. Técnicas de inteligencia artificial utilizadas en el procesamiento de imágenes y su aplicación en el análisis de pavimentos. Revista EIA, 2019, vol. 16, no 31, p. 189-207, doi: https://doi.org/10.24050/reia.v16i31.1215.

[20] ÁVILA-TOMÁS, Jose Francisco; MAYER-PUJADAS, Miguel Angel; QUESADA-VARELA, Victor Julio. La inteligencia artificial y sus aplicaciones en medicina II: Importancia actual y aplicaciones prácticas. Atención Primaria, 2021, vol. 53, no 1, p. 81-88, doi: https://doi.org/10.1016/j.aprim.2020.04.014.

[21] JONES, Paula; VIOLA, Paul; JONES, Michael. Rapid object detection using a boosted cascade of simple features. En University of Rochester. Charles Rich. 2001. Disponible: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.6807

[22] SIERRA ZAPATA, María. Estudio comparativo de modelos de identificación facial basados en correlación. Trabajo Fin de Grado en Ingeniería de las Tecnologías de Telecomunicación (pp. 204), 2015. Disponible: http://hdl.handle.net/11441/28426

[23] Microsoft. (2021, November 3). Visual Studio Code. Visualstudio.com; Microsoft. Disponible: https://code.visualstudio.com/docs

[24] Welcome to Python.org. (2022, May 17). Python.org; Python.org. Disponible: https://www.python.org/about/

[25] Pylance: nuestra nueva y mejorada experiencia de Python en VS Code. (2021, May 13). Microsoft.com. Disponible: https://docs.microsoft.com/es-mx/shows/vs-code-livestreams/pylance-new-and-improved-python-experience

[26] FrontPage - Python Wiki. (2013). Python.org. Disponible: https://wiki.python.org/moin/

[27] About - OpenCV. (2020, November 4). OpenCV. Disponible: https://opencv.org/about/

[28] NumPy. (2022). Numpy.org. Disponible: https://numpy.org/

[29] MediaPipe. (2020). Mediapipe.dev. Disponible: https://mediapipe.dev/

[30] OpenCV: Face Recognition with OpenCV. (2012). Opencv.org. https://docs.opencv.org/4.2.0/da/d60/tutorial_face_main.html

[31] PEREIRA JÚNIOR, Alexandre; DONADON HOMEM, Thiago Pedro; OLIVEIRA TEIXEIRA, Fabio. Aplicación de inteligencia artificial paramonitorear el uso de mascarillas de protección. Revista Científica General José María Córdova, 2021, vol. 19, no 33, p. 205-222, doi: https://doi.org/10.21830/19006586.725