Logotipo

Descripción generada automáticamenteUniversidad Peruana de Ciencias Aplicadas

**INFORME DE LA APLICACIÓN**

**INTELIGENCIA ARTIFICIAL  
TRABAJO 02**

**Nombre de la aplicación:** MaskDetect

**Presentado por:**Gómez Sánchez, Brandon Rafael  
Mac Kay Rodriguez, Lino Raul

**ÍNDICE**

[**IDENTIFICACIÓN DEL PROBLEMA** 3](#_Toc87272699)

[**BASE DE CONOCIMIENTO** 3](#_Toc87272700)

[**ARQUITECTURA DE LA RED NEURONAL** 3](#_Toc87272701)

[**CÓDIGO FUENTE DE LA APLICACIÓN** 4](#_Toc87272702)

[**FUNCIONALIDADES Y PRUEBAS DE USO** 5](#_Toc87272703)

[**REFERENCIAS BIBLIOGRÁFICAS** 8](#_Toc87272704)

# **IDENTIFICACIÓN DEL PROBLEMA**

Para la realización de esta tarea se logró encontrar que existe un problema con el uso de las mascarillas en algunos departamentos del país. En base a nuestra experiencia logramos ver que algunos ciudadanos de a pie deciden no usar las mascarillas cuando van a realizar cualquier tarea de su día a día.

Este problema se ha encontrado desde el inicio de la pandemia y ha habido especulaciones que intentan sustentar la razón por la que no usan mascarilla. Sin embargo, estos intentos son solo desestimados en lugares con centros comerciales privados ya que estos prohíben el ingreso a su establecimiento sin el uso de la mascarilla y los pobladores tienen que acatar las reglas predispuestas para poder ingresar a estos centros comerciales privados. Pero esto no ocurre en todos los establecimientos del Perú, hay una gran cantidad de lugares en donde se ha ido perdiendo el miedo a la pandemia y simplemente no toman las medidas de seguridad, exponiendo a todos a su alrededor debido al no uso de la mascarilla.

Por eso con este trabajo intentamos solucionar este problema en pequeña escala, permitiendo predecir si una persona está portando o no mascarilla. Este trabajo es complemento de futuros proyectos que pueden integrar un modelo predictivo en tiempo real mediante el uso de cámaras de vigilancia , para así proteger a las personas dentro y fuera de los establecimientos.

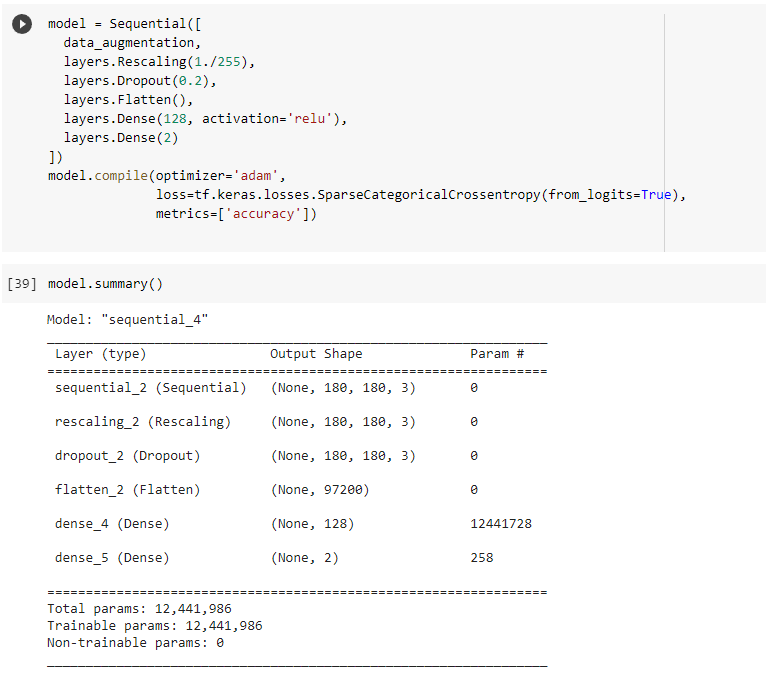
# **BASE DE CONOCIMIENTO**

El Data Set ha sido tomado de una plataforma web que permite el uso de data sets de manera libre. Nuestro conjunto de datos está principalmente compuesto por imágenes de personas que están y no están usando mascarilla. El data set tiene una longitud de más de 6000 imágenes y estas serán usadas en el modelo predictor.

# **ARQUITECTURA DE LA RED NEURONAL**

Para la creación de la red neuronal se tomó como muestra un modelo creado por Google para realizar clasificación de imágenes como se muestra en la Figura 1.

**Figura 1**  
Modelo predictivo



El modelo está basado en el uso de imágenes de 180 x 180 pixeles, en base a estas dimensiones se realizará una augmentación para generar distintas imágenes de la misma imagen.

Se realizará la normalización de los píxeles de imagen dividiéndolos entre 255.

Luego se utilizará una red neuronal de perceptrón multicapa para la clasificación de la imagen., en la que en primer lugar tendrá 128 outputs para finalizar con 2 outputs.

# **CÓDIGO FUENTE DE LA APLICACIÓN**

Debido a que se creó una aplicación web que utiliza un api creado en Flask es necesario instalar NodeJS para poder correr la SPA en la que se realizará la predicción.

Al no estar desplegado en la nube es necesario correr localmente el api creado en Flask, por ello es necesario tener instalados los módulos de Flask (*Welcome to Flask — Flask Documentation (2.0.x)*, n.d.), Flask\_cors, Tensorflow (*TensorFlow*, n.d.), Keras (*Keras: The Python Deep Learning API*, n.d.) y Numpy (*NumPy*, n.d.) para poder correr el API desarrollada.

Las instrucciones para correr el backend y el frontend están descritas en el repositorio de la aplicación (Mac Kay, n.d.).

El modelo se realizó usando Python a través de Google Colab, en un Jupyter Notebook <https://colab.research.google.com/drive/189sRd8a0UiXDjgWi-_nHixD1xqycJUjL?usp=sharing>

# **FUNCIONALIDADES Y PRUEBAS DE USO**

**Paso 1**: Entrar a la SPA con el url: <http://localhost:4200/home>

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 2**: Seleccionar la imagen a predecir

**Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente**

**Paso 3:** Al momento de hacer clic en el botón “Realizar predicción” aparecerá una ventana emergente que indicará el resultado de la predicción.

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente**

# **REFERENCIAS BIBLIOGRÁFICAS**

* *Keras: the Python deep learning API*. (n.d.). Retrieved November 4, 2021, from <https://keras.io/>
* Mac Kay, L. (n.d.). *IA-TAREA2*. Retrieved November 4, 2021, from <https://github.com/LinoMacKay/IA-TAREA2>
* Mac Kay, L. (n.d.). *IA TRABAJO.* Retrieved November 4,2021, from <https://colab.research.google.com/drive/189sRd8a0UiXDjgWi-_nHixD1xqycJUjL?usp=sharing>
* *NumPy*. (n.d.). Retrieved November 4, 2021, from <https://numpy.org/>
* *TensorFlow*. (n.d.). Retrieved November 4, 2021, from <https://www.tensorflow.org/>
* *Welcome to Flask — Flask Documentation (2.0.x)*. (n.d.). Retrieved November 4, 2021, from <https://flask.palletsprojects.com/en/2.0.x/>