Project Description

Problem statement: What is the problem?
The problem is that the majority of the people now don’t take virus outbreaks seriously and don’t follow the strict protocols so this is the reason why we are creating this app/machine.

Pitch: Why should people buy or listen to the app/idea?
Tired of people not wearing a mask and increasing the spread of viruses? Well don’t think about it again since we have created the Face Mask Detector!

Benefit: How will this benefit the AI community?
This will benefit the AI community a lot since people can use it for experimental reasons and can also be used for research. Another reason why it is useful and beneficial is because it is compatible with all devices, although it takes a few lines of code to set up, if you pre-make it, you're all good to go!

Tools & Technology Used

1. Python : The code is entirely written with python.

2. Google Colab : It is an online/open-source editor for python and many other programming languages.

3. Keras : Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the Tensorflow library

4. TensorFlow : TensorFlow is an open-source library developed by Google primarily for deep learning and machine learning applications.

5. Haar Cascade : Haar cascade is an algorithm that can detect objects in images regardless of their size. In our case, we train the algorithm to detect masks on people’s faces.

Data Description

- There are two separate folders in the dataset:

- One for images with mask:
  This contains 1,915 images

- One for images without masks:
  This contains 1,918 images

The age of the people in these images vary anywhere from 10 to 70.

- Why is age important:

- It is important because someone at the age of 50 has a different face structure than someone at age 12. This variety ensures that our model will be more accurate.

Methodology

Steps :

1. Installing/Importing Dependencies

If you are on Google Colab, you may want to start by importing the dependencies :

```
import cv2
import os
import numpy as np
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from google.colab.patches import cv2_imshow
```

2. Loading the model

We first need to clone this github repo : https://github.com/misbah4064/face_mask_detection.git. We used this as a reference since it contains “haarcascade_frontalface_alt2.xml”. This is a not so complex algorithm used to detect anything in an image. In our case, we trained it to detect masks. The github repo also contains “mask_recog.h5”, (the model that we used). The code below shows how we loaded the haar cascade and the model.

```
face_cascade=cv2.CascadeClassifier("haarcascade_frontalface_alt2.xml")
model=load_model("mask_recog.h5")
```

3. Making The Function

We started by defining our function:

```
def mask_detector(frame):
```

“frame”, in this case, is the image that we are giving the function. After this, we convert the image color to grayscale.

```
gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
```

Then, we started framing the faces of the people in the images.

```
for (x,y,w,h) in faces:
   faceframe=frame[y:y+h,x:x+w]
   faceframe=cv2.cvtColor(faceframe,cv2.COLOR_BGR2RGB)
   faceframe=cv2.resize(faceframe,(224,224))
   faceframe=img_to_array(faceframe)
   faceframe=np.expand_dims(faceframe,axis=0)
   faceframe=preprocess_input(faceframe)
   facelist.append(faceframe)
```

The code above creates a green or red frame depending on whether the person has a mask or not.

4. Labeling The Images/Frames

In this step, we will be labeling the frames. If the frame is green, the label will display “Mask”. If the frame is red, the label will display “No Mask”.

```
label="mask" if mask>no_mask else "no_mask"  #Labeling the images
color=(0,255,0) if label=="mask" else (255,0,0) #Coloring the labels
label="{}: {:.2f}%".format(label,max(mask,no_mask)*100)
cv2.putText(frame,label,(x,y-10),cv2.FONT_HERSHEY_SCRIPT_COMPLEX,7,color,2)
cv2.rectangle(frame,(x,y),(x+w,y+h),color,3)
```

5. Running The Code

To run the code that we have written until now, we have to give an input and call the function.

```
var=input("Enter the image name : ")
input_img=cv2.imread(var)
results=mask_detector(input_img)
cv2_imshow(results)
```

Reference

https://github.com/balajisrinivas/Face-Mask-Detection ----- This github repository contains the dataset

https://github.com/misbah4064/face_mask_detection.git ----- This github repository contain the model that we used

https://www.kaggle.com/datasets/andrewmvd/face-mask-detection?select=images ----- Kaggle link for extra resources

https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08 ----- A website used for extra resources