MASK IMMUNE FACE RECOGNITION SYSTEM:

PROBLEM STATEMENT:

Face is a common biometric trait used nowadays for identifying or verifying the identity of an individual in photos, video, or in real-time and thus facial recognition systems are proving to be an excellent way in providing touchless access control mechanisms and improving security systems. These systems use computer algorithms to pick out certain distinguishable

details about a person's face. However, facial occlusion proves to be a hindrance to this system.

For instance, the entire scenario of the spread of the recent COVID-19 pandemic has popularized the need of wearing a face mask to prevent the spread of infection. As preventative masks become more common around the world, detection and accurate recognition of partially occluded faces becomes necessary in order for facial recognition systems to be prevalent in their usage.

We have designed a **"mask immune"** facial recognition system that will recognize individuals wearing a mask. The emergence of such a system can benefit law enforcement, help in identifying criminals, improving surveillance systems, controlling access in smartphone and laptop unlock, etc. but most importantly, keeping in mind the current worldwide situation, it can allow healthcare workers and other individuals to be recognized and move through secured areas without removing personal protective equipment.

THEME:

FACE RECOGNITION:

Facial recognition is a part of biometric software that maps and traces an individual's facial features mathematically and stores the data as a faceprint. The software uses deep learning algorithms to compare a live capture/digital image to the stored faceprint in order to verify and recognize an individual's identity.

The software identifies few nodal points on a human face. In this context, nodal points are used to measure variables of a person’s face, like the length or width of the nose, the depth of the eye sockets and the shape of the cheekbones. The system works by capturing data for nodal points on a digital image of an individual’s face and storing the resulting data as a faceprint which is then compared in order to carry out face recognition.

PROPOSED SOLUTION:

The proposed system has been built and trained upon the Microsoft Azure AI platform and will use a recognized dataset for training and testing. Face occlusion by the wearing of mask makes the face only partially visible. The

algorithm thus will be trained to focus on whatever face landmarks are visible and are helpful in distinguishing one face from another in order to determine an accurate match from the existing dataset. The system thus intends at capturing the prominent visible details of the partially occluded face such as distance between the eyes or shape of the forehead, eyebrows, nose, colour of the pupil, etc and then produce a nearly accurate match from the existing dataset.

IMPACT AND ADVANTAGES:

This project can prove to be extremely beneficial for the people in terms of providing a touchless access mechanism as an alternative to biometrics. It is a significant step towards modernizing security systems so that it becomes difficult to deceive them. With increasing crime rates over the past few years most of the metropolitan cities are completely under surveillance and our project would help in the identification of personnel even with masks occluding the human face, thus in enhancing security. Moreover, the same project can be used in face lock present in phones without having the users remove their mask in order to unlock them. As mentioned before, it can also allow healthcare workers to be recognized and move through secured areas

without removing protective equipment and alert them to put their masks on when they're not.

STAKEHOLDERS:

The goal we strive for can be achieved with minimal funds and a little human effort. However, if we get the opportunity to collaborate with the Government of India in order to implement this project on a larger and wider scale, we would need the Government's cooperation in order to make the required datasets available to us. Another example could be, that in the city of Delhi, there are approximately 1.4 lakh cameras, one at every 10.6 m. Thus, by uploading the AI in these security systems, we can save the amount of installing new devices for our purpose. This system can also be installed in companies, industries and hospitals with the assistance of the management of the corporation. The manpower required should be qualified in implementing deep learning, machine learning, image and face recognition techniques and/or well versed with the necessary programming languages. Our servers will be cloud-based which massively reduces electricity consumption. Moreover, no server-related e-waste will be generated. Environmental sustainability will be our priority and we will work towards achieving it to our model's maximum potential.

EXECUTION AND IMPLEMENTATION:

We have deployed a multiclass image classifier which is successfully trained by our self-compiled and pre tagged dataset and is able to unmask the identity of a masked person by successfully recognizing him/her. We have used **MICROSOFT AZURE AI CUSTOM VISION MODULE**. We have created a large self-compiled dataset of **1650 pictures**. It has a collection of **masked and unmasked pictures of 33 people** which includes some famous celebrities as well. Some of the collected photographs have been masked by uploading them online on a site [2] while the other pictures were run through a code [3] that detected the key points of each face such as nose, chin, etc. and then pasted the mask on the pictures. The plus point of our dataset is that the pictures have been collected "in the wild". Moreover, we have ensured that our dataset is not skewed. It is balanced and has the same number of unmasked and masked pictures of each person. The proposed dataset was then uploaded on the custom vision portal and was "advance" trained for 1 hour. We have tested our model a number of times using both masked and unmasked images and for that we designed a website as well. (https://ai-face-recognition.vercel.app/)

Our project has a precision of 96.2%, a recall percentage of 92.1% and an AP of 97.5%

The way we have implemented this project developing a website where our database would expand day by day as more and more people use it. As our database expands people from all over the world would be able to upload their photos and (masked or unmasked) and the AI would be able to recognize the person if his/her photos are already in our database.

Another implementation of this project in the app is by providing a feature that allows people to be recognized by the AI once their webcams are on (this is still under development).

The second part of the project that involves the use of python has three separate codes i.e. the Setup, Image Test and LiveVideoTest. The setup.py file is used to execute the initial training of the project which involves 20 checkpoints for training. The dataset which has been used for training has about 1376 images with about 685 each of masked and unmasked. These images can be used for testing the project as well.

1. **Training: The primary focus over here was on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk**

2. **Deployment: Once the face mask detector is trained, we can then move on to loading the mask detector, performing face detection, and then classifying each face as with or without mask. Facial landmarks allow us to automatically infer the location of facial structures, including eyes, eyebrows, nose, mouth, jawline, texture and more.**

To use facial landmarks to build a dataset of faces wearing face masks, we started with an image of a person *not* wearing a face mask. From there, we applied face detection to compute the bounding box location of the face in the image. Once we know *where* in the image the face is, we can extract the face Region of Interest (ROI). And from there, we applied facial landmarks, allowing us to localize the eyes, nose, mouth, etc. Next we used an image of a mask to apply on the ROI and resize for different images.

The three different .py files are described below:

- **Setup.py - Accepts our input dataset and fine-tunes MobileNetV2 upon it to create our mask_detector.model. A training history plot.png containing accuracy/loss curves is also produced**
- **Image test.py - Performs face mask detection in static images.**
- **LiveVideo test.py - Using your webcam, this script applies face mask detection to every frame in the stream.**

After having achieved this benchmark the other two codes can be run depending of the purpose for the project. They can be run by their separate start command that needs to be used for the beginning of the project. A specific % is also returned as an output depending on the overall area covered by the mask and the expected area covered by the mask.
The image test file needs an input by the user for the working of the AI to judge if the person in image is masked and the Live Video file uses a webcam to identify if the person is masked. A good web cam is expected or else it is there are chances that the output percentage would be not at expected.

PLATFORM (MICROSOFT AZURE COGNITIVE SERVICES):

Cognitive services are a set of machine learning algorithms that Microsoft has developed to solve problems in the field of artificial intelligence (AI). The goal of cognitive Services is to democratize AI by packaging it into discrete components that are easy for developers to use in their own apps. Web and Universal Windows Platform developers can consume these algorithms through standard REST calls over the internet to the Cognitive services APIs. The Cognitive Services APIs are grouped into five categories:

- **Vision- analyze images and videos for content and other useful information.**
- **Speech- tools to improve speech recognition and identify the speaker.**
- **Language- understanding sentences and intent rather than just words.**
- **Knowledge- tracks down research from scientific journals for the user.**
- **Search- applies machine learning to web searches.**

PLATFORM (PYTHON):

The other part of the project is being used to detect if a person is masked or not with two different real life uses one being the Live Stream check and other being a Picture format. It has been developed on Python while using various different libraries including Keras,TensorFlow, Face_Recognition, dlib, cmake and much more. A .py file has been created to set up the AI and other two codes present are for their respective purposes.

All these features have been compiled into a single website based up on JavaScript with separate pages for various offered uses of the project.

GRAPHICAL USER INTERFACE:

The website is made with:-

1. HTML 5 – it was used for structuring and defining the elements in the page like images, text, buttons

2. CSS - for styling, positioning, scaling, and adding special effects

3. JavaScript - is used for XMLHTTPRequesting with defining headers within the URL that contain data for retrieving information in the form of JSON.

4. JSON - used for organizing data received like probabilities, name tags, their codes, etc.

5. in JavaScript, using algorithms, I sorted out the highest probability and changed properties of HTML,CSS to display the name and probability.

6. I used Github Pages to publish my source code, so that the website can be used by anyone through the link

7. bit.ly for the URL shortener

FUTURE DEVELOPMENT:

In a world that is constantly subjected to change, we have created our project in a way so that necessary amendments can be made depending on the needs of the society. We have also analysed the possible future developments in the proposed model. We want to explore if there is any scope of including thermal facial recognition. The underlying theoretical concept beneath this idea is the fact that each person has a unique system of capillaries on their face which can be detected by thermal scanners present in thermal cameras.

We are also working towards analyzing the skin colour and setting up nearby temperature detectors and incorporating the feature of real-time analysis of faces even outside the provided datasets.

SUSTAINABLE DEVELOPMENT GOALS:

Herman E Daly rightly said and I quote, “Presumably, technology has made man increasingly independent of his environment. But, in fact, technology has merely substituted non-renewable resources for renewables, which is more an increase than a decrease in dependence.” It is imperative for us to ensure environmental protection while embracing technology and as

responsible citizens, we have ensured that our project goes hand-in-hand with the SDGs that is testimonial of the fact that we have built an AI for the present as well as the future.

INDUSTRY, INNOVATION AND INFRASTRUCTURE:

1. **A considerably new technique**

GOOD HEALTH AND WELL BEING:

1. **Touchless, prevents spread of contagious diseases.**
2. **No removal of mask, no exposure to polluted air particles.**
3. **Protective equipment intact, safety ensured**

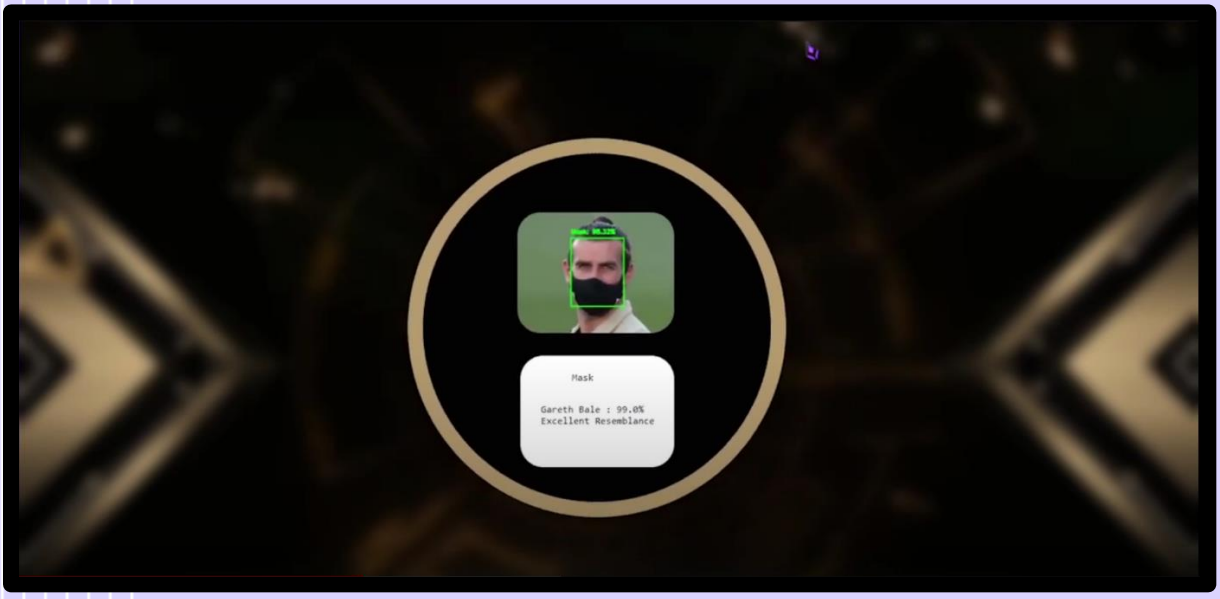
PARTNERSHIP FOR THE GOALS:

1. **healthy collaboration between corporations and human resource**
2. **Contributes to greater good of the people.**

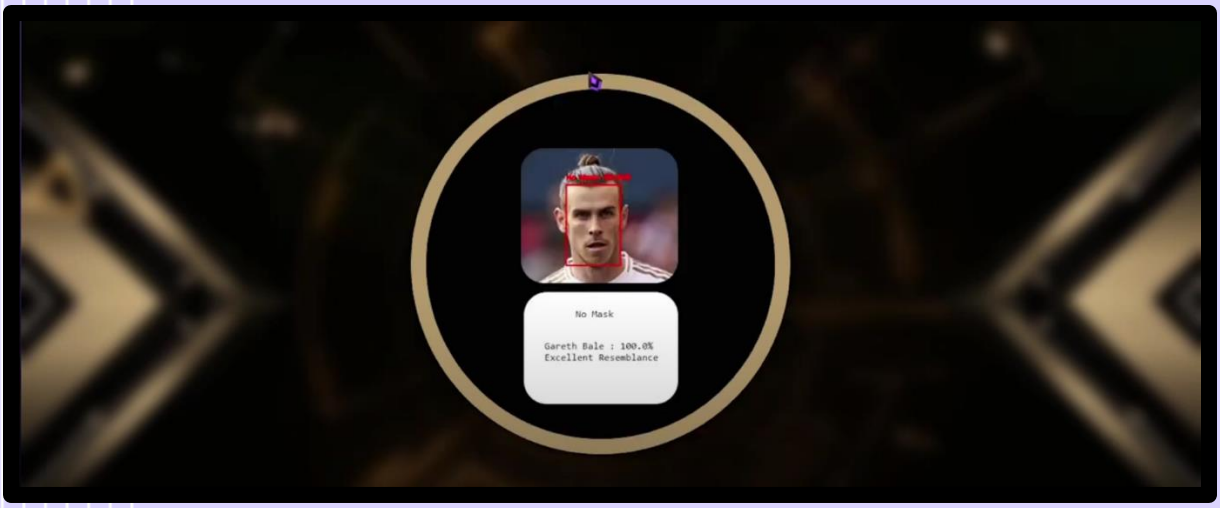
“Technology is best when it brings people together.” and we, as a team are constantly striving for the best. We hope to be able to achieve all our goals for this project. Thank you for giving us an opportunity to share with you our creation, a result of our effort, dedication and collaboration.

DETAILED IMPLEMENTATION:

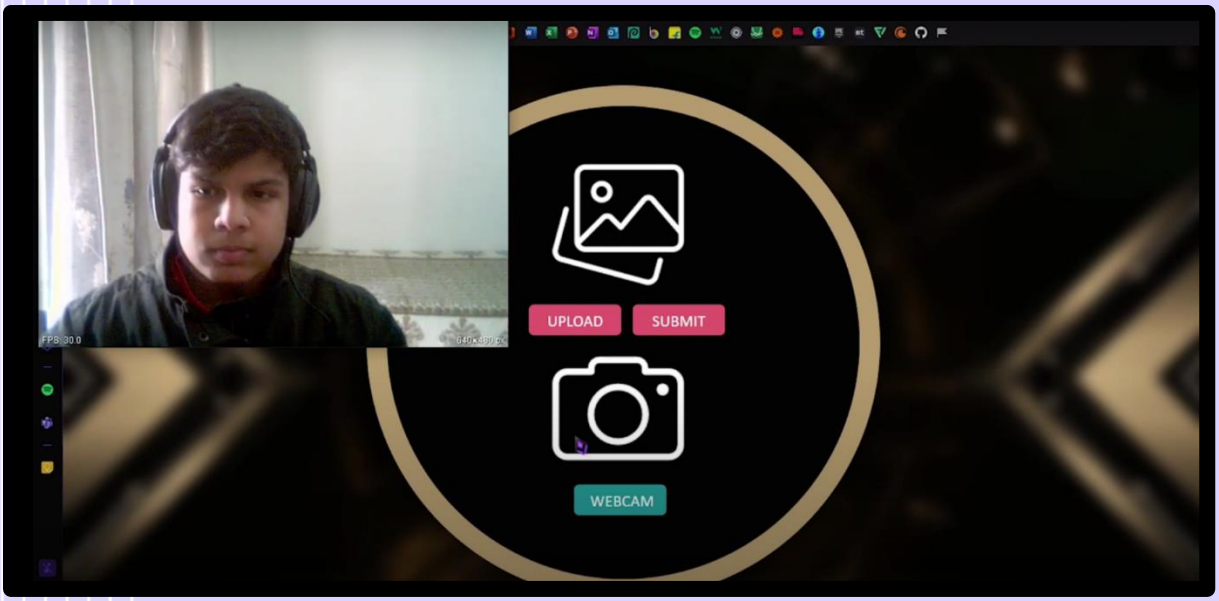
We team syntax terminators of Bluebells School International have come up with a mask immune facial recognition system. Our system has been made by our self-compiled and pre tagged dataset with masked and unmasked images of 33 people including some celebrities. Let's test our system by randomly picking up a picture of garath bale with a mask on from google. We've uploaded this picture on our GUI and are awaiting the results. Guess what our system has identified Gareth Bale with remarkable accuracy and not only that it also indicates whether a person has mask on or not.

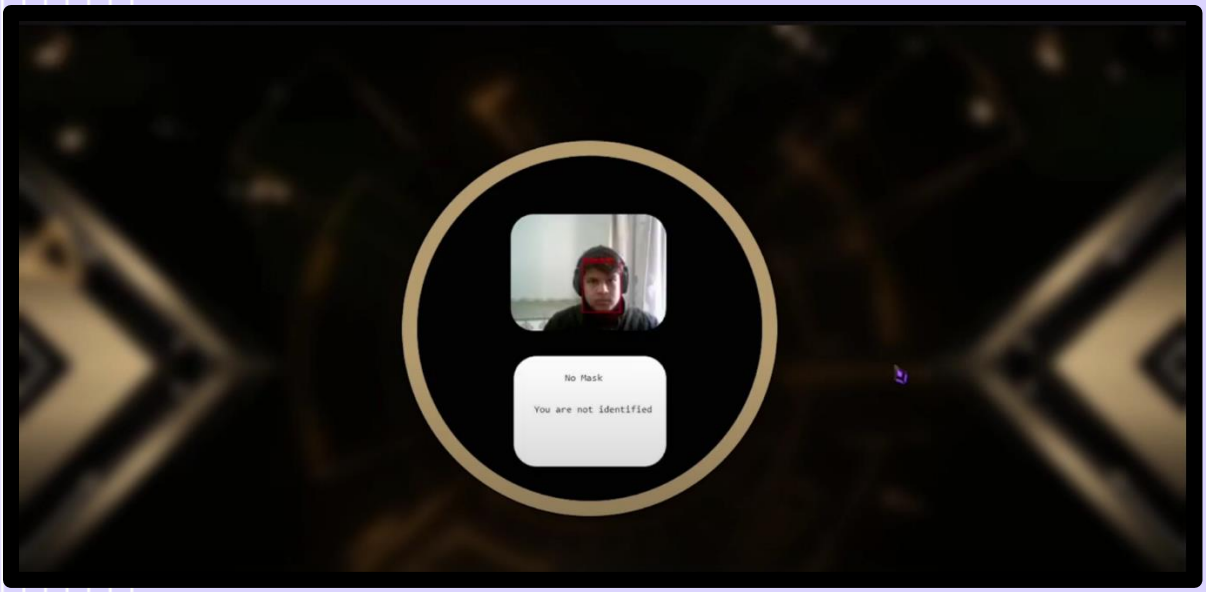


Let us repeat this process to check if the system works for the unmasked of the same person.

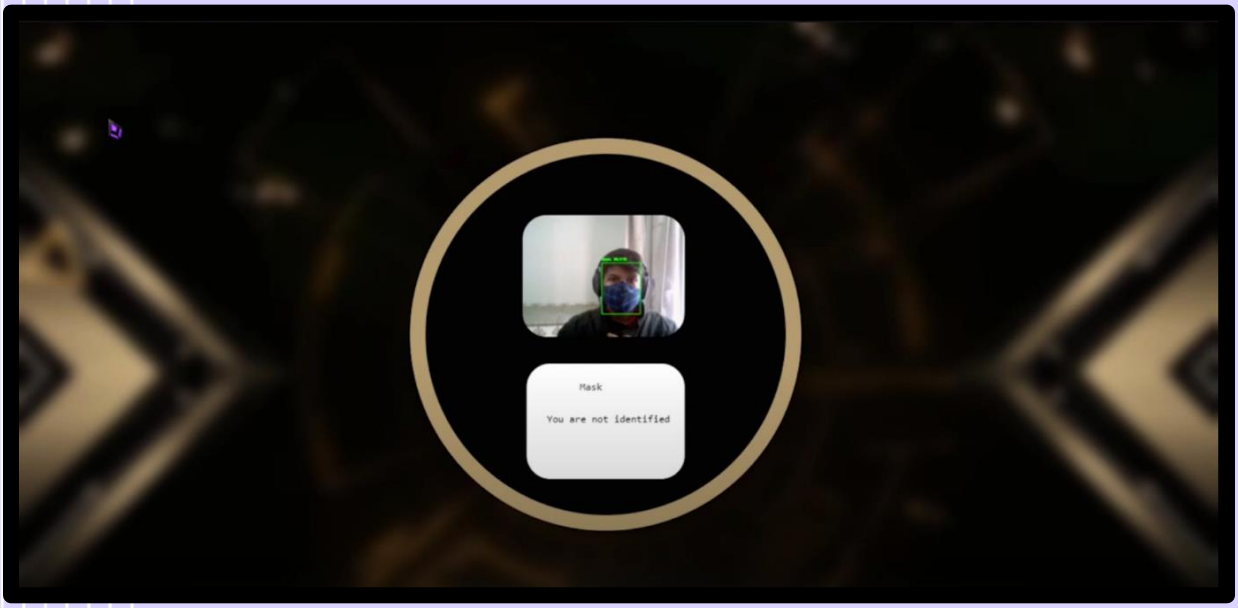


Can you notice the indication that there is no mask put on, this is the additional feature that our system provides. Yet again we were successful in recognizing the person but is this where we stopped? Of course not .The more the better so let's test the live camera feature.

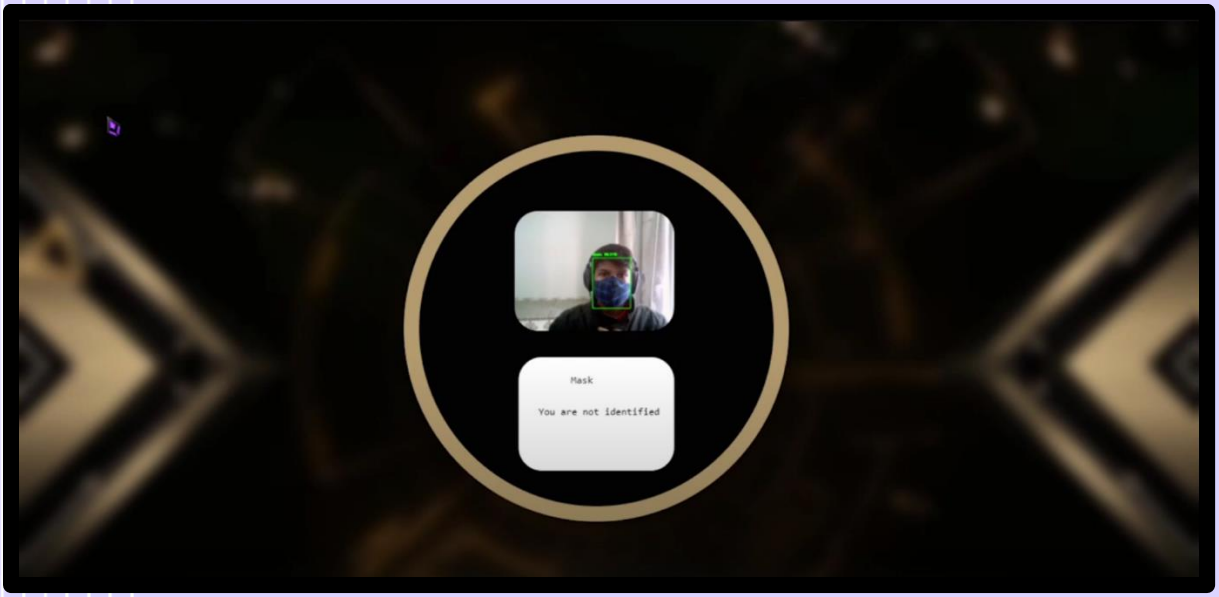




Here our friend was not recognized because he doesn't belong in the list people that can be identified by our system, hence we got an indication and message that he wasn't recognized. This time he is testing himself with mask on let us see if he gets the green box indicating the presence of the mask..



and that he does Though with this remarkable technology we still have a long way to go but we are doing our best.



CODES:

C:\Users\HP\Downloads\train_mask_detector (1).py • - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

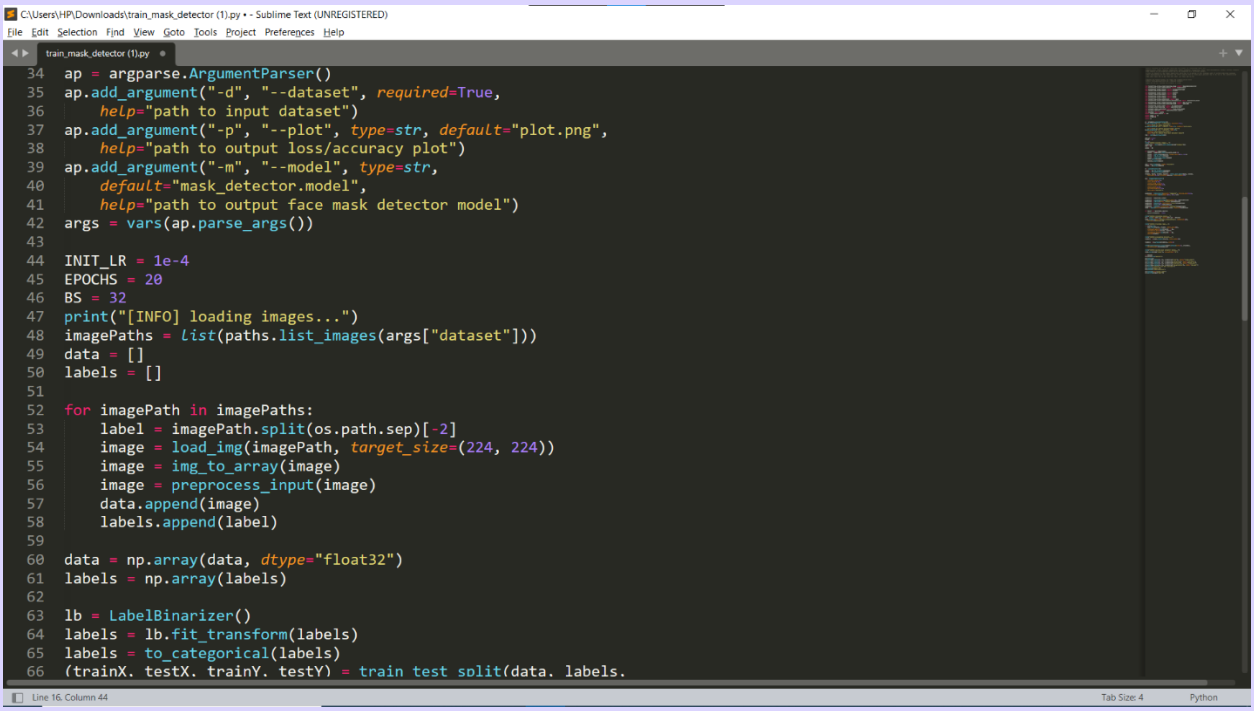
train_mask_detector (1).py

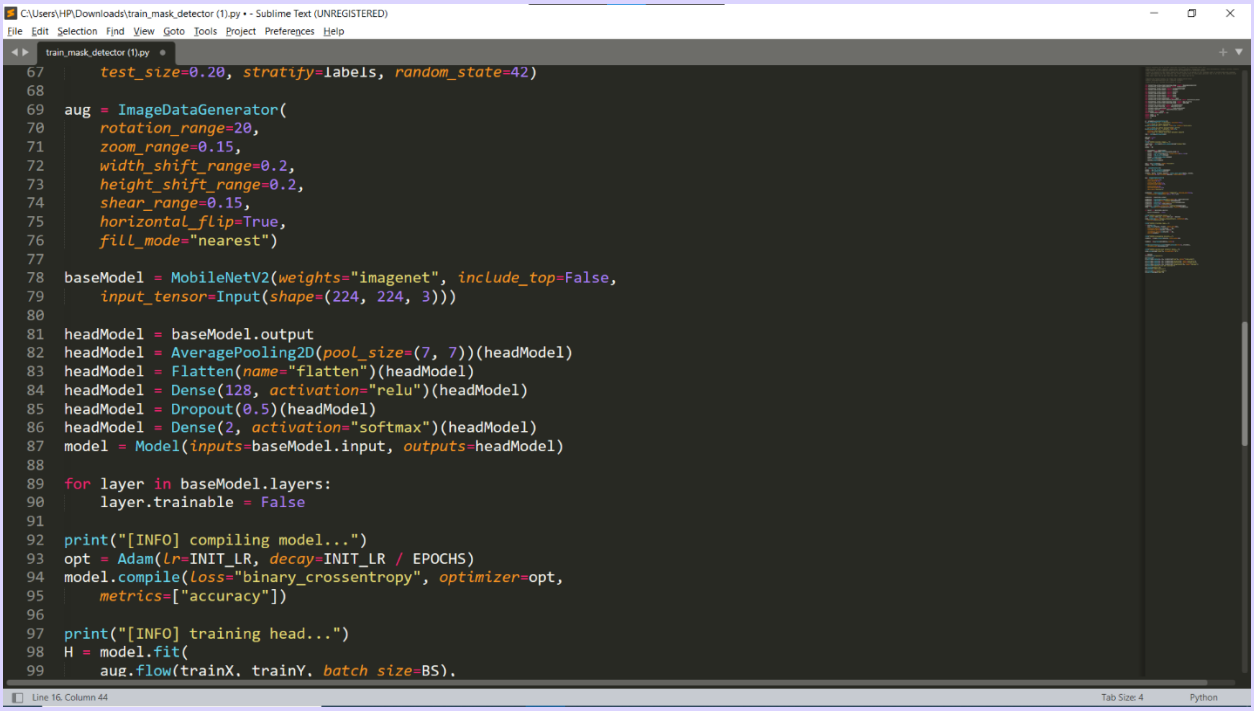
```
# Before training the AI certain libraries are needed for processing which are:
# OpenCV, dlib, cmake, imutils, matplotlib, numpy, argparse, tensorflow, keras, face-recognition, pillow, scripy, sklearn
# dlib library is an exception which has to be installed in a different manner.
# A file is present in the folder named dlib which has to be pasted in the original path of python/libs/site packages
# After downloading all the libraries the first program that is train_mask_detector has to be run by the command given
# This will train the AI and then the other .py files can be run

# specify the folder before you input the command given below
# python train_mask_detector.py --dataset dataset
# input in cmd for running the program to train

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import argparse
import os
```

Line 16, Column 44 Tab Size: 4 Python


C:\Users\HP\Downloads\train_mask_detector (1).py • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
train_mask_detector (1).py
ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dataset", required=True,
help="path to input dataset")
ap.add_argument("-p", "--plot", type=str, default="plot.png",
help="path to output loss/accuracy plot")
ap.add_argument("-m", "--model", type=str,
default="mask_detector.model",
help="path to output face mask detector model")
args = vars(ap.parse_args())
INIT_LR = 1e-4
EPOCHS = 20
BS = 32
print("[INFO] loading images...")
imagePaths = list(paths.list_images(args["dataset"]))
data = []
labels = []
for imagePath in imagePaths:
label = imagePath.split(os.path.sep)[-2]
image = load_img(imagePath, target_size=(224, 224))
image = img_to_array(image)
image = preprocess_input(image)
data.append(image)
labels.append(label)
data = np.array(data, dtype="float32")
labels = np.array(labels)
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)
(trainX, testX, trainY, testY) = train_test_split(data, labels,
Line 16, Column 44
Tab Size: 4
Python


C:\Users\HP\Downloads\train_mask_detector (1).py • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
train_mask_detector (1).py
test_size=0.20, stratify=labels, random_state=42)
aug = ImageDataGenerator(
rotation_range=20,
zoom_range=0.15,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.15,
horizontal_flip=True,
fill_mode="nearest")
baseModel = MobileNetV2(weights="imagenet", include_top=False,
input_tensor=Input(shape=(224, 224, 3)))
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)
model = Model(inputs=baseModel.input, outputs=headModel)
for layer in baseModel.layers:
layer.trainable = False
print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,
metrics=["accuracy"])
print("[INFO] training head...")
H = model.fit(
aug.flow(trainX, trainY, batch_size=BS),
Line 16, Column 44
Tab Size: 4
Python

C:\Users\HP\Downloads\train_mask_detector (1).py • - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

train_mask_detector (1).py

```
    steps_per_epoch=len(trainX) // BS,
    validation_data=(testX, testY),
    validation_steps=len(testX) // BS,
    epochs=EPOCHS)

print("[INFO] evaluating network...")
predIdxs = model.predict(testX, batch_size=BS)

predIdxs = np.argmax(predIdxs, axis=1)

print(classification_report(testY.argmax(axis=1), predIdxs,
    target_names=lb.classes_))

print("[INFO] saving mask detector model...")
model.save(args["model"], save_format="h5")

N = EPOCHS
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, N), H.history["accuracy"], label="train_acc")
plt.plot(np.arange(0, N), H.history["val_accuracy"], label="val_acc")
plt.title("Training Loss and Accuracy")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend(loc="lower left")
plt.savefig(args["plot"])
```

Line 16, Column 44 Tab Size: 4 Python

—

C:\Users\HP\Downloads\Program (1).cs - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Program (1).cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;

namespace ai_web
{
    public class Program
    {
        public static void Main(string[] args)
        {
            CreateHostBuilder(args).Build().Run();
        }

        public static IHostBuilder CreateHostBuilder(string[] args) =>
            Host.CreateDefaultBuilder(args)
                .ConfigureWebHostDefaults(webBuilder =>
                {
                    webBuilder.UseStartup<Startup>();
                });
    }
}
```

Line 27, Column 1 Spaces: 4 C#

REFERENCES:

https://towardsdatascience.com/precision-vs-recall-386cf9f89488

2. [2] https://app.generative.photos/masks4all

3. [3] https://github.com/Prodesire/face-mask

4. https://arxiv.org/pdf/2003.09093.pdf

5. https://repository.iiitd.edu.in/jspui/handle/123456789/772

6. https://arxiv.org/abs/1811.04507

7. https://ieeexplore.ieee.org/abstract/document/8698571

8. https://www.healthline.com/health-news/tech-thermal-face-scanning-newest-way-to-identify-at-a-distance-071713#1

9. https://www.dailymail.co.uk/sciencetech/article-3178864/Nowhere-hide-Facial-recognition-technology-identify-people-complete-DARKNESS-reading-thermal-signatures.html

10. https://www.cin.ufpe.br/~rps/Artigos/Face%20Recognition%20Using%20Eigenfaces.pdf

11. https://www.customvision.ai/projects/68e29f1e-edea-4108-85e4-4a92480efe57#/performance