Detecting people not following covid norms and masks over face

A

Project Report

Submitted for the partial fulfillment

of B.Tech. Degree

in

COMPUTER SCIENCE & ENGINEERING

by

Navin Chandra (1805210027)

Arnab Debnath (1805210011)

Tushar Giri (1805210059)

Under the supervision of

Mr. Abhishek Singh

Dr. Pawan Kumar Tiwari

Department of Computer Science and Engineering

Institute of Engineering and Technology

Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh.

July 2022

Contents

DECLARATION..i
CERTIFICATE..ii
ACKNOWLEDGEMENT..iii
ABSTRACT..iv
LIST OF FIGURES..v
1. INTRODUCTION 8
2. LITERATURE REVIEW 11
3. METHODOLOGY …………………………………………………………14
3.1 EXISTING SYSTEM ……………………………………………………..14
3.2 PROPOSED SYSTEM …………………………………………………....14
3.3 HARDWARE REQUIREMENTS……………………………….………...14
3.4 SOFTWARE REQUIREMENTS………………………………………….15
3.5 SYSTEM ARCHITECTURE ……………………………………………..15
3.6.MODULE DESCRIPTION………………………………………………..15
3.6.1 IMAGE PREPROCESSING …………………………………………......16
3.6.2 DATA AUGMENTATION…………………………………………………17
3.6.3 MODEL TRAINING………………………………………………………18
3.6.4 TESTING THE MODEL…………………………………………………18
3.6.5 IMAGE SEGMENTATION USING MASK CNN ……………………………………..19
3.6.6.IMPLEMENTING THE MODEL IN OPENCV. ……………………………………........19
3.6.7 IMPLEMENTING FACE RECOGNITION. ………………………….…………………...20
3.6.8 SEND SMS AND EMAIL TO A PERSON WHO IS NOT WEARING A MASK. …..............22
4. EXPERIMENTAL RESULTS …………………………………………………...23
5. CONCLUSIONS ……………………………………………………………..27
REFERENCES
ANNEXURE

List Of Figures

Fig 3.1:- Testing the Model Evaluation Metrics…………………………………………19

Fig 3.2:- Flow chart for implementation of facial recognition..…………………………22

Fig 3.3:- SMTP protocol……….………………..…………………………………………..23

Fig 4.1:- Testing the Model………..…………………………………………………….24

Fig 4.2:- Detecting people wearing masks…...………………….....……………………25

Fig 4.3:- Sending text messages to people not wearing masks………………………….26

Fig 4.4:- Email template sent to people…………………………………………………….27

Declaration

We hereby declare that this submission is our own work and that, to the best of our belief and knowledge, it contains no material previously published or written by another person or material which to a substantial error has been accepted for the award of any degree or diploma of university or other institutes of higher learning, except where the acknowledgment has been made in the text. The project has not been submitted by us at any other institute for the requirement of any other degree.

Submitted by: - Date: 26-05-2022

(1) Name: Navin Chandra

Roll No.: 1805210027

Branch: Computer Science and Engineering

Signature:

(2) Name: Arnab Debnath

Roll No.: 1805210011

Branch: Computer Science and Engineering

Signature:

(3) Name: Tushar Giri

Roll No.: 1805210059

Branch: Computer Science and Engineering

Signature:

Certificate

This is to certify that the project report entitled “Recognizing people who aren’t wearing masks” presented by Navin Chandra, Arnab Debnath, and Tushar Giri in the partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering, is a record of work carried out by them under my supervision and guidance at the Department of Computer Science and Engineering at Institute of Engineering and Technology, Lucknow.

It is also certified that this project has not been submitted to any other institute for the award of any other degrees to the best of my knowledge.

Mr. Abhishek Singh

Dr. Pawan Kumar Tiwari

Department of Computer Science and Engineering

Institute of Engineering and Technology, Lucknow

Acknowledgment

We would like to express our gratitude to Dr. Divakar Singh Yadav, Head, CSE Department, IET, Lucknow, and our humble Supervisors Dr. Pawan Kumar Tiwari and Mr. Abhishek Singh for their continuous support and suggestions so far in this project. They have been very constructive, supportive, kind and made it easier to write up this thesis work. Their continuous support gave us the motivation to work on this project and helped us to gain knowledge in various fields. I would also like to thank them for suggesting changes. They allowed us to locate areas of improvement that helped me to write this interim report.

Navin Chandra (1805210027):

Tushar Giri(18015210059):

Arnab Debnath(1805210011):

Abstract

Our earth is going through a pandemic phase where people are getting affected by the COVID-19 virus and its different variants. This has resulted in a large number of casualties and security concerns. People have been instructed to wear masks whenever they are in public places to prevent the rapid spread of this virus. Therefore, we designed a model which will help us to make people aware of the importance of wearing masks. The primary goal was to come up with an idea that can motivate people to wear masks in public places by detecting the people who have not worn masks. In this report, in order to address the issue of the masked face region, we propose a model based on discarding the masked region and deep learning-based features. The first step is to detect the people who are not wearing masks by discarding the masked faces. For this, we used CNN. Then for recognizing the people who have not worn masks we used the facial recognition algorithm which will help us find the details of those people. For awarding those people we will be sending a precautionary message to wear masks in public places. All these can be observed through live streaming from cameras at various public places like malls, markets, classrooms, etc.

Chapter 1
Introduction

1.1 OVERVIEW

It has been almost 3 years that we are facing the wrath of COVID-19 all over the world affecting the lives of many people. India is also not far away from this. India recorded one of the worst effects of COVID leading to the economical crisis. Lots of people lost their lives. Lockdown was imposed leading to strain on the world economy. After the two waves got over, relaxations were given. WHO gave instructions on how to take care of this virus. They said that mask is a great weapon to keep coronavirus away. Proper sanitization should be maintained whenever people go out. People after getting relaxed in covid regulations started going out. Often we saw people not maintaining social distancing when going out in public places. As we know how important it is to maintain social distancing, sanitization, and wearing masks when we go out in public. After the revocation of covid rules, we have seen people going out in crowded places without masks and maintaining social distance. Surveys tell that almost 90% of people are aware whereas only 40 percent of people wear masks. From these surveys, we will say that people often neglect to wear masks either because they are not comfortable or they don’t believe that there is such a thing called covid. This kind of behavior by people leads to the quick spread of covid viruses rapidly affecting the lives of many people. The WHO has thus said that only wearing masks and maintaining social distancing is the key to keeping away the virus. The areas where people are freely moving without masks are the key culprits. Also, the people living in rural areas where there is no proper awareness, need to be aware of these facts and tips to keep away these viruses. This uncertain outbreak of the COVID-19 virus has led us to the importance of wearing masks in public places and keeping us safe from such viruses.

1.2 OBJECTIVE

The objective of this model is to design and develop a model that helps us to identify those people who aren’t wearing masks either due to lack of awareness or negligence. It is created by the use of MobileNet V2, Face Recognition Library, and Twilio.

1.3 PROBLEM STATEMENT

Nowadays, carrying masks is a critical want in public places, because, to a large extent, it will become hard for protection officers to check each and every individual who is no longer wearing a mask. Therefore, it is important to identify those people and make them aware of the importance of wearing masks.

1.4 PROJECT DESCRIPTION

The MobileNet V2 algorithm will be used to create face masks in this project. We'll be training our mannequin using images of people wearing masks and those who aren't wearing masks, which will be sent as an entered dataset, and the segmented photo of the equation will be purchased as an output. The most important phase in our model is pre-processing, which involves resizing the image to a particular size and converting it to a NumPy array. One warm encoding is applied to the pictures. Finally, the data are broken down into two categories: checking out and training. The second phase is data enhancement, which involves flipping, zooming, and finally flipping the shot horizontally. The third stage is training; the bottom model's region is loaded with imageNet weights, and the closing layer of the pre-trained model is fine-tuned as a result. The usual pooling, flattening, dense, activation function (relu and softmax), and the dropout rate are all entered and processed inside the closing totally associated layer. The model summary is then obtained, and the configuration is also recorded. The optimizer loss entropy and accuracy metric are specified in the fourth phase, and the trained model is then assessed and stored. The schooling loss and training accuracy data are plotted inside the fifth phase. Following that, a prediction on the teaching set is completed. Finally, the model is put to the test. Image segmentation is the sixth phase, in which the dataset's proximity is imported, ID mapping is complete in the region of the masks, and no masks are issued. Following that, a TensorFlow session is created, and the Mask RCNN model is loaded. After that, appropriate detection of Boxes, Classes, Scores, and Masks is completed. The results of the detection are then shown after event segmentation. Finally, the mannequin is created with the use of a webcam, which allows the video to be viewed through the body and scaled as needed. The preprocessing feature is then used to determine the impacts of persons wearing masks and not wearing masks collectively, with a % accuracy. Then we identify those of us who aren't wearing masks and match them with real individuals in our database to gain access to their contact information. For this, we used a biometric identification library to match that person and collect their information. Then, using an API request, we'll utilize Twilio to deliver messages and mail to these men and women who are no longer wearing masks.

1.5 MOTIVATION

The world has been turned upside down over the past two years, owing to the very fatal virus Covid-19. The spread of covid is still visible in most areas, with new varieties of the same virus-like Omicron emerging. The mask is one item that must be worn frequently when going out these days. However, many individuals are ignorant of genuineness and refuse/avoid putting on a mask. This exercise aids in identifying those who do not appear to be wearing a mask. Masks have been the most recent addition to a list of common items that includes wallets and purses.

Chapter 2
Literature Review

The Literature Survey is hired to deliver a quick evaluation and explanation of the reference papers. The literature survey conveys the technical important points associated with the task in an instead ideal and precise manner. Xinbei Jiang, Tianshan Gao, Zichen Zhu, and Yukang Zhao [1] proposed a real-time masks detection tool with the usage of YOLOv3. This paper makes use of the Masked Face Detection (PWMFD) dataset containing 9205 sports activities masks image samples. Relationships among channels are executed via means of integrating Darknet53's eye mechanism with the usage of SE blocks to permit the network to goal this feature. A Glou loss is carried out to account for spatial variations among scoring and floor packing containers and to enhance the robustness of bounding box regression. Excessive foreground/history kind imbalance is dealt with as a lack of focus. The relaxation of the outcomes showed that SEYOLOv3 changed into more healthy than YOLOv3 and different current PWMFD detectors. When evaluated with YOLOv3, the proposed dummy confirmed an 8.6% extra mAP and detection rate. Samuel Ady Sanjaya and Suryo Adi Rakhmawan [2] advanced mask detection with the usage of MobileNetV2. Within the article, the laptop learns the MobilenetV2 set of rules for mask identification. The steps concerned in the manikin introduction consist of facts gathering, pre-processing, facts separation, version testing, and version application. The proposed dummy can offer an accuracy of 96.85%. Sunil Singh, Umang Ahuja, Munish Kumar, Krishna Kumar, and Monika Sachdeva [3] proposed a mask detection tool for the usage of YOLOv3 and quicker RCNN models. This newspaper draws the eye to packaging packing containers whether or not or now no longer the humans on the window in purple or green sun sunglasses are carrying masks, persevering with the percentage of individuals who put on masks in regular life.

G. Jignesh Chowdary, Narinder Singh Puni, Sanjay Kumar Sonbhadra, and Sonali Agarwal [4] developed a mask sensing device using InceptionV3 transfer learning. Inside the paper, a switch-controlled dummy has been proposed to automate the approach of detecting a person not wearing a sports mask. The dummy uses deep knowledge of the Inception V3 algorithm to observe the mask. The Simulated Masked Face Dataset is used for training and testing. Due to constrained availability, the picture graph growth approach is used for better training and version validation. The dummy became 99.9% curate for the duration of schooling and 100% curate for the duration of testing.

Shilpa Sethi, Mamta Kathuria, and Trilok Kaushik [5] carried out masks to hit upon using deep mastering. To gain immoderate precision and brief inference time, the proposed approach makes use of a single-degree detector and a -degree detector. ResNet50 and furthermore, sooner or later in this article, the idea of a switcher mastering high-degree semantic statistics comes into play. During mask detection, for you to improve localization performance, bounding area transformation is hired. Three well-known baseline models viz. ResNet50, AlexNet, and MobileNet are used for experimenting with the version. The proposed together with those models can produce immoderate accuracy in a whole lot much less inference time. The proposed technique 6 achieved an accuracy of 98.2% while carried out with ResNet50. in evaluation with the presently published Retina facemask detector, the proposed model achieves 11.07% and 6.44% extra precision and bear in mind in mask detection. The proposed model is fantastically prepared for video surveillance devices. Riya Chiragkumar Shah and Rutva Jignesh Shah [6] proposed a tool for the Detection of Face masks with the usage of a Convolutional Neural Network. The model proposed properly right here is supposed to be modeled using python libraries, specifically TensorFlow, Keras, and OpenCV. The model used is the MobileNetV2 of convolutional neural networks. Throughout this paper, a model is advanced using the above-stated libraries. The model is tested for a variety of prerequisites with distinct hyperparameters. the maximum critical dataset is fed into the version, and run by the academic program, which trains the model at the given dataset. Then the detection utility is run, which activates the video stream, and captures the frames continually from the video waft with an anchor area using an item detection process. The output is then equipped MobileNetV2 layers the vicinity it's far labeled into humans wearing a mask surrounded thru green containers and different people now not carrying an area surrounded thru red boxes. Safa Teboulbi, Seifeddine Messaoud, Mohamed Ali Hajjaji, and Abdellatif Mtibaa [7] advanced a tool in Real-Time Implementation of AI-Based Face mask Detection and Social Distancing measuring gadget for COVID-19 Prevention. This research paper makes a specialty of imposing a mask and Social Distancing Detection model as an embedded vision system. The pre-educated models just like the MobileNet, ResNet Classifier, and VGG are used. This paper includes fundamental blocks. The important block includes the 7 training and moreover the attempting out models, while the second block includes the whole framework testing. This give-up end result detects people carrying a mask and now not wearing a mask and guarantees social distancing. Xueping Su, Meng Gao, Jie Ren, Yunhong Li, Mian Dong, and Xi Liu [8] carried out mask detection and class thru deep transfer mastering. This paper describes an enterprise new set of rules for mask detection that integrates transfer learning and Efficient-Yolov3, using EfficientNet because of the truth of the backbone feature extraction network, and also you because of the truth of the loss function to restrict the number of network parameters and beautify the accuracy of mask detection. This paper divides the mask into instructions for licensed masks and unqualified masks create a mask class statistics set and advise an enterprise new mask class set of rules then combines transfer reading and MobileNet, improves the generalization of the version, and solves the rely on small information size and clean overfitting. Mohamed Almghraby and Abdelrady Okasha Elnady [9] proposed a tool for mask Detection in Real-Time using MobileNetv2. The created model for detecting face masks at some stage in this paper uses deep mastering, TensorFlow, Keras, and OpenCV. The MobilenetV2 set of rules is hired sooner or later in this paper to find out the face mask. this model dedicates 80 percent of the education dataset to education and 20% to testing and splits the training dataset into 80% education and 20% validation, primary to a final model with sixty 5 percent of the dataset for schooling, 15 percent for validation, and 20% for testing. Stochastic Gradient Descent (SGD) is hired as an optimization approach with a getting-to-know rate of 0.001 and momentum of 0.85. 8 Chhaya Gupta and Nasib Singh Gill [10] proposed a tool of Corona masks: A mask Detector for Real-Time Data. Convolutional Neural Network (CNN) set of rules is hired sooner or later of this challenge to erect faces. During this paper, a dataset has been created which includes 1238 pictures which might be divided into lessons "masks" and "no masks". Live streaming films may also moreover be used as input and extraordinary people wearing a mask and now not wearing a mask may also be detected. The convolutional neural network is knowledgeable about the dataset and it gives 95% of accuracy.

2.1 SUMMARY OF THE SURVEY

With Covid-19 cases on the rise all over the world, it's more important than ever to follow the guidelines provided by the WHO and local health authorities. The above-mentioned document was useful in visualizing and perceiving the overall project's flow, as well as defining its architecture, input, and layout to be provided for each module, as well as identifying the expected output. The above-mentioned articles are focused on recognizing persons who aren't wearing a mask. In light of all of these articles, our project is also focused on using MobileNetV2 to detect people who aren't wearing face masks.

Chapter 3
Methodology

SYSTEM DESIGN

Software design should be a method for converting requirements into the appropriate representation on a regular basis. This graphic will be used to construct the structure and targeted plan diagrams, as well as the mission method.

3.1 EXISTING SYSTEM

Yolov3 is now being used to construct a system that is aware of face masks. They introduced a spine community that will distribute more resources inside the current system, and they used Glou and focus loss to speed up the coaching process and improve performance. It has an accuracy rate of 86 percent. When using the appliance with one-of-a-kind algorithms, the accuracy will be doubled.

3.2 PROPOSED SYSTEM

The mask detection in the proposed machine has been done using MobileNet V2. In comparison to the previous system, MobileNet V2 will be able to detect face masks in a large group of men and women and provide greater accuracy. As a result of the entry dataset and therefore the segmented photo of the equation being received as output, provide an image of some persons wearing masks and now without wearing masks. The mannequin is then performed employing a camera, with the video being checked by way of body and scaled as needed. The preprocessing feature is then used to encourage the impacts of those wearing masks and those who aren't.

3.3 HARDWARE REQUIREMENTS

The following hardware specifications are required to run this model:

● RAM – 8 GB

● Operating System(OS) – Windows 10

● Processor – Intel Core i5

● Processor speed – 3.60 GHz

3.4 SOFTWARE REQUIREMENTS

The Jupyter Notebook IDE and the Python programming language were used to create this application.

● Programming Language – Python

● Python IDE – Sublime Text

● Deep Learning Framework - Tensorflow.

3.5 SYSTEM ARCHITECTURE

The structural plan denotes the many methods that will be employed in the job, while the exact layout denotes the whole operation of each module that can be used for better perception and execution of the project to achieve the desired results.

3.6 MODULE DESCRIPTION

The system can be divided into six modules.

Module1: Image Preprocessing

Module2: Data Augmentation

Module3: Model Training

Module4: Testing the Model

Module5: Image Segmentation using Mask CNN

Module6: Implementing the Model in Opencv.

Module7: Implementing Face Recognition .

Module8: Sending SMS and Email to Person who is not wearing Mask.

3.6.1 Input images

Dataset consists of two classes such as with mask and without a mask.

3.6.2 Image Preprocessing

Preprocessing is a common and basic procedure that is followed throughout this undertaking. Preprocessing has the goal of reducing unwanted distortions and improving photo facts as a whole by improving a few key photo points that may be used in conjunction with processing. The phrase photo scaling is used to describe the resizing of a digital image in in-camera work and digital photography. The phrase upscaling or decision enhancement is used in video technology to describe the amplification of digital content. The picture primitives in an exceeding vector image photo are scaled using geometric transformations with no loss of photo quality. When scaling a raster portraiture image, a brand-new snapshot of the top or a reduction in the number of pixels is expected. When the number of pixels is reduced, a visible first-class loss is expected. One of the two-dimensional instances of pattern charge conversion in connection to digital sign processing is thirteen scaling of raster graphics, which comprises the conversion of discrete indications from one charge to another. The conversion of RGB to grey involves a few steps. The gimp photo software tool has three methods for this. The lightness technique is a method that uses the most common of the most notable and least exceptional colorings. The frequent method refers to a technique that uses a simple three-color common. The luminosity approach is another cutting-edge technology. Because of reason, the novice is heavily weighted throughout this strategy. For human perception, frequency is used in addition to the weighted average. Gray to black and white conversion is accomplished entirely through a binary picture consisting of pixels with exactly one of the two colorations (black and white). Each pixel in a binary picture is recorded as one bit, either a zero or a one. Black-and-white, monochromatic, and other terms are frequently used to describe these photographs. In Photoshop, the bitmap mode is equivalent to a binary image. These pictures are utilized as masks, thresholding, and dithering in digital photo processing. Bi-level pictures can only be handled by a few input/output devices, such as laser printers, fax machines, and bilevel laptop computer displays. A bitmap, which is a dense array of bits, will be used to store a picture.

3.6.3 Data Augmentation

A mask detection tool could no longer take an input file, randomly transform it, and then return every input and altered data. Keras' image records generator takes the input picture graph and changes it into 14 modified data at random. Data augmentation may also be defined as a set of strategies for injecting random oscillations and perturbations and generating a new schooling sample from an existing one. The model's generalizability is increased by the way facts augmentation is used.

3.6.4 Model Training

For cellular visual recognition, MobileNetV2 improves on MobileNetV1 in areas like as classification, item identification, and semantic segmentation. MobileNetV2 is provided as part of the TensorFlowSlim image classification library. MobileNetV2 modules with pre-trained checkpoints are also available as TFHub modules. Two new architectural elements have been added to MobileNetV2:

● linear bottlenecks between layers

● fast links between bottlenecks

Convolutional Neural Network

Convolutional Neural Networks are an artificial neural community that is optimized to technique pixel information for image cognizance and processing (CNN). The Convolutional Neural Network is the fundamental and building block of the computer's imaginative and prophetic assignment of image segmentation.

● **Convolutional layer**: Filters and kernels are used to create a characteristic map that summarizes the entered image.

● **Pooling layer**: This layer is used to summarize the existence of capabilities in function map patches, which aids in function map downsampling.

● **Fully connected layer**: Every neuron in a single layer is hooked up to every neuron in every other layer.

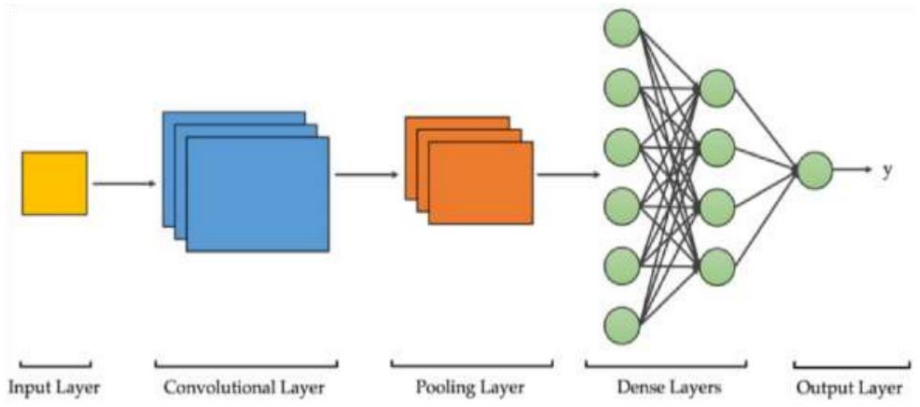


Fig 3.1:- Testing the Model Evaluation Metrics

Accuracy

The degree of general correctly identified samples out of all the samples is known as accuracy. It's characterized as follows:

TP+TN/TP+FP+FN+TN = Accuracy

Where True effectiveness (TP) indicates that something has been successfully acknowledged. False effective (FP) = erroneously identified True negative (TN) indicates that something has been successfully rejected. False-negative (FN) indicates that something was rejected mistakenly.

Precision

The amount of effective magnificence forecasts that obviously belong to the effective magnificence is determined using a precision approach.

Precision = TP/TP+FP

Recall

The number of effective magnificence predictions built from all effective samples within the dataset is determined using the recall technique.

Recall = TP/TP+FN F1-Score F1- Score is the common suggest of Precision and Recall F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Macro Average

The approach is quite simple. Simply adopt the common sense of precision and maintain the device with distinct setups in mind. Then Macro-common may be honestly the common recommendation of Macro-common precision and macro-common remember.

Weighted Average

The F1 Scores are generated for each label, and then their not unusual position is weighted using a guide - that is, the diversity of real situations for each label. It has the potential to produce an F1 Score that is no longer amid precision and keep in mind.

3.6.6 Image Segmentation using Mask CNN

The Mask R-CNN is a CNN for image segmentation and event segmentation that is built on top of a quicker R-CNN that can come across objects and limits. Instance segmentation or instance recognition is the process of detecting all objects in a photograph and segmenting each event. It emerges as a result of the harsh realities of object detection, localization, and classification penalties. During this method of segmentation, a clear separation between each item classified as a similar circumstance is noted. Everyone is treated as a single entity during the event segmentation process. It's also known as foreground segmentation because it works on the picture's subject matters rather than the background-CNN and can produce two outputs for each object, a class label, and a bounding field offset, whereas Mask R-CNN can produce three outputs, including article masks in addition to the class label and bounding container offset. The more masks output is exclusive from the opposite two outputs, implying that the finer the spatial design of an item, the more masks output is required. Mask R-CNN is a faster version of R-CNN that includes an output for object masks in addition to current outputs such as classification labels and bounding boxes.

3.6.7 Implementing the Model in OpenCV

Finally, the model is constructed with the help of a webcam, which reads the video frame by frame and resizes it as needed. Then, in connection with the accuracy in %, the preprocessing feature is known as achieving the final result of human people wearing masks and now not wearing a mask.

3.6.8 Implementing the Face Recognition

The method of detecting a human face using technology is known as facial reputation. Biometrics is used in a facial reputation device to map face functions from a picture or video. To find a match, it compares the information to a database of known faces.

Step 1: Face detection

Both by myself and in a crowd, the digital digicam recognizes and locates a snapshot of a face. The image might also show the character searching ahead of time or in profile.

Step 2: Face analysis

Following that, a snapshot of the face is taken and examined. Because it may be simpler to integrate a 2D image with public photographs or those in a database, most facial interest period is based on 2D as a potential rather than 3D images. The gap between your eyes, the depth of your eye sockets, the space from the forehead to the chin, the form of your cheekbones, and the outline of your lips, ears, and chin are all crucial factors that the program interprets. The objective is to learn the facial landmarks, which are perhaps the most important aspect of facial recognition.

Step 3: Converting the image to data

Face capture mode converts analog facts (a face) into a collection of digital data (data) depending on the person's facial traits. The appraisal of your face has evolved into a mathematical formula. A face print is a name for the digital code. Every character has their own faceprint in the same manner as thumbprints are unique.

Step 4: Finding a match

After that, your faceprint is compared to a database of other recognized faces. Any photo tagged with a person's name on Facebook becomes a part of Facebook's database, which may be used for face recognition as well. Self-discipline is achieved when your faceprint matches an image in a facial interest database.

The face center of attention is the most natural of all the biometric measures. This makes intuitive sense because we normally recognize ourselves and others by gazing out at faces rather than thumbprints and irises.

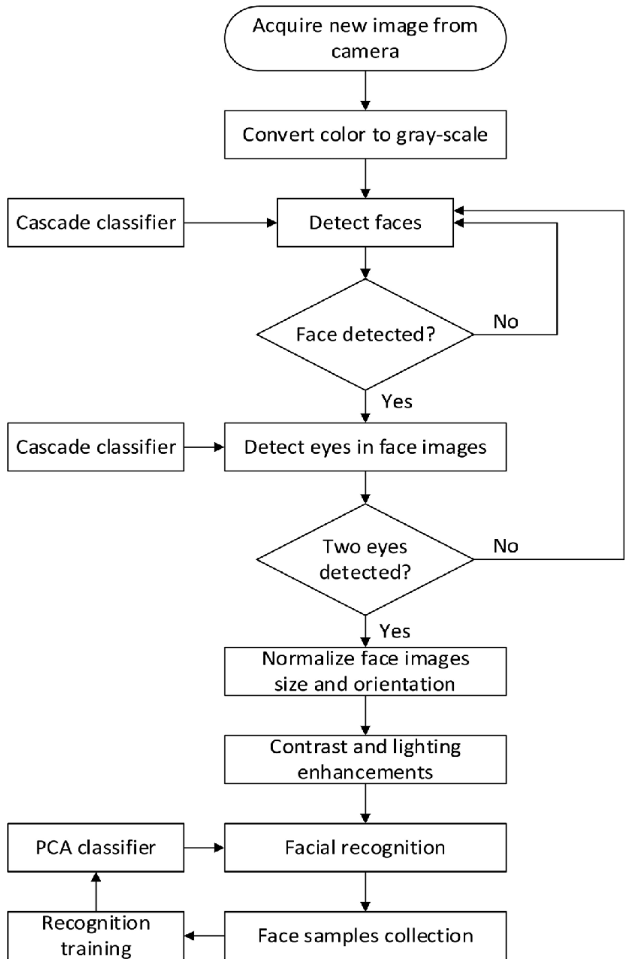


Fig 3.2: - Flow chart for implementation of facial recognition.

3.6.9 Sending SMS and Email to Person who is not wearing Mask

We're employing a face recognition system to identify folks who aren't wearing masks, which will aid us in locating their information. We will send a preventive message in the form of SMS and email to such persons in order to reward them, advising them to wear masks in public. We utilized Twilio for delivering SMS, which provides programmable communication tools for sending and receiving text messages via its web service APIs, and the smtplib module for email, which simply establishes an SMTP session for the client object and can then be used to send mail to another computer.

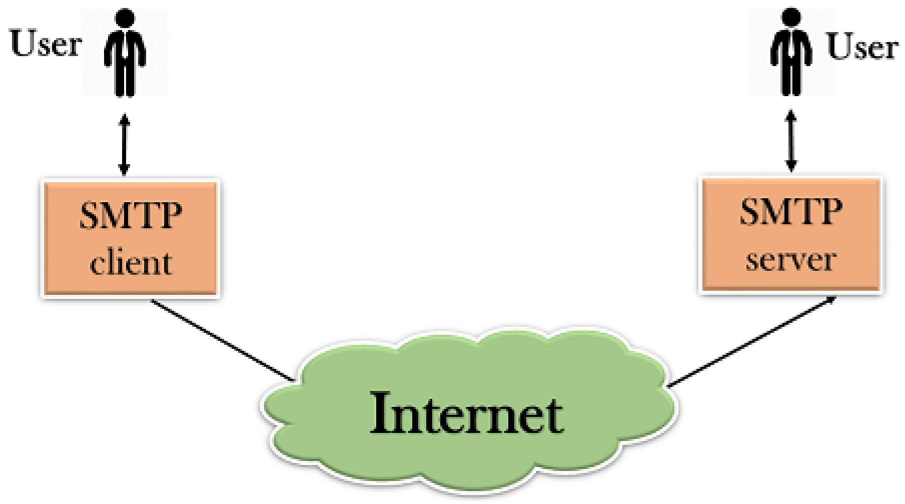
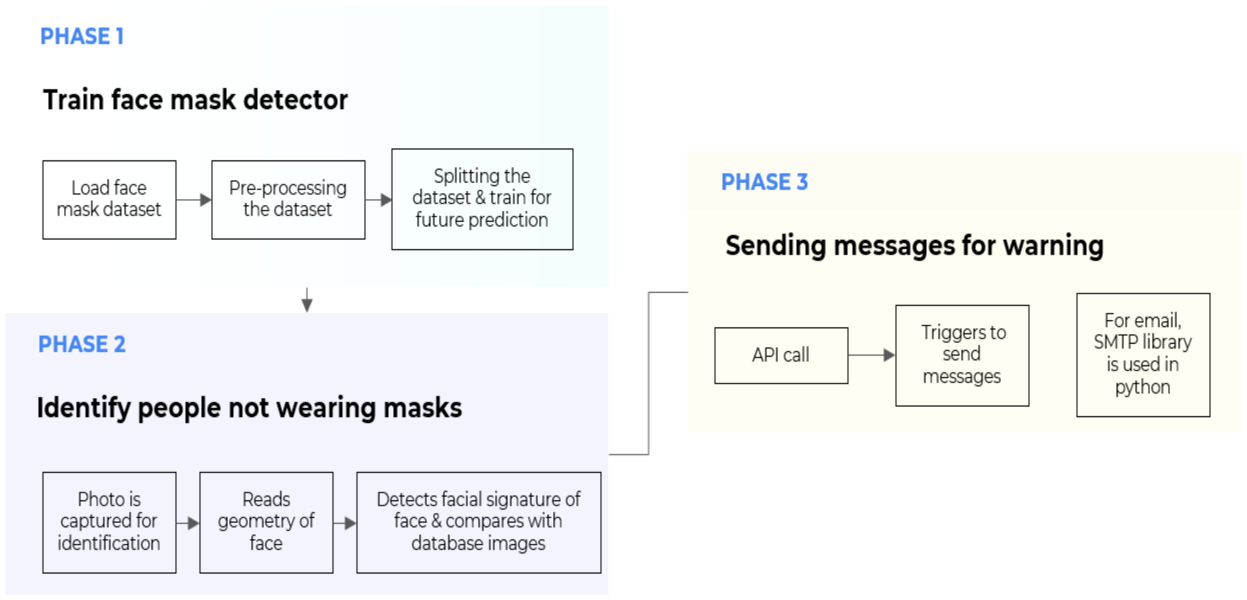


Fig 3.3:- SMTP protocol

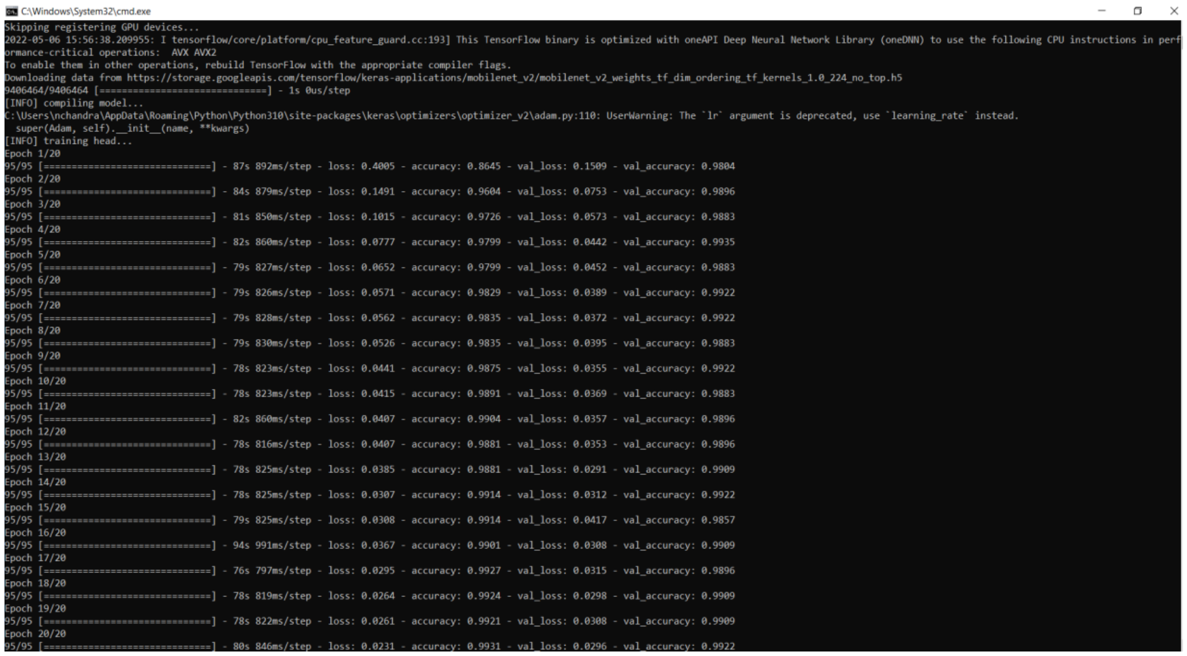
3.7 General Frame of connection of the above modules


PHASE 1
Train face mask detector
Load face mask dataset
Pre-processing the dataset
Splitting the dataset & train for future prediction
PHASE 2
Identify people not wearing masks
Photo is captured for identification
Reads geometry of face
Detects facial signature of face & compares with database images
PHASE 3
Sending messages for warning
API call
Triggers to send messages
For email, SMTP library is used in python

Chapter 4

Experimental Results

1.Testing the Model



C:\Windows\System32\cmd.exe

```
Skipping registering GPU devices...
2022-05-06 15:56:38.209955: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224_no_top.h5
9406464/9406464 [==============================] - 1s 0us/step
[INFO] compiling model...
C:\Users\nchandra\AppData\Roaming\Python\Python310\site-packages\keras\optimizers\optimizer_v2\adam.py:110: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
[INFO] training head...
Epoch 1/20
95/95 [==============================] - 87s 892ms/step - loss: 0.4005 - accuracy: 0.8645 - val_loss: 0.1509 - val_accuracy: 0.9804
Epoch 2/20
95/95 [==============================] - 84s 879ms/step - loss: 0.1491 - accuracy: 0.9604 - val_loss: 0.0753 - val_accuracy: 0.9896
Epoch 3/20
95/95 [==============================] - 81s 850ms/step - loss: 0.1015 - accuracy: 0.9726 - val_loss: 0.0573 - val_accuracy: 0.9883
Epoch 4/20
95/95 [==============================] - 82s 860ms/step - loss: 0.0777 - accuracy: 0.9799 - val_loss: 0.0442 - val_accuracy: 0.9935
Epoch 5/20
95/95 [==============================] - 79s 827ms/step - loss: 0.0652 - accuracy: 0.9799 - val_loss: 0.0452 - val_accuracy: 0.9883
Epoch 6/20
95/95 [==============================] - 79s 826ms/step - loss: 0.0571 - accuracy: 0.9829 - val_loss: 0.0389 - val_accuracy: 0.9922
Epoch 7/20
95/95 [==============================] - 79s 828ms/step - loss: 0.0562 - accuracy: 0.9835 - val_loss: 0.0372 - val_accuracy: 0.9922
Epoch 8/20
95/95 [==============================] - 79s 830ms/step - loss: 0.0526 - accuracy: 0.9835 - val_loss: 0.0395 - val_accuracy: 0.9883
Epoch 9/20
95/95 [==============================] - 78s 823ms/step - loss: 0.0441 - accuracy: 0.9875 - val_loss: 0.0355 - val_accuracy: 0.9922
Epoch 10/20
95/95 [==============================] - 78s 823ms/step - loss: 0.0415 - accuracy: 0.9891 - val_loss: 0.0369 - val_accuracy: 0.9883
Epoch 11/20
95/95 [==============================] - 82s 860ms/step - loss: 0.0407 - accuracy: 0.9904 - val_loss: 0.0357 - val_accuracy: 0.9896
Epoch 12/20
95/95 [==============================] - 78s 816ms/step - loss: 0.0407 - accuracy: 0.9881 - val_loss: 0.0353 - val_accuracy: 0.9896
Epoch 13/20
95/95 [==============================] - 78s 825ms/step - loss: 0.0385 - accuracy: 0.9881 - val_loss: 0.0291 - val_accuracy: 0.9909
Epoch 14/20
95/95 [==============================] - 78s 825ms/step - loss: 0.0307 - accuracy: 0.9914 - val_loss: 0.0312 - val_accuracy: 0.9922
Epoch 15/20
95/95 [==============================] - 79s 825ms/step - loss: 0.0308 - accuracy: 0.9914 - val_loss: 0.0417 - val_accuracy: 0.9857
Epoch 16/20
95/95 [==============================] - 94s 991ms/step - loss: 0.0367 - accuracy: 0.9901 - val_loss: 0.0308 - val_accuracy: 0.9909
Epoch 17/20
95/95 [==============================] - 76s 797ms/step - loss: 0.0295 - accuracy: 0.9927 - val_loss: 0.0315 - val_accuracy: 0.9896
Epoch 18/20
95/95 [==============================] - 78s 819ms/step - loss: 0.0264 - accuracy: 0.9924 - val_loss: 0.0298 - val_accuracy: 0.9909
Epoch 19/20
95/95 [==============================] - 78s 822ms/step - loss: 0.0261 - accuracy: 0.9921 - val_loss: 0.0308 - val_accuracy: 0.9909
Epoch 20/20
95/95 [==============================] - 80s 846ms/step - loss: 0.0231 - accuracy: 0.9931 - val_loss: 0.0296 - val_accuracy: 0.9922
```

Fig 4.1: - Testing the Model

2. Detecting Face Mask

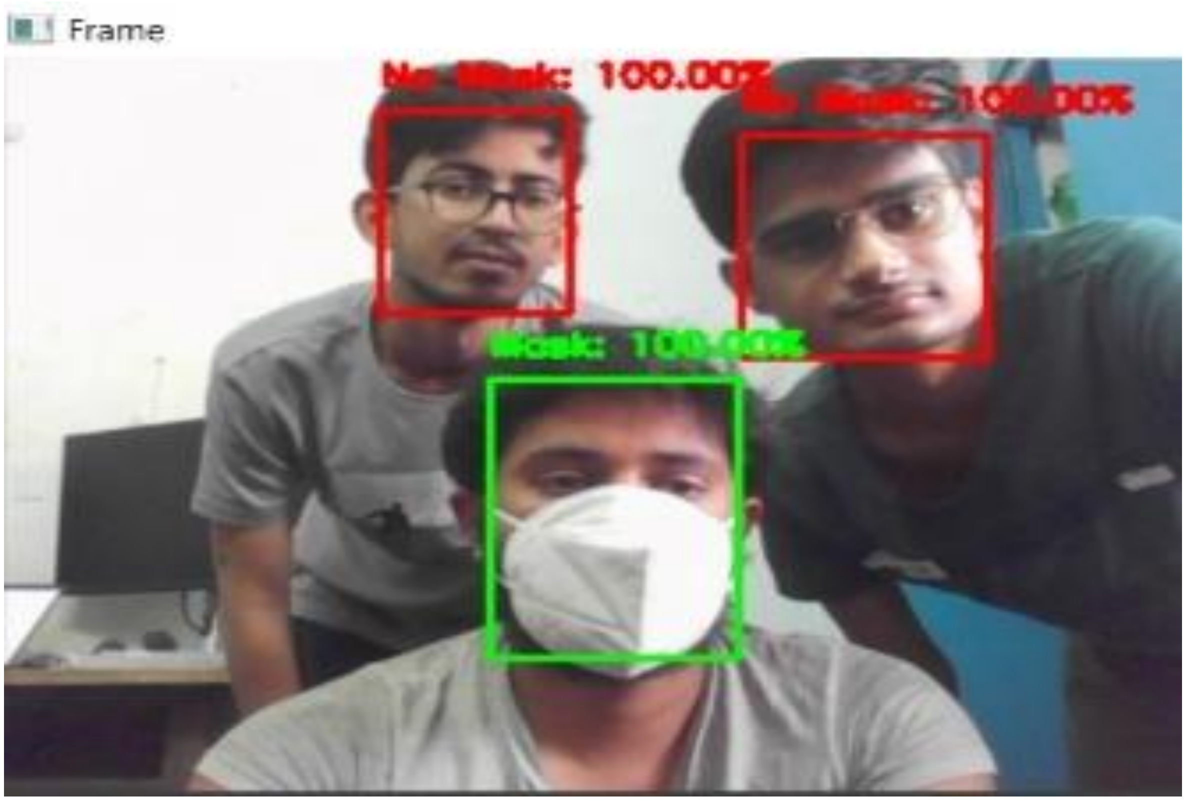
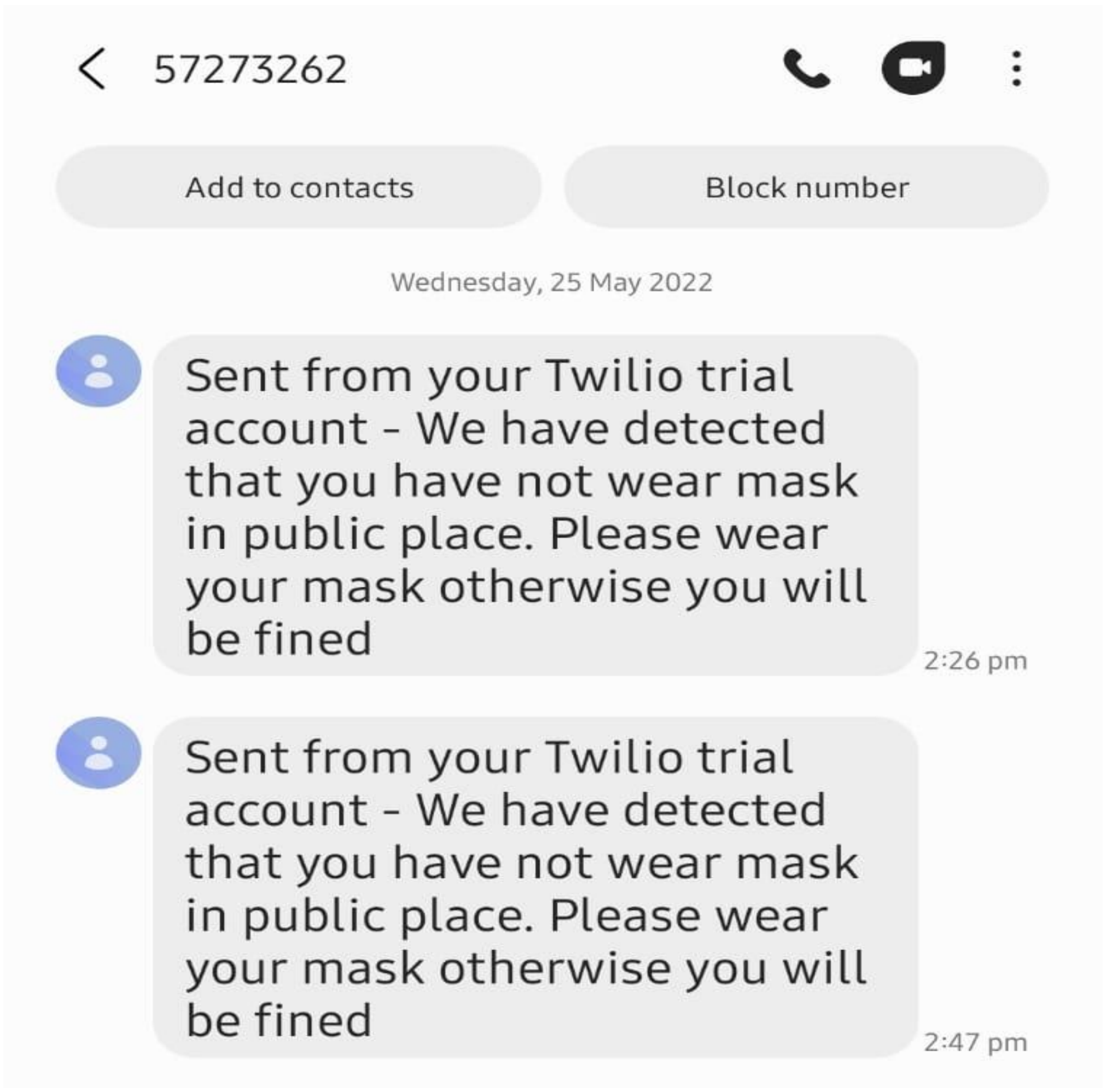


Fig 4.2: - Detecting people wearing masks

3. Sending Messages via SMS/Email



Wednesday, 25 May 2022

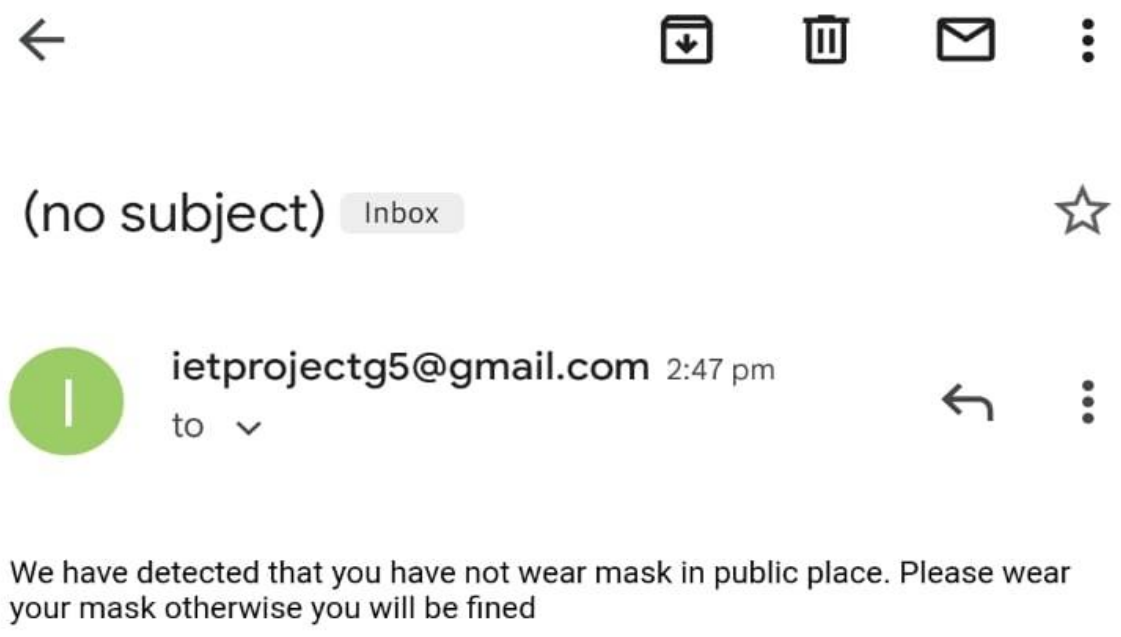
Sent from your Twilio trial account - We have detected that you have not wear mask in public place. Please wear your mask otherwise you will be fined

2:26 pm

Sent from your Twilio trial account - We have detected that you have not wear mask in public place. Please wear your mask otherwise you will be fined

2:47 pm

Fig 4.3:- Sending text messages to people not wearing masks


(no subject) Inbox

ietprojectg5@gmail.com 2:47 pm
to

We have detected that you have not wear mask in public place. Please wear your mask otherwise you will be fined

Fig 4.4:- Email template sent to people

Chapter 5
Conclusions

5.1 Conclusions

In this mask detection, we used the MobilenetV2 set of rules to correctly distinguish persons wearing masks and those without masks, as well as send an email to those involved. Its overall performance in photographs is generally accurate, and our detecting effects were also rather accurate. This detection will be utilized for video flow or digital digicam-fed inputs, as well. This can be used in places of work and establishments by way of training the database with personnel photos or students' photos and by way of face reputation, and the character is diagnosed by way of a mobile number that is unique for each person, and other information about the individual is obtained from the database, and it will be simple to tell that specific character or beneficial for taking any actions concerned. The suggested version can be improved by including other aspects such as the number of persons and the social distance between individuals.

5. 2 Future Work

Facial recognition with a mask is becoming more and more important over the past year thanks to the spread of the COVID-19 virus. All through this project, used the MobilenetV2 algorithm and different deep gaining knowledge of methods to spot humans no longer sporting a mask. This situation has been tested using a webcam and entering a database, In the future, this function is often used in conjunction with various AI methodologies and may be performed by units such as Raspberry Pi, Autonomous drone structures, etc., to improve efficiency and reduce vision time taken to ensure that people no longer play the mask. and in addition we prefer to combine fines for those who do not wear the mask properly and measure social distance.

This project will be very useful and can be used in many places like hospitals, airports, schools offices, shops,, colleges, supermarkets, theaters, temples, houses, etc., and may also be used to manage free Covid events.

References

[1] Xinbei Jiang, Tianshan Gao, Zichen Zhu, and Yukang Zhao., Real-Time Face Mask Detection Method Based on YOLOv3, Electronics, 7, pp.130-147, 2021.

[2] Samuel Ady Sanjaya and Suryo Adi Rakhmawan., Face Mask Detection Using MobileNetV2, International Journal of Engineering and Advanced Technology, 4, pp.2249-8958, 2021.

[3] G. Jignesh Chowdary, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal, Face Mask Detection using Transfer Learning of InceptionV3, IEEE Access, 20, pp. 456-665, 2021.

[4] Shilpa Sethi, Mamtha Kathuria and Trillok Kaushik., Face mask detection using deep learning, Multimedia Tools and Application, 8, pp.42-72, 2020.

[5] Chunli Qin, Demin Yao, Yonghong Shi and Zhijian Song, "Computer-aided detection in chest radiography based on artificial intelligence: a survey," BioMedical Engineering Online, vol. 17, pp. 1-23, 2018.

[6] Riya Chiragkumar Shah and Rutva Jignesh Shah., Detection of Face Mask using Convolutional Neural Network, Mobile Information System, 43, pp.382-487, 2019.

[7] Safa Teboulbi, Seifeddine Messaoud, Mohamed Ali Hajjaji, and Abdellatif Mtibaa., Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention, Scientific Programming, 32, pp.167-254, 2021.

[8] Xueping Su, Meng Gao, Jie Ren, Yunhong Li, Mian Dong, and Xi Liu., Face mask detection and classification through deep transfer learning, Multimedia Tools and Applications, 53, pp.11042-11772, 2021.

[9] Mohamed Almghraby and Abdelrady Okasha Elnady., Face Mask Detection in Real-Time using MobileNetv2, International Journal Of Engineering and Advanced Technology, 6, pp.49-89, 2021

[10] Chhaya Gupta and Nasib Singh Gill., Corona mask: A Face Mask Detector for Real-Time Data, International Journal of Advanced Trends in Computer Science and Engineering, 9, pp.2278-3091, 2021.

[11] Face Mask Dataset – Kaggle Repository.

[12] Source Code Of Project – https://github.com/tushar007giri/Detecting-Mask-over-faces-and-alerting

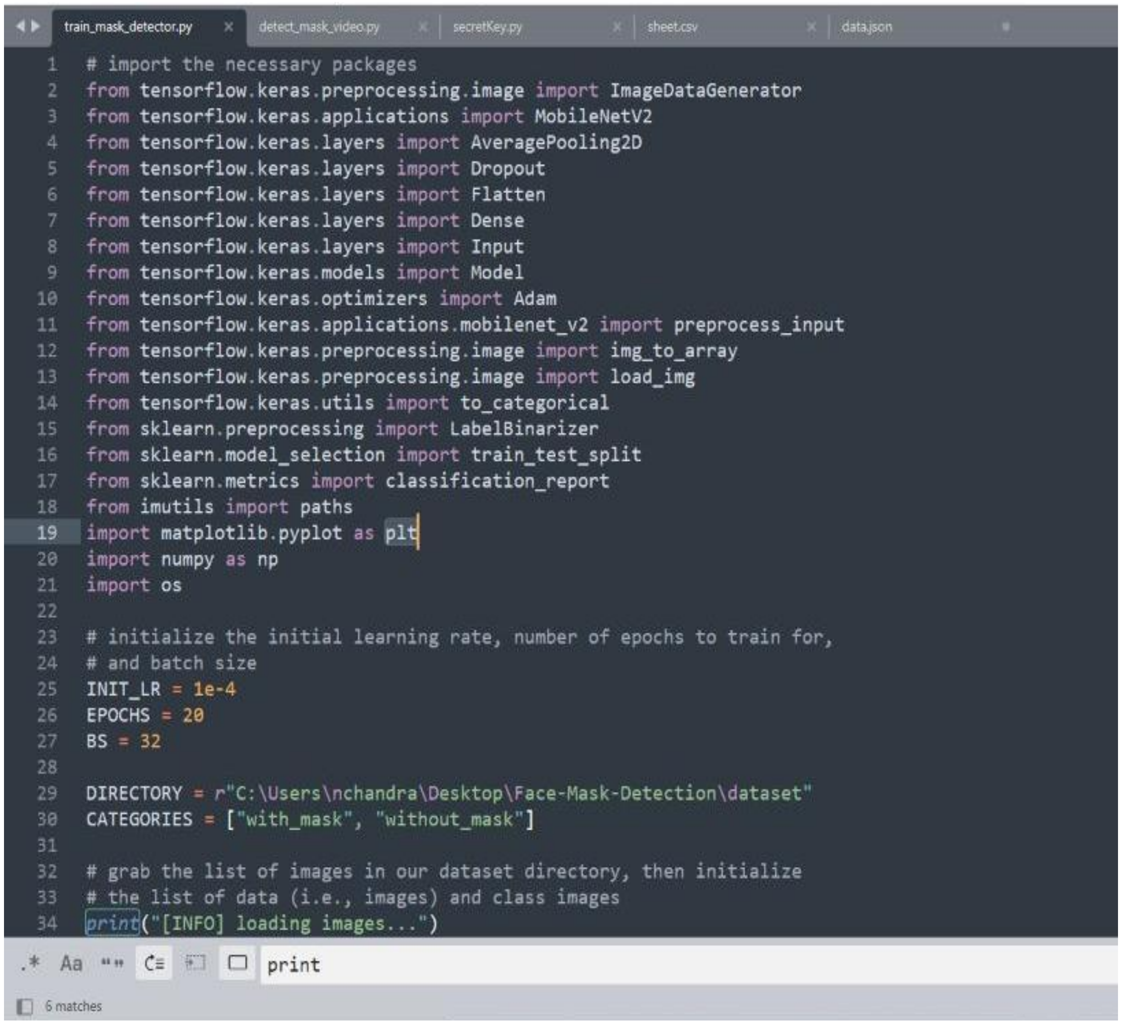
Annexure

Source Code of Project –

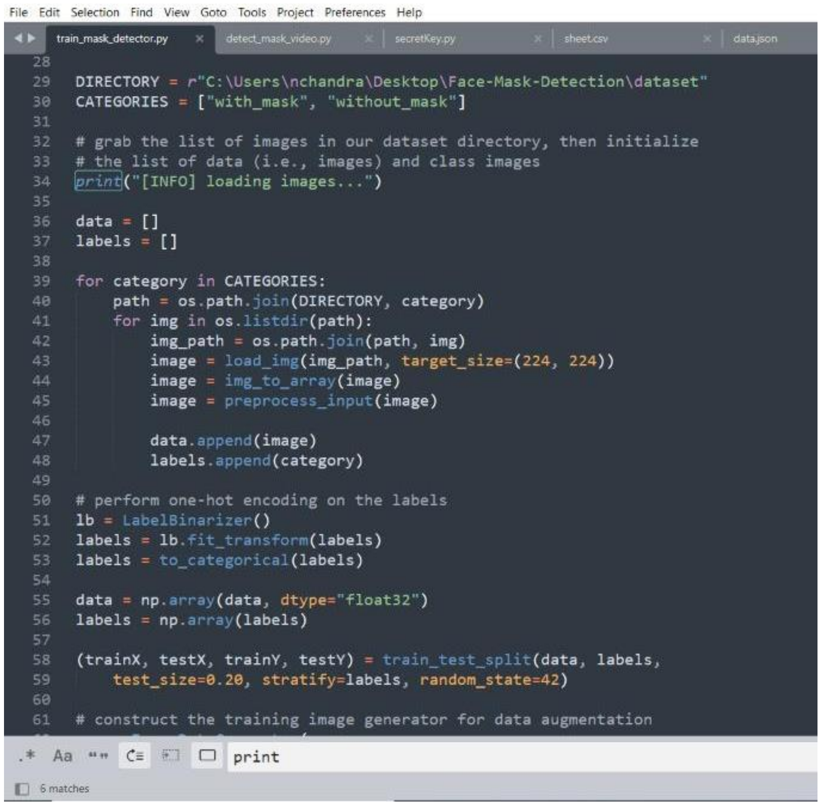
https://github.com/tushar007giri/Detecting-Mask-over-faces-and-alerting

Annexure A

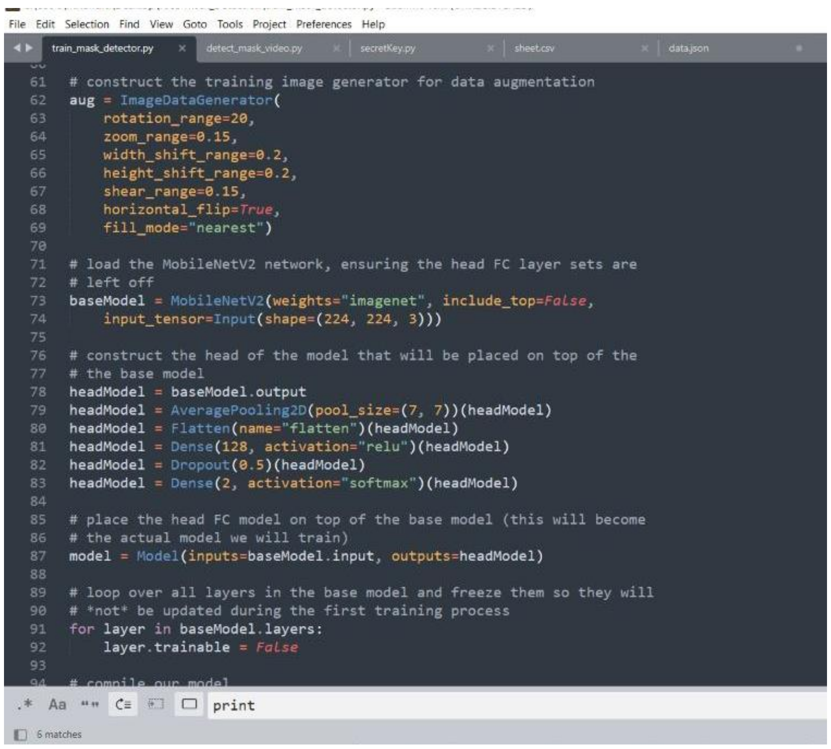
Train mask detector



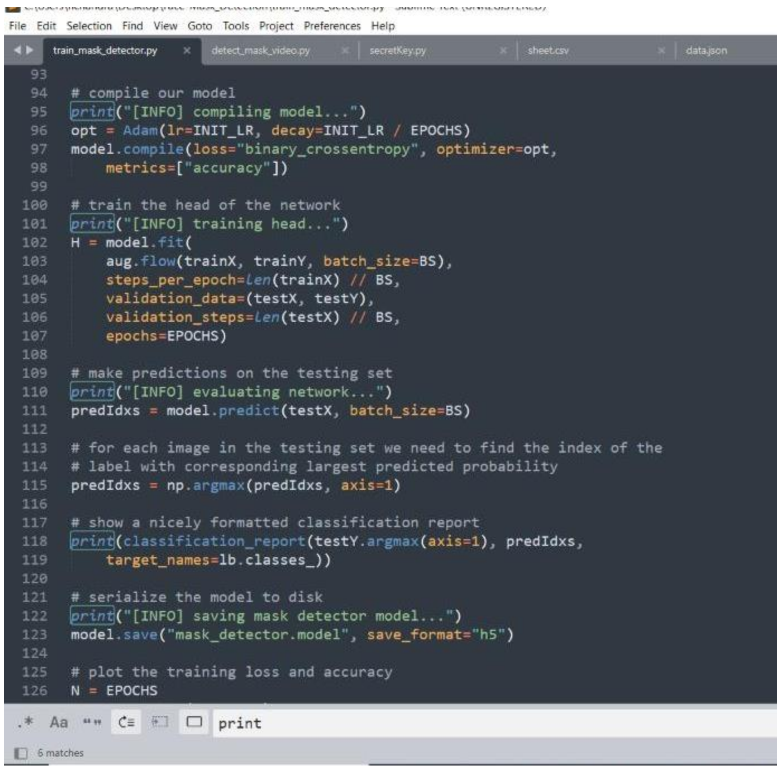
```python
# import the necessary packages
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
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 os

# initialize the initial learning rate, number of epochs to train for,
# and batch size
INIT_LR = 1e-4
EPOCHS = 20
BS = 32

DIRECTORY = r"C:\Users\nchandra\Desktop\Face-Mask-Detection\dataset"
CATEGORIES = ["with_mask", "without_mask"]

# grab the list of images in our dataset directory, then initialize
# the list of data (i.e., images) and class images
print("[INFO] loading images...")
```



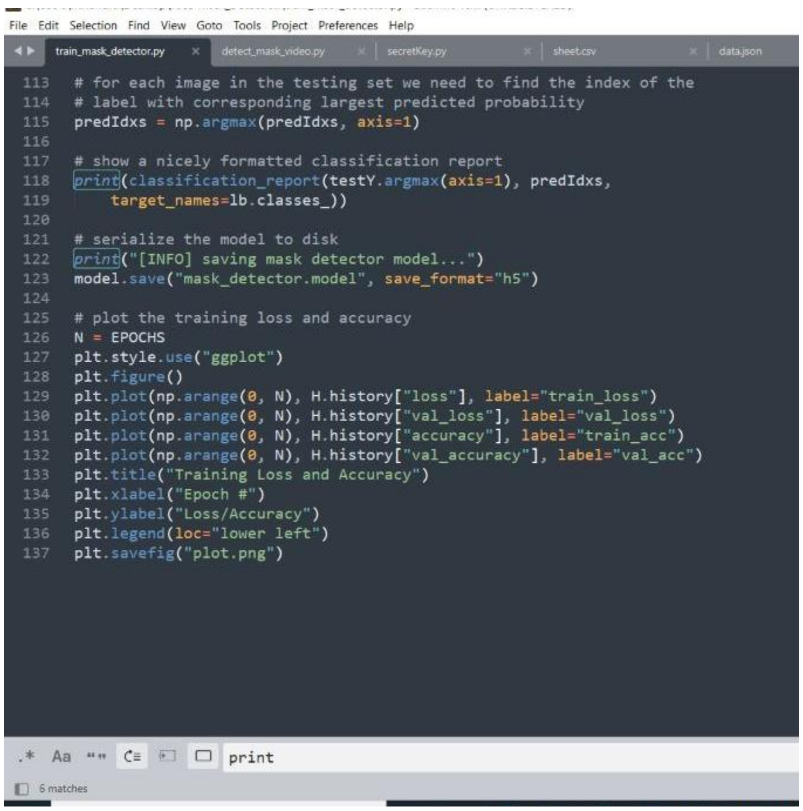
```python
DIRECTORY = r"C:\Users\nchandra\Desktop\Face-Mask-Detection\dataset"
CATEGORIES = ["with_mask", "without_mask"]

# grab the list of images in our dataset directory, then initialize
# the list of data (i.e., images) and class images
print("[INFO] loading images...")

data = []
labels = []

for category in CATEGORIES:
    path = os.path.join(DIRECTORY, category)
    for img in os.listdir(path):
        img_path = os.path.join(path, img)
        image = load_img(img_path, target_size=(224, 224))
        image = img_to_array(image)
        image = preprocess_input(image)

        data.append(image)
        labels.append(category)

# perform one-hot encoding on the labels
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

data = np.array(data, dtype="float32")
labels = np.array(labels)

(trainX, testX, trainY, testY) = train_test_split(data, labels,
    test_size=0.20, stratify=labels, random_state=42)

# construct the training image generator for data augmentation
```



```python
# construct the training image generator for data augmentation
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")

# load the MobileNetV2 network, ensuring the head FC layer sets are
# left off
baseModel = MobileNetV2(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

# construct the head of the model that will be placed on top of the
# the base model
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)

# place the head FC model on top of the base model (this will become
# the actual model we will train)
model = Model(inputs=baseModel.input, outputs=headModel)

# loop over all layers in the base model and freeze them so they will
# *not* be updated during the first training process
for layer in baseModel.layers:
    layer.trainable = False

# compile our model
```



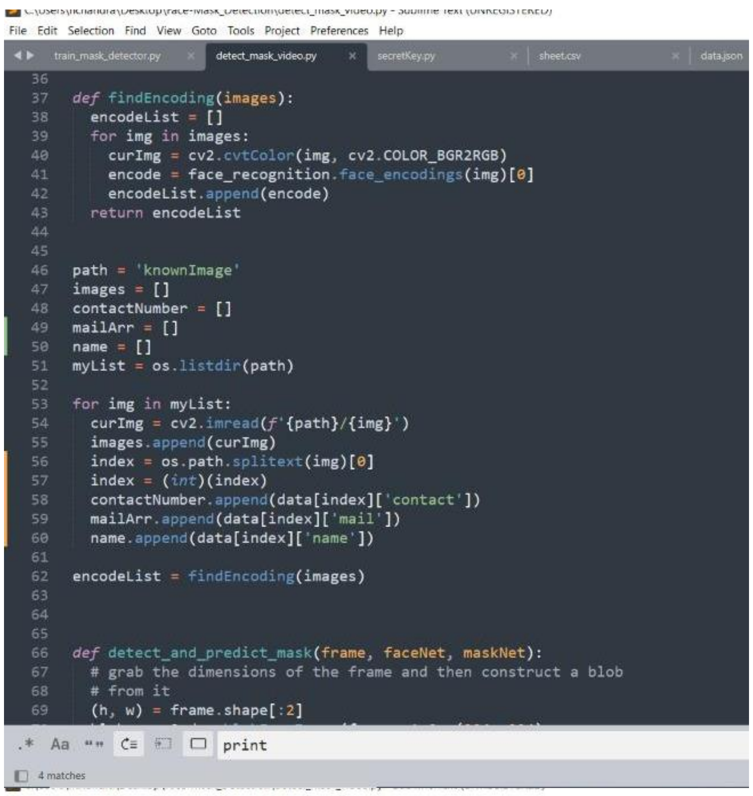
```python

# compile our model
print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,
    metrics=["accuracy"])

# train the head of the network
print("[INFO] training head...")
H = model.fit(
    aug.flow(trainX, trainY, batch_size=BS),
    steps_per_epoch=len(trainX) // BS,
    validation_data=(testX, testY),
    validation_steps=len(testX) // BS,
    epochs=EPOCHS)

# make predictions on the testing set
print("[INFO] evaluating network...")
predIdxs = model.predict(testX, batch_size=BS)

# for each image in the testing set we need to find the index of the
# label with corresponding largest predicted probability
predIdxs = np.argmax(predIdxs, axis=1)

# show a nicely formatted classification report
print(classification_report(testY.argmax(axis=1), predIdxs,
    target_names=lb.classes_))

# serialize the model to disk
print("[INFO] saving mask detector model...")
model.save("mask_detector.model", save_format="h5")

# plot the training loss and accuracy
N = EPOCHS
```



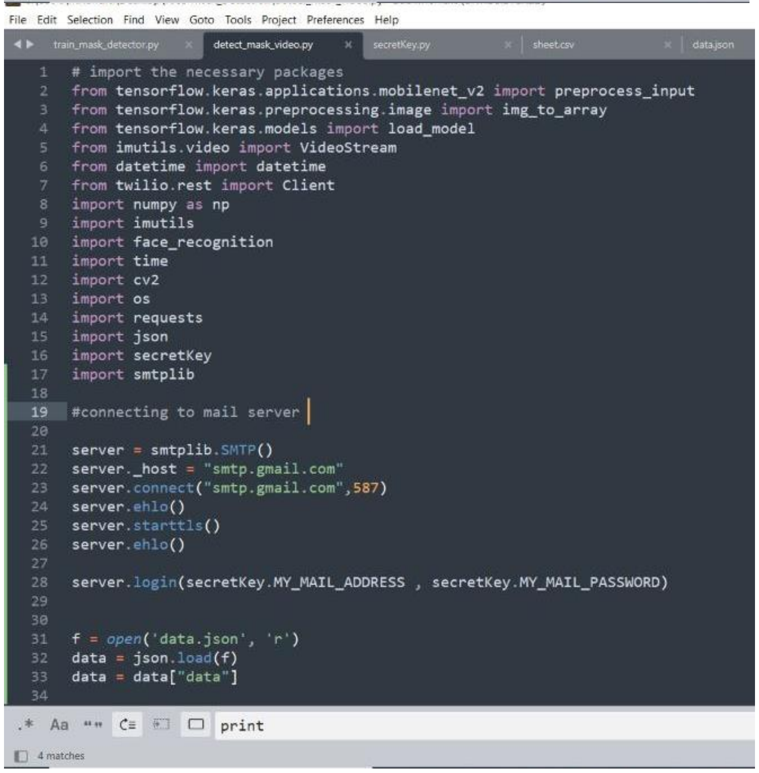
```python
# for each image in the testing set we need to find the index of the
# label with corresponding largest predicted probability
predIdxs = np.argmax(predIdxs, axis=1)

# show a nicely formatted classification report
print(classification_report(testY.argmax(axis=1), predIdxs,
    target_names=lb.classes_))

# serialize the model to disk
print("[INFO] saving mask detector model...")
model.save("mask_detector.model", save_format="h5")

# plot the training loss and accuracy
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("plot.png")
```

Annexure B

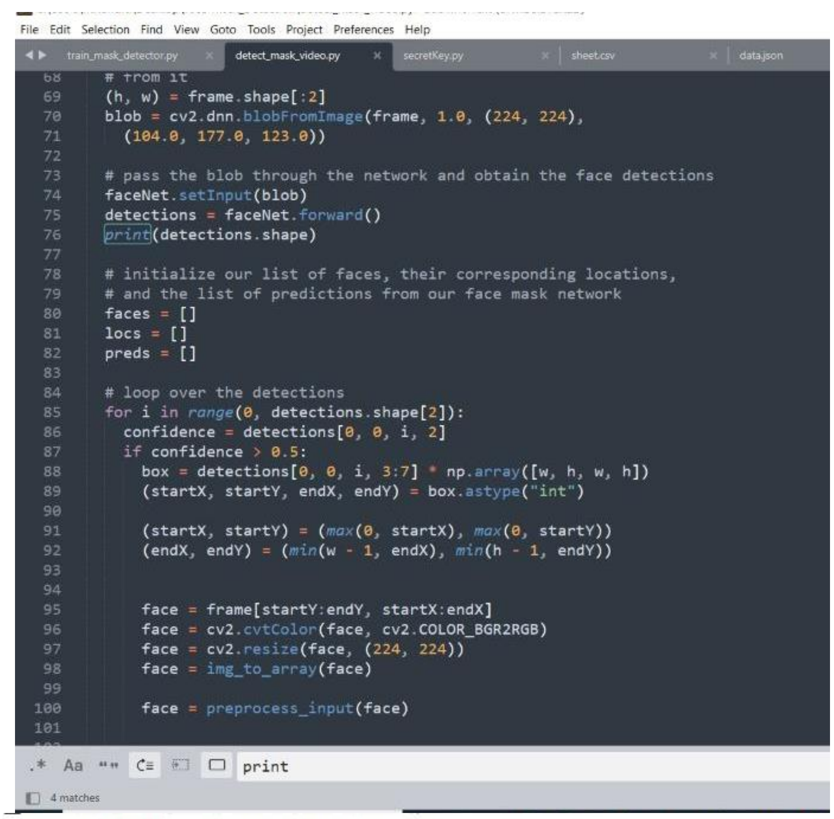
Indentifying and matching the details of the persons who have not worn masks.



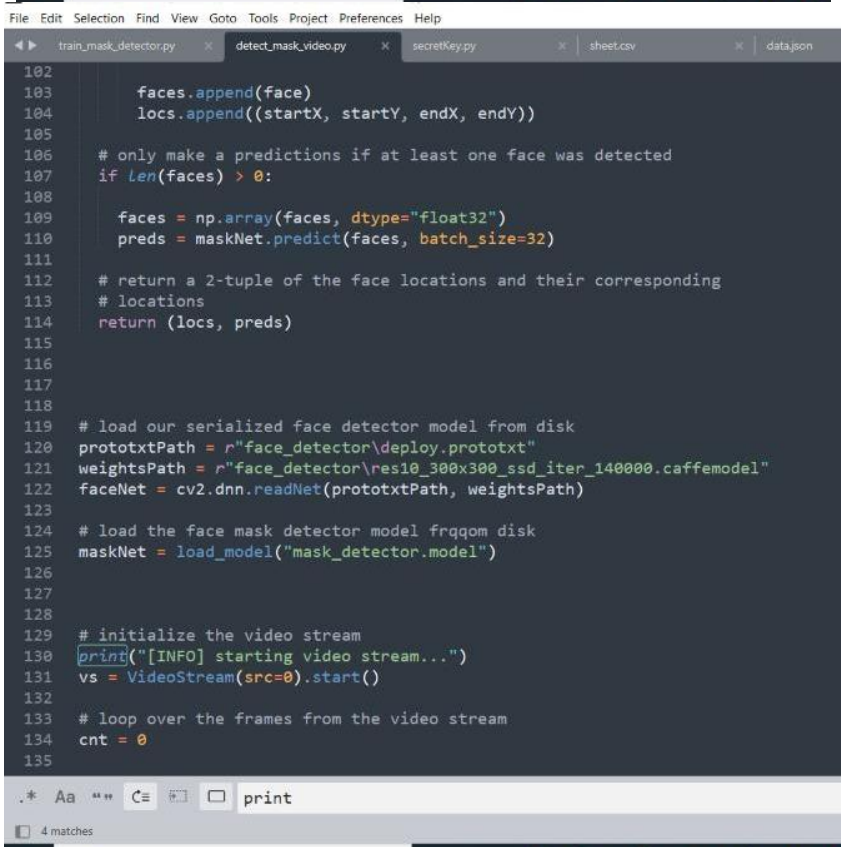
```
def findEncoding(images):
  encodeList = []
  for img in images:
    curImg = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    encode = face_recognition.face_encodings(img)[0]
    encodeList.append(encode)
  return encodeList


path = 'knownImage'
images = []
contactNumber = []
mailArr = []
name = []
myList = os.listdir(path)

for img in myList:
  curImg = cv2.imread(f'{path}/{img}')
  images.append(curImg)
  index = os.path.splitext(img)[0]
  index = (int)(index)
  contactNumber.append(data[index]['contact'])
  mailArr.append(data[index]['mail'])
  name.append(data[index]['name'])

encodeList = findEncoding(images)



def detect_and_predict_mask(frame, faceNet, maskNet):
  # grab the dimensions of the frame and then construct a blob
  # from it
  (h, w) = frame.shape[:2]
```



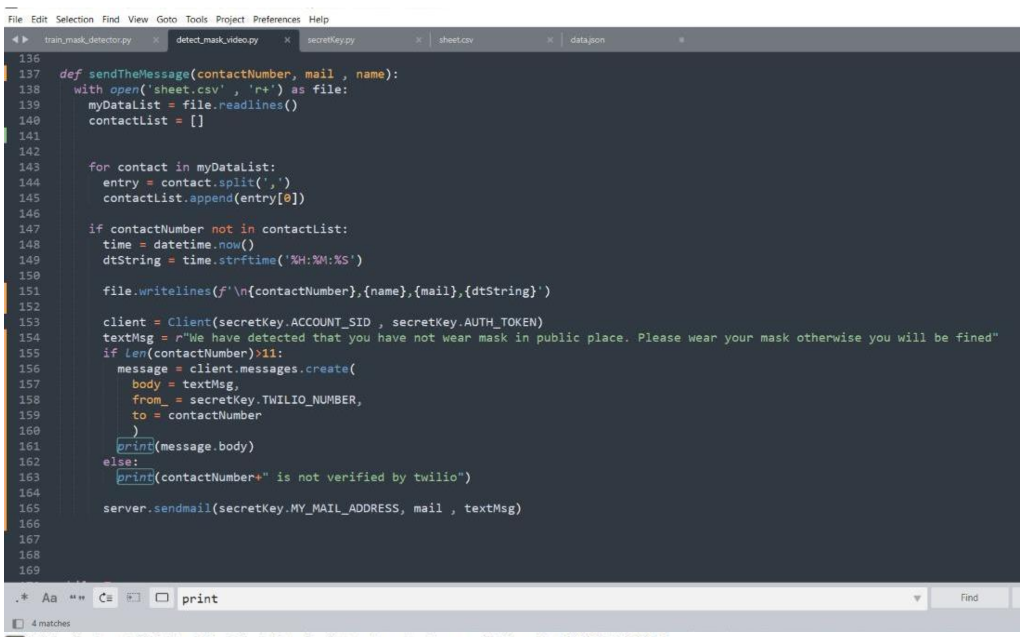
```
# import the necessary packages
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
from datetime import datetime
from twilio.rest import Client
import numpy as np
import imutils
import face_recognition
import time
import cv2
import os
import requests
import json
import secretKey
import smtplib

#connecting to mail server

server = smtplib.SMTP()
server._host = "smtp.gmail.com"
server.connect("smtp.gmail.com",587)
server.ehlo()
server.starttls()
server.ehlo()

server.login(secretKey.MY_MAIL_ADDRESS , secretKey.MY_MAIL_PASSWORD)


f = open('data.json', 'r')
data = json.load(f)
data = data["data"]
```



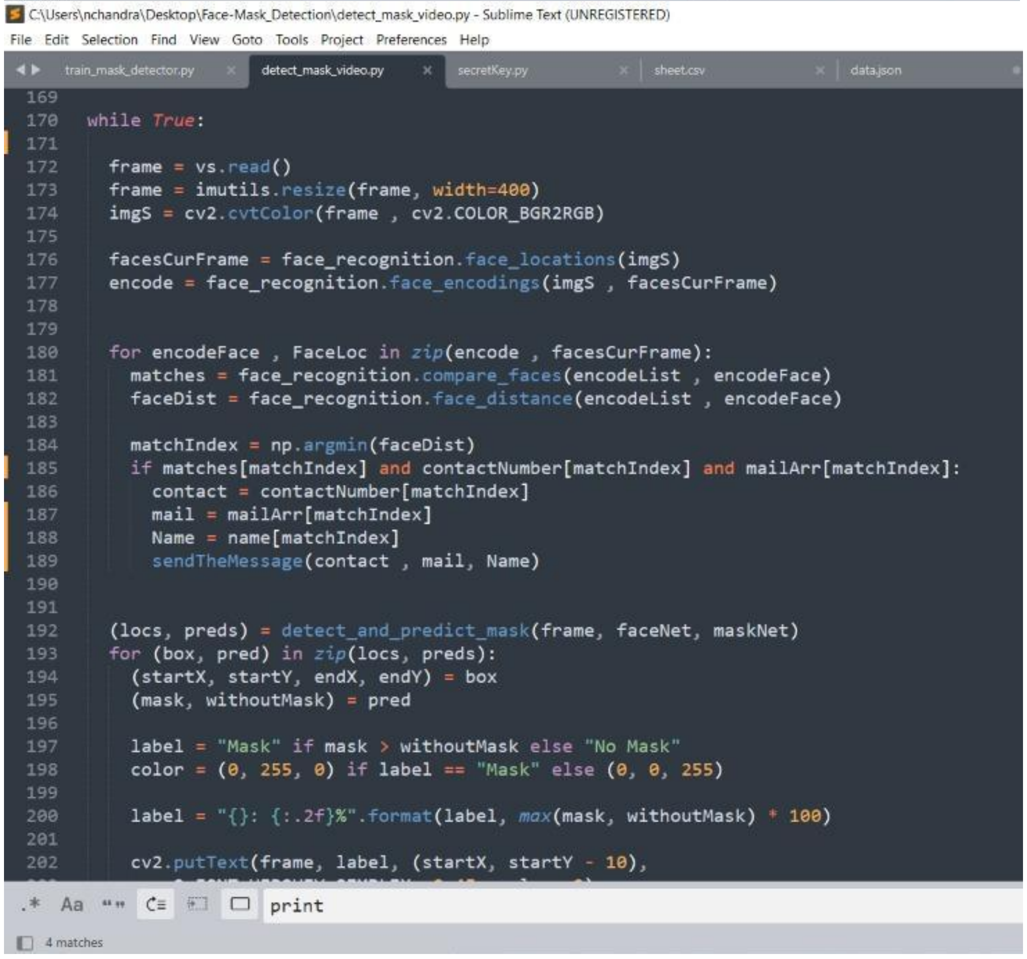
```
  # from it
  (h, w) = frame.shape[:2]
  blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
    (104.0, 177.0, 123.0))

  # pass the blob through the network and obtain the face detections
  faceNet.setInput(blob)
  detections = faceNet.forward()
  print(detections.shape)

  # initialize our list of faces, their corresponding locations,
  # and the list of predictions from our face mask network
  faces = []
  locs = []
  preds = []

  # loop over the detections
  for i in range(0, detections.shape[2]):
    confidence = detections[0, 0, i, 2]
    if confidence > 0.5:
      box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
      (startX, startY, endX, endY) = box.astype("int")

      (startX, startY) = (max(0, startX), max(0, startY))
      (endX, endY) = (min(w - 1, endX), min(h - 1, endY))


      face = frame[startY:endY, startX:endX]
      face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
      face = cv2.resize(face, (224, 224))
      face = img_to_array(face)

      face = preprocess_input(face)

```



```

      faces.append(face)
      locs.append((startX, startY, endX, endY))

  # only make a predictions if at least one face was detected
  if len(faces) > 0:

    faces = np.array(faces, dtype="float32")
    preds = maskNet.predict(faces, batch_size=32)

  # return a 2-tuple of the face locations and their corresponding
  # locations
  return (locs, preds)




# load our serialized face detector model from disk
prototxtPath = r"face_detector\deploy.prototxt"
weightsPath = r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model frqqom disk
maskNet = load_model("mask_detector.model")



# initialize the video stream
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()

# loop over the frames from the video stream
cnt = 0

```



```python
def sendTheMessage(contactNumber, mail , name):
  with open('sheet.csv' , 'r+') as file:
    myDataList = file.readlines()
    contactList = []


    for contact in myDataList:
      entry = contact.split(',')
      contactList.append(entry[0])

    if contactNumber not in contactList:
      time = datetime.now()
      dtString = time.strftime('%H:%M:%S')

      file.writelines(f'\n{contactNumber},{name},{mail},{dtString}')

      client = Client(secretKey.ACCOUNT_SID , secretKey.AUTH_TOKEN)
      textMsg = r"We have detected that you have not wear mask in public place. Please wear your mask otherwise you will be fined"
      if len(contactNumber)>11:
        message = client.messages.create(
          body = textMsg,
          from_ = secretKey.TWILIO_NUMBER,
          to = contactNumber
          )
        print(message.body)
      else:
        print(contactNumber+" is not verified by twilio")

      server.sendmail(secretKey.MY_MAIL_ADDRESS, mail , textMsg)
```



C:\Users\nchandra\Desktop\Face-Mask_Detection\detect_mask_video.py - Sublime Text (UNREGISTERED)

```python
while True:

  frame = vs.read()
  frame = imutils.resize(frame, width=400)
  imgS = cv2.cvtColor(frame , cv2.COLOR_BGR2RGB)

  facesCurFrame = face_recognition.face_locations(imgS)
  encode = face_recognition.face_encodings(imgS , facesCurFrame)


  for encodeFace , FaceLoc in zip(encode , facesCurFrame):
    matches = face_recognition.compare_faces(encodeList , encodeFace)
    faceDist = face_recognition.face_distance(encodeList , encodeFace)

    matchIndex = np.argmin(faceDist)
    if matches[matchIndex] and contactNumber[matchIndex] and mailArr[matchIndex]:
      contact = contactNumber[matchIndex]
      mail = mailArr[matchIndex]
      Name = name[matchIndex]
      sendTheMessage(contact , mail, Name)


  (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)
  for (box, pred) in zip(locs, preds):
    (startX, startY, endX, endY) = box
    (mask, withoutMask) = pred

    label = "Mask" if mask > withoutMask else "No Mask"
    color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

    label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

    cv2.putText(frame, label, (startX, startY - 10),
```