VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA

"REAL TIME FACE MASK DETECTION AND ALERT SYSTEM USING FACE RECOGNITION TO FIGHT AGAINST COVID – 19"

A Project Report

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Engineering

In

Computer Science & Engineering

Submitted by

LAVANYA S P	**(4NN17CS400)**
MOHAMMAD USMAN SHARIF	**(4NN17CS402)**
PAVAN P	**(4NN17CS403)**
SHILPA RANI	**(4NN17CS407)**

Under the guidance of

Gowtham M

Assistant Professor

Department of CSE

ESTD-2008

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NIE INSTITUTE OF TECHNOLOGY MYSURU – 570018

2020-2021

Department of Computer Science and Engineering

NIE Institute of Technology

ESTD-2008

CERTIFICATE

This is to certify that the project work entitled ***"REAL TIME FACE MASK DETECTION AND ALERT SYSTEM USING FACE RECOGNITION TO FIGHT AGAINST COVID-19"*** is carried out by ***LAVANYA SP (4NN17CS400), MOHAMMAD USMAN SHARIF (4NN17CS402), PAVAN P (4NN17CS403), and SHILPARANI (4NN17CS407)*** the bonafide students of **NIE INSTITUTE OF TECHNOLOGY** in partial fulfilment for the award of **Bachelor of Engineering in Computer Science & Engineering** of the **Visvesvaraya Technological University, Belagavi** during the academic year **2020-2021**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

Signature of the Guide	Signature of the HOD	Signature of the principal
Mr. GOWTHAM M Assistant Professor, Dept. of CSE, NIEIT, Mysore	**MS. USHA M S** Associate Professor & HOD Dept. of CSE, NIEIT, Mysore	**DR. BANSILAL** Principal NIEIT, Mysore.

External viva

Name of the examiners	Signature with date
1……...	1……
2……...	2……

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose consistent guidance and encouragement crowned our efforts with success.

We consider ourselves proud to be a part of **NIE Institute of Technology Mysuru** family, the institution which stood by our side in all our endeavors.

We wish to express our gratitude to **Prof. Bansilal,** Principal, **NIE Institute of Technology Mysuru,** for providing a congenial working environment.

We express our sincere thanks to **Smt. Usha** HOD, Department of Computer Science and Engineering, for her support and encouragement.

We would like to thank our guide**, Mr. Gowtham M.** Assistant Professor, Department of Computer Science and Engineering, for his inspiration, guidance, constant supervision, direction and discussions in the successful completion of this project work.

We are thankful to the Department of Computer Science and Engineering staff members and non-staff members for their co-operation extended towards this work.

Finally, our heartfelt gratitude to our **family members, relatives** and **friends** for their constant support, motivation and encouragement throughout this project. Above all we would like to thank God Almighty, for having showered his blessings on us.

LAVANYA S P	**(4NN17CS400)**
MOHAMMAD USMAN SHARIF	**(4NN17CS402)**
PAVAN P	**(4NN17CS403)**
SHILPARANI	**(4NN17CS407)**

ABSTRACT

The corona virus COVID-19 pandemic is causing a global health crisis so the effective protection methods is wearing a face mask in public areas according to the World Health Organization (WHO). The COVID-19 pandemic forced governments across the world to impose lockdowns to prevent virus transmissions. Reports indicate that wearing face masks while at work clearly reduces the risk of transmission. An efficient and economic approach of using AI to create a safe environment setup. Face Mask detection has turned up to be an astonishing problem in the domain of image processing and computer vision. Face detection has various use cases ranging from face recognition to capturing facial motions, where the latter calls for the face to be revealed with very high precision. Due to the rapid advancement in the domain of machine learning algorithms, the jeopardies of face mask detection technology seem to be well addressed yet. This technology is more relevant today because it is used to detect faces in real-time inspection and supervision. To develop a model for detection of individual person who not wearing mask in public places. This model will identify whether the person is wearing mask or not. Develop a model to send notification to individual who not wearing mask using his/her details. The technique of detecting people's faces and segregating them into two classes namely the people with masks and people without masks is done with the help of image processing and deep learning. With the help of this project, a person who is intended to monitor the people can be seated in a remote area and still can monitor efficiently and give instructions accordingly.

LIST OF CONTENT

LIST OF FIGURE

LIST OF TABLE

CHAPTER 1

INTRODUCTION

The trend of wearing face masks in public is rising due to the COVID- 19 corona virus epidemic all over the world. Before Covid-19, People used to wear masks to protect their health from air pollution. While other people are self-conscious about their looks, they hide their emotions from the public by hiding their faces. Scientists proofed that wearing face masks works on impeding COVID-19 transmission. COVID-19 (known as corona virus) is the latest epidemic virus that hit the human health in the last century. In 2020, the rapid spreading of COVID-19 has forced the World Health Organization to declare COVID- 19 as a global pandemic.

More than five million cases were infected by COVID-19 in less than 6 months across 188 countries. The virus spreads through close contact and in crowded and overcrowded areas. The corona virus epidemic has given rise to an extraordinary degree of worldwide scientific cooperation. Artificial Intelligence (AI) based on Machine learning and Deep Learning can help to fight Covid-19 in many ways. Machine learning allows researchers and clinicians evaluate vast quantities of data to forecast the distribution of COVID-19, to serve as an early warning mechanism for potential pandemics, and to classify vulnerable populations. The provision of healthcare needs funding for emerging technology such as artificial intelligence, IoT, big data and machine learning to tackle and predict new diseases. In order to better understand infection rates and to trace and quickly detect infections, the AI's power is being exploited to address the Covid-19 pandemic. People are forced by laws to wear face masks in public in many countries. These rules and laws were developed as an action to the exponential growth in cases and deaths in many areas. However, the process of monitoring large groups of people is becoming more difficult. The monitoring process involves the detection of anyone who is not wearing a face mask.

Here we introduce a mask face detection model that is based on deep learning. The proposed model can be integrated with surveillance cameras to impede the COVID-19 transmission by allowing the detection of people and recognition who are wearing masks not wearing face masks and sending alert message. The model is integration between deep learning and classical machine learning techniques with opencv, tensorflow and keras. We find the most suitable algorithm that achieved the highest accuracy and consumed the least time in the process of training, detection and recognition of face.

1.1 Related Work

It took entire face into account instead of specific features. The face image was represented in a low dimension subspace formed by the PCA of the image database. Eigenvectors of the covariance matrix forms the eigenfaces of the image set. The eigenvalues of this training set and weights were calculated respectively. The test images go across the similar procedure and then the distance between training images and testing images were calculated.

proposed two algorithms for face recognition Geometric feature-based Template matching based. The results were obtained on frontal faces. LDA based feature extraction for face recognition was proposed by Etemadi and Chalupa. In this method, Eigen value analysis was executed on separation matrix other than on covariance matrix. The performance and computational aspects of PCA was presented by Moon and Phillips They proposed a new algorithm based on PCA. The components of proposed algorithm are illumination normalization, low pass filter, first low order Eigen vector removal, angle and mahala Nobis similarity measure matching. They introduced a generic modular PCA algorithm. Jones

1.2 Aim

Our aim is to break the chain link of the COVID-19 19 infection from one person to another and reduce this pandemic and protect people from being infected. Once this automatic fine technology implemented then there will no question of people breaking the rules of wearing mask Face is one of the variables that are very easy to remember in real life. Generally, humans can remember and recognize a person based on his face. However, face is one of a complex variables when viewed from the perspective of computer vision. Human faces have different features and characteristics of each person so that face recognition is very good to be applied in various areas, including entertainment, smart cards, information security etc.

applies the human face recognition system using the eigenface approach. Eigenface is one of the facial recognition methods based on the Principal Component Analysis (PCA) algorithm. PCA involved a mathematical procedure to derive a set of features for face recognition. Face recognition stage begins with face detection process using cascade classifier method, face preprocess, collect and train the face detected and finally the face recognition

1.3 Organization of Report

The sequel of pages and their hierarchical arrangement play a pivotal role in structuring the project report properly and interlinking the vital elements of the report in the best possible format. This project report consists of 8 chapters as mentioned below:

- Introduction
- Literature survey
- System requirements
- System analysis
- System design
- System methodology

Introduction: provides the background information about the project and the basic idea of what this project is expected to do. Literature survey: gives the detailed study of all the existing systems and its disadvantages. System requirements*:* tells about the detailed description on system requirements including both hardware and software. System analysis*:* provides a detailed description about the system analysis, why is it required, method of analysis of existing system, proposed system and its components. System design: It is involved in giving a description on how the system is going to be designed, how exactly the system would be developed. System implementation*:* It is all about the implementation part of the project that describes the critical coding of the project. Testing: gives information about testing the project in the real time scenarios and determines the efficiency of the system. Snapshots: It consists of snapshots of software and hardware modules. Conclusion & Future scope: It includes the extensions that could be made to this project. References*:* consists of the papers, books, and websites we have referred to.

1.4 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, Video Surf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

1.5 TENSORFLOW

Tensor Flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google, Tensor Flow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, While the reference implementation runs on single devices, Tensor Flow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units).

Tensor Flow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

1.6 KERAS

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel. Keras is a minimalist Python library for deep learning that can run on top of Theano or Tensor Flow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license.

Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles:

- ✓ **Modularity**: A model can be understood as a sequence or a graph alone. All the concerns of a deep learning model are discrete components that can be combined in arbitrary ways.
- ✓ **Minimalism**: The library provides just enough to achieve an outcome, no frills and maximizing readability.
- ✓ **Extensibility**: New components are intentionally easy to add and use within the framework, intended for researchers to trial and explore new ideas.
- ✓ **Python**: No separate model files with custom file formats. Everything is native Python. Keras is designed for minimalism and modularity allowing you to very quickly define deep learning models and run them on top of a Theano or Tensor Flow backend.

CHAPTER 2

LITERATURE SURVEY

A literature survey or review which combines both summary and synthesis of specific conceptual categories. Literature survey gives conclusion about how one can analyses and understand gaps exist in how a problem has been researched to date.

2.1 Survey Papers

In [1] system is proposed that featured histogram methods to recognize face, by using Viola Jones method with computer vision to detect face by taking features of eyes, nose and mouth. For matching factors for recognition histogram of individual feature is computed. Local features of histogram are explored from database image for face recognition based on feature. The results show that the proposed method is with precision of 97% for the recognition of faces. Here face detection method detect the face, their facial feature, extracted feature and their respective histogram are calculated. The algorithm given worked for face recognition with success rate of 97%. Skin colour detection can detect all face images but it also contains the neck and clothing whose colour is similar to the skin. So the error rate is high. So the result is not good. It is also possible to improve the accuracy of the matching method by hybridizing the feature base and template based approaches.

In[2] novel decolorization method is proposed to convert color images into grayscale. The proposed method, called CorrC2G, estimates the three global linear weighting parameters of the color to gray conversion by correlation. These parameters are estimated directly from the correlations between each channel of the RGB image and a contrast image. The proposed method works directly on the RGB channels; it does not use any edge information nor any optimization or training. The objective and subjective experimental results on three available benchmark datasets of color to gray conversion, e.g., Cadim, CSDD, and Color250, show that the proposed decolorization method is highly efficient and comparable to recent state-of-the art decolorization methods

In [3] is the most memorable part of the body in real life that makes it an important variable. In this research, we use face recognize method that integrated in Ry-UJI robot. The robot is recognized by a detected voice command looking for someone and when a person's face has been found, face recognition is complete. This article will apply the human face recognition system using the eigenface approach. Eigenface is one of the facial recognition methods based on the Principal Component

Analysis (PCA) algorithm. PCA involved a mathematical procedure to derive a set of features

for face recognition. Face recognition stage begins with face detection.

The [4] main advances in object detection were achieved thanks to improvements in object representations and machine learning models. A prominent example of a state-of-the-art detection system is the Deformable Part-based Model (DPM) It builds on carefully designed representations and kinematically inspired part decompositions of objects, expressed as a graphical model. Using discriminative learning of graphical models allows for building high-precision part-based models for variety of object classes. Manually engineered representations in conjunction with shallow discriminatively trained models have been among the best performing paradigms for the related problem of object classification as well In the last years, however, Deep Neural Networks (DNNs) have emerged as a powerful machine learning model

In [5] that DNN-based regression is capable of learning features which are not only good for classification, but also capture strong geometric information. We use the general architecture introduced for classification by [14] and replace the last layer with a regression layer. The somewhat surprising but powerful insight is that networks which to some extent encode translation invariance, can capture object locations as well. Second, we introduce a multi-scale box inference followed by a refinement step to produce precise detections. In this way, we are able to apply a DNN which predicts a low-resolution mask, limited by the output layer size, to pixel-wise precision at a low cost the network is a applied only a few dozen times per input image.

In [6] contrast to the conventional methods designed for natural images that aim to preserve contrast between different classes in the converted gray image, the proposed conversion method reduces as much as possible the contrast (i.e. intensity variance) within the text class. It is based on learning a linear filter from a predefined dataset of text and background pixels that: i) when applied to background pixels, minimizes the output response; and ii) when applied to text pixels, maximizes the output response, while minimizing the intensity variance within the text class. Our proposed method (called here LC2G for Learning-based Color-to-Gray) is conceived to be used as pre-processing for document image binarization. A dataset of forty-six (46) historical document images is created and used to evaluate subjectively and objectively the proposed method. The method demonstrates drastically its effectiveness and impact on the performance of state-of-the-art binarization methods. Four other web-based image datasets are created to evaluate the scalability of the proposed Method.

The [7] human face is an important biometric quantity which can be used to access a user-based

system. As human face images can easily be obtained via mobile cameras and social networks, user-based access systems should be robust against spoof face attacks. In other words, a reliable face- based access system can determine both the identity and the liveness of the input face. To this end, various feature-based spoof face detection methods have been proposed. These methods generally apply a series of processes against the input image(s) in order to detect the liveness of the face. In this paper, a deep-learning-based spoof face detection is proposed. Two different deep learning models are used to achieve this, namely local receptive fields (LRF)-ELM and CNN. LRF-ELM is a recently developed model which contains a convolution and a pooling layer before a fully connected layer that makes the model fast. CNN, however, contains a series of convolution and pooling layers. In addition, the CNN model may have more fully connected layers. A series of experiments were conducted on two popular spoof face detection databases, namely NUAA and CASIA. The obtained results were then compared, and the LRF-ELM method yielded better results

[8] Present days masked face recognition is more important. Mainly terrorists and criminals covered their face with mask for disguise. Besides this, sunglass, hat, color festoon etc. also act like mask. Using different types of masks or occlusions the key features to identify a person is decreasing. Lower numbers of face features in the masked face cause difficulties than other normal face recognition technique Consequently the accuracy rate of recognition is decreasing. That's why masked face is being one of the major's concerned factor in the field of face recognition. Some of masked face.

CHAPTER 3

SYSTEM REQUIREMENTS

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to satisfy these requirements may result in installation problems or performance problems. The former may prevent a tool or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or may be to hang or crash. System requirements are also referred to as minimum system requirements. System requirements can be broadly classified as hardware requirements, software requirements, functional requirements and non-functional requirements. The hardware system requirements often specify the OS version, processor type, memory size, available disk space and extra peripherals, if any, needed. The software system requirement consistent of all necessary requirements required for project development. A functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs. A Non-functional requirement impose constraints on the planning or implementation like performance engineering requirements, quality standards, or design constraints.

3.1 Hardware requirement

- Primary Camera or Secondary Camera
- Internal memory: 6GB or more
- Processor: Intel core i5
- DNN

3.2 Software requirements

- Anaconda navigator
- Frame work Open c
- Programming language: Python 3.7
- IDE: Python idle, Spyder
- MySQL
- Windows 10

CHAPTER 4

SYSTEM ANALYSIS

4.1 Existing System

Now in the present situation there is no technology to detect whether a person is wearing a mask or not. And people are taking advantage over this and will be travelling without mask which causes lot of inconvenience to others and they may be in the trouble at the end of the day. This is also one of the reasons for this pandemic to spread to this level.

4.2 Proposed System

This application identify whether the person is wearing mask or not in real-time video stream. When the person is not wearing a mask then the face of that particular person will be detected, and the captured image will be matched with the image stored in data base. And sends a warning message to the identified person.

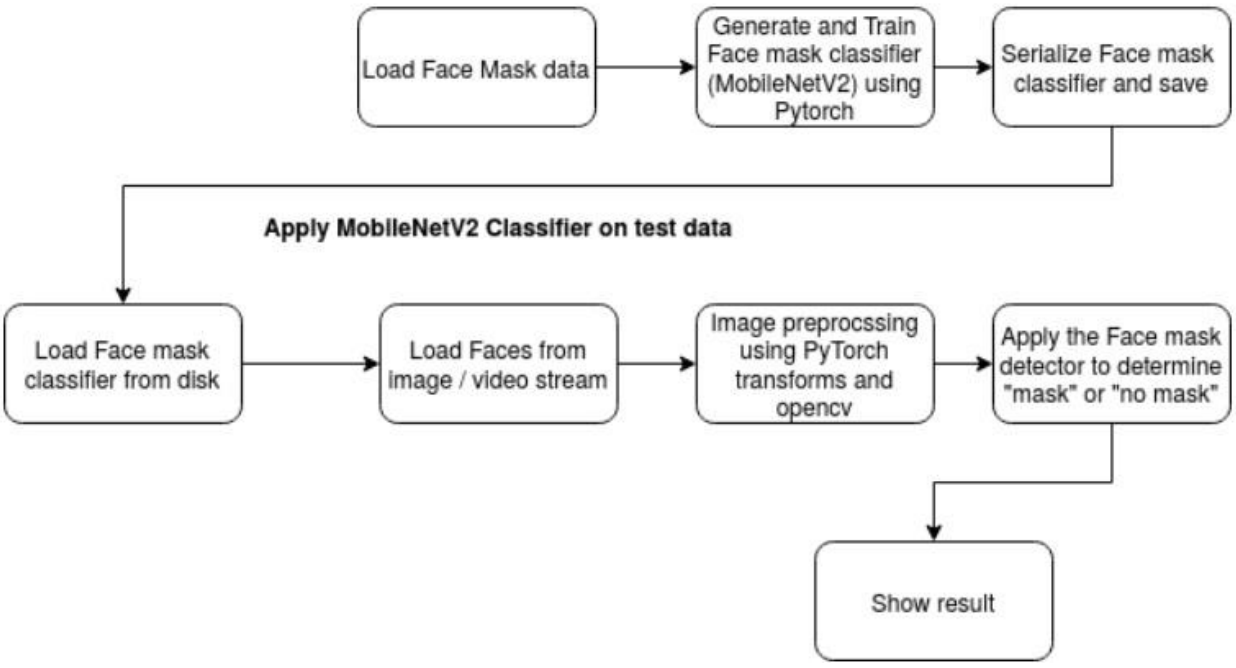

Figure 4.2.1: Proposed System Architecture

Approach

1. Train Deep learning model (MobileNetV2)

2. Apply mask detector over images / live video stream

Data at Source

The majority of the images were augmented by OpenCV. The set of images were already labeled “mask” and “no mask”. The images that were present were of different sizes and resolutions, probably extracted from different sources or from machines (cameras) of different resolutions.

Data preprocessing

Preprocessing steps as mentioned below was applied to all the raw input images to convert them into clean versions, which could be fed to a neural network machine learning model.

1. Resizing the input image (256 x 256)
2. Applying the color filtering (RGB) over the channels (Our model MobileNetV2 supports 2D 3 channel image)
3. Scaling / Normalizing images using the standard mean of MobileNetV2
4. Center cropping the image with the pixel value of 224x224x3 **5**.Finally Converting them into tensors (Similar to NumPy array)

CHAPTER 5

SYSTEM DESIGN

System design is that the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of system theory for the product development.

5.1 Methodology

Face detection: The downside of face mask detection is all regarding face detection. However, before face mask detection to faithfully notice a face and its landmarks. This can be basically a segmentation problem and in sensible system, most finding this task. After all the particular detection supported option extracted from these facial landmarks is barely a minor step.

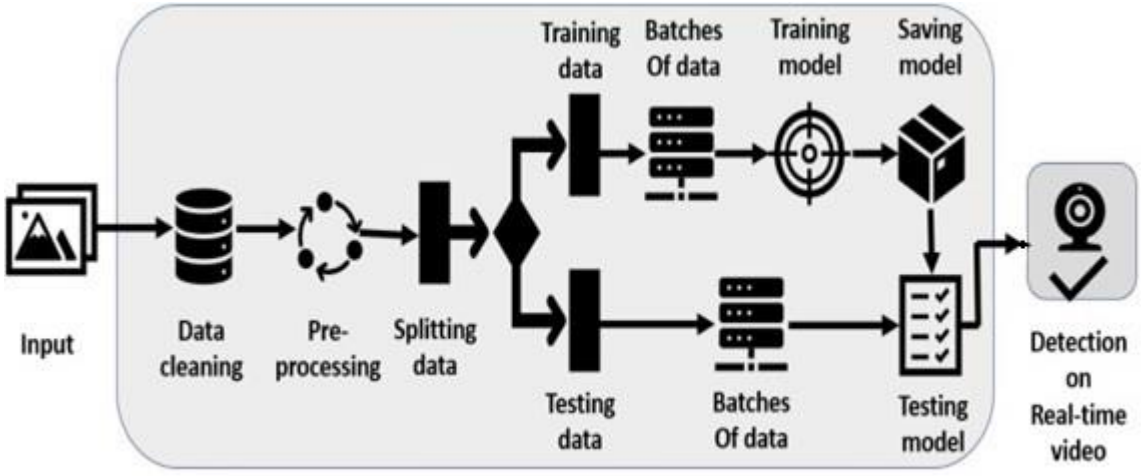

Figure 5.1.2: Flow Diagram

Methods:

1. Train deep learning model
2. Apply face mask detector over live stream.

5.2: Algorithm's used:

- Training phase for head and shoulder detection
- Determine Head-and Shoulder
- Face mask Detection Algorithm
- Training for presence of human complexion
- Identification of the covered face

5.3 Face Mask Detection

A proper and efficient face mask detection algorithm always enhances the performance of face mask detection system. It is performed on the basis of Face mask Detection Algorithm. This algorithm focusses more on speed and reliability. This detector uses Haar-like features and a cascade classifier. The cascade object detector is pre-trained to detect faces, noses and other objects.

5.4 Pre-processing

The detected face is extracted and subjected to preprocessing. This pre-processing step involves cropping and converting color image into grayscale and resize the image into 64*64 pixels. The main reason why grayscale representations are often used is for extracting descriptors rather than operating on color images directly, grayscale simplifies the algorithm and reduces computational requirements.

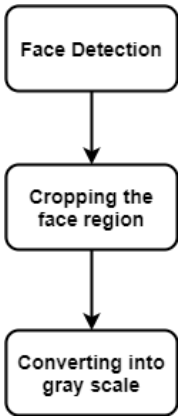

Figure 5.4.1 Pre-Processing

5.5 MobileNetV2

MobileNetV2 is a Deep Neural Network that has been deployed for the classification problem. Pre-trained weights of ImageNet were loaded from Tensor Flow. Then the base layers are frozen to avoid impairment of already learned features. Then new trainable layers are added, and these layers are trained on the collected dataset so that it can determine the features to classify a face wearing a mask from a face not wearing a mask. Then the model is fine-tuned, and then the weights are saved. Using pre-trained models helps avoid unnecessary computational costs and helps in taking advantage of already biased weights without losing already learned features.

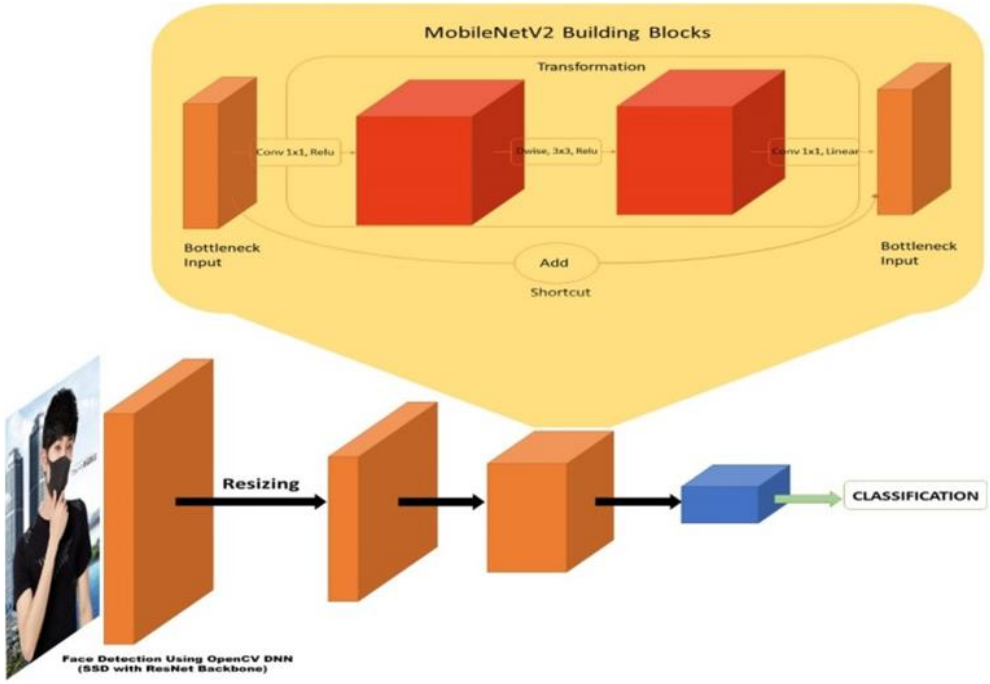

Figure 5.5.1: Building Blocks of MobileNetV2

This layer is the fundamental block of the Convolutional Neural Network. The term convolution implies a mathematical combination of two functions to get the third function. It works on a sliding window mechanism, which helps in extracting features from an image. This helps in generation feature maps. The convolution of two functional matrices, one being the input image matrix A and the other being convolutional kernel B give us the output C as:

$$C(T)=(A*B)(x)=\int_{\infty-\infty} A(T)\times B(T-x)\, dT$$

ALGORITHM 1: Pre-processing and Training on Dataset

INPUT: Images along with their pixels values
OUTPUT: Trained Model

STEP 1: Load Images and their pixel values.
STEP 2: Process the images, i.e., resizing, normalization, and conversion to a 1D array.
STEP 3: Load the Filenames and their respective labels.
STEP 4: Perform Data augmentation and then split data into training and testing batches.
STEP 5: Load MobilenetV2 model from Keras. Train it on training batches and compile it using Adam optimizer.
STEP6: Save the model for future use.

5.6 Dataset development

The dataset development phase consists of capturing image of each individual and extracting the specified features, in our case it is face, and later it is enhanced using pre- processing techniques and stored in a folder. This dataset is used at the time of recognizing the individuals present in the classroom. The accuracy of this algorithm depends on the way we train the system.

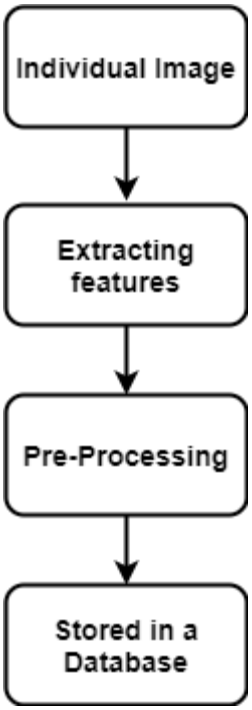

Figure 5.6.1 Database Development

5.7 Face Recognition

The proposed system uses principal components analysis (PCA) for face Recognition. After detecting the faces from the input image, the feature is extracted using the Principle Component Analysis method. This method is used to reduce the dimensionality of data space to the similar feature space. For automatic recognition we need to create a face database. Various image samples were taken for each person and their features are extracted and stored in the database as a train image. For an input image, face detection and feature extraction is performed and features of each face class of trained image is compared and stored in the database.

5.8 Post Processing

Post processing method involves sending message to the particular person who is not wearing a mask, to establish this process it requires information of that person and it is stored inside database. The gray scale image extracts from dataset and compare that person with stored data using machine learning model. And send warning message to that person.

ALGORITHM 2: Deployment of Face Mask Detector

INPUT: Choice of deployment and Files(optional).
OUTPUT: Images classified into the mask and no mask or Classification in Real-time.

STEP 1: Load saved classifier from disk. Also, load face detector from OpenCV.
STEP 2: If the choice is classification on image:
Load Image(s)
STEP 2.1: Apply face detection model to Detect faces in an image
STEP 2.2: If Faces are detected:
Crop face to bounding box coordinates from face detection model
Get predictions from the face classifier model.
Show predictions and save resultant image.
Else:
Show no output
STEP 3: If the choice is classification in real-time:
Load real-time feed from OpenCV
Read the feed frame by frame.
STEP 3.1: Apply face detection model to Detect faces in Frames read in real-time
STEP 3.2: If Faces are detected:
Crop face to bounding box coordinates from face detection model
Get predictions from the face classifier model.
Show output in a real-time feed
Else:
Show normal feed
STEP 4: End stream when q is pressed

CHAPTER 6

IMPLEMENTATION

6.1 Camera Placement at Public Places:

Camera would be placed in public places like markets, footpath, public malls etc., It is important to place the camera to detect faces with mask or without mask. Faces will be detect in different luminous. It will help to detect their face whether they are in mask or not. Camera will work 24X7 and take continuous live stream And this completely based on live stream only.

Figure 6.1.1: At Public Place

Figure 6.1.2: At Public Mall

6.2 Data Set Collection

Dataset collection is a process of collecting required data from all the relevant sources as needed for the research problem, test the hypothesis and evaluate the outcomes.

In our proposed system we have collected dataset from google to train a model, and extracted into our project. The dataset is going to stored inside folder. And it will be extracted from program. Each dataset is going to store with different ID and names.

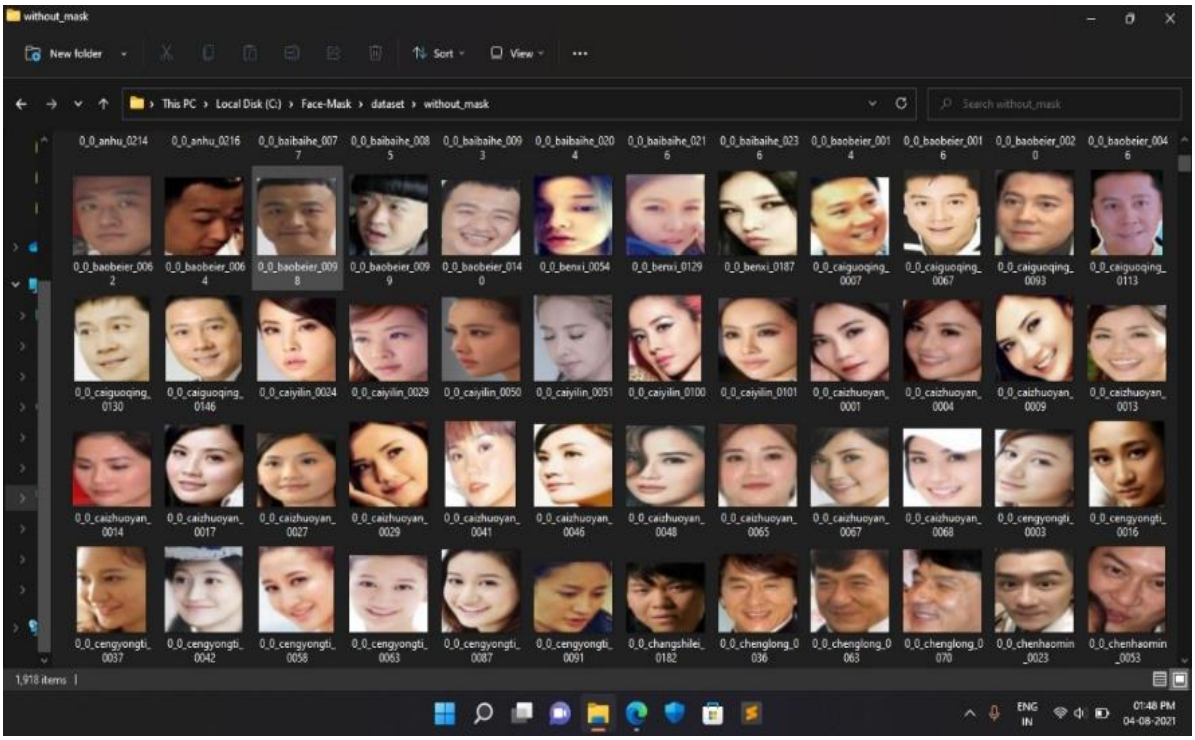

Figure 6.2.1: Dataset Collected Without Mask

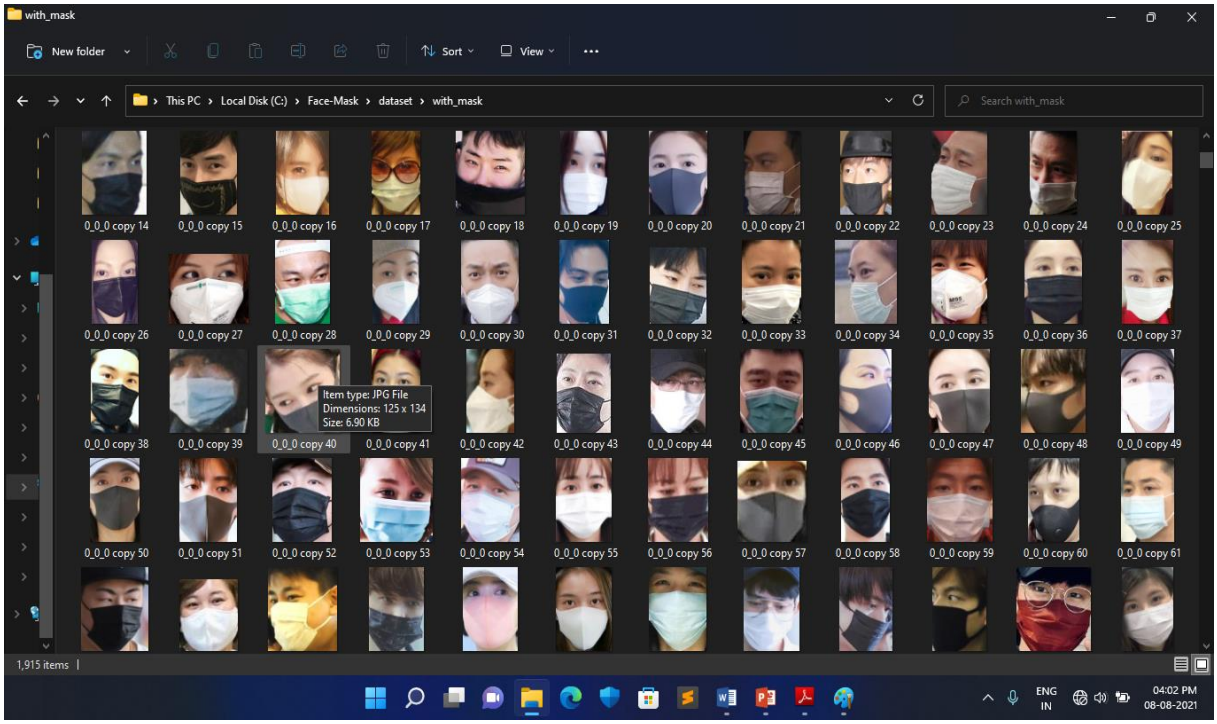

Figure 6.2.2: Dataset Collected With Mask

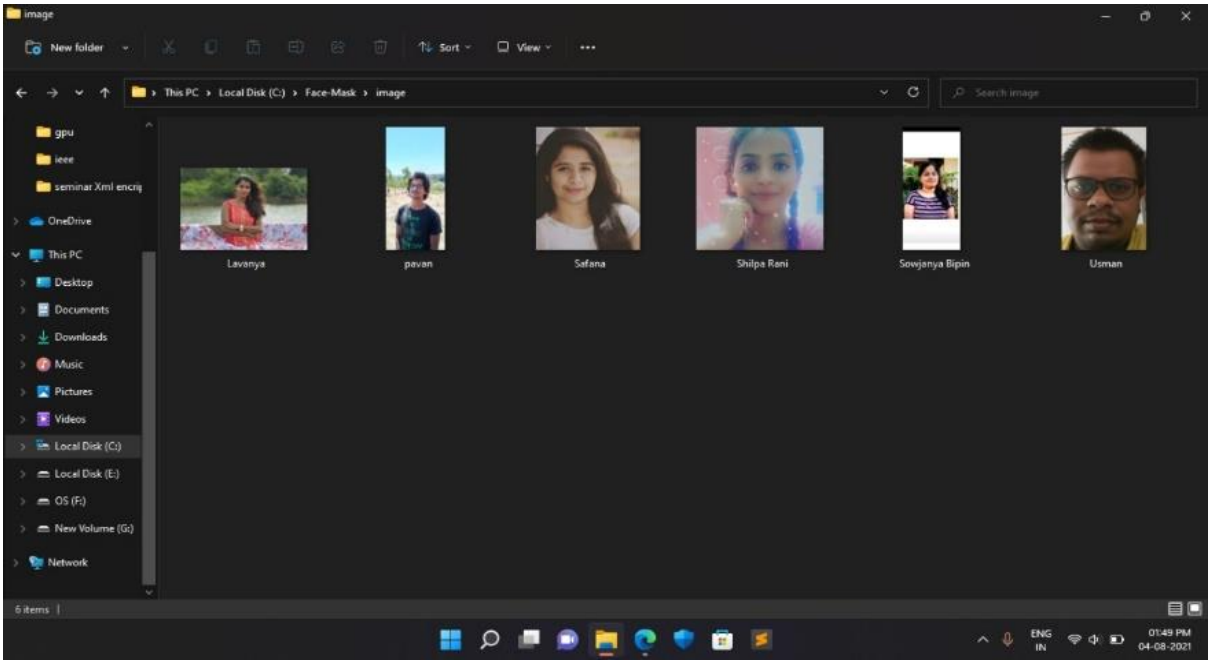

Figure 6.2.3: Data Set Used For Face Recognition

6.3 Training and Creating A Face Mask Detector Model

The first step is to create a training model from the data sets that are collected to able and build a perfect face mask detector and the next step is to create the main program to use the build model in that main program file. The creation of mask detector model will help in the main program file for detecting the face mask whenever a person's face is encounter in the main Real time video stream frame. The training will take more time to complete which make the model to work accurately. It will read all image from the directory that is specified and the images are resize to fit only face image in it. While creating a face mask detector model it will undergo several training and testing process and after creating the model it will list the accuracy and loss of the face mask detector model.

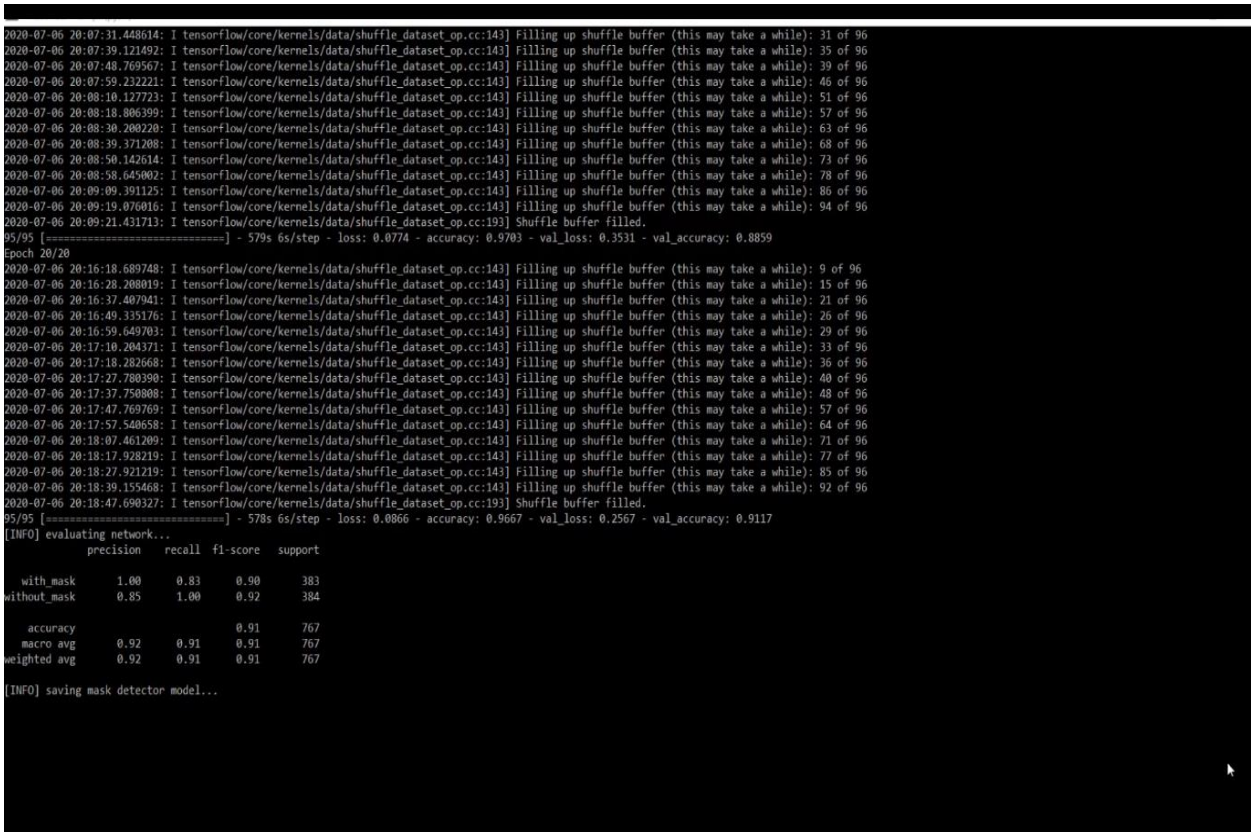

Figure 6.3.1: Creating the Mask Detector Model

And also the face mask detector model which has created is plotted using graphical representation.

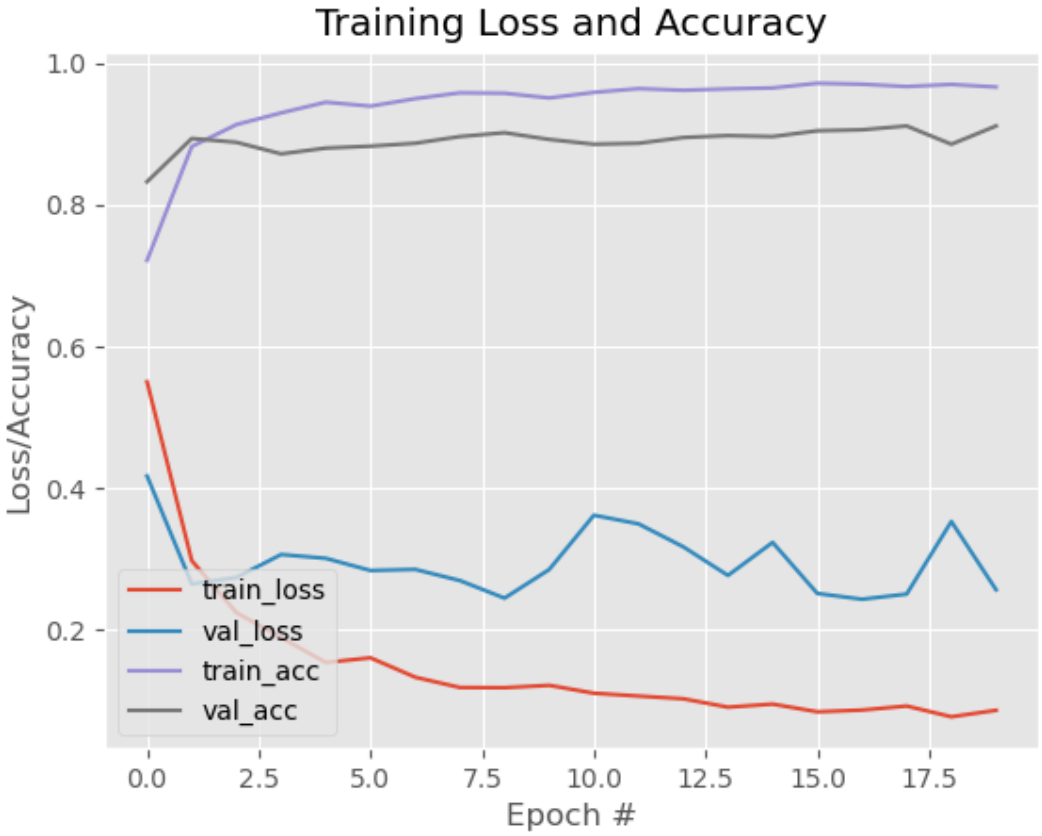

Figure 6.3.2: Training Loss and Accuracy

6.4: Creating Database For Users

The details of the user like name, age, email, phone number, address are needed for the project for that purpose we need a database to store the user data. So we will create a database that hold details of person's. the details of person will help the project to get the detected person's details to send the alert message to that person.

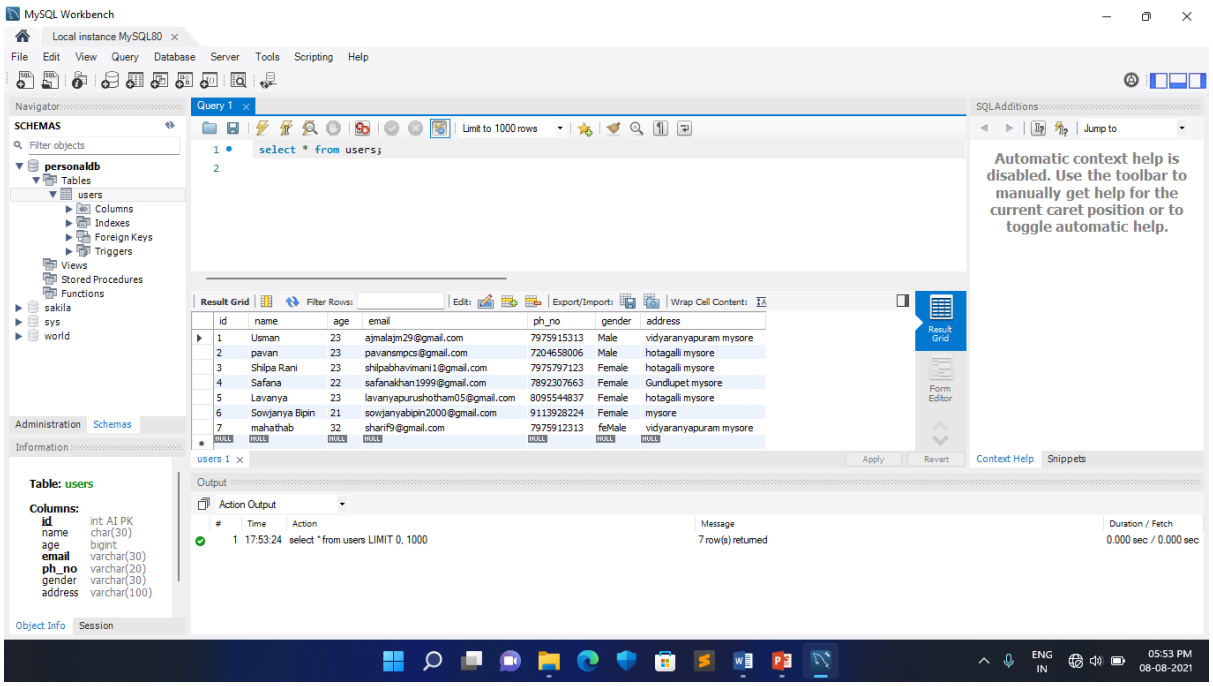

Figure 6.4.1: Database for Storing details

6.5 Face Mask Detection and Recognition Of Face

The main aim of this project is to create a face mask detection and recognition of face without mask and alerting the person how not wearing a mask by sms and email. Know the first step is to detect the face mask like detecting the person how interferer inside the real time video stream frame the project has to detect the person wearing mask or not. And if the person is wearing a mask then the face of that person has to be surrounded with green frame and labeled as mask and percentage and if not wearing mask means a red fame and with labeled as no mask and percentage has to be appear. The person who have not wear a mask has to be identified or recognized like how is not wearing a mask and name of that recognized person has to be displayed bellow is face in live video stream.

Figure 6.5.1: Face Mask Detected

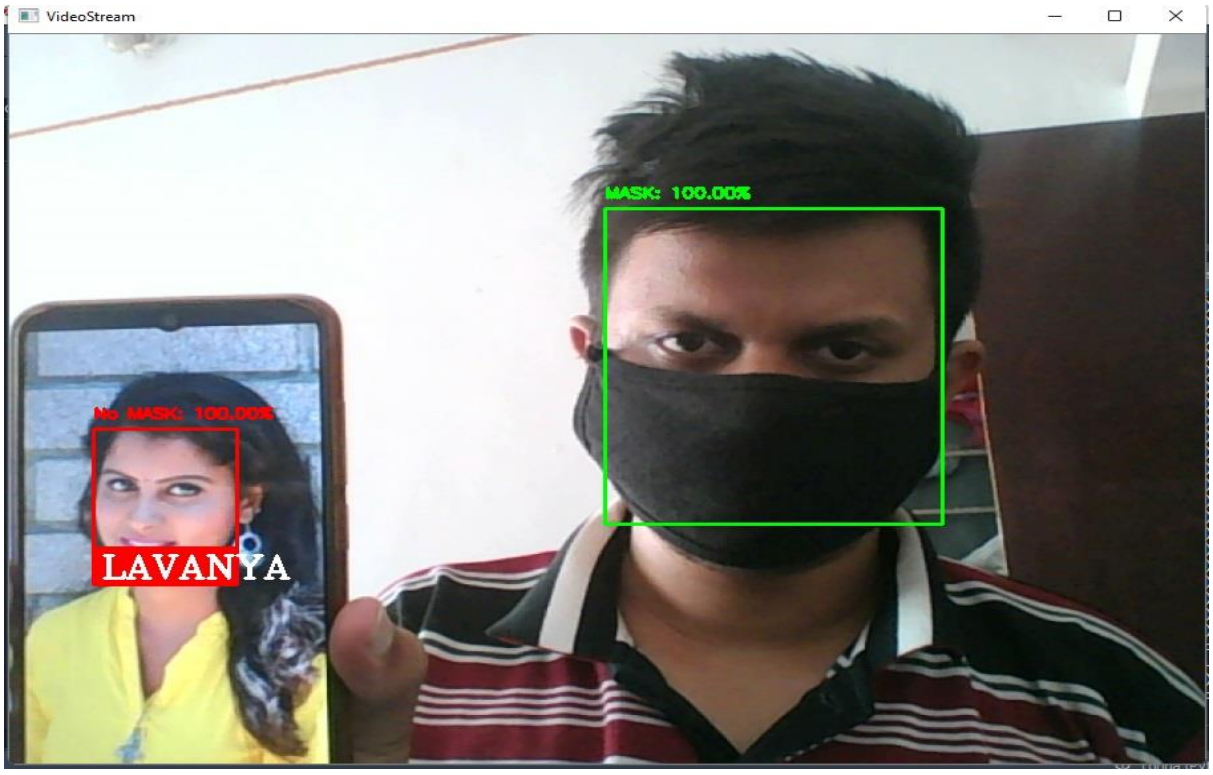

Figure 6.5.2: Face Mask and Without Face Mask

6.6 Sending Email Who Not Weared Face Mask

It is also one of the main part which will help to spared covid-19 awareness among the people and alerting people that if they don't wear mask then a serious action will be taken against them so whenever the people step out of their house the have to wear a mask to safe them self for covid-19. An email to the email address of the person is send to alert that person.

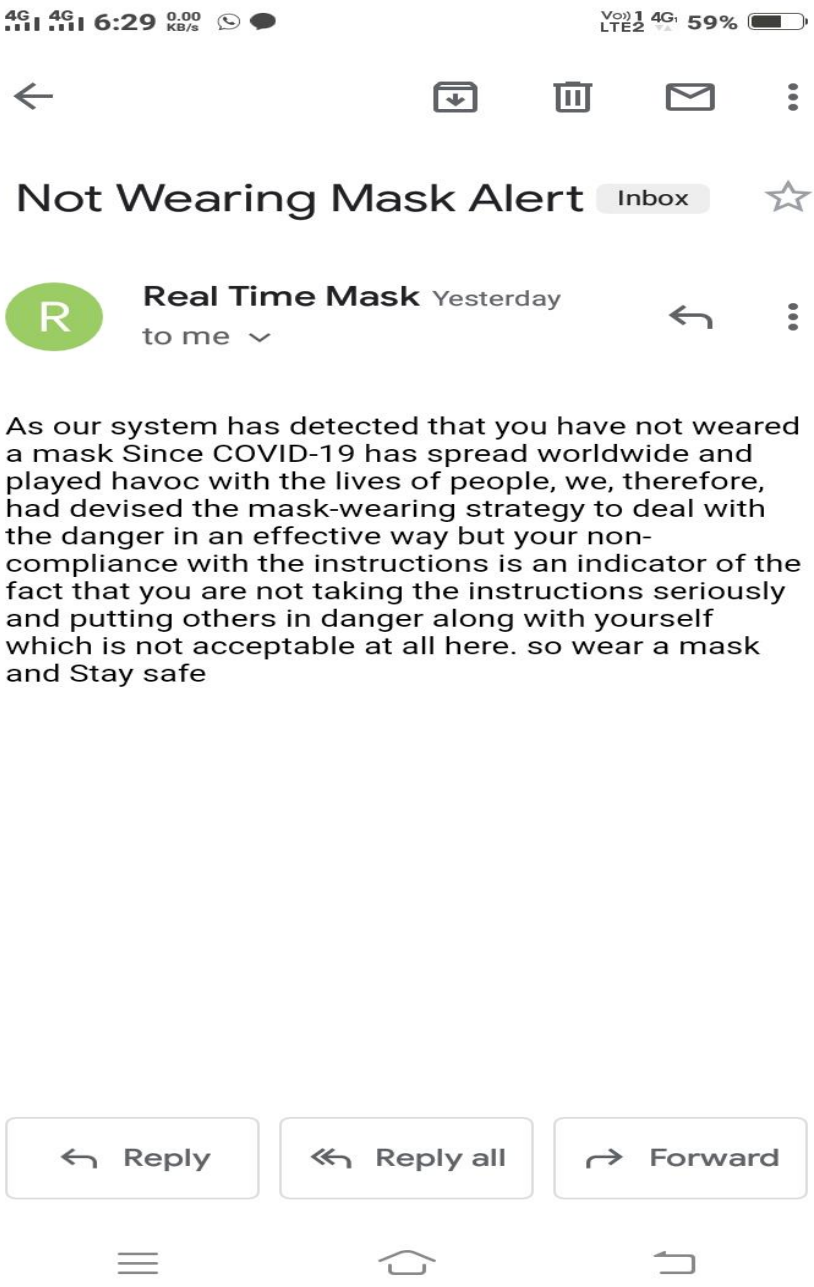

Figure 6.6.1: Email Message

6.7 Sending SMS to Phone No

The sms is send to the phone no to the persons who not wearing a mask. In this project we are using twilio service for sending message

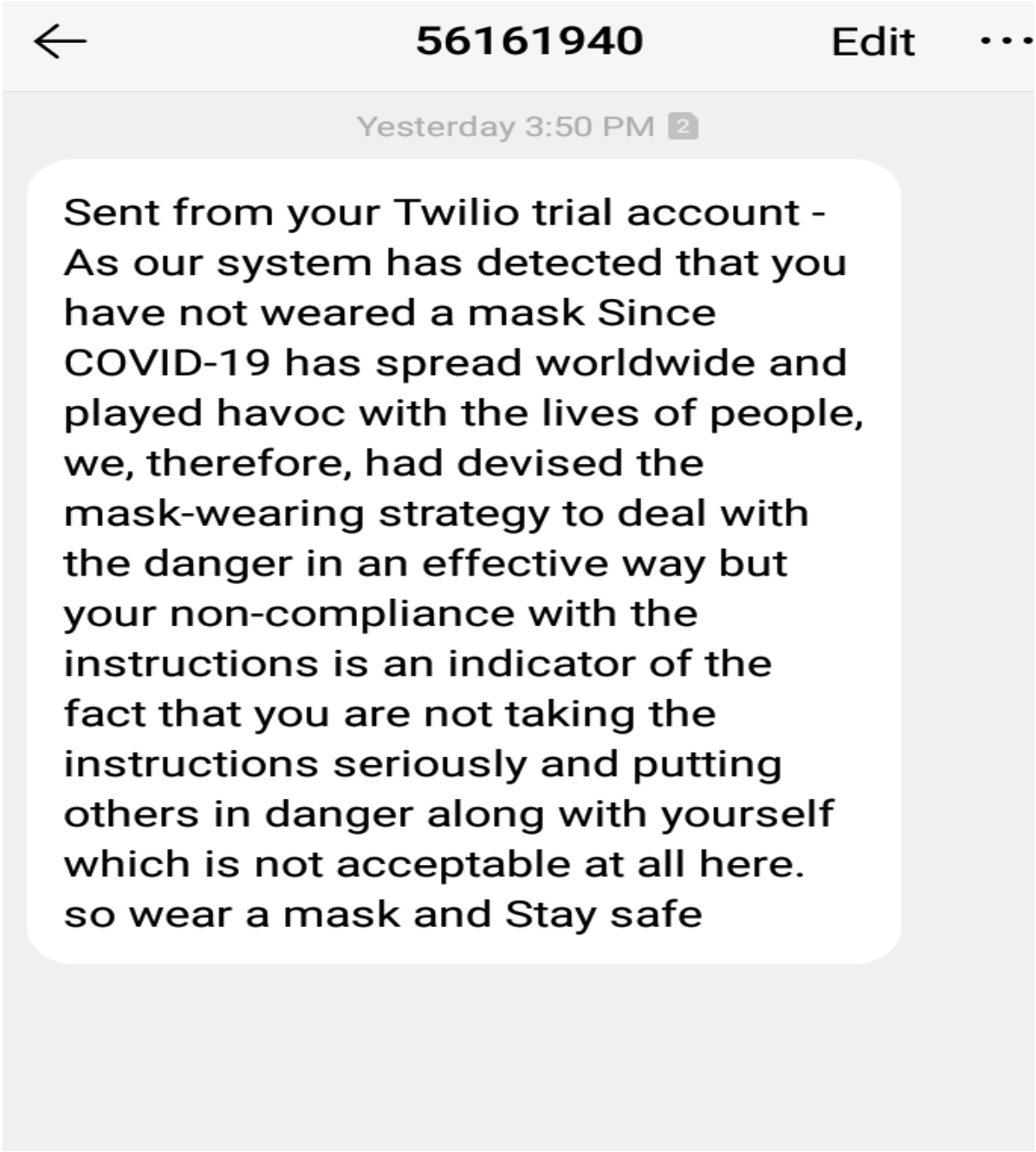

Figure 6.7.1: SMS Sending

6.8 The Actual Implementation

```
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

import numpy as np

import imutils

import time

import cv2

import os

maskNet = load_model("mask_detector.model")

print("[INFO] starting video stream...")

vs = VideoStream(src=0).start()

while True:

        frame = vs.read()

if mask > withoutMask:

                label = "MASK"

                color = (0, 255, 0)

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

                cv2.putText(frame, label, (startX, startY -
10),cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
```

```
                cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
            else:
                label = "No MASK"
                color = (0, 0, 255)
                label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
                cv2.putText(frame, label, (startX, startY -
10),cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
                cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
                for encodeface,faceLoc in zip(encodesCurFrame,faceCurLoc):
                    matches =
face_recognition.compare_faces(encodeListKnown,encodeface)
                    faceDis =
face_recognition.face_distance(encodeListKnown,encodeface)
                    matchIndex = np.argmin(faceDis)
                    if matches[matchIndex]:
name = names[matchIndex].upper()
                        cv2.rectangle(frame,(startX,endY-
35),(endX,endY),(0,0,255),cv2.FILLED)
                        cv2.putText(frame,name,(startX+6,endY-
6),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),2)
                        p="select email from Users where name = %s"
                        mycur.execute(p,(name, ))
                        row = mycur.fetchall()
```

```
                                        de = row[0][0]
                                        print(de)
                                        email_alert(sub, bod, de)
                                        print("[INFO] Send Successfully.....")
        cv2.imshow("VideoStream", frame)
        key = cv2.waitKey(1) & 0xFF
        if key == ord("q"):
                break
```

6.8.1: sms function and email function

Sms function to send sms:

```
account_sid = "xxxxxxxxxxxxxxxxx2058b19bd18cbb"
auth_token = "xxxxxxxxxxxxxxxxxxxxxf4c0668d481"
client = Client(account_sid, auth_token)
message = client.messages \
.create(
body="Message to be send",
from_='+XXXXXX0051'
to=pto
)
print(message.sid)
```

email function to send emails:

```
def email_alert(subject, body, to):
    mesg = EmailMessage()
    mesg.set_content(body)
    mesg['subject'] = subject
    mesg['to'] = to

    user = "xyzabc@gmail.com"
    mesg['from'] = "Real Time Mask"
    #pass is app password
    password = "xxxxxxxxxxxxxxxbhz"
    #port no 587
    server = smtplib.SMTP("smtp.gmail.com", 587)

    server.starttls()
    server.login(user, password)
    server.send_message(mesg)
    server.quit()
```

CHAPETER 7

SYSTEM TESTING

System testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Although crucial to software quality and widely deployed by programmers and testers, software testing still remains an art, due to limited understanding of the principles of software. The difficulty in software testing stems from the complexity of software: we cannot completely test a program with moderate complexity.

Testing is more than just debugging. Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs (errors or other defects). The purpose of testing can be quality assurance, verification and validation, or reliability estimation.

Software testing can be stated as the process of validating and verifying that a computer program/application/product:

- Meets the requirements that guided its design and development
- Works as expected
- Can be implemented with the same characteristics
- Satisfies the needs of stakeholders.

Testing can be used as a generic metric as well. Software testing is usually a trade-off between budget, time and quality. Software testing, depending on the testing method employed, can be implemented at any time in the software development process. Traditionally most of the test effort occurs after the requirements have been defined and the coding process has been completed, but in the agile approaches most of the test effort is on-going. As such, the methodology of the test is governed by the chosen software development methodology.

7.1 Testing Methodologies

Software Testing can be of different types, depending on the purpose and entities being tested. The various kinds are:

- **Unit Testing:** It involves testing individual software components or modules. It's typically done by the programmer and not by testers, as it requires detailed knowledge of internal program design and code. It may require developing test driver modules or test harnesses.

- **Integration Testing:** It involves the testing of integrated modules to verify combined functionality after integration. Modules tested are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

- **System Testing:** The entire system is tested as per the requirements. It is a type of Functional testing that is based on overall requirements specifications and covers all combined parts of a system.

- **End-to-end Testing:** Similar to system testing, it involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

- **Acceptance Testing:** Normally this type of testing is done to verify if system meets the customer specified requirements. In engineering and its various sub disciplines, acceptance testing is a test conducted to determine if the requirements of a specification are met. It may involve chemical tests, physical tests or performance tests.

- **Unit testing**

Test Id	Test Case	Expected Output	Actual Output	Pass/Fail
1	Training mask detector	Model has to be build	Model build	Pass
2	Mask detection	The face mask has to be detected	Face mask will be detected	Pass
3	Face recognition	Face has to be recognized	Face will recognize	Pass
4	Email sending	Email has to be send	Email send	Pass
5	Sms sending	Sms must be send	Sms send	Pass
6	Exit	Q pressed then close camera and execution	camera closed and execution stop	Pass

Table 6.1: Unit Testing

- **Integration Testing:** It involves the testing of integrated modules to verify combined functionality after integration. Modules tested are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems. The integration testing performed on the project are:

Test Id	Test Case	Expected Output	Actual Output	Pass/Fail
1	Detect Multiple Faces mask	Multiple Faces mask are detected	Multiple Faces mask are detected	Pass
2	Label display	Label to display in frame	Label will be displayed near face	Pass
3	Recognize the faces	Recognized faces	Faces are recognized	Pass

Table 6.2: Integration Testing.

- **System Testing:** The entire system is tested as per the requirements. It is a type of Functional testing that is based on overall requirements specifications and covers all combined parts of a system. The system tests performed on the project are:

Test Id	Test Case	Expected Output	Actual Output	Pass/Fail
1	Real time face	Detect multiple person's in the Public place	Multiple faces mask and no mask are detected	Pass
		Recognize the detected face	The detected face are recognized	
		Retrieve the name, email and phno of the person that is recognized	Name, email, phno of the recognized person is retrieved	
		Send sms and email to recognized person	Email and text message is send to the recognized person	

Table 6.3: System Testing

CHAPTER 8

EXPERIMENT RESULTS AND ANALYSIS

This project mainly has three modules, Detection of human faces mask, Recognition of the detected Faces and Sending sms and emails to detected face person. For our experiments. For training we have collected separate dataset from google that is with mask and without mask we have 1915 images of with mask and 1918 images of without mask for training the mask detector model.

8.1 Mask Detection Result

The detection results were found to be better when the lightings are from the same sideas the camera placed. The detection rates were also found to be better. The average detection rate is **98.40%**

	PRECISION	RECALL	F1-SCORE	SUPPORT
WITH MASK	0.99	1.00	0.99	383
WITHOUT MASK	1.00	0.99	0.99	384
ACCURACY			0.99	767
MACRO AVG	0.99	0.99	0.99	767
WEIGHTED AVG	0.99	0.99	0.99	767

Table 8.1.1: accuracy and loss

CHAPTER 9

SNAPSHOTS

Figure 9.1: Mask Dataset

Figure 9.2: No Mask Dataset

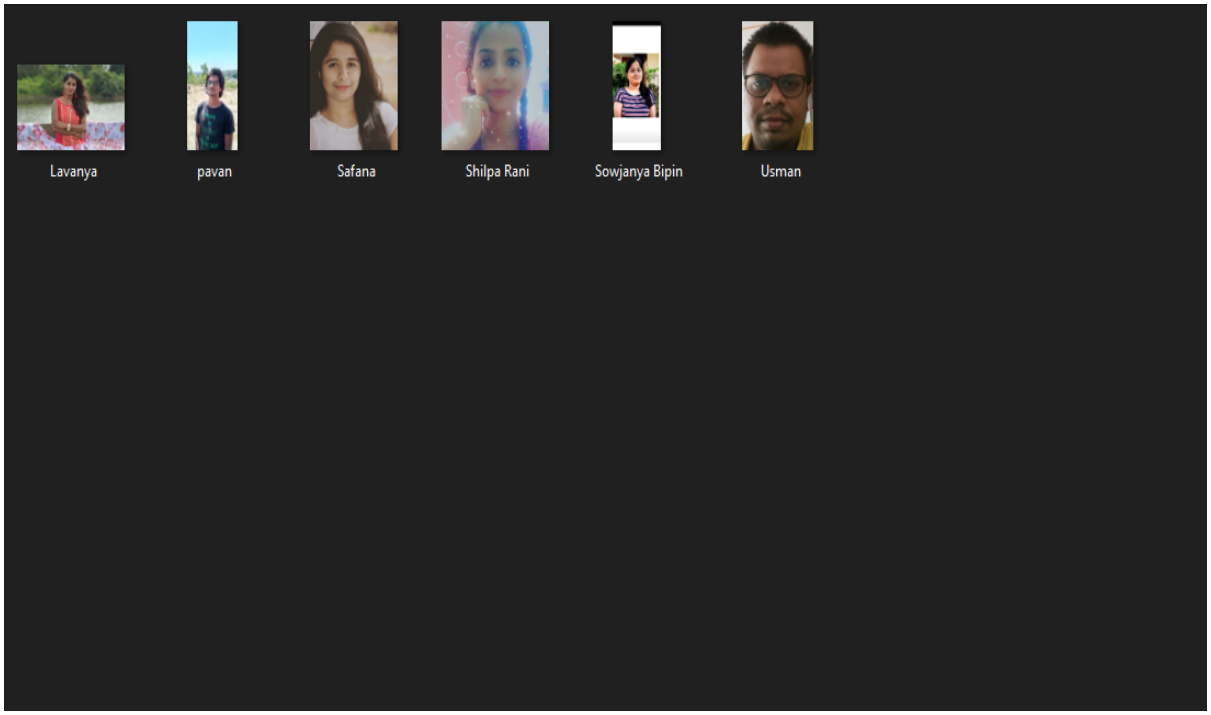

Figure 9.3: Database Registered Users

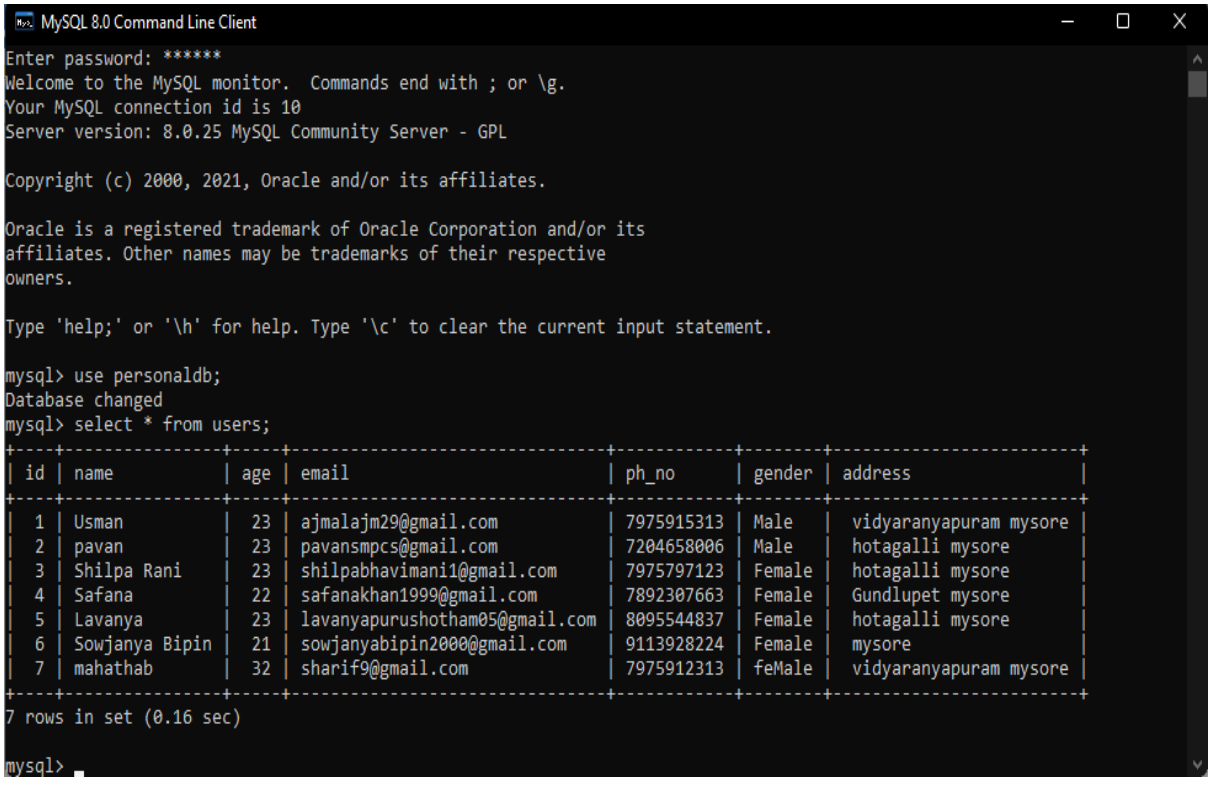

Figure 9.4: MySQL Database

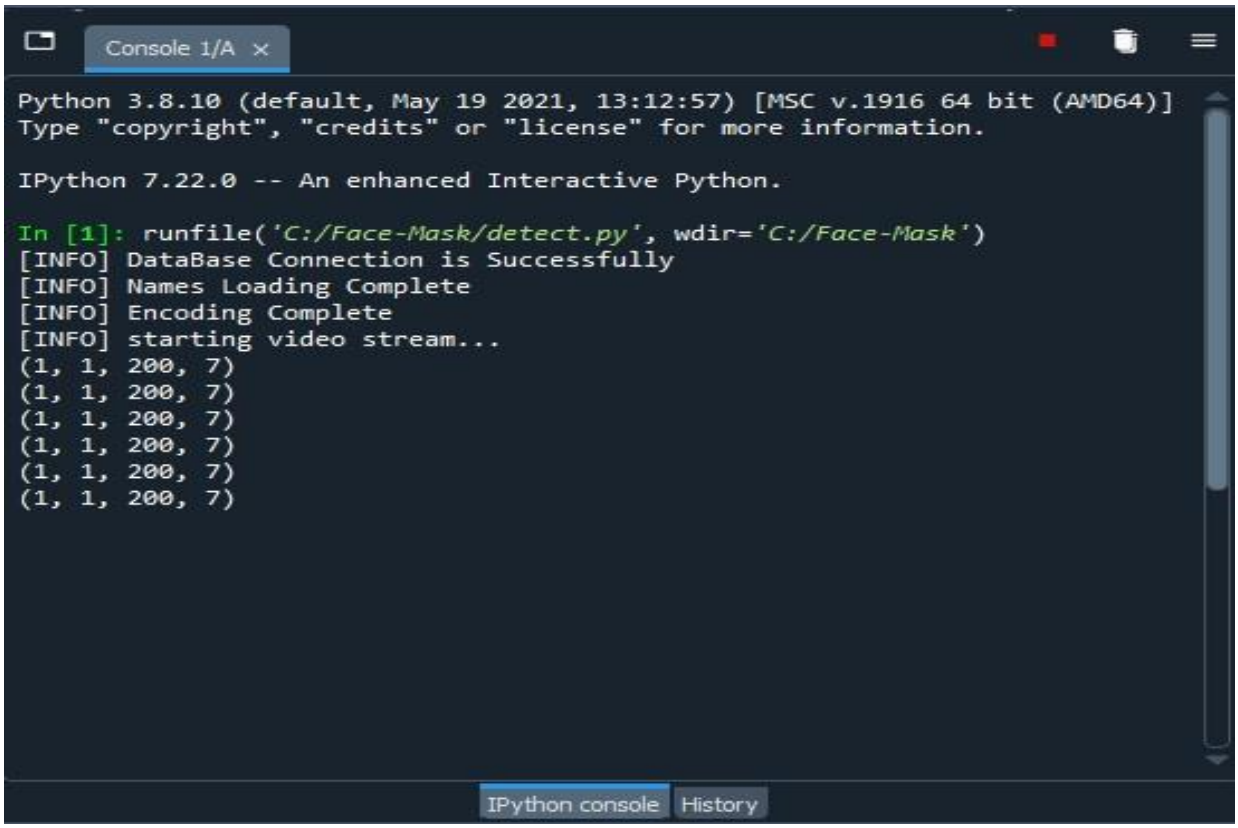

Figure 9.5: Staring Console of Project

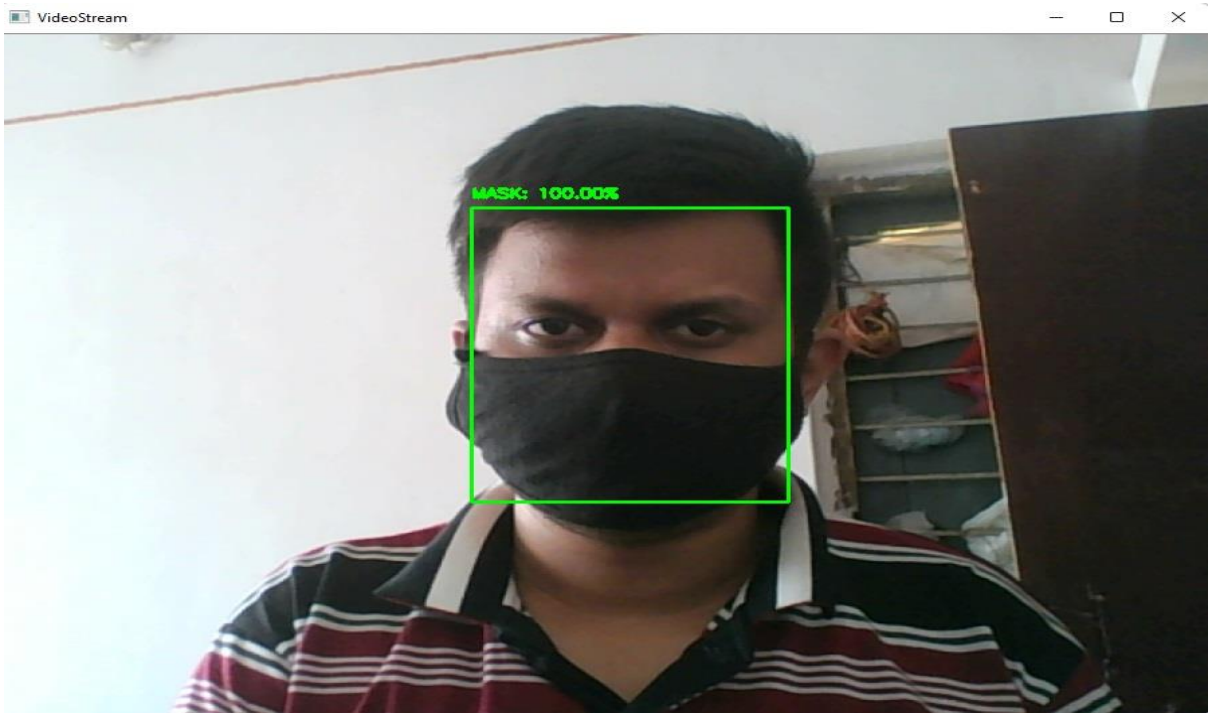

Figure 9.6: Mask Detection on Live Stream

Figure 9.7: No Face Mask and Face Recognition

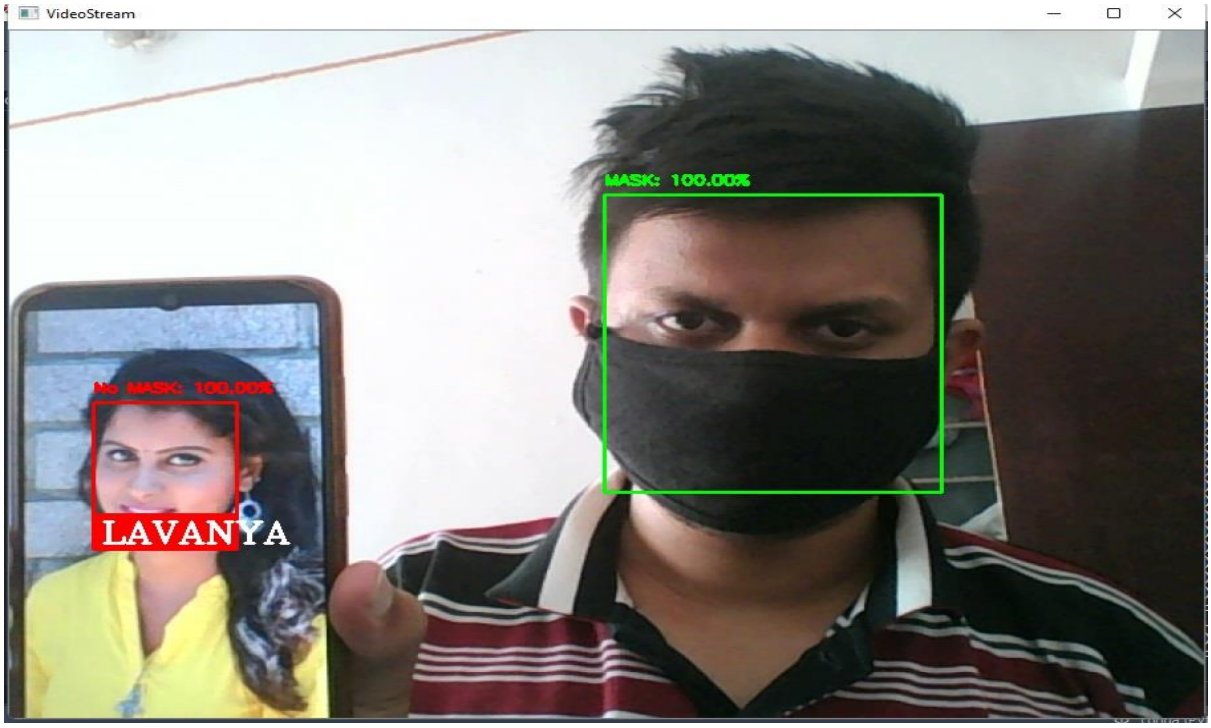

Figure 9.8: multiple face detection with mask and without mask

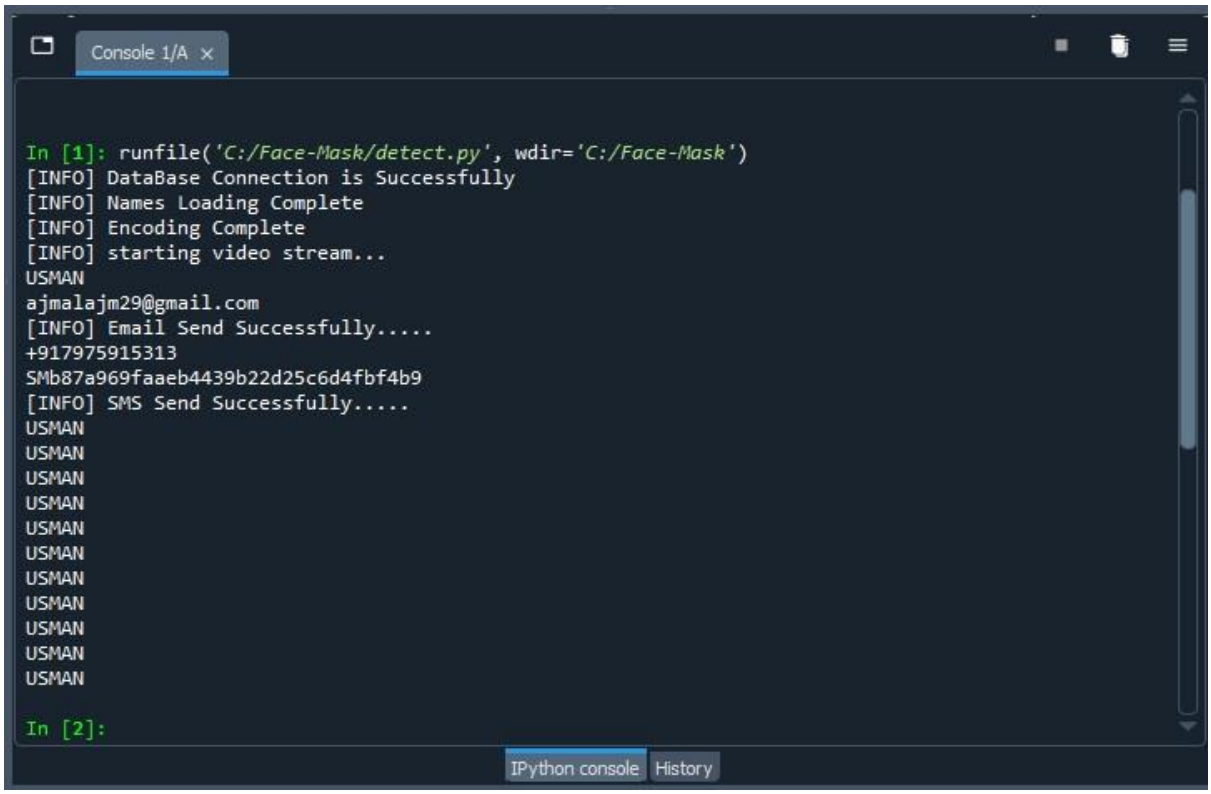

Figure 9.9: Email and Sms Sending

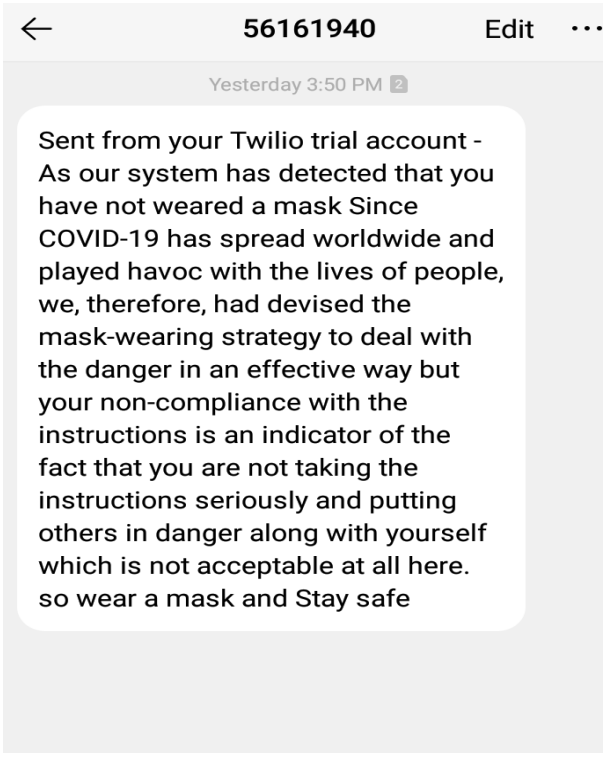

Figure 9.10: Sms Received

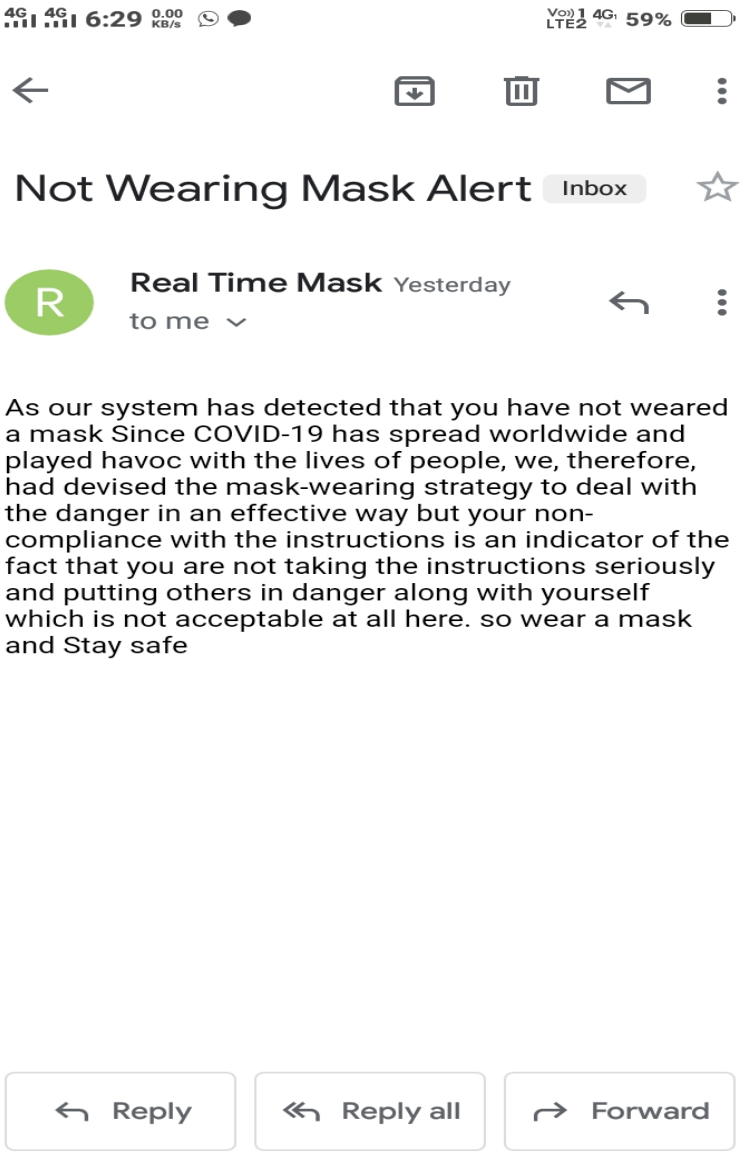

Figure 9.11: Email Received

CONCLUSION

As the technology are blooming with emerging trends the availability so we have novel face mask detector which can possibly contribute to public healthcare. The technique of OpenCV deep neural networks used in this model generated fruitful results. Classification of images was done accurately using the MobilenetV2 image classifier, which is one of the uniqueness of the proposed approach. The architecture consists of Mobile Net as the backbone it can be used for high and low computation scenarios. In order to extract more robust features, we utilize transfer learning to adopt weights from a similar task face mask detection, which is trained on a very large dataset.

We used OpenCV, tensor flow, keras and DNN to detect whether people were wearing face masks or not. if the person not wearing mask then the face will be recognized and the recognized person will get alert message in the form of sms and email. The models were tested with images and real-time video streams. The accuracy of the model is achieved and, the optimization of the model is a continuous process and we are building a highly accurate solution by tuning the hyper parameters. This specific model could be used as a use case for edge analytics. Furthermore, the proposed method achieves state-of-the-art results on a public face mask dataset. By the development of face mask detection we can detect if the person is wearing a face mask and allow their entry would be of great help to the society.

FUTURE ENHANCEMENTS

In the future if some very good accurate algorithm which give more accuracy from existing one then it can be used. The future work is to improve the recognition rate of algorithms when there are unintentional changes in a person like tonsuring head, using scarf, beard. The system developed detect and recognizes face up to 15 degrees' angle variations considering the left direction as 0degree, which has to be improved further. Gait recognition can be fused with face recognition systems in order to achieve better performance of the system. The concept of SURF features canbe implemented to increase the detection rate. The system will be deployed as a standalone which could be used by others. This will now be done using the MATLAB App builder. Some security features can be added to encrypt data such as sms and email credentials.

REFERENCE

[1]Deep Spiking Neural Network for Video-Based Disguise Face Recognition Based on Dynamic Facial Movements, By Daqi Liu , Nicola Bellotto , Member, IEEE, and Shigang Yue Senior Member.

[2]Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network. By, Mohammad A. Abuznied1 and Ausif Mahmood .

[3]CorrC2G: Color to Gray Conversion by Correlation, By Hossein Ziaei Nafchi, Atena Shahkolaei, Rachid Hedjam.

[4]Influence of color-to-gray conversion on the performance of document image binarization: toward a novel optimization problem, Rachid Hedjam, Hossein Ziaei Nafchi, Margaret Kalacska and Mohamed Cheriet.

[5]Deep Learning based Face Liveness Detection in Videos, Yaman AKBULUT, Abdulkadir SENGÜR, Ümit BUDAK, Sami EKICI.

[6]Face recognition on surgically altered faces usig principal component analysis, By Radhakrishnan.B, Dr. L. Padma suresh.

[7]An advancement towards efficient face recognition using live video feed, By Jalendu Dhamija, tanupriya Choudhary, Praveen kumar, Yogesh singh rathore.

[8]Video Face Recognition Using Siamese Networks with Block-Sparsity Matching, By Fania Mokhayeri, Student Member, IEEE, and Eric Granger.

http://mathworks.com

https://www.wikipedia.org

https://towardsdatascience.com

https://youtube.com