FACE MASK DETECTION AND RECOGNITION WITH EMAIL-VERIFICATION

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

Mayank Goel (111816023)

Saksham Sachdeva (111815031)

Shreyansh Gupta (111816034)

Semester: VII

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Indian Institute of Information Technology Pune

Near Bopdev Ghat, Kondhwa Annexe, Yewalewadi, Pune, Maharashtra 411048

NOVEMBER **2021**

BONAFIDE CERTIFICATE

This is to certify that the project report entitled FACE MASK DETECTION AND RECOGNITION WITH EMAIL VERIFICATION submitted by **Mayank Goel, Saksham Sachdeva and Shreyansh Gupta** bearing the **MIS No: 111816023, 111815031 and 111816034** respectively, in com-pletion of their project work under the guidance of **Dr. Bhupendra Singh** is accepted for the project report submission in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering in the, Indian Institute of Information Technology, Pune, during the academic year 2021-22.

Dr. Bhupendra Singh
Project Guide
Department of CSE
IIIT Pune

Dr. Tanmoy Hazra
Head of the Department
Department of CSE
IIIT Pune

Project Viva-voce held on November 13,2021

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

This project would not have been possible without the help and cooperation of many. We would like to thank the people who helped us directly and indirectly in the completion of this project work.

First and foremost, we would like to express our gratitude to our respected director, **Dr.Anupam Shukla**, for providing his kind support in various aspects.

We would like to express our gratitude to our project guide **Dr.Bhupendra Singh**, Professor, Department of CSE, for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this B.Tech project.

We would like to express our gratitude to the head of department **Dr. Tanmoy Hazra**, Professor, Department of CSE, for providing his kind support in various aspects.

We would also like to thank all the faculty members in the Dept. of CSE and our classmates for their steadfast and strong support int this project.

ABSTRACT

The purpose of this document is to present a detailed description of our product, face mask detection and recognition with email verification. It will explain the purpose and features of the system, what the system does and the constraints under which it must operates. This document is intended for both the clients and the developers of the system.

It is a platform for corporates and employers who want to keep the professional working environment safer during the pandemic time. In this we have considered face mask detection and face recognition for generating an alert through email. The interface aims to accurately detect the face cover and significantly generate an e-mail warning to the person and the office for safekeeping the public interactions at workplace via trained machine learning model.

The major technologies used in development of the project include OpenCV, numpy to collect the data, and Python, Keras, Tensorflow to train and test our model.

Keywords : Python, OpenCV, Keras, Tensorflow.

TABLE OF CONTENTS

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Welcome

Our platform is a user friendly interface which can help various organizations to maintain a safe working environment. In this pandemic crisis one needs to wear mask and maintain personal hygiene, so we have developed an autonomous software which detects face mask and recognizes the person, if not.

Additionally it raises an email warning to the person violating the rules. In this way one will always maintains social distancing resulting in less covid cases within the organization.

1.2 Outline

The report is organized as follows:

Chapter 1 provides a general introduction to the report.

Chapter 2 introduces the motivation, necessary background and the practical knowledge required.

Chapter 3 talks about concepts of the technology stack used.

Chapter 4 explains structure and working of the interface.

Chapter 5 provides a brief overview the proposed work and it analyses the use cases for various users.

Chapter 6 provides an overview of the working of the platform and the results obtained.

Chapter 7 concludes our work and gives the future work which can be done to improve this scheme.

Chapter 2

Motivation

Due to the subsequent increase in the covid cases during the pandemic, safe-guarding the public was a major issue to put light on. Wearing mask while being in the public place/ organization would reduce the chances of virus transmission. So our product, helps in maintaining the safety measures in any professional environment workplace. Using official alert warnings through office emails will further ensure their protection and at the same moment preserving the office from any further transmission of Covid.

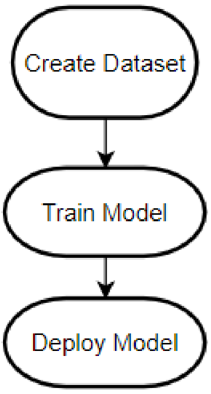


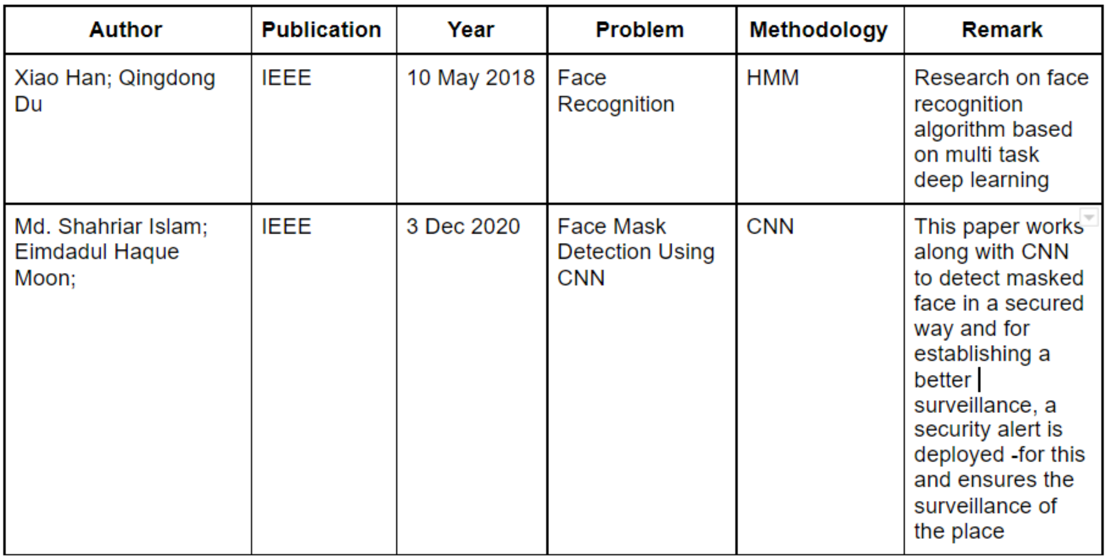
Figure 2.1: Project Flow

No necessary background knowledge is required. The software automatically detects the violations through the surveillance cameras. A good knowledge of Machine learning and related algorithms is required to train a model. In this interface we proposed a feasible software for face mask detection with knowledge of Convolutional Neural Networks (CNN), and classification algorithms. The interface is user friendly and can be understood easily, just like any other API.

Chapter 3

Literature Review

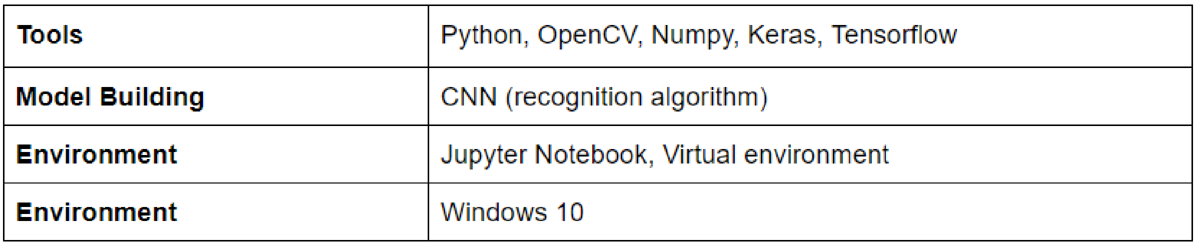
In this chapter, we will be discussing the details about the research paper and the technologies used.



Author	Publication	Year	Problem	Methodology	Remark
Xiao Han; Qingdong Du	IEEE	10 May 2018	Face Recognition	HMM	Research on face recognition algorithm based on multi task deep learning
Md. Shahriar Islam; Eimdadul Haque Moon;	IEEE	3 Dec 2020	Face Mask Detection Using CNN	CNN	This paper works along with CNN to detect masked face in a secured way and for establishing a better surveillance, a security alert is deployed -for this and ensures the surveillance of the place

Table 3.1: Literature Review

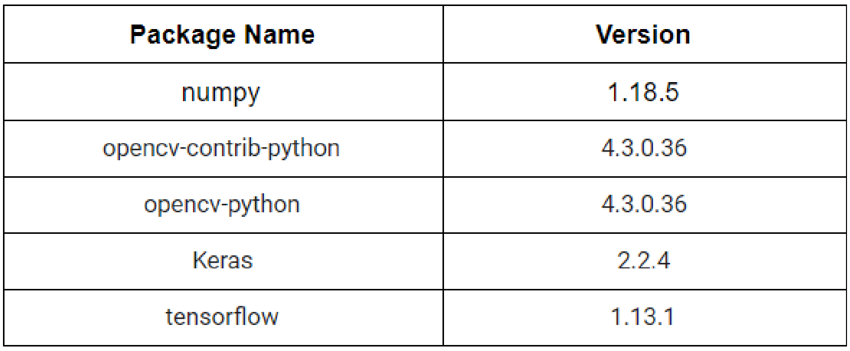
In this project, we focus on the research hotspots of face recognition based on depth learning in the field of bio metrics. Using convolutional neural network gives higher accuracy than the rest of other algorithms, also it quite fast compares with other algorithms. Detection of a masked face, combined with the relevant theory and methods of depth learning based on the face recognition, showed a higher accuracy rate and capable of faster detecting the mask face and without mask face of a person, which helps systematically detection of a person over the visual detection.



Tools	Python, OpenCV, Numpy, Keras, Tensorflow
Model Building	CNN (recognition algorithm)
Environment	Jupyter Notebook, Virtual environment
Environment	Windows 10

Table 3.2: Technology Stack

In our project we have used OpenCV which is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection. For training and testing our model we have used CNN which an efficient recognition algorithm and is widely used in pattern recognition and image processing. Generally, the structure of CNN includes two layers one is feature extraction layer, the input of each neuron is connected to the local receptive fields of the previous layer, and extracts the local feature.



Package Name	Version
numpy	1.18.5
opencv-contrib-python	4.3.0.36
opencv-python	4.3.0.36
Keras	2.2.4
tensorflow	1.13.1

Table 3.3: Packages Used

The environment we used to compile our code is created through Anacond using conda environment and installing all required packages and then using Jupyter Notebook to compile and run our model.

Chapter 4

Analysis And Design

This chapter describes basic steps of design and structure that we followed during the development of this project.

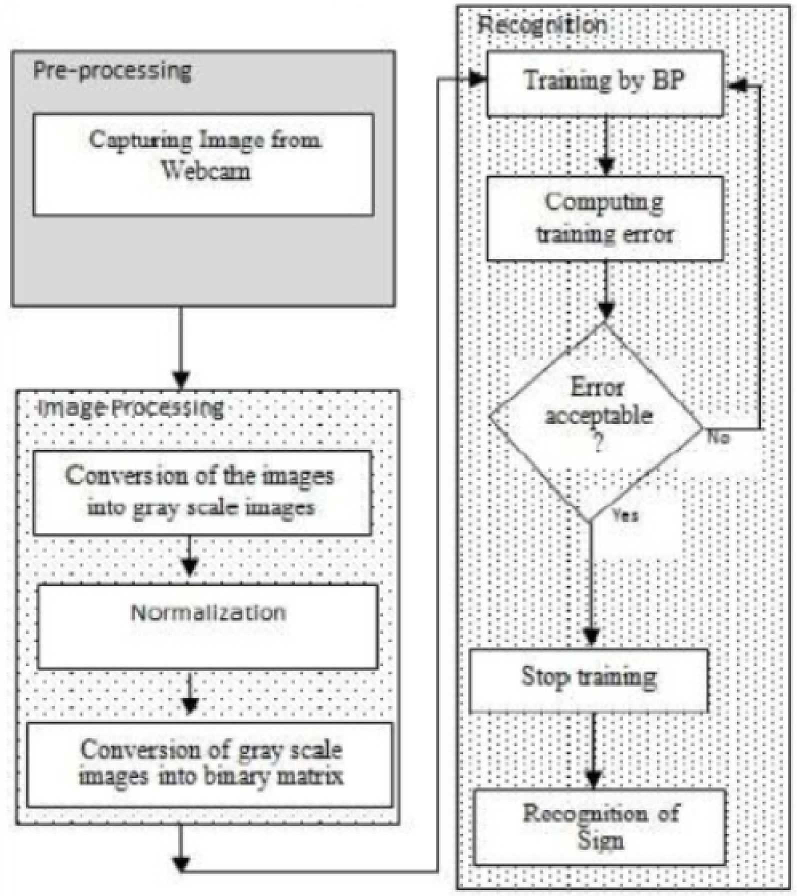
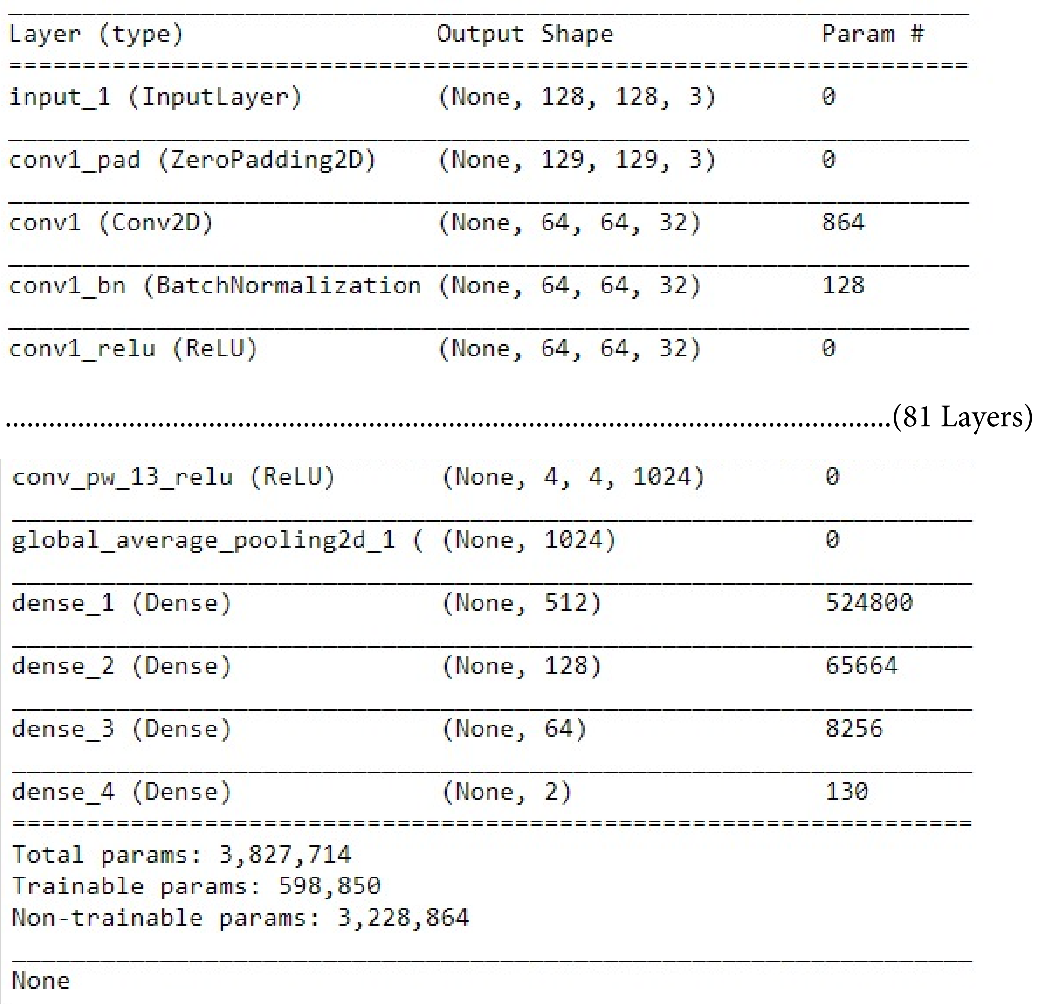


Figure 4.1: Steps stating the development design

Face Mask Detection and Recognition with Email Verification is developed in two steps, data acquisition and classification. First of all we needed to gather data on which we would train our model. In our project we have used publicly available dataset for model training. Now, we have our mask detection model ready. Secondly we will create a model for face recognition to recognise the person found violating the rules, for sending mail warnings. We use surveillance cameras to gather images.

Currently we have a working model for face mask detection which correctly identifies whether a person has a face mask on or not. We have used the concept of Transfer Learning, mainly, MobileNet for model training along with our custom FC layers to produce predictions in our desired format.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 128, 128, 3)       0
_________________________________________________________________
conv1_pad (ZeroPadding2D)    (None, 129, 129, 3)       0
_________________________________________________________________
conv1 (Conv2D)               (None, 64, 64, 32)        864
_________________________________________________________________
conv1_bn (BatchNormalization (None, 64, 64, 32)        128
_________________________________________________________________
conv1_relu (ReLU)            (None, 64, 64, 32)        0
```

....(81 Layers)

```
conv_pw_13_relu (ReLU)       (None, 4, 4, 1024)        0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               524800
_________________________________________________________________
dense_2 (Dense)              (None, 128)               65664
_________________________________________________________________
dense_3 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 130
=================================================================
Total params: 3,827,714
Trainable params: 598,850
Non-trainable params: 3,228,864
_________________________________________________________________
None
```

Figure 4.2: Layers used in training model

In the second step we train the model after adding FC layers with 512 units then 128 units then 64 units and finally 2 units to classify among the required classes.

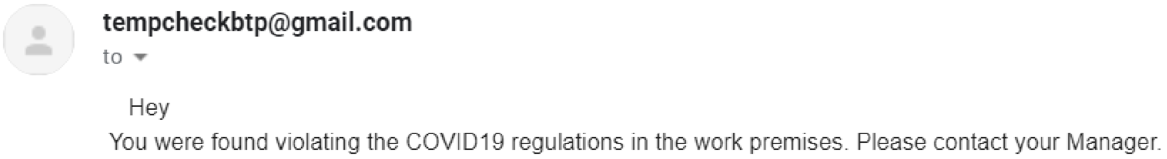
The next step is to extract features from a face using a face embedding model. A face embedding is a vector that represents the features extracted from the face and we an use these vectors to recognize faces. Note that face embedding for the same face may be really close in the vector space, whereas the face embeddings of two different faces may be really far away. We get a face embedding after passing the image through a face embedding model.

Figure 4.3: Face Embedding

Comparing faces: Now that we have face embeddings for every face in our data saved in a file, the next step is to recognise a new t image that is not in our data. So the first step is to compute the face embedding for the image using the same network we used above and then compare this embedding with the rest of the embeddings we have

Listi

Now after the face is recognized, we need to alert the person using e-mails. Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP. To send the mail you use smtpObj to connect to the SMTP server on the local machine and then use the sendmail method along with the message, the from address, and the destination address as parameters.



tempcheckbtp@gmail.com
to

Hey
You were found violating the COVID19 regulations in the work premises. Please contact your Manager.

Figure 4.4: Email Sent to Person

Chapter 5

Proposed Work

This chapter explains our progress regarding the implementation of the interface. Now we will discuss how we created the model for face mask detection which we have considered and then we will use the dataset to train our model.The dataset is as follows:

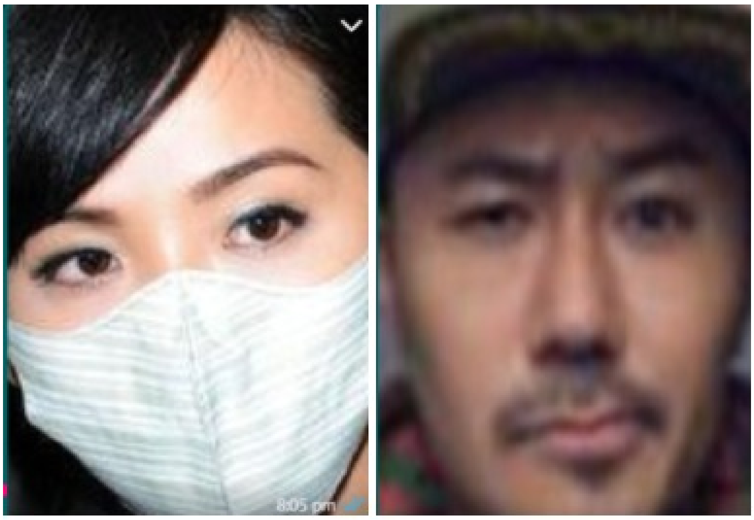
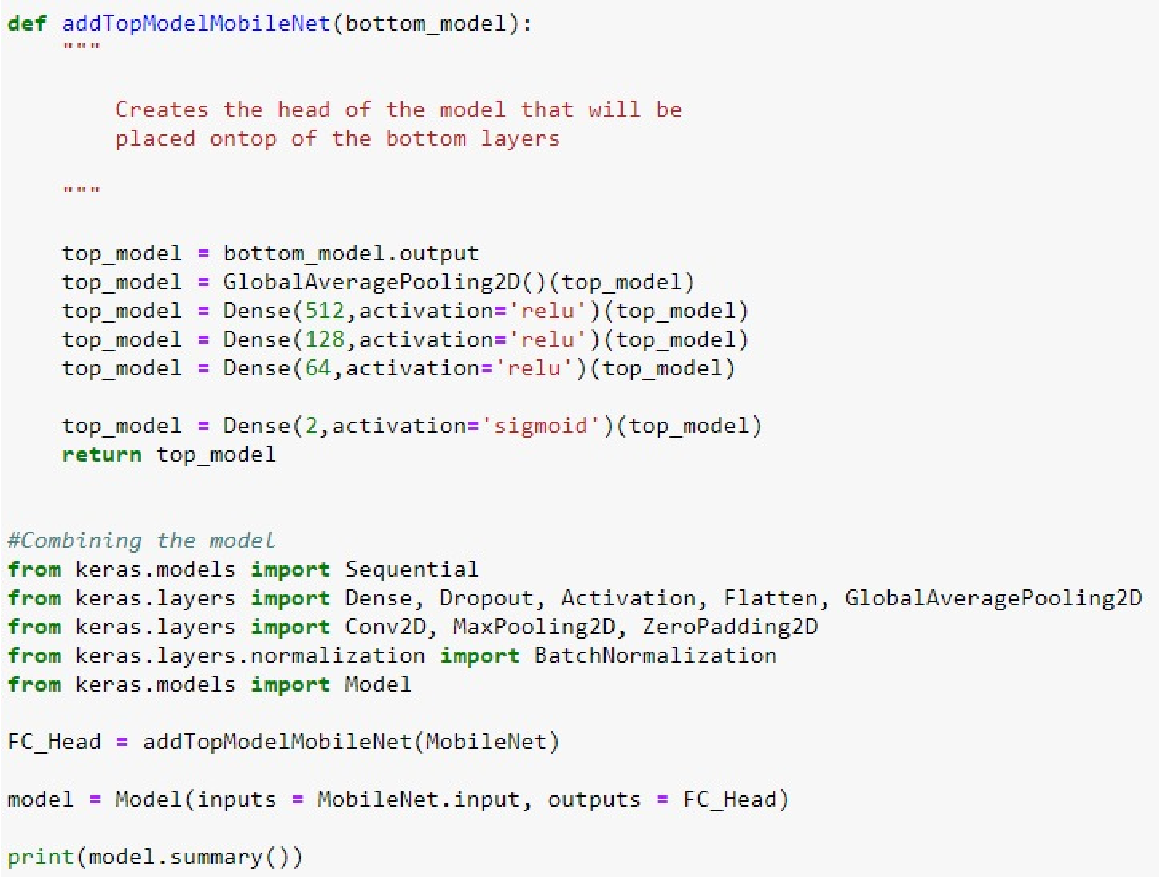


Fig 5.1: Data Set

The Dataset consists of facial images divided into two classes: with mask and without mask. After generating data set for training and testing respectively we will train our model. Our model consists of various convolutional layers which helps to find feature map of all signs and map it to particular class for which that image belongs to. Involving different CNN layers determine the accuracy and precision of our model. In Subsequent model we will predict person name without mask and will provide email verification system.

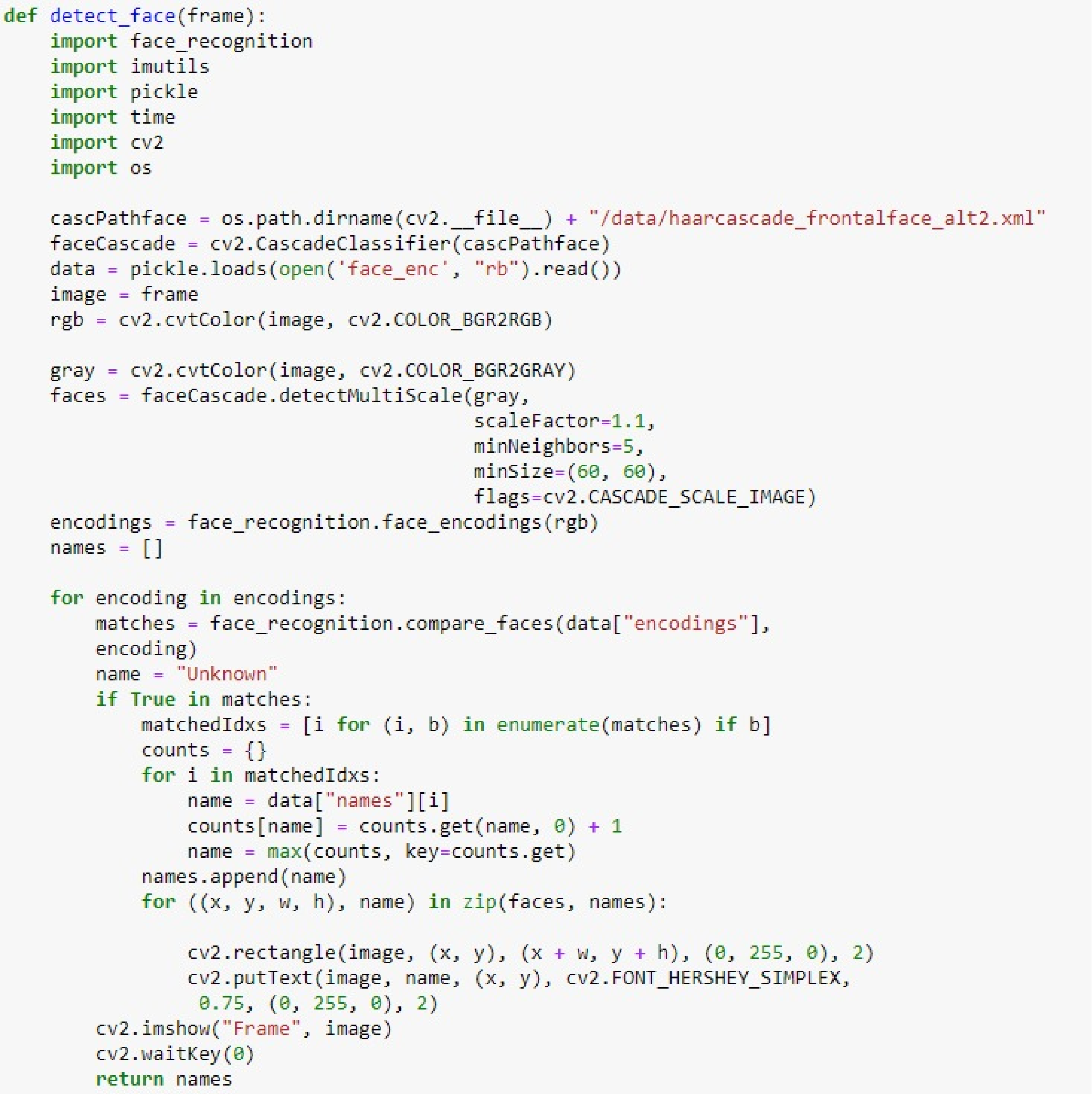


```python
def addTopModelMobileNet(bottom_model):
    """

        Creates the head of the model that will be
        placed ontop of the bottom layers

    """

    top_model = bottom_model.output
    top_model = GlobalAveragePooling2D()(top_model)
    top_model = Dense(512,activation='relu')(top_model)
    top_model = Dense(128,activation='relu')(top_model)
    top_model = Dense(64,activation='relu')(top_model)

    top_model = Dense(2,activation='sigmoid')(top_model)
    return top_model


#Combining the model
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten, GlobalAveragePooling2D
from keras.layers import Conv2D, MaxPooling2D, ZeroPadding2D
from keras.layers.normalization import BatchNormalization
from keras.models import Model

FC_Head = addTopModelMobileNet(MobileNet)

model = Model(inputs = MobileNet.input, outputs = FC_Head)

print(model.summary())
```

Listing 5.1: Model Architecture

Currently we have a working model for face mask detection which correctly identifies whether a person has a face mask on or not. We have used the concept of Transfer Learning, mainly, MobileNet for model training along with our custom FC layers to produce predictions in our desired format. So we train the model after adding FC layers with 512 units then 128 units then 64 units and finally 2 units to classify among the required classes. We found that we have 92 layers in our model, with over three million parameters

In out next phase, we try to detect the person's face if he/she is not wearing mask, which comes under our face recognition model. Firstly, if the person is found with the mask, then he passes the security without any violation. In the second case, if he/she is found with any obligation objective to the rules, we try to detect the person's face considering if he/she is wearing mask or not, which comes under our face recognition model.

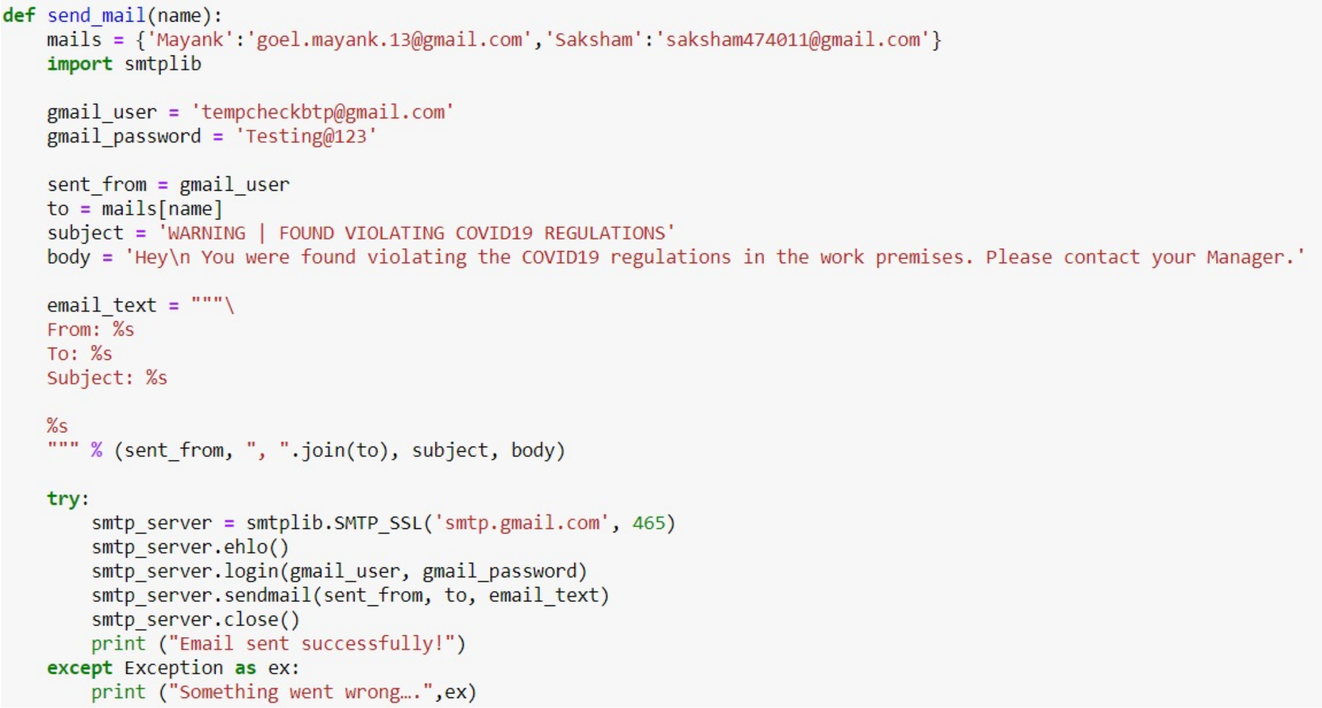


```python
def detect_face(frame):
    import face_recognition
    import imutils
    import pickle
    import time
    import cv2
    import os

    cascPathface = os.path.dirname(cv2.__file__) + "/data/haarcascade_frontalface_alt2.xml"
    faceCascade = cv2.CascadeClassifier(cascPathface)
    data = pickle.loads(open('face_enc', "rb").read())
    image = frame
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray,
                                         scaleFactor=1.1,
                                         minNeighbors=5,
                                         minSize=(60, 60),
                                         flags=cv2.CASCADE_SCALE_IMAGE)
    encodings = face_recognition.face_encodings(rgb)
    names = []

    for encoding in encodings:
        matches = face_recognition.compare_faces(data["encodings"],
        encoding)
        name = "Unknown"
        if True in matches:
            matchedIdxs = [i for (i, b) in enumerate(matches) if b]
            counts = {}
            for i in matchedIdxs:
                name = data["names"][i]
                counts[name] = counts.get(name, 0) + 1
                name = max(counts, key=counts.get)
            names.append(name)
            for ((x, y, w, h), name) in zip(faces, names):

                cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
                cv2.putText(image, name, (x, y), cv2.FONT_HERSHEY_SIMPLEX,
                 0.75, (0, 255, 0), 2)
        cv2.imshow("Frame", image)
        cv2.waitKey(0)
        return names
```

Listing 5.2: Face Recognition Design

At first,we take the input from as frames for the person who was found violating rules in the first phase. We take there picture captures and pass it through our face recognition model built on our own dataset, based on known personalities and can be improved on need bases.

After running through our phases in which at first, we ran through our face-mask detection for mask verification. After checking it, if the person was found breaching the rules, we try to detect the name from the capture of the frontal face of person. After recognising the person, if the person got identified in the list from our database, we send a firm admonition to that person found violating rules through email.



```
def send_mail(name):
    mails = {'Mayank':'goel.mayank.13@gmail.com','Saksham':'saksham474011@gmail.com'}
    import smtplib

    gmail_user = 'tempcheckbtp@gmail.com'
    gmail_password = 'Testing@123'

    sent_from = gmail_user
    to = mails[name]
    subject = 'WARNING | FOUND VIOLATING COVID19 REGULATIONS'
    body = 'Hey\n You were found violating the COVID19 regulations in the work premises. Please contact your Manager.'

    email_text = """\
    From: %s
    To: %s
    Subject: %s

    %s
    """ % (sent_from, ", ".join(to), subject, body)

    try:
        smtp_server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
        smtp_server.ehlo()
        smtp_server.login(gmail_user, gmail_password)
        smtp_server.sendmail(sent_from, to, email_text)
        smtp_server.close()
        print ("Email sent successfully!")
    except Exception as ex:
        print ("Something went wrong….",ex)
```

Listing 5.3: Send Email Function

Hereby, the count will be maintained for every person respectively. If the number of alerts generated exceeds the threshold, a minimum amount of penalty will be taken in account from the employee as per the regulations of the employer.

Chapter 6

Results and Discussion

This chapter provides the results of the Face Mask Detection and Recognition with Email Verification project.In this interface we proposed a feasible software for face mask detection and face recognition with knowledge of Convolutional Neural Networks (CNN), and classification algorithms. Due to the subsequent increase in the covid cases during the pandemic, safe-guarding the public was a major issue to put light on. Wearing a mask while being in a public place/ organization would reduce the chances of virus transmission.

Here we are able to successfully build a model that would distinguish a person wearing a face mask from a person not wearing it.

After classifying the people based on mask, our model will let those people with masks, pass without any objection. The person found violating the rules will be considered and their photo capture from the video footage moves on further in our model.
This frame is processed and moved ahead for face recognition. We use this picture and process with running it through the database first. If the person is found and verified from our database, we check for their credentials and send an email warning to the person, and further the steps will be followed as per the demand from the employee to safegaurd their work environment from employees violating rules.

Chapter 7

Conclusion

To conclude, with our implementation of this platform employers who want professionalism and hygiene at the workplace can use our product to maintain the safety measures with ease through our software.

And further we will be training our model to detect the face of the employee as input and give an output by raising an alert through an email to the employee and the employer. The interface would be developed from classifying only the employee images, to the system that can successfully recognize dynamic mask movements that come in continuous sequences of images.

Chapter 8

References

https://ieeexplore.ieee.org/document/8356995

[1] Xiao Han; Qingdong Du. An order of depth learning to classify face recognition technology. (IEEE 2019)

[2] Md. Shahriar Islam; Eimdadul Haque Moon. This paper works along with CNN to detect masked face in a secured way and for establishing a better surveillance, a security alert is deployed - for this and ensures the surveillance of the place (IEEE 2020)

[3] Maliha Khan; Sudeshna Chakraborty. Framework for the use of face recognition application authentication. (IEEE Oct. 2019) (ICCCIS Conference 2019)