A

Major Project

On

VIDEO BASED MASK DETECTION FOR COVID

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

BY

Ryan Julius	(17601A0570)
Vivek Mittal	(177R1A0560)
Rugved R	(177R1A0549)

Under the Guidance of

M.MALYADRI REDDY

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2017-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled "**VIDEO BASED MASK DETECTION FOR COVID**" being submitted by **RYAN JULIUS (17601A0570),VIVEK MITTAL (177R1A0560)** and **RUGVED R (177R1A0549)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2020-21.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

INTERNAL GUIDE
Mr. M.Malyadri Redddy
Associate Professor

DIRECTOR
Dr. A. Raji Reddy

HOD
Dr. K. Srujan Raju

EXTERNAL EXAMINER

Submitted for viva voce Examination held on ____________________ ___

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

Mr. M.Malyadri Reddy, Assoc. Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mrs. Najeema Afrin, Mr. B.Deepak Kumar** and **Mr. J. Narasimha Rao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project

The guidance and support received from all the members of **CMR TECHNICAL CAMPUS** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

RYAN JULIUS (17601A0570)
VIVEK MITTAL (177R1A0560)
RUGVED R (177R1A0549)

ABSTRACT

COVID-19 pandemic has rapidly affected our day-to-day life disrupting many different sectors of the industry. Wearing a protective face mask has become a new normal. An individual that doesn't wear a mask poses a threat to the safety of the group of people in the area. As one year has passed living with the virus it has also been shown that many individuals don't wear their mask if they are not told to do so. At this stage it is very important to ensure that each and every one of us wears a mask. In a vicinity that consists of a group of people such as schools, colleges, movie theatres etc. we can use image processing to recognize how many individuals are consistently wearing their masks. This proposed system also ensures that if an individual is to ever remove his/her mask, the relevant authority is informed about it.If deployed correctly, our product could potentially be used to help ensure our safety and the safety of others around us.This allows the college/org to be more efficient and save time compared to the standard methods available.

LIST OF FIGURES

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

Using facial landmarking to solve issues of mask detection that are simple to accomplish for the consumer.This project aims to ensure organizations can detect if the individuals in the room are wearing a mask, we want to ensure that this takes place in the most safe and efficient way possible which is also relatively contactless.

1.2 PROJECT PURPOSE

This has been developed to facilitate the detection of students or employees of an organisation using facial landmarking in order to ensure all individuals are wearing their masks.There are no Standard systems in the market as it is too expensive to make the product on a large scale.

Our project aims to combat all these issues while also making sure to keep the total cost down.

1.3 PROJECT FEATURES

Some of the main features of this project are :

<u>Quick Implementation:</u>Once the program is running it detects the face mask almost immediately and gives the score out of 100 based on which necessary actions can be performed

<u>High Accuracy:</u>The model that we use has been given a good rating in terms of how accurately the mask gets recognized.

<u>Low Cost:</u>The model can run efficiently even on low end hardware which decreases the cost of the project.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.2 PROBLEM DEFINITION

A detailed study of the process must be made by various techniques like interviews,questionnaires etc.The data collected by these sources must be scrutinized to arrive at a conclusion.The conclusion is an understanding of how the system functions. This system is called the existing system.Now the existing system is subjected to close study and problem areas are identified.The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces.The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected.The proposal is presented to the user for an endorsement by the user.The proposal is reviewed on user request and suitable changes are made.This loop ends as soon as the user is satisfied with the proposal.

2.3 EXISTING SYSTEM

As of now there has been no industrial implementation of this project as it is still in its infancy stage. This product's future also heavily revolves around how the pandemic situation will turn out in the coming years.

However, there are many facial recognition software that are in the market that leverage facial landmarks to extract the region of interest which is foundational for both facial recognition and our mask detection software.We hope to take advantage of this to come out with a product that benefits the public.

2.3.1 LIMITATIONS OF EXISTING SYSTEM

- Cannot authenticate mass group of users
- Time consuming
- Requires physical input from user

2.4 PROPOSED SYSTEM

In this project, we propose an end to end system which solves the issue of individuals not wearing masks using image processing and facial landmarking.

This proposed system can be split into the following phases:

1.Face mask Training:

a) Load Face Mask Dataset

- First load dataset from disk, Face mask sample data into the database.

b) Train Face Mask Classifier

- Train the model using (Keras /TensorFlow) on the given Dataset.

2. Facemask Detection:

a) Load Facemask Classifier

-Load Training data and process it in the Classifier.

b) Detect faces

-The Image will get Detected based on the constraints given.

c) Extract Face roi

-Only the region of interest will be extracted.

d) Apply Facemask Classifier

-Once the mask detector is trained, we can then move on to loading the mask detector, performing face detection.

e) Segregate face roi as ‘mask’ or ‘no mask’

-Classifying each face as with_mask or without_mask.

2.4.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Fully automated and does not require physical assistance
- Easy implementation and installation using containerization.
- Can verify mass groups of users.

2.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company.

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.5.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it gives an indication that the system is economically possible for development.

2.5.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.5.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

Processor	:	Intel Core i3 or higher
RAM	:	4GB and Above.

Raspberry pi 4 (optional)

High Resolution Camera (720p/1080p)

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

Operating System	:	Windows/Linux/Rasbian OS
Languages	:	Python
IDE	:	PyCharm

Technology used : 1) TensorFlow

2) imutils

3) sklearn

4) OpenCV

5) NumPy

6) argparse

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture describes how a masked_face will be identified and compared with a stored database. The detailed architecture is explained below.

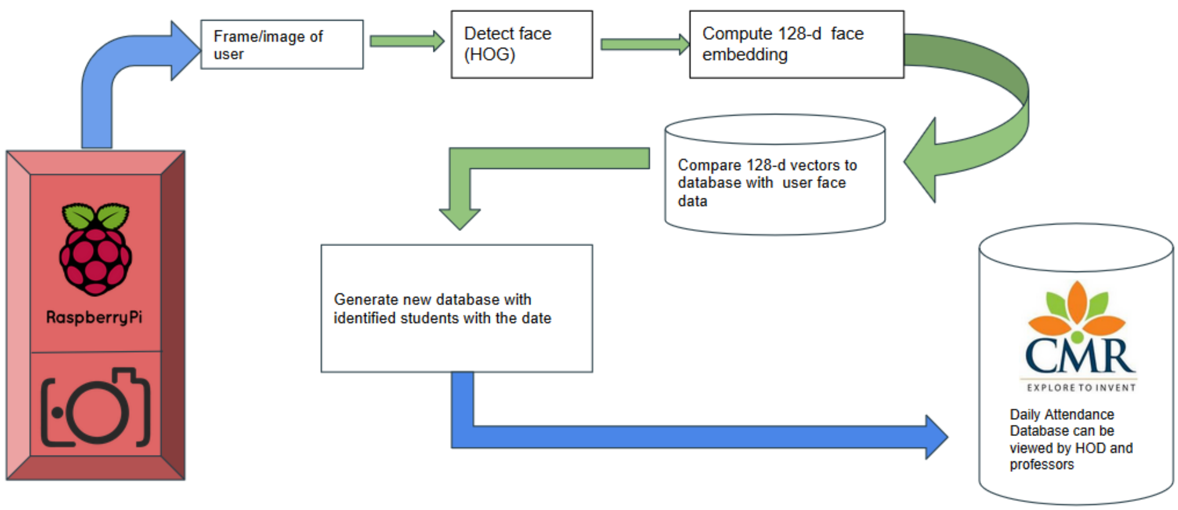

Fig. 3.1 Project Architecture of Video Based Mask Detection

3.2 MODULES DESCRIPTION

Modules

- Administrator
- User

3.2.1 ADMINISTRATOR

- create and edit user face database
- implement diff ML model(HOG CNN)/update system
- generate CSV file

3.2.2 USER

- If the user is a student:
  - add face to database:
  - view attendance:
  - report attendance issue:
- If the user is a professor:
  - view class attendance:
  - report student:

3.3 USE CASE DIAGRAM

In the use case diagram we have basically two actors who are the user and the administrator. The user face will be detected. Whereas the administrator has the access to detect whether the user is wearing a mask or not.And admin has the authority to decide whether to allow a person or not based on the score.

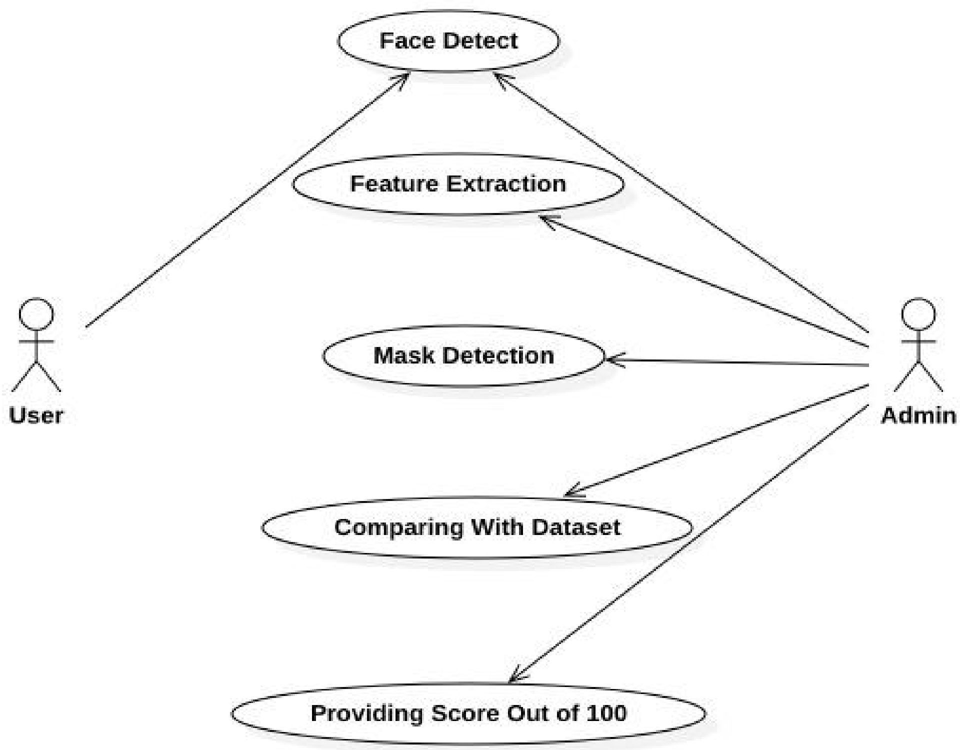

Fig. 3.2 Use Case Diagram for Admin and User for Video Based Mask Detection

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.

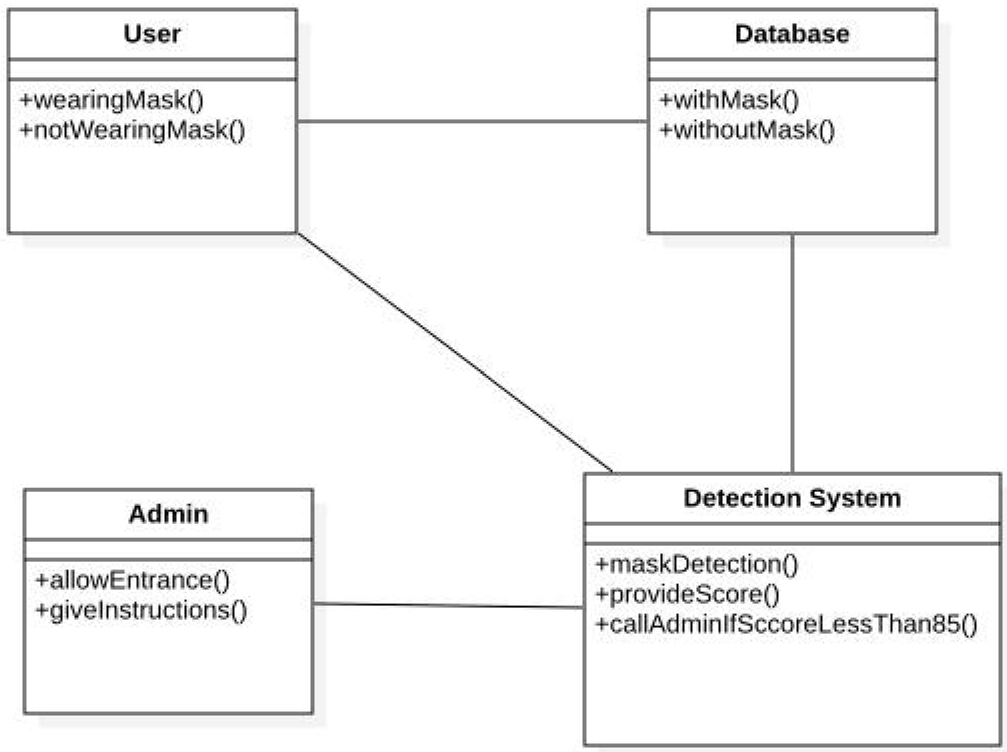

Fig. 3.3 Class Diagram for Administrator and user for Video Based Mask Detection

3.5 SEQUENCE DIAGRAM

The user face will be detected and the software identifies whether the person is wearing a mask or not.Entrance is provided based on the score.

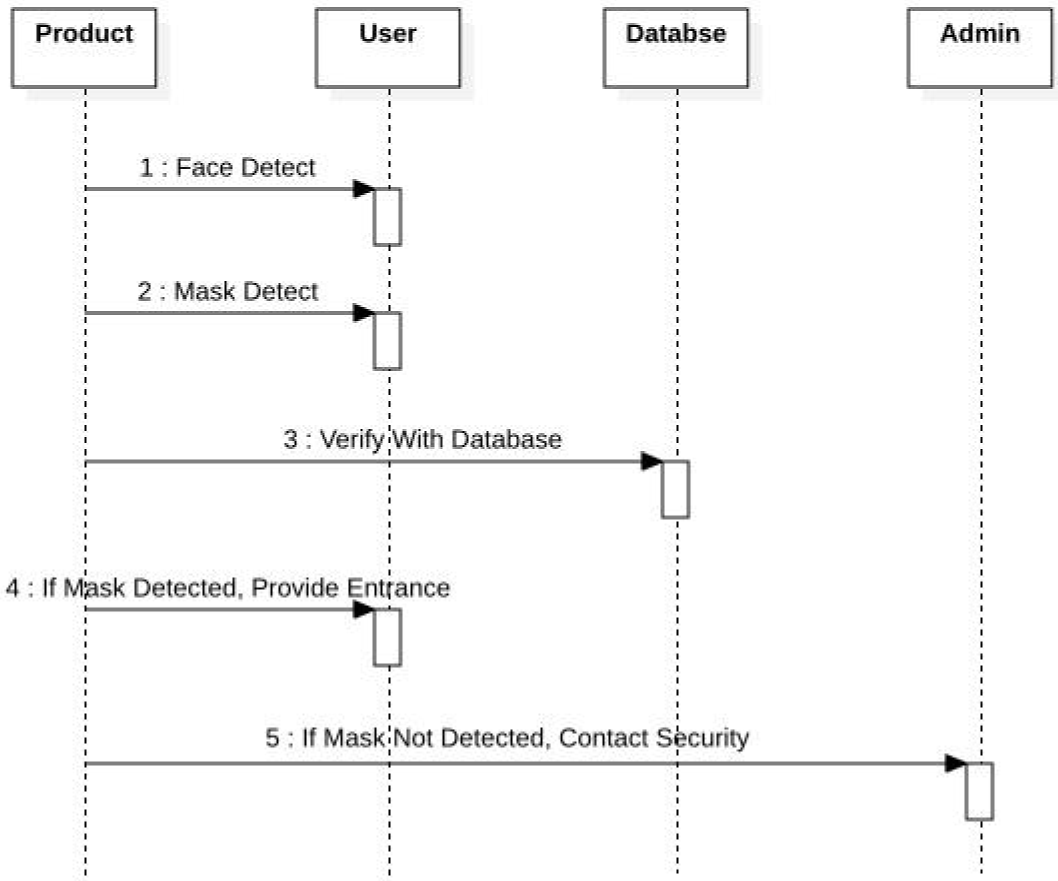

Fig.3.4 Sequence Diagram for Video Based Mask Detection

3.6 ACTIVITY DIAGRAM

It describes the flow of activity states.

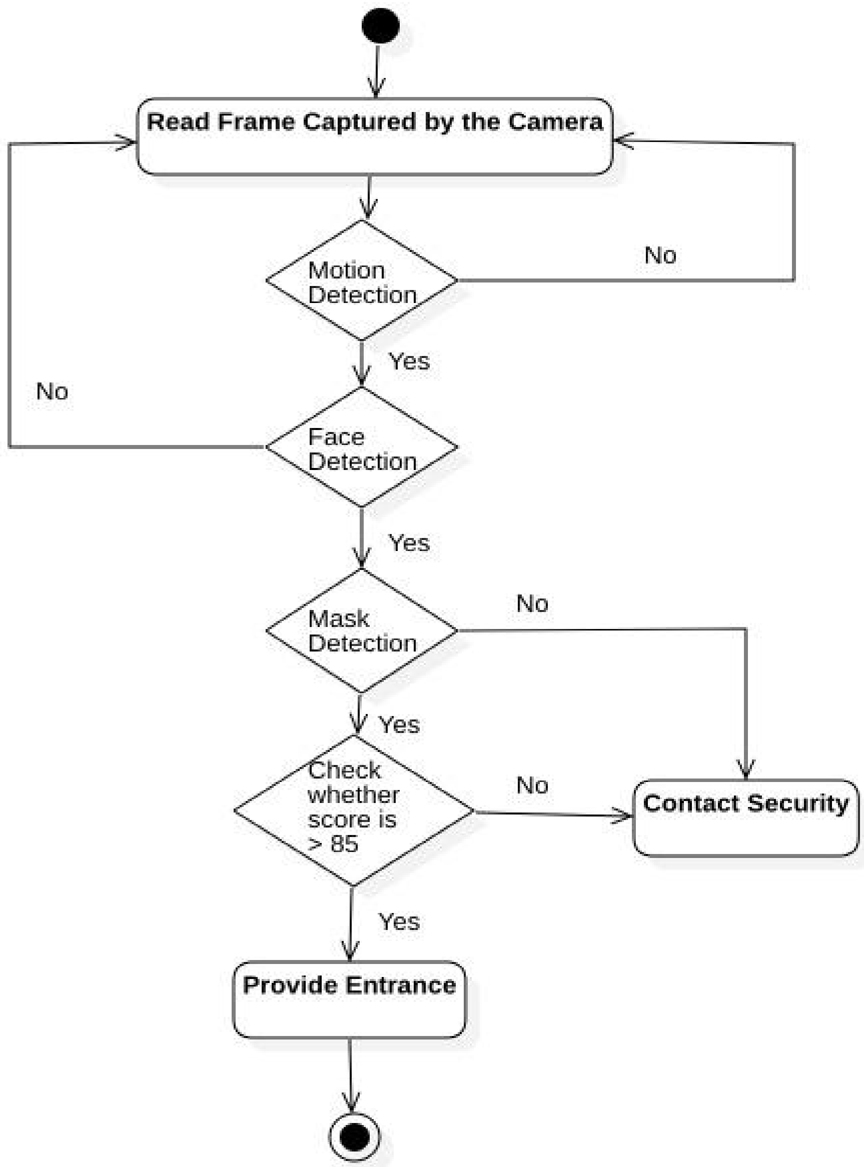

Fig. 3.5 Activity Diagram for Admin for Video Based Mask Detection

4.IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

train_mask_detector.py

```
# 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:\Mask Detection\CODE\Face-Mask-Detection-master\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
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
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
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")
```

detect_mask_video.py

```
 # 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
import numpy as np
import imutils
import time
import cv2
import os
```

```
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]
 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]):
   # extract the confidence (i.e., probability) associated with
   # the detection
   confidence = detections[0, 0, i, 2]

   # filter out weak detections by ensuring the confidence is
   # greater than the minimum confidence
   if confidence > 0.5:
     # compute the (x, y)-coordinates of the bounding box for
     # the object
     box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
     (startX, startY, endX, endY) = box.astype("int")

     # ensure the bounding boxes fall within the dimensions of
     # the frame
     (startX, startY) = (max(0, startX), max(0, startY))
     (endX, endY) = (min(w - 1, endX), min(h - 1, endY))

     # extract the face ROI, convert it from BGR to RGB channel
```

```
        # ordering, resize it to 224x224, and preprocess it
        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)

        # add the face and bounding boxes to their respective
        # lists
        faces.append(face)
        locs.append((startX, startY, endX, endY))

    # only make a predictions if at least one face was detected
    if len(faces) > 0:
        # for faster inference we'll make batch predictions on *all*
        # faces at the same time rather than one-by-one predictions
        # in the above `for` loop
        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 from 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
while True:
  # grab the frame from the threaded video stream and resize it
  # to have a maximum width of 400 pixels
  frame = vs.read()
  frame = imutils.resize(frame, width=400)




  # detect faces in the frame and determine if they are wearing a
  # face mask or not
  (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

  # loop over the detected face locations and their corresponding
  # locations
  for (box, pred) in zip(locs, preds):
    # unpack the bounding box and predictions
    (startX, startY, endX, endY) = box
    (mask, withoutMask) = pred

    # determine the class label and color we'll use to draw
    # the bounding box and text
    label = "Mask" if mask > withoutMask else "No Mask"
    color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

    # include the probability in the label
    label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

    # display the label and bounding box rectangle on the output
    # frame
    cv2.putText(frame, label, (startX, startY - 10),
      cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
    cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

  # show the output frame
  cv2.imshow("Frame", frame)
  key = cv2.waitKey(1) & 0xFF
```

```
 # if the `q` key was pressed, break from the loop
  if key == ord("q"):
     break

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

5.SCREENSHOTS

5. SCREEN SHOTS

5.1 DATABASE

Two databases,one with_mask and one without_mask are used to train the model.

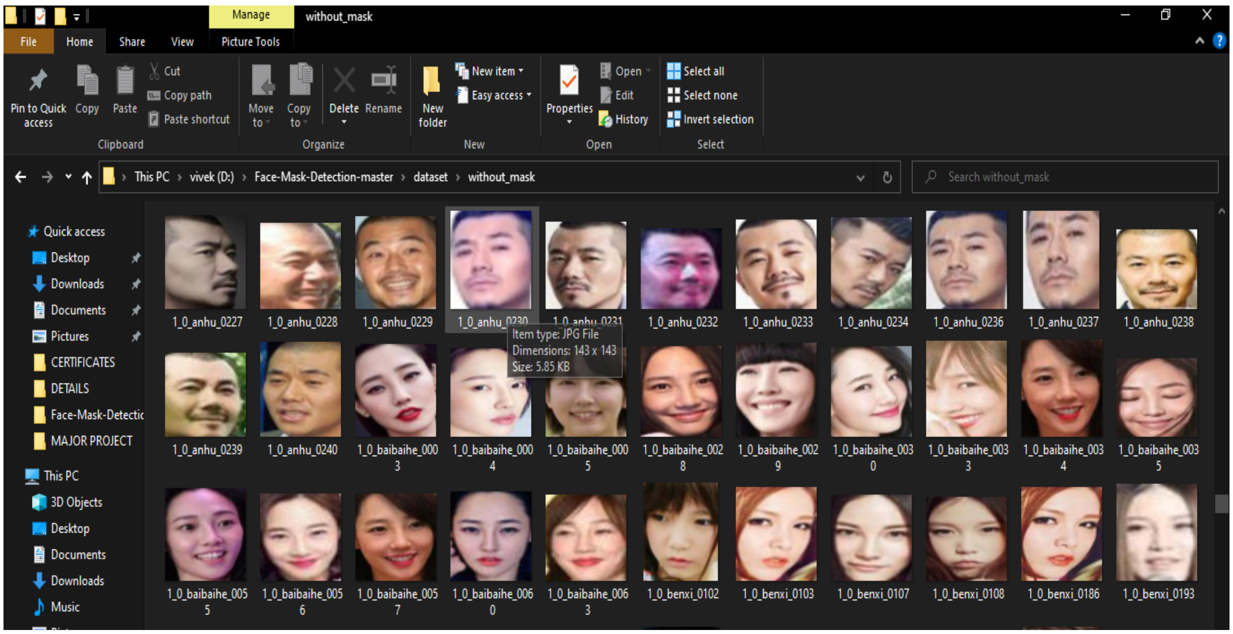

Fig 5.1 :Without_mask Database Folder

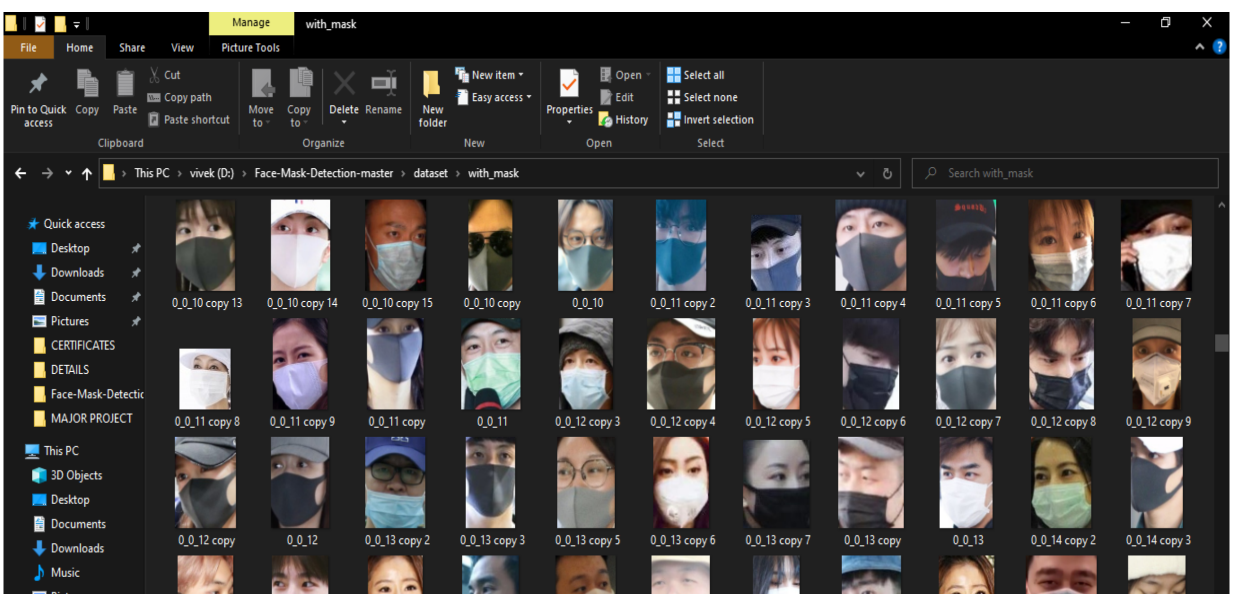

Fig 5.2 : With_mask Database Folder

5.2 WITHOUT_MASK

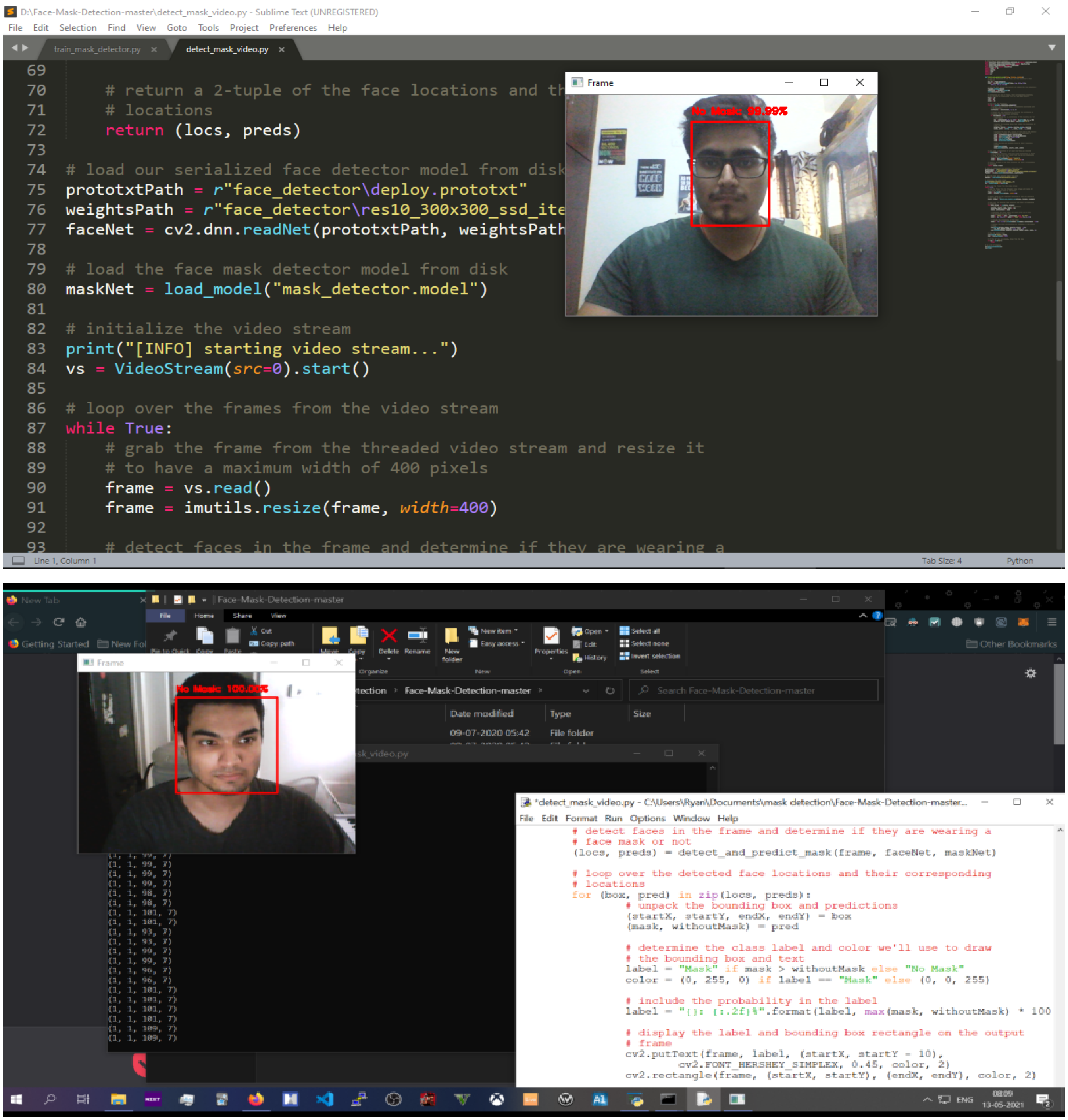

Fig 5.2.: Without_mask_detection

5.3 WITH_MASK

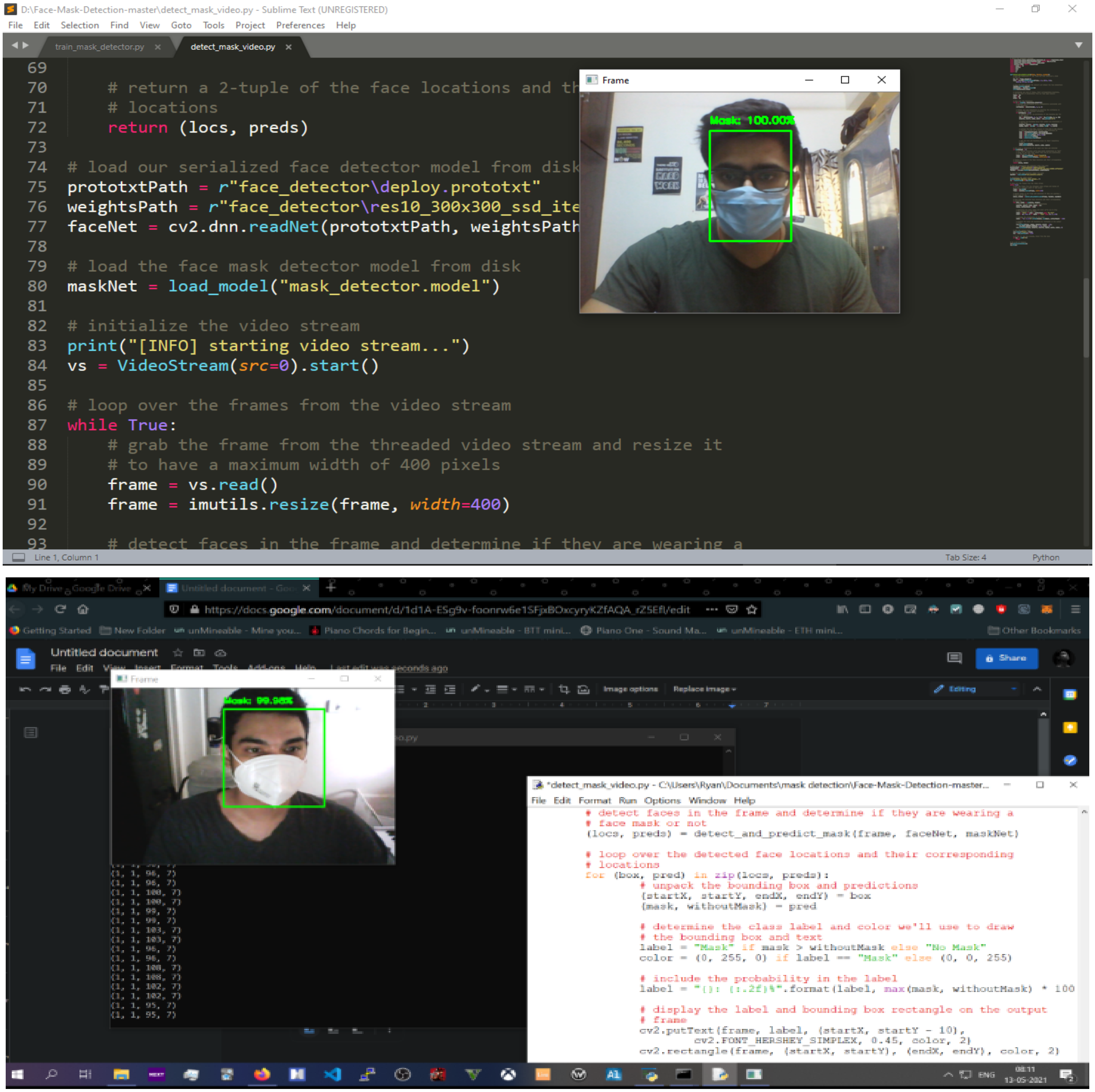

Fig 5.3.: With_mask_detection

6.TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.3 TEST CASES

6.3.1 NEW USER ID CREATION

Test Case ID	Test Case Name	Purpose	Test Case	Output
1	Without_mask	Creating a New User ID	User gives his/her name, roll number and a picture that's taken in good lighting	User ID1 for the user has been successfully created
2	New User ID Creation	Creating a New User ID	User gives his/her name, roll number and a picture that's taken in good lighting	User ID1 for the user has been successfully created
3	New User ID Creation	Creating a New User ID	User gives his/her name, roll number and a picture that's taken in good lighting	User ID1 for the user has been successfully created

7.CONCLUSION

7. CONCLUSION & FUTURE ENHANCEMENTS

7.1 PROJECT CONCLUSION

The project titled as “Video Based Mask detection” is a software application.As Covid-19 is a rising concern, it is necessary to ensure that everyone wears a mask, our system ensures it is done in the most handsfree way as possible and our aim as a team to ensure we work towards that problem while also tackling other objectives such as efficiency, cost , portability and implementation.

With enough development we believe this is a product that can be used in different industries all over the world. This software is developed with scalability in mind. Additional modules can be easily added when necessary. The software is developed with a modular approach. All modules in the system have been tested with valid data and invalid data and everything works successfully.Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

7.2 FUTURE ENHANCEMENTS

Some of the few enhancements we have in mind is to connect IOT devices such as the raspberry pi as an I/O device that sends the images to be detected to the organization's server. Using the gpio pins we can connect a display that displays all the individuals that don't wear a mask.As an alternative to local based processing , cloud based image recognition can also be added to improve the cost of implementation. This also allows the organizations to take up the product on a monthly/yearly service.

8.BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 References

[1]. World Wide Web

[2]. Pattern Recognition and Machine Learning by Christopher M. Bishop

[3]. https://github.com/17601a0570/majorproject

8.2 Websites

- www.opencv.org
- www.dlib.net
- www.wikipedia.com
- www.geeksforgeeks.com
- www.machinelearningmastery.com

ISSN (Online): 2581-5792

IJRESM

IJRESM

International Journal of Research in Engineering, Science and Management

www.ijresm.com support@ijresm.com

SJIF Impact Factor: 4.308

Certificate

It is here by certified that the manuscript entitled

Video Based Mask Detection

by

Ryan D. Julius

has been published in

Volume 4, Issue 6, June 2021

in

International Journal of Research in Engineering, Science and Management

Editor-in-Chief (IJRESM)

All the best for your future endeavors

ISSN (Online): 2581-5792

IJRESM

IJRESM

International Journal of Research in Engineering, Science and Management
www.ijresm.com support@ijresm.com
SJIF Impact Factor: 4.308

Certificate

It is here by certified that the manuscript entitled

Video Based Mask Detection

by

Vivek Mittal

has been published in

Volume 4, Issue 6, June 2021

in

International Journal of Research in Engineering, Science and Management

Editor-in-Chief (IJRESM)

All the best for your future endeavors

ISSN (Online): 2581-5792

IJRESM

IJRESM

International Journal of Research in Engineering, Science and Management

www.ijresm.com support@ijresm.com

SJIF Impact Factor: 4.308

Certificate

It is here by certified that the manuscript entitled

Video Based Mask Detection

by

R. Rugved

has been published in

Volume 4, Issue 6, June 2021

in

International Journal of Research in Engineering, Science and Management

Editor-in-Chief (IJRESM)

All the best for your future endeavors

International Journal of Research in Engineering, Science and Management
Volume 4, Issue 6, June 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

Video Based Mask Detection

Ryan D. Julius[1], Vivek Mittal[2], R. Rugved[3*]
[1,2,3]*Student, Department of Computer Science and Engineering, CMR Technical Campus, Hyderabad, India*

***Abstract*: COVID-19 pandemic has rapidly affected our day-to-day life disrupting many different sectors of the industry. Wearing a protective face mask has become a new normal. An individual that doesn't wear a mask poses a threat to the safety of the group of people in the area. As one year has passed living with the virus it has also been shown that many individuals don't wear their mask if they are not told to do so. At this stage it is very important to ensure that each and every one of us wears a mask. In a vicinity that consists of a group of people such as schools, colleges, movie theatres etc. we can use image processing to recognize how many individuals are consistently wearing their masks. This proposed system also ensures that if an individual is to ever remove his/her mask, the relevant authority is informed about it. If deployed correctly, our product could potentially be used to help ensure our safety and the safety of others around us. This allows the college/org to be more efficient and save time compared to the standard methods available.**

***Keywords*: Deep Learning, facial land marking, MobileNet, NumPy, TensorFlow object detection module.**

1. Introduction

Using facial land marking to solve issues of mask detection that are simple to accomplish for the consumer. This project aims to ensure organizations can detect if the individuals in the room are wearing a mask, we want to ensure that this takes place in the safest and efficient way possible which is also relatively contactless.

As of now there has been no industrial implementation of this project as it is still in its infancy stage. This product's future also heavily revolves around how the pandemic situation will turn out in the coming years. However, there are many facial recognition software that are in the market that leverage facial landmarks to extract the region of interest which is foundational for both facial recognition and our mask detection software. We hope to take advantage of this to come out with a product that benefits the public.

In this project, we propose an end to end system which solves the issue of individuals not wearing masks using image processing and facial landmarking.

This proposed system can be split into the following phases: the first phase is face mask training where the model is trained using datasets. The second phase is face mask detection where the faces will be detected and also mask will be detected using face mask classifier.

2. Methodology

A. System Architecture

In this project, a real time face mask detection model was developed with the help of deep learning, object detection. In the first step, a frame/image of the user is taken by the camera and the single or multiple faces will be detected in any scenarios using HOG algorithm. The Region of interest for detected faces is considered (in this case, bottom part of the face) and are grouped together. Then a face mask classifier is applied which is CNN based on the faces which determines if the mask is present or not for all the faces in the image and a particular score is given based on how the person is wearing the mask .100 is the maximum score means mask is present correctly, less than 90 means that the person is wearing the mask in a wrong way. Then the model alerts the authority or gives necessary instructions to non-mask users or masked users with less score. As we are using the object classifier where the object is a mask, all the people will be tracked using CCTV to provide a safe environment.

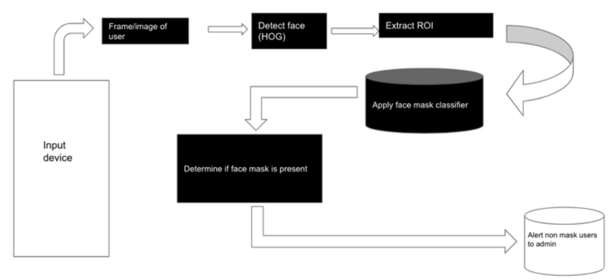

Fig. 1. System architecture

B. Training and Testing

First we need to take the two datasets, one with mask and other without mask. The more images we have the better will be the model. In the pre-processing part we are converting all our images into a collection of arrays and with the help of those arrays we will create the deep learning model. We need to create 2 empty lists named data and labels. We will append all the images inside the data list. Labels list contain the labels of all the images indicating if they are masked or not and we will loop through them. Once we get the path of the images, the images will be joined with the corresponding paths. Then a function (load_image) is used which is a function of KERAS that is used to size the images. Those images are converted to array with the help of (img to_array) it is also a function of the KERAS

*Corresponding author: rugved170549@gmail.com

module. Once the image is converted to an array we use the process (preprocess_input) which is a part of mobilenet (these are classes of low latency low power model that can be used for classification and other common task convolutional neural networks). Then append the image (which is an array) to the data list same with their corresponding labels inside the labels list, now we have all the image and labels array inside the list. All the data is in numerical values but labels are still Alphabetical values, so that needs to be converted into a binary array. To do that you need to use the LabelBinarizer method which is from the sklearn module. After completion of conversion from alphabets to 1's and 0's.

Those arrays are converted to NumPy arrays. We are training the data with the help of convolution neural network but with a little twist, our idea here is, we will ignore the convolution which we usually do and use mobile net instead of that because mobile net is very light as compared to the convolutional neural network and they use less parameters. After the images are processed as an array, we will use them as input to an array, it is sent to the mobilenet. We will perform the maxpooling (it is used for reducing the spatial size of the convoluted features.). max pooling uses the maximum value from each cluster of neurons at the prior layer. Lastly, we will get the fully connected layer of the result and get the output. We need to give less learning rate so the loss in data will be less. Which will result in all over better accuracy. We are also using Image Data Generator which will help in generating more images with a single image by applying various properties such as rotating the image, zooming on it and changing its height and width.

To run the model, go to command prompt and locate where the file is and run it using python filename.py command as shown in Fig. 2.

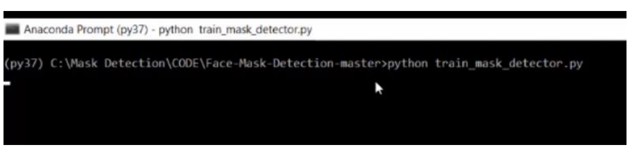

Fig. 2. Running the model

If the training process is successful, the output will be the same as shown in Fig. 3.

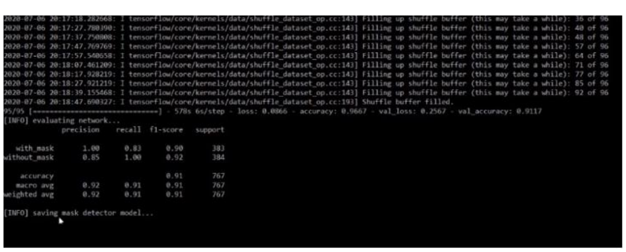

Fig. 3. After running the model

Fig. 4. represents the training and loss graph, which indicates how accurate and how much data we have lost.

To open the camera on your system, use the command python filename.py as shown in Fig. 5.

Once the video starts streaming, it will produce these readings (as shown in Fig. 6) that are updated every second.

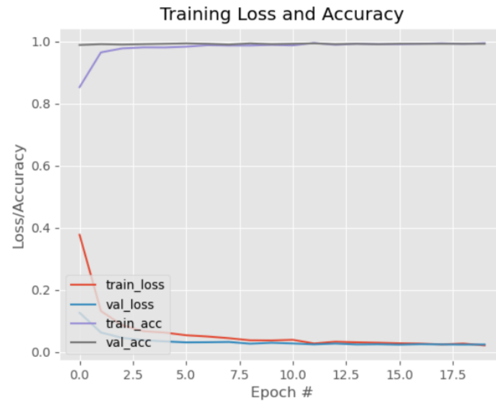

Fig. 4. Training and Loss graph

Fig. 5. Command to open the camera

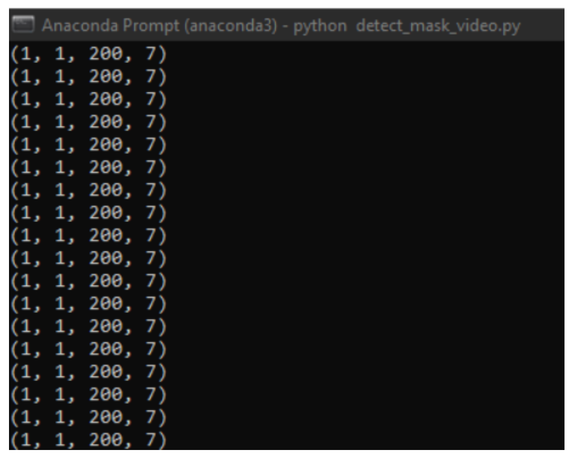

Fig. 6. Readings produced on video streaming

3. Results and Discussions

A real time video based mask detection model has been introduced.

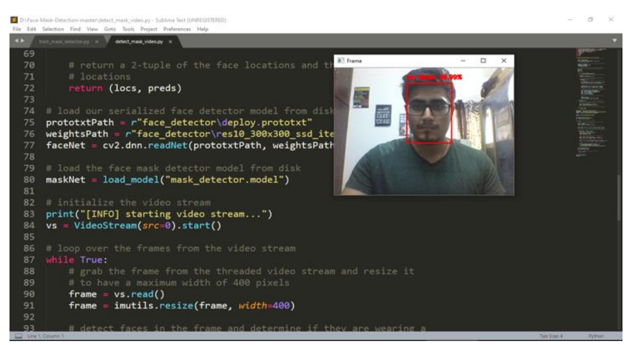

Fig. 7. No mask detection

Fig. 8. shows the accuracy of mask_detection if it's not worn properly.

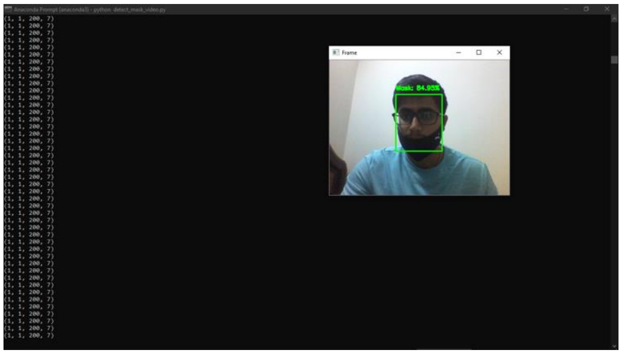

Fig. 8. Mask detection for mask not worn properly

Fig. 9. shows the accuracy of mask_detection which is 100%.

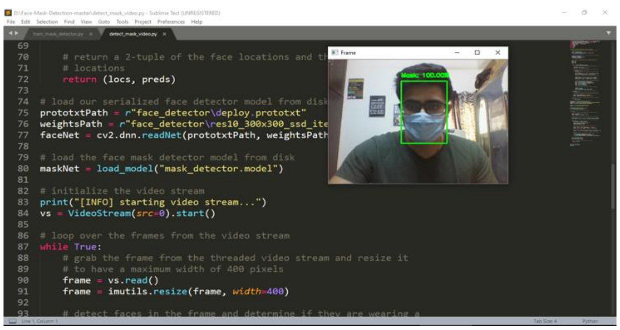

Fig. 9. Mask detection

4. Conclusion

The paper titled as "Video Based Mask detection" is a software application. As COVID-19 is a rising concern, it is necessary to ensure that everyone wears a mask, our system ensures it is done in the most hands-free way as possible and our aim as a team to ensure we work towards that problem while also tackling other objectives such as efficiency, cost, portability and implementation.

With enough development we believe this is a product that can be used in different industries all over the world. This software is developed with scalability in mind. Additional modules can be easily added when necessary. The software is developed with a modular approach. All modules in the system have been tested with valid data and invalid data and everything works successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

References

[1] Abadi Martín, Barham Paul; Chen Jianmin; Chen Zhifeng; Davis Andy; Dean Jeffrey; Devin Matthieu; Ghemawat Sanjay; Irving Geoffrey; Isard, Michael; Kudlur Manjunath; Levenberg Josh; Monga Rajat; Moore, Sherry; Murray, Derek G.; Steiner, Benoit; Tucker, Paul; Vasudevan, Vijay; Warden, Pete; Wicke, Martin; Yu, Yuan; Zheng, Xiaoqiang, "TensorFlow: A System for Large-Scale Machine Learning," 2016.

[2] Tensor flow "API Documentation."

[3] Chen, S. K; Chang, Y. H. (2014). 2014 International Conference on Artificial Intelligence and Software Engineering (AISE2014). DEStech Publications, Inc. p. 21. ISBN 9781605951508.

[4] Yan Holly, "Why face mask guidance has changed so much -- and how wearing masks can protect the economy," June 2020.

[5] "Python 3.9.2 documentation". docs.python.org. March 2021.

[6] Sebastian Raschka "Chapter 1: Introduction to Machine Learning and Deep Learning," 2020.