# VALIDATING THE USAGE AND DETECTION OF MASK USING YOLO V3

**A MINOR PROJECT REPORT**

***Submitted by***


**K. PUGAZHVENTHAN RA1811003020391**
**M. SRINATH RA1811003020445**
**FAHAD AHMED RA1811003020441**


**Under the guidance of**


**DR. SHINY DUELA. J**

**(Associate Professor, Department of Computer Science and Engineering)**


***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

**of**

FACULTY OF ENGINEERING AND TECHNOLOGY


**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**
**RAMAPURAM CAMPUS, CHENNAI -600089**

## SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

**(Deemed to be University U/S 3 of UGC Act, 1956)**

## BONAFIDE CERTIFICATE

Certified that this project report titled **"VALIDATING THE USAGE AND DETECTION OF MASK USING YOLO V3"** is the bonafide work of **K. PUGAZHVENTHAN [REG NO: RA1811003020391], M. SRINATH [REG NO: RA1811003020445], FAHAD AHMED [REG NO: RA1811003020441]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an occasion on this or any other candidate.

SIGNATURE

**DR. SHINY DUELA. J**

**Associate Professor,**

Computer Science and Engineering,
SRM Institute of Science and Technology,

Ramapuram Campus, Chennai.

SIGNATURE

**Dr. K. RAJA, M.E., Ph.D.,**

**Professor and Head**

Computer Science and Engineering,
SRM Institute of Science and Technology,

Ramapuram Campus, Chennai.

Submitted for the project viva-voce held on 24-11-2021 at SRM Institute of Science and Technology, Ramapuram Campus, Chennai -600089.

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

## SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

## RAMAPURAM, CHENNAI - 89

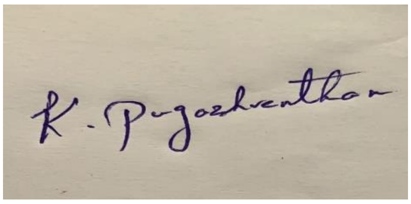
# DECLARATION

We hereby declare that the entire work contained in this project report titled "**VALIDATING THE USAGE AND DETECTION OF MASK USING MACHINE LEARNING**" has been carried out by **K. PUGAZHVENTHAN** [REG NO: RA1811003020391], **M. SRINATH** [REG NO: RA1811003020445], **FAHAD AHMED** [REG NO: RA1811003020441] at SRM Institute of Science and Technology, Ramapuram Campus, Chennai- 600089, under the guidance of **DR. SHINY DUELA. J, Associate Professor**, Department of Computer Science and Engineering.

**Place: Chennai**

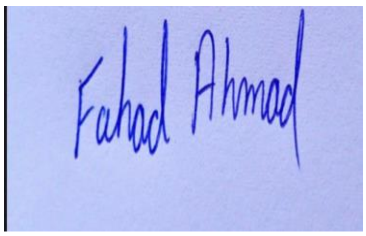
**Date: 24 – 11 – 2021**

**K. PUGAZHVENTHAN**



**M.SRINATH**

**FAHAD AHMED**



# ABSTRACT

Corona virus disease have made world seriously effected in recent times. After the break out of this Covid19 pandemic, there raised in the need of primary protection methods, face mask usage is the fundamental one. To avoid the spread of virus, many regulatory rules like checking temperature, maintaining social distancing, insisting the people to wear masks properly are being implemented around the world. While using public services, service providers need their clients to use facemasks in fulfillment of emerging government rules. Though the existing systems have been very efficient in this domain, usually a prediction says that the modern technology can detect the face masks very clearly and efficiently. To effectively prevent the spread of the corona virus, an object detection method based on YOLO V3is proposed. This proposed framework focuses on real-time face masks detection in the densely populated areas like communities, organizations where huge number of people gather together. In this project we propose a system on basis of two techniques, SSD-single shot detection and deploying our application in public surveillance cameras. Finally, we implemented a complete work for recognizing the face masks and to check whether they are worn correctly or not. To understand this we need a large data set and we have gathered that from kaggle data sets.

# TABLE OF CONTENTS

LIST OF FIGURES

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| SSD | SINGLE SHOT DETECTION |
| MB V2 | MOBILENET VERSION 2 |
| CCTV | CLOSED CIRCUIT TELEVISION |
| CNN | CONVOLUTIONAL NUERAL NETWORK |
| API | APPLICATION PROGRAM INTERFACE |
| SMFRD | SIMULATED MASKED FACE RECOGNITION DATASET |

# Chapter 1

# INTRODUCTION

## 1.1 Introduction

The purpose of this project is to develop a face mask detection softwarethat can be used in a wide variety of places such as, corporate offices, shopping malls, theatres and other places with dense crowds. To develop this application we’ve used, Darknet53 a period object detection model. we tend to use the in public available face mask detection dataset from Kaggle. We trained on an initial network resolution of 224*224 pixels. Higher resolution will result in higher detection. To detect whether a person is wearing face mask or not, “Validating the usage and detection of mask using Machine learning” is used. Medical Experts have been advising to wear mask for the people who are working in large groups and people who have mild symptoms and people who are taking care of the one having symptoms. Depending on the necessities the system is customized to YOLO V3 technology. It can also analyze what quantity share of individuals is carrying masks and not. With this model the outcome is correct and economical.

### 1.1.1 Problem Statement

Firstly, we need to understand the problem in an efficient way. We need to build a model that can detect the face masks in pictures taken from the CCTC cameras. The pictures taken in the cameras are low resoluted and blur and can't detect the face efficiently. But the model we have built can detect the face masks in a low resoluted picture clearly by using YOLO V3. And can detect whether the person is wearing mask or not by using Image classifiers provided.

## 1.2 Aim of the project

The aim of the project is to develop an software to avoid the spread of (Covid-19) and detect the person wearing mask or not by capturing images through surveillance cameras

## 1.3 Project Domain

The domain of the project is Machine Learning. The progress of machine learning techniques have been challenging when it comes computer vision and Image processing. Machine learning uses various algorithms based on the requirement of the project.

## 1.4 Scope of the Project

This kind of face mask detections can be used in various real time application like Airports, Schools, Organizations, etc., to ensure security, spreading of corona virus and create awareness among public

## 1.5 Methodology

The methodology carried out in this model is, firstly a frame that can access camera will take the input as a image and image classifiers are loaded and then that image classifiers will zoom into certain of pixel's to detect the person with or without mask. The Primary method we implemented in our system is Darknet53 methodology. This methodology is a method based on YOLO V3 (You Look Only Once). The face mask recognition in this system is developed with machine learning Algorithms. This system involves a several steps to generate the face mask recognition model. These are the steps like Data Collecting, Pre-Processing, Split the Data, building the model, Testing the Model, Implementing the model. These are the steps to generate the model. The data which has been collected labeled into two group's. secondary thing is pre-processing phase. Before training and testing Data four steps are there in pre-processing.in this steps re- sizing plays a main key role in preprocessing state. It will make im- age into 224*224 pixels. Tertiary thing is Training data-75 Test Data- 25, Constructing the training image generator for augmentation the base model parameters, Compiling the Model, Training the model, saving the model at last implementing the our model .

# Chapter 2

# LITERATURE REVIEW

| S.NO | TITLE | AUTHOR | METHODOLOGY | TECHNICAL GAP |
|---|---|---|---|---|
| 1. | **Face Mask Detector (2020)** | Arjya Das, Mohammad Wasif Ansari, Rohini Basak | **OpenCV, Keras/TensorFlow, and Deep Learning using CNN** | Poor image quality<br>Small image size make facial recognition more difficult.<br>Different face angles can throw off facial recognitions reliability.<br>IT has accuracy of 85-90%<br>It doesn't has show bounding box |

| | | | | |
|---|---|---|---|---|
| **2** | **Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for COVID-19. (2020)** | **Shi, F.; Wang, J.; Shi, J.; Wu, Z.; Wang, Q.; Tang, Z.; He, K.; Shi, Y.; Shen, D** | **IMAGE DATA ACQUISITION** | **Several studies repor little or no power of generalization, when evaluating the traine models in their own sets. Even the models that were trained usi pre-processing techniques, which tri to eliminate the biase belonging to the data sets, showed limited results. Therefore, most of the results achieved so far** |
| 3 | **Face Detection and Recognition System (2020)** | Gurlove Singh<br>Amit Kumar Goel | DIGITAL IMAGE PROCESSING | Massive data storage burden. The ML technology used in fac detection requires powerful data storage that may not be available to all users. Detection is vulnerabl ...<br>A potential breach of privacy. |

| | | | | |
|---|---|---|---|---|
| **4.** | **Human Recognition with Face Mask(2021)** | **Dr. Vandana S. Bhat , Arpita Durga Shambavi , Komal Mainalli, K M Manushree, Shraddha V Lakamapur** | **In this journal They have tried haar cascade algorithm** | **Ababoost is the algorithm used in this paper but it can be less susceptible to overcome of all the remaining algorithms.The main disadvantage of Adaboost  is more sensitive to noisy data.The challenge occured in this task is more complexive as the model should target on the face mask but not on the human faces.It also acquires a high performance and more economic.** |
| 5. | Masked face Recognition Dataset and Application | Zhongyuan Wang, Guangcheng Wang, Baojin Huang,Zhangyang Xiong | Simulated masked face recognition data set(SMFRD),Real world masked face recognition data set(RMFRD) are the technical models implemented in this journal | The main risky factor here is if the human faces turned up side or not faced to the camera then the detection is critical as the model is not trained to detect the faces from a certain angle. |

| | | | | |
|---|---|---|---|---|
| **6.** | **Detecting Faces in Images(2019)** | **M.H.Yang,D.J.Kriegman, and N.Ahuja** | **This model also caused advancement in the computer vision** | **The factors that make the work complicate should be evaluated and a lot of training data sets needs to be taken as sample and perform testing.After considering the result of testing phase,one can conclude that the method can be efficient or not.** |
| 7. | A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2 (2021) | Preethi Nagrath,Rachana jain,Agam madan,Rohan Arora | In this journal deployment of the model was done by using two categories called image data set and training data set. The proceedure was carried out by using OpenCV deep nueral networks.Image classification was implemented by using MobileNetV2 image classifiers | Real world projects can be so challenging to the technology to come with the desired output and with no errors |

| | | | | |
|---|---|---|---|---|
| **8.** | A deep learning based unmanned approach for real-time monitoring of people wearing medical masks | Roy, B., Nandy, S., Ghosh, D., Dutta, D., Biswas | In this journal they used Faster R-CNN and evaluated them on Moxa3K benchmark dataset | **The problem with these methods is that Mask-RCNN is computationally heavy and requires adequate computing power to run the model in real time. especially in places where such devices are not affordable.** |
| 9. | Jiang, M., Fan, X., Yan, H. (2020). | A face mask detector | In this journal they used Retina FaceMask which is a high-accuracy and efficient face mask detector | **lack of control over their personal information** Many of the training datasets uses for facial recognition are dominating by light-skinned individuals. |

| | | | | |
|---|---|---|---|---|
| **10.** | **Fast In-browser Face Mask Detection with Serverless Edge Computing for COVID-19** | Zekun Wang, Pengwei Wang, Peter C. Louis, Lee E. Wheless, Yuankai Huo | In this journal they used yolo version 1 | it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev where as yolo v3 is on par with Focal Loss but about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required! |

# Chapter 3

# PROJECT DESCRIPTION

## 3.1 Existing System

The existing system runs based on the mechanism of Voila Jones Algorithm, which is mainly ineffective when the image is turned up side. Mask is generally not the key point in means of detection techniquesso we have included detection of face mask in our project. The existingsystem is also incompatible to lightning conditions. The existing system uses a computer generated images which are used to detect face mask but not instantly. Existing system may have data set errors whichcauses defects in the detection of face mask.

## 3.2 Proposed System

The proposed system contains the main contendor is it supports vector machines. It uses an algorithm called Darknet53 algorithm.In this system YOLO V3 technology is used. whereas the detection rate and filtering the faces result will be very high and accurate. SSD technology single shot detection where within single shot entire image capturing process will be finished it results in efficient way of capturing the images. It also comprises of many complex processes like face, feature extraction etc. we can perform all these complex processes with less efficient webcams. It Easily detects the log of objects. Processing rate is higher. When there is a high processing rate theresult will be efficient.

### 3.2.1 Advantages

- Easily detects the log of objects.
- Processing rate is higher.
- single shot Detection
- object detection model.

## 3.3 Feasibility Study

A Feasibility study is carried out to check the viability of the project and to analyze the strengths and weaknesses of the proposed system. The application of usage of mask in crowd areas must be evaluated. The feasibility study is carried out in three forms

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

### 3.3.1 Economic Feasibility

The proposed system does not require any highcost equipment. This project can be developed within the available software.

### 3.3.2 Technical Feasibility

The proposed system is completely a Machine learning model. The main tools used in this project are Anaconda prompt, Visual studio, Kaggledata sets, Jupyter Notebook And the language used to execute the process in Python. The above mentioned tools are available for free and technical skills required to use this tools are practicable. From this we can conclude that the project is technically feasible.

### 3.3.3 Social Feasibility

Social feasibility is a determination of whether project will be acceptable or not. our project is Eco-friendly for society and there is no social issues. our project must not threatened by the system instead must accept it as a necessity. since our project is applicable for every individuals in the society to take care about the society and environment. The level of the acceptance of System is very high and it depends on the methods deployed in the system. our system is highly familiar with the society.

## 3.4 System Specification

### 3.4.1 Hardware Specification

- Processor - Intel i5-8250 CPU @1.60GHz 1.80GHz
- 512 GB SSD
- NVIDIA GEFORCE RTX
- CPU QUAD CORES

### 3.4.2 Software Specification

- ANACONDA
- ANACONDA PROMPT
- PYTHON
- VISUAL STUDIO
- GIT

# Chapter 4

# MODULE DESCRIPTION

## 4.1 General Architecture

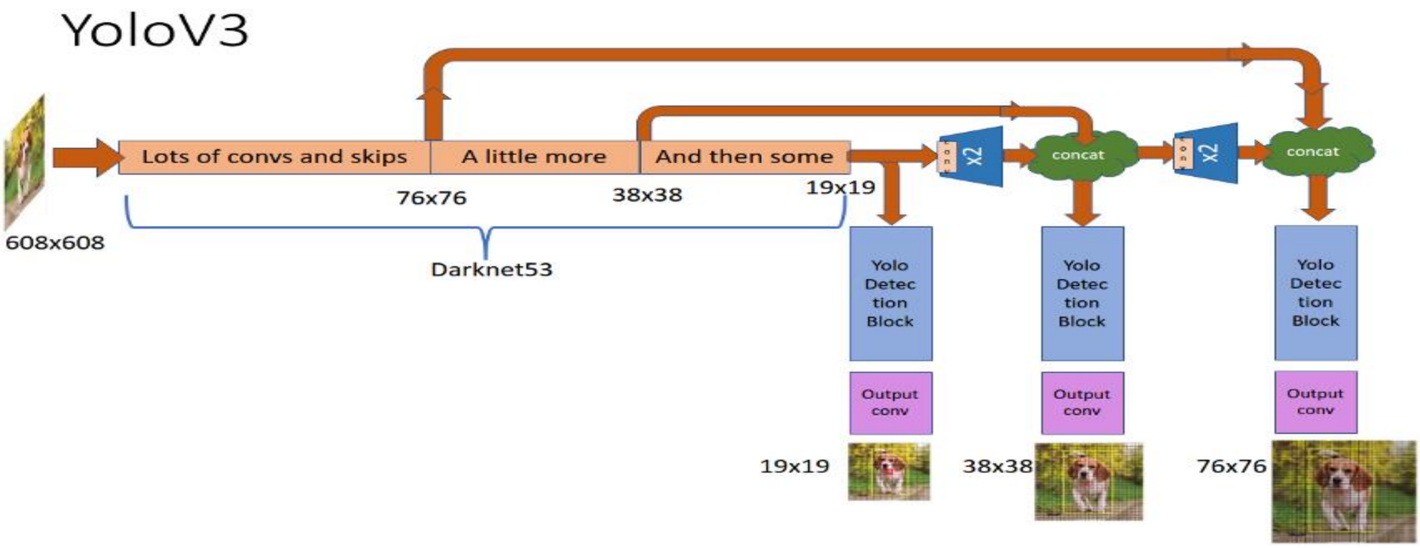



Figure 4.1: **Architecture Diagram**

Figure 4.1 represents the architecture diagram of the project. Firstly, theinput is taken from the camera and algorithm is trained using data sets, which is used as training data. The input taken from the camera will extract the images in the capture. Then the images are forwarded to SSD model and detection model is loaded using MobileNetV2.Then image processing techniques are applied to the captured image and then now mask classifiers are used to determine whether the person is wearing mask or not by outlining the face area.

## 4.2 Design Phase

### 4.2.1 Data Flow Diagram

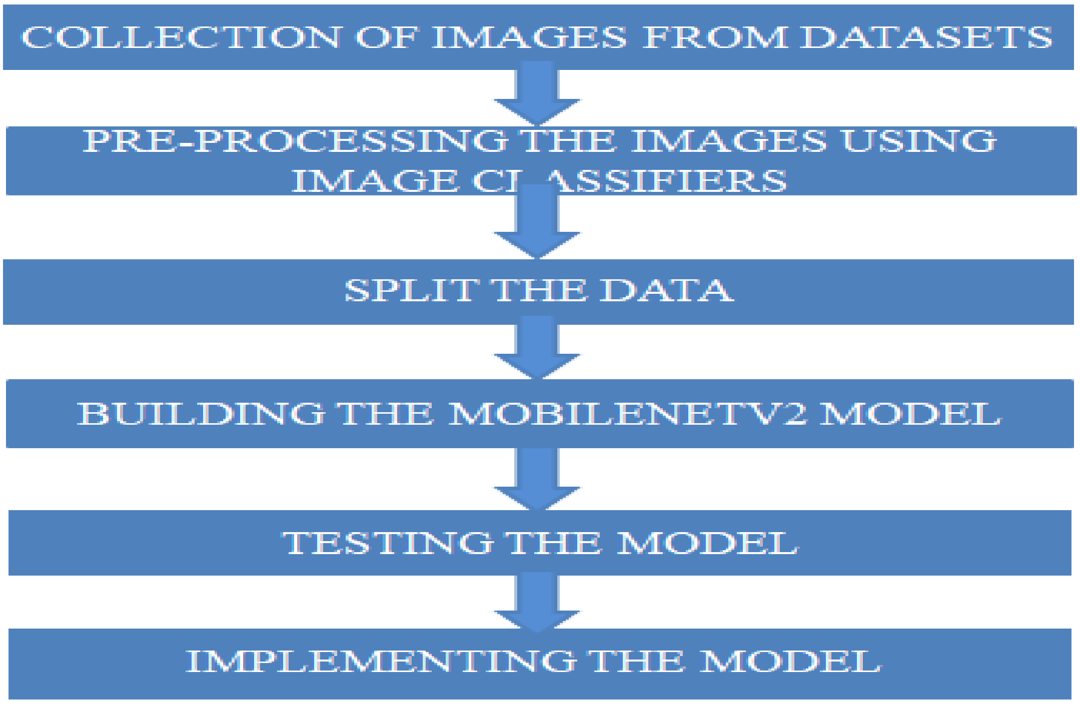



Figure 4.2: **Data Flow Diagram**

Figure 4.2 represents the flow diagram of our project. The data sets of persons who are wearing mask and not are collected using Kaggle data sets. The Preprocessing step includes resizing the image as according to the system requirement, image is pushed into the array, processing using Darknet53 model, and hot encoding is performed to labels. The data splitted into 75 and 25 percent to meet the requirements of the model.75 percent is allocated to training whereas 25 percent is allocated to testing the data. And the next step is building the model, it is carried out by using MobilenetV2.Testing is done to check the viability of the project. Finally, the model is implementedby carrying out all this processes.

### 4.2.2 UML Diagram

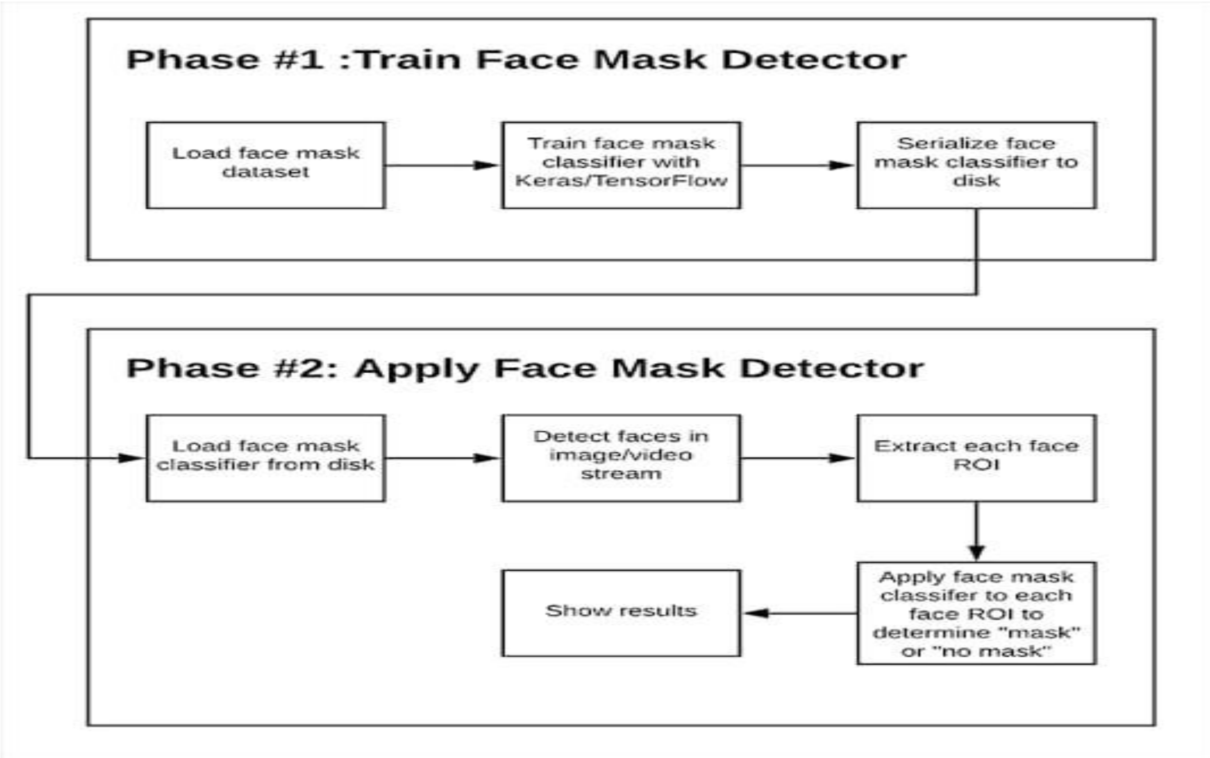



Figure 4.3: **UML Diagram**

Figure 4.3 repesent the UML diagram of our model. In this first video/image is captured from CCTV cameras. Then the captured video/image is divided into frames or images. The images or frames captured in CCTV cameras transferred to the processing block in this it calculates distance of person from the camera by using NumPy objects and it detects the face. The model detects if there is a mask on the face of the person and categorizes the data as masked and unmasked faces.

### 4.2.3 Use Case Diagram

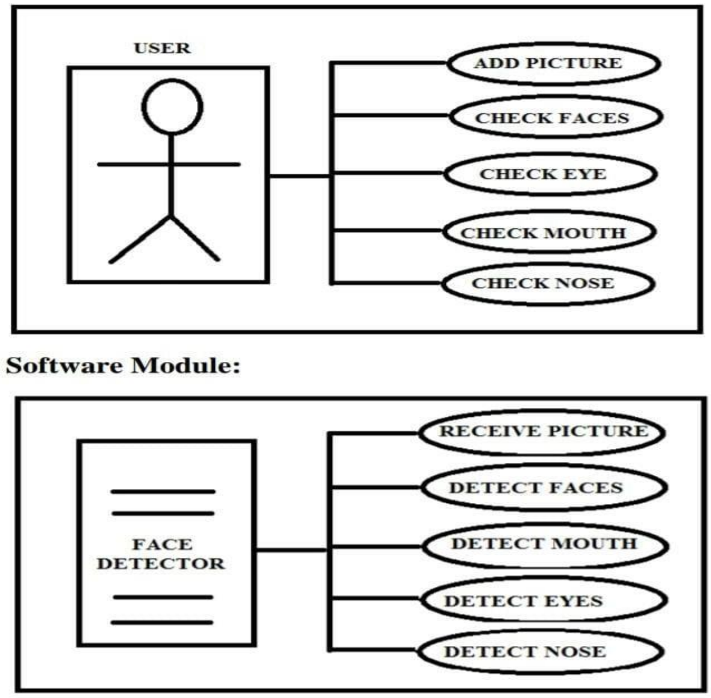



Figure 4.4: **Use Case Diagram**

Figure 4.4 represents the Use Case diagram of our model. The captured images are collected and carried out to the image processing techniques and to fed into the model. The model classifies the images into ”Wearing mask” and ”Not wearing mask” by using image classifiers like Haar Cascade classifiers. Then the according action is responded as the person is wearing mask or not.

### 4.2.4 Sequence Diagram

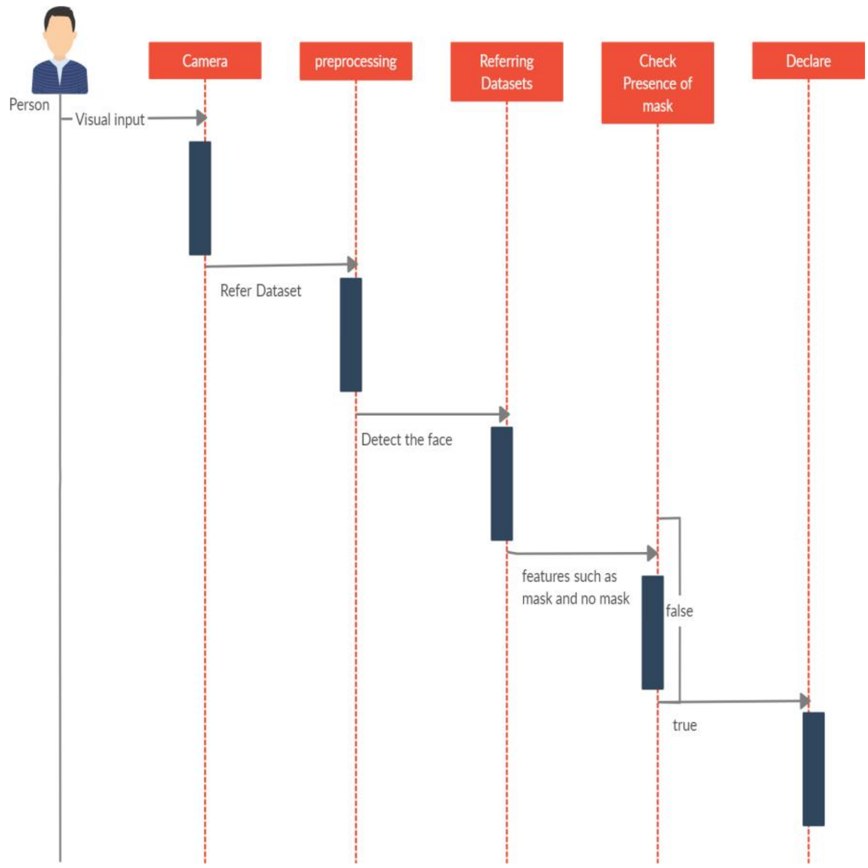



Figure 4.5: **Sequence Diagram**

Figure 4.5 represents sequence diagram, in this will camera captures the visual input of people and the input is transferred for processing. In processing it can detect the face mask by referring the data sets sample provided. By referring data sets it detects the features such as people wearing mask or not. By checking the presence of mask if trueit can declare person wearing mask if false it can declare person as notwearing mask

## 4.3 Module Description

Our entire project is divided into two modules.

### 4.3.1 MODULE1: DATA COLLECTION AND TRAINING DATA

Data Collection and training using Machine Learning Algorithms

### 4.3.2 Step:1 Data collecting

- The development of the Face Mask Recognition model begins with collecting the data
- The data set train data on people who use masks and who do not.


Figure 4.6: **Test Image**

### 4.3.3 Step:2 Processing of data

- The Pre-processing section could be a section before the coaching and testing of the info. There are four steps within the pre-processing that area unit resizing image size, changing the image to the array, pre-processing input victimization MobileNetV2, and the last one is performing hot encoding on labels.

- The resizing image could be a vital Pre-processing step in pc vision because of the effectiveness of training model. The next step is to method all the images within the data set into an array. The im- age is born-again into the array for line them by the loop perform. After that, the image are going to be accustomed Pre-process in- put victimization MobileNetV2. and therefore the last step during this section is playacting hot secret writing on labels as a result of several machine learning algorithms cannot operate knowledge labeling directly. They need all input variables and output variables to be numeric, as well as this algorithm. The labeled knowledge are going to be remodeled into a numerical label, that the algorithm will perceive it.

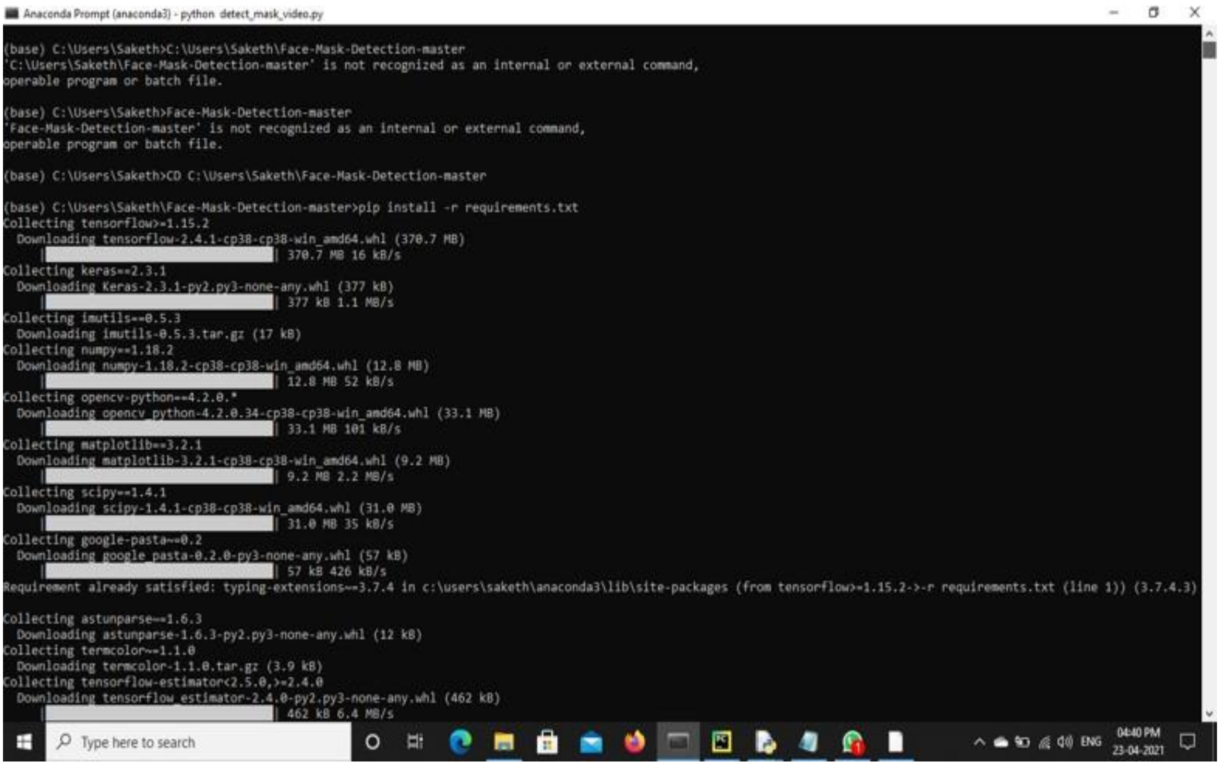



Figure 4.7: **Pre processing of Data**

### 4.3.4 Step:3 Split the Data

- After the pre-processing part, the information is split into 2 batches, that are training data specifically seventy five percent, and the rest is testing knowledge. every batch is containing each of with-mask and without-mask pictures.

### 4.3.5 DATASETS SAMPLE

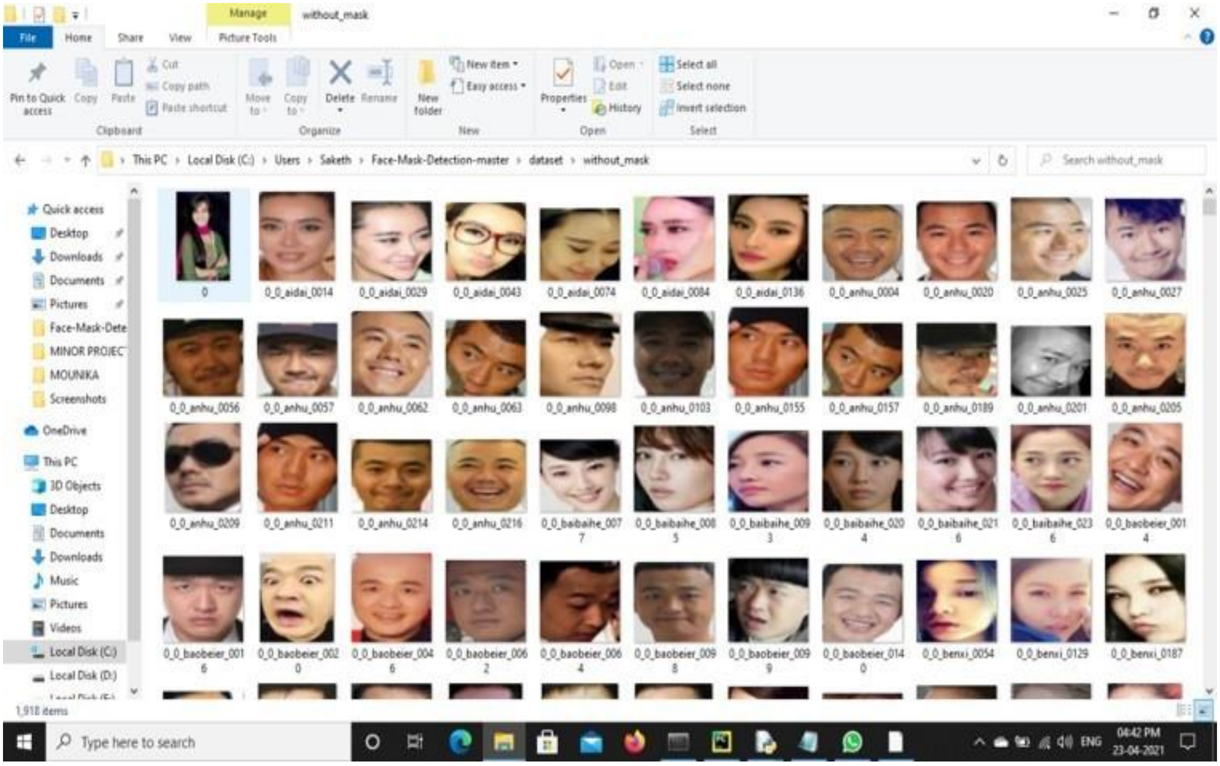



Figure 4.8: **Without mask**

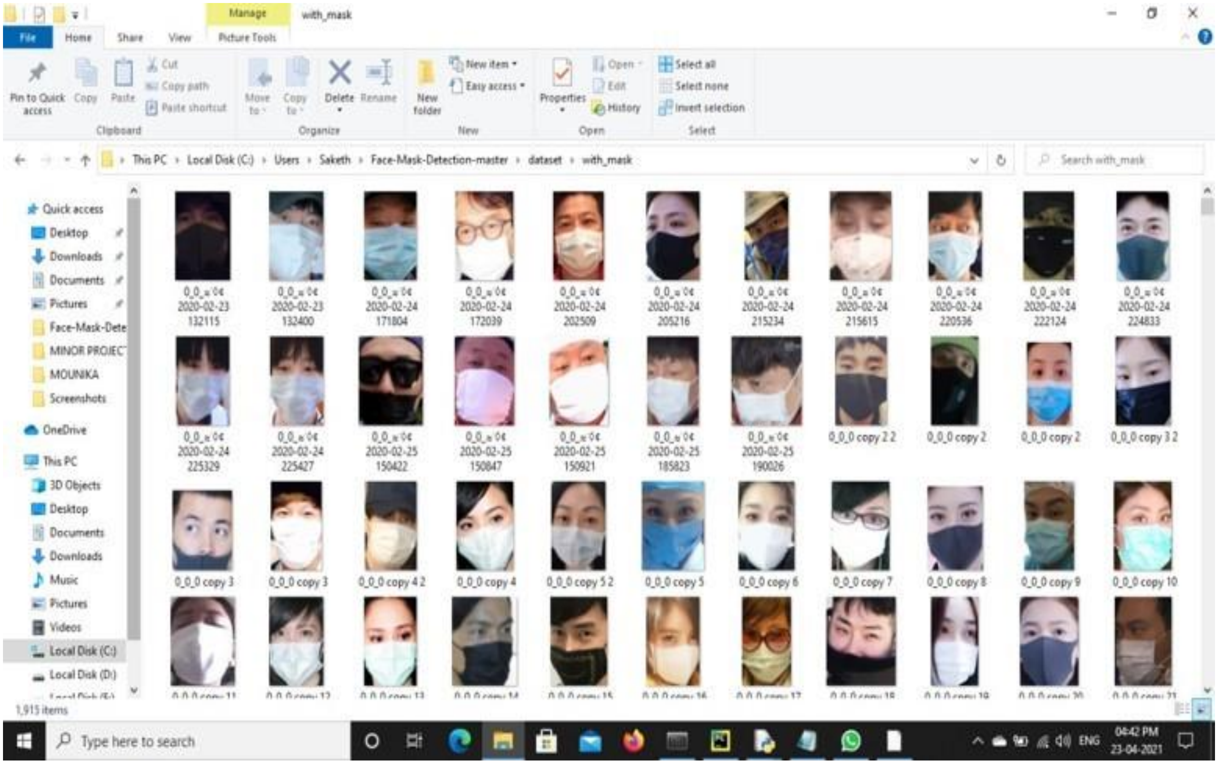


Figure 4.9: **With Mask**

### 4.3.6 Step:4 Building the Model

- The next part is building the model. There are six steps in building the model that are constructing the training image generator for augmentation, the base model with MobileNetV2, adding model parameters, collecting the model, coaching the model, and therefore the last is saving the model for the long run prediction method.

### 4.3.7 Step:5 Testing the Model

- To make sure the model can predict well, there are steps in testing model. The first step is making predictions on the testing set.

### 4.3.8 Step:6 Implementing the model.

- The model enforced within the video. The video scan from frame to border, then the face detection formula works. If a face is detected, it takings to future method. From detected frames containing faces, reprocessing are going to be disbursed together withresizing the image size, changing to the array, preprocessing in- put victimization MobileNetV2. future step is predicting input file from the saved model. Predict the input image that has been processed employing a antecedently designed model. Besides, the video frame also will be labelled that the person is sporting a maskor not alongside the predictive proportion.

# Chapter 5

# IMPLEMENTATION AND TESTING

## 5.1 Input and Output

### 5.1.1 View of a person without mask

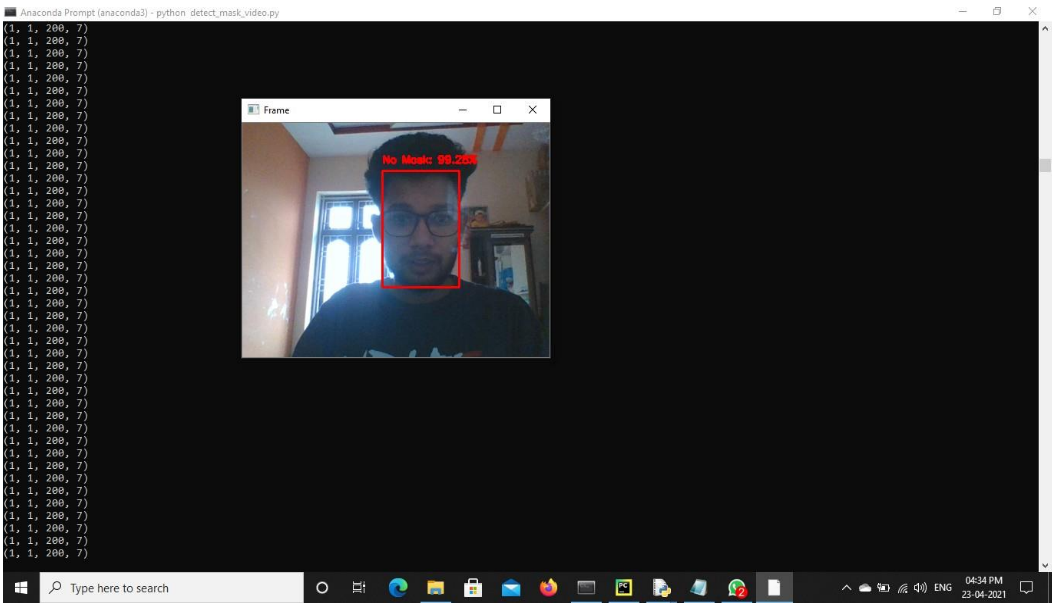



Figure 5.1: **Person without mask**

### 5.1.2 View of a person with mask

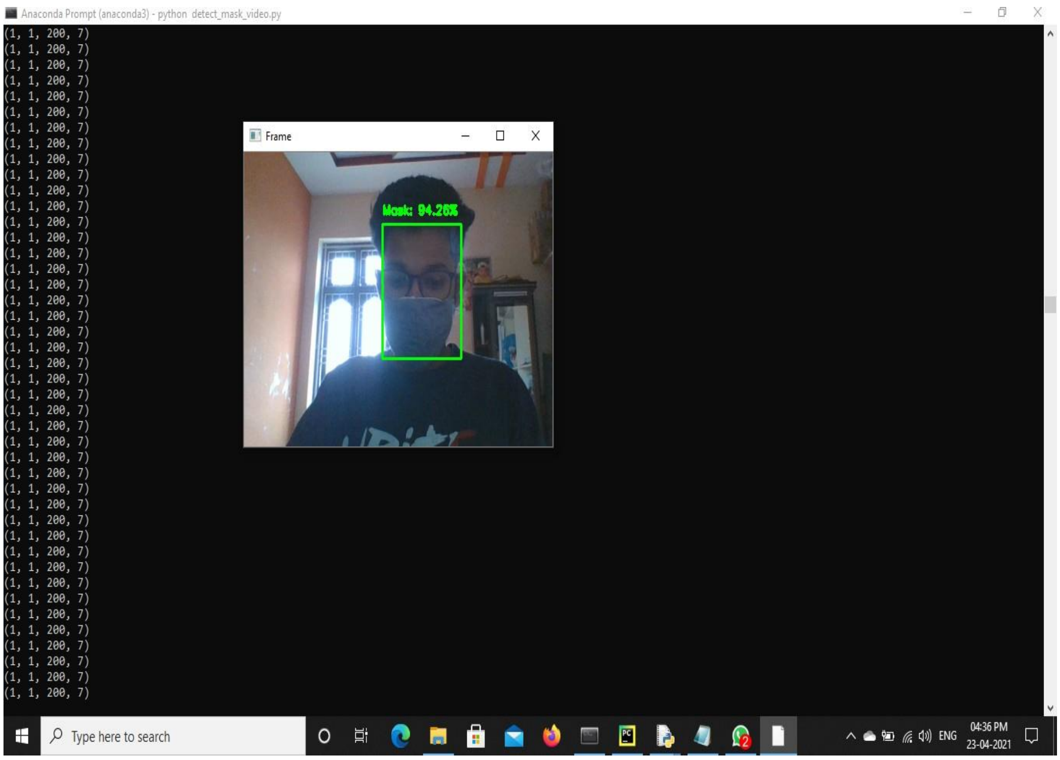



Figure 5.2: **Person with mask**

## 5.2 Testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not.

### 5.2.1 Types of Testing

### 5.2.2 Unit testing

Unit testing is a benificiable software testing method where the units of source code is tested to check the efficiency and correctness of the program.

**Input**

```
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)
```

**Test result**

- Data sets images are accessed.
- Images of size 224*224 pixels are considered
- The considered images are loaded into an array for preprocessing.

### 5.2.3 Integration testing

**Input**

```
frame = vs.read()
  frame = imutils.resize(frame, width=400)
  (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)
  for (box, pred) in zip(locs, preds):
  (startX, startY, endX, endY) = box
    (mask, withoutMask) = pred
    label = "Mask" if mask > withoutMask else "No Mask"
```

```
    color = (0, 255, 0) if label == "Mask" else (0, 0, 255)
```

**Test result**

- A frame of size 400 pixels is created to take the input and display the output.
- A label is defined as a square box.It attains green colour when a mask is present on the face and red colour when there is no mask.

### 5.2.4 Functional testing

**Input**

```
print("[INFO] training head ...")
H = model.fit(
    aug.flow(trainX, trainY, batch_size=BS),s
    tepsperepoch=len(trainX)// BS,
    validation_data=(testX, testY),
    validationsteps=len(testX)// BS,
    epochs=EPOCHS)
print("[INFO] evaluating network ...")
predIdxs = model.predict(testX, batch_size=BS)
print("[INFO]saving mask detector model ...")
model.save("mask_detector.model", save_format="h5")
```

**Test Result**

- All the images from data sets are loaded into the model and carried out for training.
- Training is done by considering each image and saving the characteristics of the image.
- After completing the training the MobilenetV2 saves the detector model and can compete with the input given.

### 5.2.5 Test Result

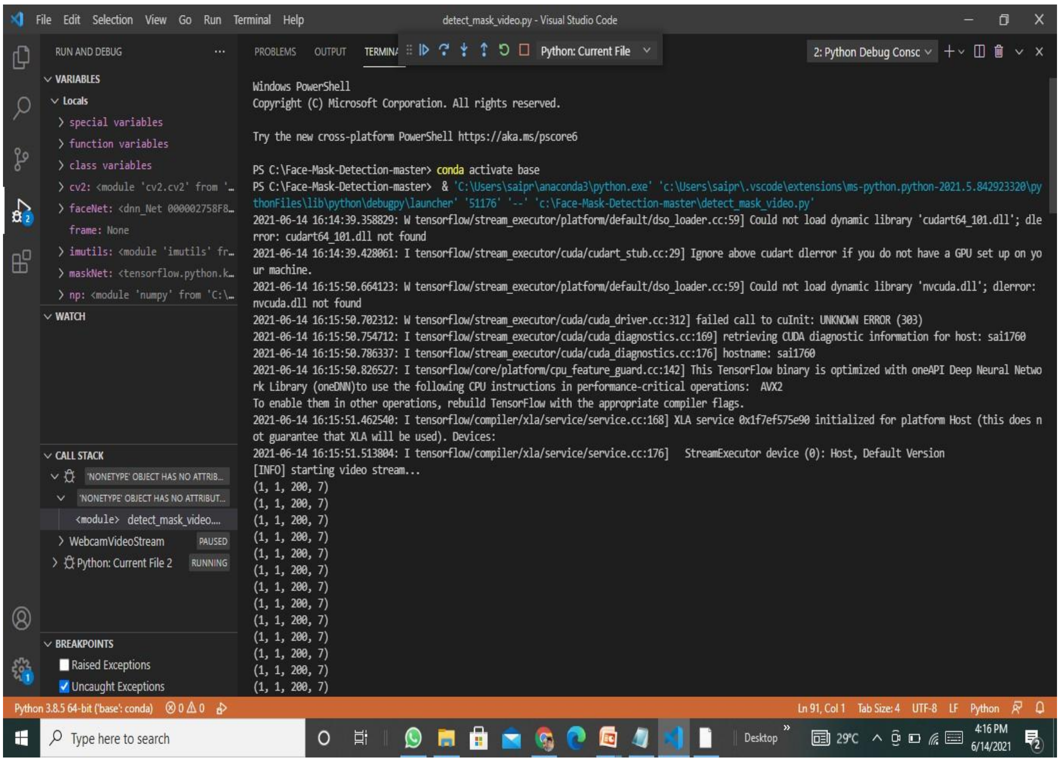



Figure 5.3: **Test Image**

## 5.3 Testing Strategy

- Unit testing: Unit testing verifies the bits of code to check the viability of the code.
- Integration testing: Integration testing is carried out to the efficiency of the modelwith functional requirements.
- Functional testing: The functional testing is done to verify the output with theprovided input against the functional requirements.

# Chapter 6

# RESULTS AND DISCUSSIONS

## 6.1 Efficiency of the Proposed System

The goal of these experiment is to create awareness among the citizens and to decrease humans activity and to stop spread of virus, provide insight into the performance of detection techniques with masked-face images and to evaluate performance of recognition models that in addition to the presence of face-masks also determine whether the masks are wearing properly or not within in single shot capturing using SSD-single shot detection and mobile net-v2 technology where we can see high performance and accuracy of results.

## 6.2 Comparison of Existing and Proposed System

The existing system has no key factor like mask detection and there is no specific software to deploy in surveillance cameras and detection with less efficient cameras are highly impossible and performance rate and accuracy is very low. only single direction capturing. No object detection Technique. In our proposed System the technology was used is highly accurate and processing is very high. when there is a high processing rate. we can say our system is efficient. latest technology SSD–single shot detection using object detection model. deployable in less efficient web cam technology SSD-single shot detection using object detection model. High performance rate and high accuracy of results. Processing speed is high. Easily detects the log of objects. Applicable for less efficient camera's like web cams.

## 6.3 Sample Code

```
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):
  (h, w) = frame.shape[:2]
  blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
    (104.0, 177.0, 123.0))
  faceNet.setInput(blob)
  detections = faceNet.forward()
  print(detections.shape)
  faces = []
  locs = []
  preds = []
  for i in range(0, detections.shape[2]):
    confidence = detections[0, 0, i, 2]
    if confidence > 0.5:
      box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
      (startX, startY, endX, endY) = box.astype("int")
      (startX, startY) = (max(0, startX), max(0, startY))
      (endX, endY) = (min(w - 1, endX), min(h - 1, endY))
      face = frame[startY:endY, startX:endX]
      face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
      face = cv2.resize(face, (224, 224))
      face = img_to_array(face)
```

```
            face = preprocess_input(face)
            faces.append(face)
            locs.append((startX, startY, endX, endY))
    if len(faces) > 0:
        faces = np.array(faces, dtype="float32")
        preds = maskNet.predict(faces, batch_size=32)
    return (locs, preds)
prototxtPath = r"face_detector\deploy.prototxt"
weightsPath = r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)
maskNet = load_model("mask_detector.model")
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
while True:
    frame = vs.read()
    frame = imutils.resize(frame, width=400)
    (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)
    for (box, pred) in zip(locs, preds):
        # unpack the bounding box and predictions
        (startX, startY, endX, endY) = box
        (mask, withoutMask) = pred


        label = "Mask" if mask > withoutMask else "No Mask"
        color = (0, 255, 0) if label == "Mask" else (0, 0, 255)
        label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
        cv2.putText(frame, label, (startX, startY - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
        cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
cv2.destroyAllWindows()
vs.stop()
```

## Output

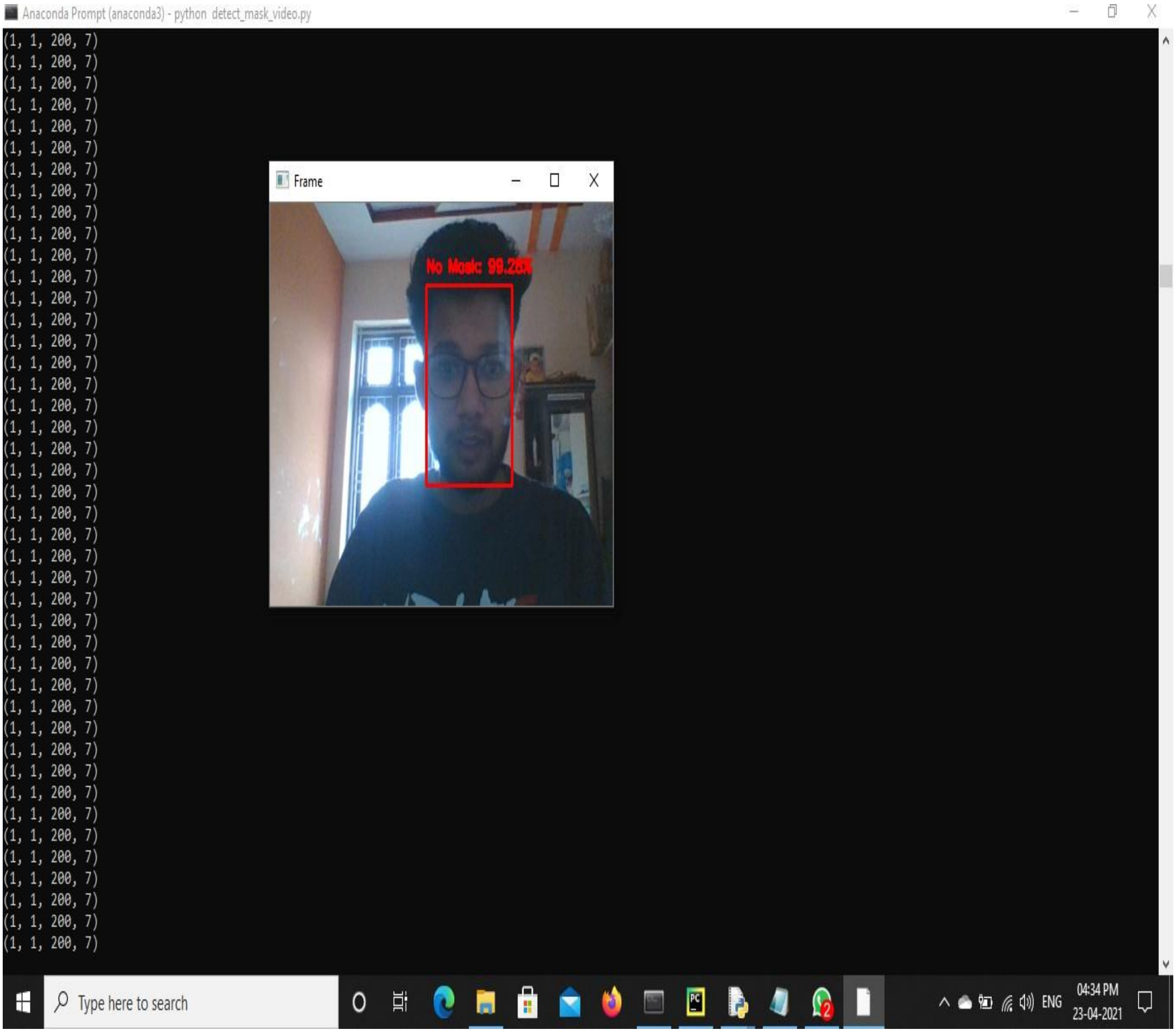



Figure 6.1: **Model detected person is without mask**

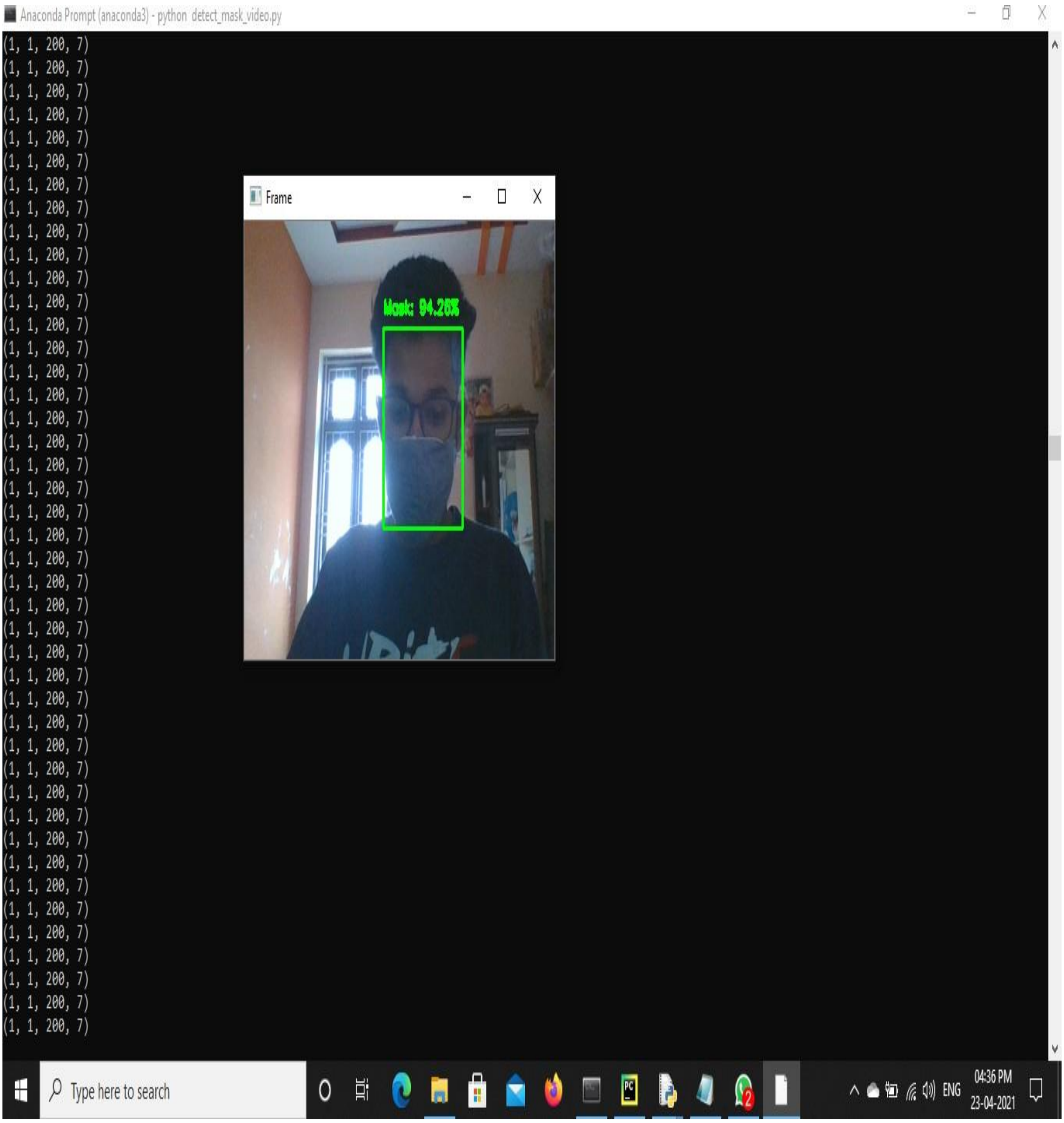



Figure 6.2: **Model detected person is without mask**

# Chapter 7

# CONCLUSION AND FUTURE ENHANCEMENTS

## 7.1 Conclusion

We conclude that by using machine learning techniques validating the usage and detection of mask using machine learning with this we can filter the people in community areas to avoid the spread of corona virus. In proposed mask detection work, the training and testing data sets are implemented successfully by categorizing as masked and unmasked MobilenetV2 image classifiers are used to classify the images as masked faces and unmasked faces, it is the important factor in the implementation. The accuracy of the proposed model was good and can be implemented at any time. The model can be implemented in any organizations, schools, offices, malls, densely crowded areas etc to stop the spreading of virus.

## 7.2 Future Enhancements

- This project is not only limited to this current pandemic but can be used as a protective technology to prevent from the micro organisms and pollution by wearing a mask.

- With the data gathered through this work we can use this for creating a survey, by analyzing how much percentage of people are wearing masks and not.
- If the person is not wearing mask, then we can create a software that sends the warning messages to wear a mask and it can used to create awareness among the people.

# Chapter 8

# SOURCE CODE & POSTER PRESENTATION

## 9.1 Sample Code

```
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")
```

# References

[1] Dr. Vandana S. Bhat , Arpita Durga Shambavi , Komal Mainalli , K M Manushree, Shraddha V Lakamapur, “Review on Literature Survey of Human Recognition with Face Mask”,Issue 1,20 Mar,2021

[2] Zhongyuan Wang, Guangcheng Wang, Baojin Huang, Zhangyang Xiong, Qi Hong, Hao Wu, Peng Yi, Kui Jiang,Nanxi Wang, Yingjiao Pei, Heling Chen, Yu Miao, Zhibing Huang, and Jinbi Liang, “Masked Face Recognition Dataset and Application” ,version2,23 Mar,2020

[3] M.H.Yang,D.J.Kriegman, and N.Ahuja,”Detecting Faces in Images:A survey”,IEEE Trans.pattern analysis and machine intelligence,vol.24,01Jan,2002.

[4] Preethi Nagrath,Rachana jain,Agam madan,Rohan Arora ,SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2,Dec31,2020

[5] X. Zhang, S. Ren and J. Sun, ”Deep residual learning for image recognition”,June,2016.

[6] H.Qu,X.Fu ,”Research on semantic segmentation of high-resolution remote sensing image based on full convolutional neural network”, 2018 12th International Symposium on Antennas Propagation and EM Theory (ISAPE), pp. 1-4, Dec 2018.

[7] S. Kumar, A. Negi, J. N. Singh and H. Verma, ”A deep learning for brain tumor mri images semantic segmentation using fcn”, 2018 4th International Conference on Computing Communication and Automation (ICCCA), pp.1-4,Dec 2018

[8] K. Li, G. Ding and H. Wang, ”L-fcn: A lightweight fully convolutional network for biomedical semantic segmentation”, 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Dec 2018

[9] T. Poggio and L. Stringa, “A project for an intelligent system: Vision and Learning,” Int. J. Quantum Chem., vol. 42,2017

[10] Mingjie Jiang, Xinqi Fan and Hong Yan, RETINAFACEMASK: A FACE MASK DETECTOR,8 Jun,2020.