DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

(FACE DETECTION WITH COVID MASK ON)

By
Vishnu Prasad M – ENG18CS0324.
Ashish Ranjan - ENG19CS1003.
S Abhishek Reddy - ENG19CS1012.
Soundarya R - ENG19CS1018.

Under the supervision of

Dr. Debanjali Bhattacharya
Assistant Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,

(2021-2022)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **"FACE DETECTION WITH COVID MASK ON"** is carried out by **Vishnu Prasad M (ENG18CS0324), Ashish Ranjan (ENG19CS1003), S Abhishek Reddy (ENG19CS1012), Soundarya R (ENG19CS1018)**, bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2021-2022**.

Dr. Debanjali Bhattacharya	**Dr Girisha G S**	**Dr. A Srinivas**
Assistant Professor Dept. of CS&E, School of Engineering Dayananda Sagar University Date:	Chairman CSE School of Engineering Dayananda Sagar University Date:	Dean School of Engineering Dayananda Sagar University Date:

Name of the Examiner **Signature of Examiner**

1.

2.

DECLARATION

We, **Vishnu Prasad M (ENG18CS0324), Ashish Ranjan (ENG19CS1003), S Abhishek Reddy (ENG19CS1012)** and **Soundarya R (ENG19CS1018)** are student's of eighth semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the phase-II project titled **"Face Detection with COVID Mask On"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**.

Student	**Signature**
Vishnu Prasad M	
USN : ENG18CS0324	
Ashish Ranjan	
USN : ENG19CS1003	
S Abhishek Reddy	
USN : ENG19CS1012	
Soundarya R	
USN : ENG19CS1018	

Place : Bangalore

Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science and Engineering**, **Dayananda Sagar University,** for providing right academic guidance that made our task possible.

We would like to thank our guide **Dr. Debanjali Bhattacharya**, **Assistant Professor**, **Dept. of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.

We would like to thank our Project Coordinators Dr. Meenakshi Malhotra and Dr. Bharanidharan N, and all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

We have seen that the world has gone into a very depressed state due to the COVID-19 pandemic. Thousands of people die every day due to this disease and there is very little that can be done to bring about its eradication. In the field of Computer Science, a lot of research is going on to invent different techniques to analyze the statistics of the disease which helps in planning steps to eradicate it. Machine learning plays a huge role in health care by being able to utilize many algorithms to predict and analyze diseases and recommend remedies.

Face Detection is a technique that uses machine learning to recognize features on a person’s face by analyzing different sets of images and training itself. Using this technique, we plan to build a software for Face Detection with COVID Mask on. Face Detection is performed in three stages, namely, Pre-processing, Face Detection and Feature Extraction.

In this project we have used Convolutional Neural Networks (CNN) to detect the facial features and the presence of a face mask.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

Public use of face masks has been common in China and other nations in the world since the beginning of the new coronavirus disease outbreak. We now know from recent studies that a significant portion of individuals with coronavirus lack symptoms (“asymptomatic”) and that even those who eventually develop symptoms (“pre-symptomatic”) can transmit the virus to others before showing symptoms, according to the advisory published by the World Health Organization (WHO). “This means that the virus can spread between people interacting in close proximity — for example, speaking, coughing, or sneezing — even if those people are not exhibiting symptoms”. Recent information also gives trace of new strains of the corona virus, in which the virus has changed its structure and become mutant. Some of these new strains cannot even be detected using the RT-PCR test that is commonly used. So it is inevitable for the people of a large and overpopulated country like India to wear masks and let the work go on.

1.1 SCOPE

The scope of our project is to implement a face mask detection system that can accurately predict if a person is wearing a mask or not and thus help enforce public health and safety protocols.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

- **Problem :** Some systems are not able to accurately identify presence of face masks accurately and hence does not help in identifying violators.

- **Solution :** Implement a face mask detection system that can accurately recognize whether a person is wearing a mask or not using machine learning and facial feature extraction technologies.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

Under the current Covid-19 situation such a system is crucial for use in various private and public facilities. Following are some use cases which can benefit from this system –

Hospitals: The system can be integrated with CCTV cameras to see if their staff is wearing masks while coming in contact with patients. This mitigates the spread of the disease among patients who are at much more risk due to the presence of comorbidities.

Offices: The system can help in maintaining safety standards to prevent the spread of Covid-19 or any such air borne disease among employees who regularly come in close contact with each other and also in a place the presence of air conditioning may further accelerate the spread.

The choice of a system must be based on the best performance. Hence the above performance metrics may be considered for coming out with the best system so that it can be implemented at large scale.

3.1 STATE OF THE ART WORK

Author's Name/ Paper Title	Conference/ Journal Name and year	Technology/ Design	Results shared by author	What you infer
Detection of Face Mask using Convolutional Neural Network	Published: 2020	Convolutional Neural Network, MobileNetV2	Accurate prediction of Presence of a Mask.	How the validation accuracy and validation loss affects the face detection.
Wearmask:Fast In-browser face mask detection with serverless edge computing for COVID-19.	Published:2021	Convolutional Neural Network, SRCNet	Facial image classification	Image pre processing , facial detection and cropping.

Table 3.1 – State of the Art Work

CHAPTER 4

PROJECT DEFINITION

CHAPTER 4 PROJECT DEFINITION

4.1 DESIGN

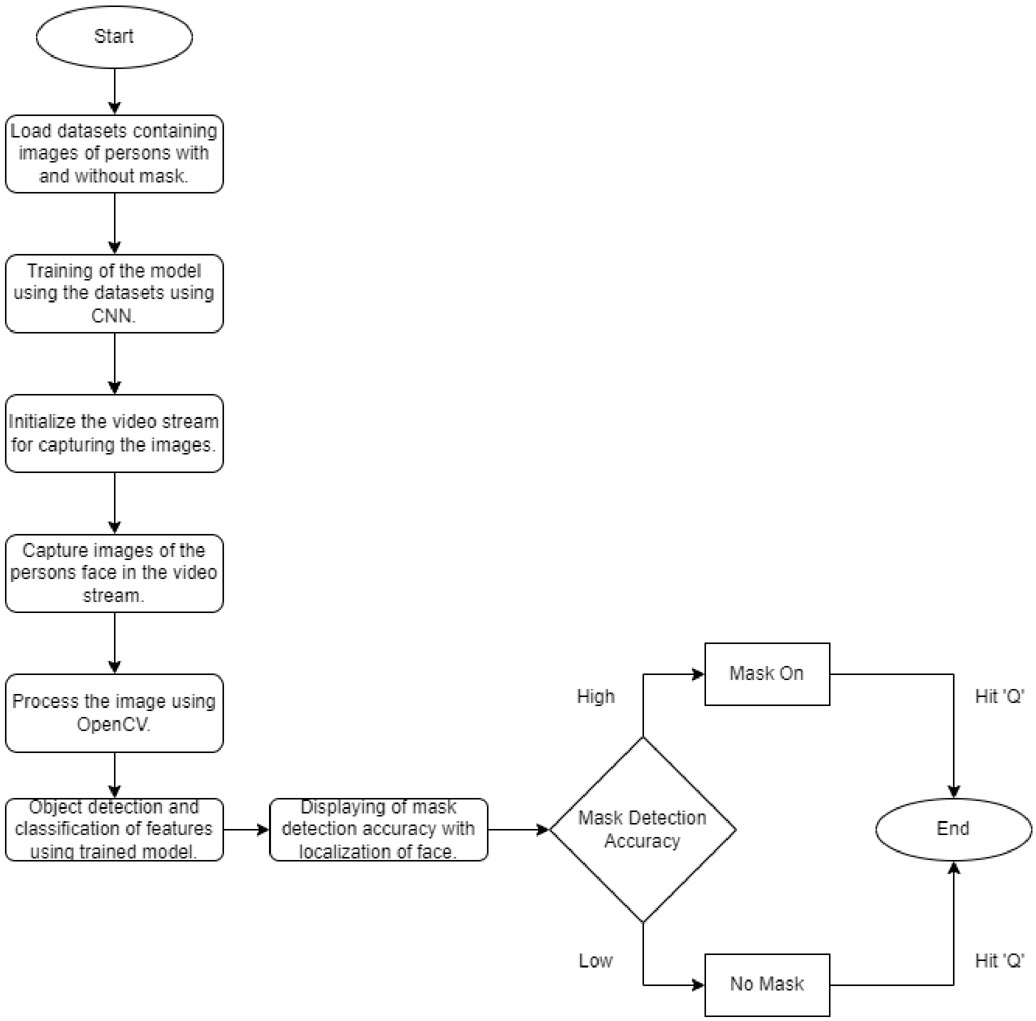


Figure 4.1 – Data Flow Diagram for Face Mask Detection System.

4.2 STEPS INVOLVED

1. Dataset preprocessing – Checking of images in the dataset to understand which are masked and which are non-masked.
2. Face detection – Detecting the face of a person using a video stream.
3. Facial Feature extraction – Extraction of features that define a masked face using a trained model.
4. Classification based on whether the person in the video stream is wearing a mask or not.

4.3 ASSUMPTIONS AND DEPENDENCIES

The main assumption that we make in this project is that the trained model is able to detect a face properly and is able to extract the facial features. It should be able to do so while being able to overcome technical constraints related to hardware such as camera constraints (such as inability to detect face due to lack of proper lighting conditions).

The working of the project shall depend on the hardware on which it is running for the sake of good performance. Since the programming language used is platform independent, there should be no issue with the operating system. The project software needs to fulfill some requirements such as software libraries to work properly (E.g. – Python libraries like TensorFlow and Keras).

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

- The user should be able to access the interface easily.
- The program should not cause delay in detecting the face.
- It should be able to differentiate a masked and non-masked face accurately.

5.2 SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements

- Numpy
- Scipy
- Tensorflow
- Keras
- Scikit-Learn
- Imutils
- OpenCV
- MobileNetV2

Hardware Requirements

- Processor – Intel Core™ i3-8145U CPU @ 2.10GHz
- System Type – x64 based System.
- RAM – 4 GB
- Secondary Storage – 256 GB
- Graphics – Intel UHD Graphics 620

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

- The dataset for our project was collected from 'kaggle.com' and few open source image libraries.
- Our system uses machine learning technologies such as convolutional neural networks (CNN) for extracting facial features from the video stream.
- We use OpenCV technology for processing the images captured from the video stream.
- The processed images are used for training the mask detection model.
- The model is then used in a live video stream to extract the facial features and detect the presence of a face mask.
- The appropriate classification is shown in a box around the persons' face.

6.1 SOURCE CODE SNIPPETS

- The detection system uses two programs, one to train the mask detector model (train_mask_detector.py) and one to run a live video stream and detect the presence of face masks using the trained model (detect_mask_video.py).

Train_mask_detector.py

- Fetching the datasets from a directory, creating categories for and preparing them for classification.

```
DIRECTORY = r"C:\Users\vishn\Documents\DSU\Major\Code\Face-Mask-Detection-master\dataset"
CATEGORIES = ["with_mask", "without_mask"]

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)
```

- Creating a training image generator to modify the images in the datasets for training.

```
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")
```

- Create the head model and place it above 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)

model = Model(inputs=baseModel.input, outputs=headModel)
```

- Compile the model and train the head of the model.

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

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)
```

- Saving the trained model to the disk.

```
print("[INFO] saving mask detector model...")
model.save("mask_detector.model", save_format="h5")
```

Detect_mask_video.py

- Creating a blob from the video stream frame by grabbing its dimensions and passing the blob through the network.

```
(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)
```

- Loop over the detections and extract the confidence, i.e. probability of the detections. Weak detections are filtered out by ensuring that the confidences are greater than the minimum confidence.

```
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")
```

- Pre-processing the images by resizing it and converting its colour channel from BGR to RGB.

```
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)
```

- Make predictions if at least one face is detected and return their location and predictions.

```
if len(faces) > 0:
    faces = np.array(faces, dtype="float32")
    preds = maskNet.predict(faces, batch_size=32)

return (locs, preds)
```

- After loading the face mask detector model from the disk, start the video stream, detect faces and predict the presence of masks.

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

while True:
    frame = vs.read()
    frame = imutils.resize(frame, width=1050)

    (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)
```

- Loop over the detected faces and locations, draw a bounding box on the output frame and display the label and probability on the box.

```
for (box, pred) in zip(locs, preds):
    (startX, startY, endX, endY) = box
    (mask, withoutMask) = pred

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

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

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

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

- The system has been worked with the MobileNetV2 classifier.
- **MobileNetV2** is a state of the art for mobile visual recognition including classification, object detection and semantic segmentation. This classifier uses depth wise separable convolution which is introduced to dramatically reduce the complexity cost and model size of the network, and hence is suitable to mobile devices, or devices that have low computational power.
- In MobileNetV2, another best module that is introduced is inverted residual structure**.** Non-linearity in narrow layers is deleted. Keeping MobileNetV2 as backbone for feature extraction, best performances are achieved for object detection and semantic segmentation.
- Here, we use MobileNetV2 to train the base model.

```
baseModel = MobileNetV2(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))
```

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

- On training the mask detection model using our dataset, we recorded the following statistics.

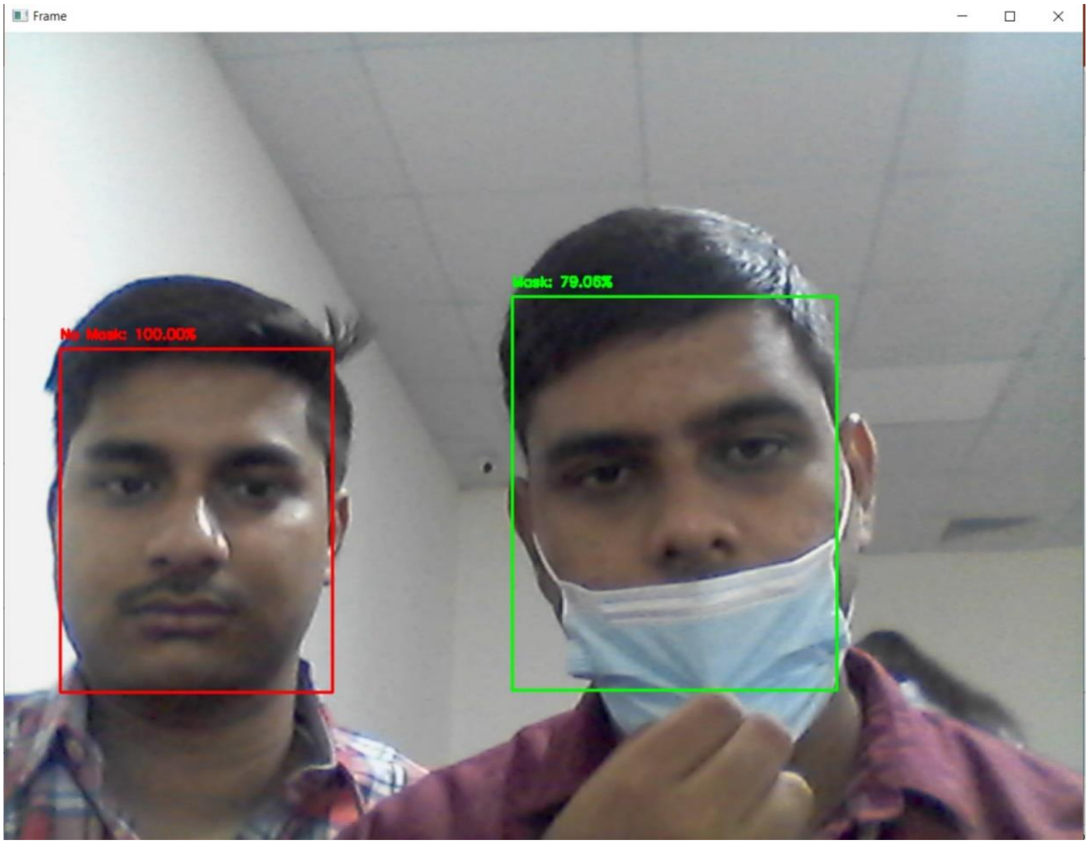
	precision	recall	f1-score	support
with_mask	0.99	1.00	0.99	383
without_mask	1.00	0.99	0.99	384
accuracy			0.99	767
macro avg	0.99	0.99	0.99	767
weighted avg	0.99	0.99	0.99	767

Figure 8.1 – Precision and Accuracy of Model after Training.

8.2 – Training accuracy and loss Graph.

- The accuracy has been recorded to be at 99.0%.
- The precision of the model in detection are 0.99 in presence of a face mask and 1.00 in its absence.
- Even if the face in the video stream is at a tilted angle, the model can detect it with high probability.

Output Screenshots



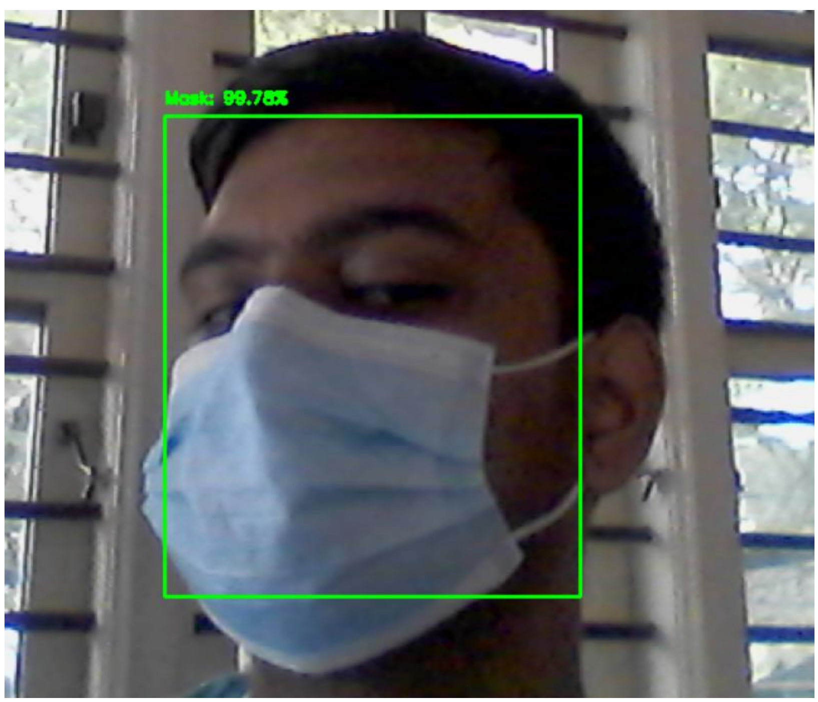


Figure 8.3 – Detection and prediction of the presence of face mask.

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

CHAPTER 9 CONCLUSIONS AND FUTURE WORK

In this work, we have implemented the face mask detection system using CNN and MobileNetV2 classification architecture that successfully detects the presence or absence of a face mask on a person's face with high accuracy in different conditions and surroundings. This system plays a crucial role in maintaining public safety during the peak of such a pandemic.

This system can be used at all public and corporate premises to keep a check on public safety violators.

In the future, this system can be further developed into a biometric authentication system which shall use face recognition technology to detect the identity of a person even when they are wearing a face mask.

REFERENCES

[1] RIYA CHIRAGKUMAR SHAH, RUTVA JIGNESH SHAH. DETECTION OF FACE MASK USING CONVOLUTIONAL NEURAL NETWORK (2020).

[2] A. G. HOWARD, M. ZHU, B. CHEN ET AL., “MOBILENETS: EFFICIENT CONVOLUTIONAL NEURAL NETWORKS FOR MOBILE VISION APPLICATIONS,” 2017.

[3] WEI WANG, YUTAO LI, TING ZOU, XIN WANG, JIEYU YOU, YANHONG LUO, "A NOVEL IMAGE CLASSIFICATION APPROACH VIA DENSE-MOBILENET MODELS", MOBILE INFORMATION SYSTEMS, VOL.2020, ARTICLEID 7602384, 8 PAGES, 2020. HTTPS://DOI.ORG/10.11 55/2020/7602384

[4] I. B. VENKATESWARLU, J. KAKARLA AND S. PRAKASH, "FACE MASK DETECTION USING MOBILENET AND GLOBAL POOLING BLOCK,"