A

Mini Project Report on

FACE MASK DETECTION

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

Bachelor of Technology in

Electronics & Telecommunication

Submitted by

Name of Students	Roll No.
Utkarsha Shatrughna Jadhav	19151108
Sagar Raju Wani	19151215
Wadavane Harshvardhan Rajkumar	19151223
Omkar Balaji Deshmukh	19151224

Under the guidance of

Prof. Sachin D. Shingade sir

Department of Electronics and Telecommunication

Government College of Engineering, Karad

(An Autonomous Institute of Govt. of Maharashtra)

Maharashtra India – 415124

CERTIFICATE

This is to be certified that this is a bonafide record of the project presented by the students whose names are given below of Third Year B.Tech V semester (2021-22) of Electronics and Telecommunication Department, Government College of Engineering, Karad have completed their Mini Project entitled

" FACE MASK DETECTION USING RAPSBERRY PI 3, PI CAM"

They have submitted their Project Report for the partial fulfilment of the curriculum of the Degree of Bachelor of Electronics and Telecommunication from Government College of Engineering, Karad.

Roll No.	Name of Candidate
Utkarsha Shatrughna Jadhav	19151108
Sagar Raju Wani	19151215
Wadavane Harshvardhan Rajkumar	19151223
Omkar Balaji Deshmukh	19151224

External Examiner

Prof. Sachin D. Shingade
Project Guide

Prof. Supriya Diwan
(Head of Department)
Dept. Of Electronics & Telecomm.
GCE, Karad

Date: / / 2021

DECLARATION

We

1. Utkarsha Shatrughna Jadhav
2. Sagar Raju Wani
3. Wadavane Harshvardhan Rajkumar
4. Omkar Balaji Deshmukh

hereby declare that the Project Report entitled,

" FACE MASK DETECTION USING RAPSBERRY PI 3, PI CAM"

done by our group under the guidance of Project guide and External Examiner In partial fulfilment of the requirement for the award of the Degree of Bachelors of Technology (Electronics and Telecommunication) of Government College of Engineering, Karad in the academic year 2021 – 2022.

Prof. S. Shingade
Project Guide
GCE, Karad

External
Examiner

ACKNOWLEDGEMENT

We take this opportunity to express my deep sense of gratitude and sincere thanks to Prof. A.T. PISE (PRINCIPLE), Government College of Engineering, Karad without whom this work was impossible. This continuous encouragement and support helped me throughout.

We would like to thank to my project guide Prof. S.D. Shingade Sir, under their guidance, constant source of encouragement, support and motivation this project was carried out. He provided me all resources and generous help and technical matters at all points of time. And more than that trust he kept upon me more than myself.

We would also like to thank Prof. Supriya Diwan (HEAD OF THE DEPARTMENT) of Government College of Engineering, Karad for giving invaluable suggestions.

We would like to thank my classmates& friends for being with me all the times in odds and evens phases of life. Whose direct and indirect support and sincere advice brought me to achieve this? We are very much thankful to all teaching and non-teaching staff members of Government College of Engineering, Karad who were directly and indirectly involve in my project work.

We wish to thank our FAMILY and FRIENDS for having raised me in such conducive and loving environment, for teaching me how consciousness can emerge from sub symbolic computation, for inspiring me to study connectionism which has enabled me to come so far. They helped me in a long way throughout the project phase; this forced me to limit and brought forward the best in me.

ABSTRACT

COVID 19 pandemic is causing a global health epidemic. The most powerful safety tool is wearing a face mask in public places and everywhere else. The COVID 19 outbreak forced governments around the world to implement lockdowns to determine virus transmission. According to survey reports, wearing a face mask at public places reduces the risk of transmission significantly. In this paper, an IoT-based that uses a machine learning model for face mask detection.

The proposed model can be used for any shopping mall, hotel, apartment entrance, etc. As an outcome a cost-effective and reliable method of using AI and sensors to build a healthy environment. Evaluation of the proposed framework is done by the Face Mask Detection algorithm using the TensorFlow software library.. This proposed system can detect face mask of the users preventing from COVID 19 by enabling the Internet of Things (IoT) technology.

ABBREVIATION

Acronym	Definition
COVID-19	Corona Virus Disease 2019
IoT	Internet of Things
OpenCV	Open Source Computer Vision Library
CNN	Convolution Neural Networks
VNC	Virtual Network Computing
API	Application programming interface
OS	Operating System

TABLE OF CONTENTS

LIST OF FIGURES:

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Detection of face masks can be a tough job. During this period, it received extra attention due to the spread of corona virus disease. This is why various nations adopt the rule "No entry without masking." Facade detection is a critical safety problem and Covid-19 prevention. Masking reduces the risk of the associate of exposure to infected patients, irrespective of the symptoms. The identification of masks is carried out in airports, hospitals, offices and academic areas.

The detection of masks has thus become a challenging and highly critical issue. Facial recognition is however quicker if not masked Detection of façades is a key safety issue and prevention of Covid-19.In the medical field, masking lowers the associate's potential risk of exposure to infected patients, whether or not they show symptoms. Airports, hospitals, offices and academic departments are used in mask detection. Mask detection has therefore become an extremely important and difficult problem. However, face recognition with mask is key as the extraction of coated face is incredibly complex compared to conventional face. Face recognition without masking is simpler. That's such a vast number of facial characteristics as the measurements of the nose, mouth and kidney within the masked face.

In the field of medicine, masks lessen the risk of potential exposures to the nursing associate. That’s such a vast number of facial characteristics as the measurements of the nose, mouth and kidney within the masked face. The mask in the medical field reduces the associate's potential risk of exposure to infected patients whether or not they have symptoms. So much mask detection is always focused in two steps. 1) Reconnaissance of face 2) Extraction of Feature The first step is face recognition; we want to find the face from a photograph here. Especially in the nursing unmasked faces in an image, the multiple mask Associate is detected. It is also solved with an old technique of detection of objects. Viola-Jones law, adaptive Boost Algorithm and GROW are the standard face detection algorithms for square meters (Histogram of Gradient). The technique of object detection is here classified as multi-stage detectors and individual short detectors (SSD). Here, a vast number of papers on the measurement of mass detection have been analyzed. Many square measurement approaches used for mask detection like video analytics, segmentation of image linguistics.

1.2 EXPECTED OUTCOME

New developments and the availability of smart technologies force to the creation of new models, which will help meet the needs of developing countries. In this work, an IoT-enabled smart door is developed to monitor body temperature and detect face masks that can enhance public safety. This will help to reduce manpower while also providing an extra layer of protection against the spread of Covid-19 infection. The model uses a real-time deep learning system using Raspberry pi to detect face masks, and temperature detection as well as monitor the count of people present at any given time.

The device performs excellently when it comes to temperature measurement and mask detection, the trained model was able to achieve a result of 97 percent. The test results demonstrate a high level of accuracy in detecting people wearing and not wearing facemasks, as well as it also generates alarms monitored and recorded. Furthermore, there are numerous techniques to enhance performance to improve results. Future development will include improving the accuracy of these steps, using a combination of various features, and improving performance, as well as producing a mobile app with a user friendly interface for monitoring. As a result, authorities will be able to take immediate action following pandemic safety standards.

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Survey of Paper1

[1] S. Joshi, S. S. Joshi, G. Kanahasabai, R. Kapil and S. Gupta, "Deep Learning Framework to Detect Face Masks from Video Footage," 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, pp. 435-440, doi: 10.1109/CICN49253.2020.9242625.

Proposed Work:

1]Proposed paper an approach for detecting facial masks in videos using deep learning.

2] Proposed framework capitalizes on the MTCNN face detection model to identify the faces and their corresponding facial landmarks present in the video frame.

3] The proposed framework was tested on a dataset which is a collection of videos capturing the movement of people in public spaces while complying with COVID-19 safety protocols.

Out - Comes :

1]The paper present new approach for detecting face masks from videos

2] Proposed approach was found to be effective as it portrayed high precision, recall, and accuracy values on the chosen dataset which contained videos with varying occlusions and facial angles

3] The effectiveness of the facial mask classifier largely confides on the ability of the face detection algorithm to accurately identify faces in the video frames.

Design Methodology :

2] A distinct facial classifier is built using deep learning for the task of determining the presence of a face mask in the facial images detected.

1] The dataset used in this work is a collection of footage videos of public places from multiple geographical locations, compiled from YouTube

2] Programs were written in Python - 3.6 and utilised OpenCV - 4.2.0, Keras - 2.3.0 and TensorFlow - 2.2.0.

3] The experiments were conducted on Google Colab with Intel(R) Xeon(R) 2.00 GHz CPU, NVIDIA Tesla T4 GPU, 16 GB GDDR6 VRAM and 13 GB RAM

2.2 Literature Survey of Paper 2

[2] X. Fan, M. Jiang and H. Yan, "A Deep Learning Based Light-Weight Face Mask Detector With Residual Context Attention and Gaussian Heatmap to Fight Against COVID-19," in IEEE Access, vol. 9, pp. 96964-96974, 2021, doi: 10.1109/ACCESS.2021.3095191.

Proposed Work:

1] Proposed a deep learning based single-shot light-weight face mask detector to meet the low computational requirements for embedded systems.

2] Propose two novel methods to enhance the model's feature extraction process as well as a novel residual context attention module.

3] Proposed model has a high potential to contribute to public health care and fight against the coronavirus disease 2019 pandemic.

4] Proposed model achieves state-of-the-art results on two public datasets, the AIZOO and Moxa3K face mask datasets.

Out - Comes :

1] Model is capable of tackling the challenges present in face mask detection

2] Experimentally, paper shows that light-weight models can achieve similar or even better performance than heavy models by using RCAM and SGHR.

3] Proposed face mask detector has a high potential to contribute to public health care to control the spread of COVID-19. With One drawback of the method is the extra computation required for generating heatmaps and, due to limitations of the datasets, the method cannot distinguish between correct and incorrect mask wearing

Design Methodology:

1] Approaches system design by Deep learning based

2]Dataset Used: AIZOO, Moxa3K

3] General architecture of the SL-FMDet, followed by two novel modules, RCAM and SGHR are used.

2.3 Literature Survey of Paper 3

[3] J. Zhang, F. Han, Y. Chun and W. Chen, "A Novel Detection Framework About Conditions of Wearing Face Mask for Helping Control the Spread of COVID-19," in IEEE Access, vol. 9, pp. 42975-42984, 2021, doi: 10.1109/ACCESS.2021.3066538

Proposed Work:

1] Proposed introduce a new practical dataset covering various conditions, which contains 8635 faces with different wearing status.

2] Proposed a novel detection framework about conditions of wearing face mask, named Context-Attention R-CNN, which enlarge the intra-class distance and shorten inter-class distance by extracting distinguishing features

Out - Comes :

1] Work is applicable to protect against other infectious diseases which can be spread by such things as coughing, sneezing, or even speaking at close range.

2] Developed Context-Attention R-CNN, a framework to detect conditions of wearing face mask, which contains three novel points: multiple context feature extractor, decoupling branches and attention module.

3] Experiments show that Context-Attention R-CNN outperforms many state-of-the-art detectors, including two-stage detectors and single-stage detectors.

Design Methodology :

1]Dataset used: MAFA

2]Framework to detect conditions of wearing face mask: Context-Attention R-CNN

3] Algorithm: Context-Attention R-CNN

4] DATA ANNOTATION format for object detection task: PASCAL VOC format

5] Network architecture: VGG16, pre-trained on ImageNet

2.4 Literature Survey of Paper 4

[4] T. M. Hoang, G. P. Nam, J. Cho and I. -J. Kim, "DEFace: Deep Efficient Face Network for Small Scale Variations," in IEEE Access, vol. 8, pp. 142423- 142433, 2020, doi: 10.1109/ACCESS.2020.3012660.

Proposed Work:

1] Proposes a novel face detector called DEFace that focuses on the challenging tasks of face detection to cope with a small size that is under 12 pixels and occlusions due to a mask or human body parts.

2] Proposed the extended feature pyramid network (FPN) module to detect small faces by expanding the range of P layer, and the network by adding a receptive context module (RCM) after each predicted feature head from the top-down pathway in the FPN architecture to enhance the feature discriminability and the robustness

3] Based on the FPN principle, the combination between the low- and high-resolutions are beneficial for object detection with different object sizes

Out - Comes :

1] The work is based on novel deep network model by adapting the RetinaNet structure with extended FPN module. proposed method balances trade-off between the performance with a fast computation speed

2] Proposed method outperforms the other state-of-the-art methods with hard face examples in the WIDER Face dataset

3] DEFace can detect the face region more accurately in comparison to the other state-of-the-art methods while maintaining the processing time

4] The experimental results for the challenging benchmarks on WIDER Face , FDDB , and MAFA dataset showed that the proposed method outperforms other state-of-the-art methods in terms of small face detection, and efficiency.

Design Methodology :

1] Dataset Used : WIDER FACE DATASET, FDDB DATASET, MAFA DATASET

2] Investigation performed the training and testing with a desktop computer that featured an Intel CoreTM i7 processor with a speed of 3.60 GHz, 16 GB of main memory, and a NVIDIA GeForce RTX 2080 Ti graphics card that includes 4,352 compute unified device architecture (CUDA R) cores and 11 GB of graphics memory

3] Approaches deep neural network for face detection inspired by RetinaNet

2.5 Literature Survey of Paper 5

[5] L. Aziz, M. S. B. Haji Salam, U. U. Sheikh and S. Ayub, "Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review," in IEEE Access, vol. 8, pp. 170461-170495, 2020, doi: 10.1109/ACCESS.2020.3021508.

Proposed Work:

1] Paper provides a comprehensive survey of recent advances in visual object detection with deep learning.

2] Proposes Dataset and performance evaluation.

3] Proposes Object Detection in various fields .

For example :-Object detection in survillance, Object detection in military, Object detection in medical, Object detection in daily life

Out - Comes :

1] Work is applicable to protect against other infectious diseases which can be spread by such things as coughing, sneezing, or even speaking at close range.

2]It contains future direction and current trends.

3]Work is applicable for efficient post processing methods and also for objct detection in multi-domain.

Design Methodology :

1] Dataset used :- PASCAL VOC 2007, PASCAL VOC2012, ImageNet , Microsoft COCO and OpenImages .

2] In this paper, the performance of various object detection methods using three benchmark datasets is compared such as PASCAL VOC 2007/2012 , Microsoft COCOand Open image, while evaluated algorithm are SPP-net, Fast R-CNN , NOC , Bayes , Mr-CNN& S-CNN , Faster R-CNN , HyperNet , ION ,MS-GR , StuffNet , SSD300, SSD512 ,OHEM , SDP CRC, GCNN , subCNN ,GBD-Net , PVANET , YOLO , YOLOv2 ,R-FCN , FPN , Mask R-CNN , DSSD ,R-CNN and DSOD .

3]The performance of object detection algorithms is evaluated with three parameters, such as recall, precision, and Frame per Second (FPS).

2.6 Literature Survey of Paper6

[6] B Varshini , HR Yogesh , Syed Danish Pasha , Maaz Suhail , V Madhumitha , Archana Sasi , IoT-Enabled Smart Doors for Monitoring Body Temperature and Face Mask Detection, Global Transitions Proceedings (2021), doi: https://doi.org/10.1016/j.gltp.2021.08.071

Proposed Work:

1] Proposed model can be used for any shopping mall, hotel, apartment entrance, etc

2] IoT-enabled smart door is developed to monitor body temperature and detect face masks that can enhance public safety

Out - Comes :

1] Cost-effective and reliable method of using AI and sensors to build a healthy environment

2] IoT-enabled smart door is developed to monitor body temperature and detect face masks that can enhance public safety

3] Model uses a real-time deep learning system using Raspberry pi to detect face masks, and temperature detection as well as monitor the count of people present at any given time

4] Demonstrate a high level of accuracy in detecting people wearing and not wearing facemasks, as well as it also generates alarms monitored and recorded

Design Methodology :

1] Evaluation of the proposed framework is done by the Face Mask Detection algorithm using the TensorFlow software library

2] Work was carried out on a computer running the 62-bit Windows10 operating system and equipped with an Intel ® CoreTM i5-8265U CPU running at 1.60GHz and 8 GB of RAM

3] Python 3.7, Keras as the backend and the Tensor-flow platform,

4] Raspberry Pi 3 Model V , IR (2), MLX90614, Pi Cam

5] Algorithm: CNN

6]Dataset created by Prajna Bhandary contains 1,376 photos divided into two classes: 690 images with masks and 686 images without masks

2.7 Literature Survey of Paper 7

[7] S. V. Militante and N. V. Dionisio, "Real-Time Facemask Recognition with Alarm System using Deep Learning," 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC), 2020, pp. 106-110, doi: 10.1109/ICSGRC49013.2020.9232610.

Proposed Work:

1] Paper manuscript presented a study on real-time facemask recognition with an alarm system through deep learning techniques by way of Convolutional Neural Networks

2] Paper a useful tool in fighting the spread of the COVID-19 virus by detecting a person who wears a facemask or not and sets an alarm if the person is not wearing a facemask

Out - Comes :

1] System develops a Raspberry Pi-based real-time facemask recognition that alarms and captures the facial image if the person detected is not wearing a facemask

2**]** Test results show a distinguished accuracy rate in detecting persons wearing a facemask and not wearing a facemask.

3] VGG-16 CNN model achieving a 96% result for performance accuracy

Design Methodology :

1] Research study uses deep learning techniques in distinguishing facial recognition and recognize if the person is wearing a facemask or not

2] VGG-16 CNN model achieving a 96% result for performance accuracy

3] dataset collected contains 25,000 images using 224x224 pixel resolution and achieved an accuracy rate of 96% as to the performance of the trained model

4]Conducted on Intel Processor Core i7- 8750 2.20 GHz base speed, Graphics Card NVIDIA Geforce GTX 1050 4 GB, the memory size of RAM is 6GB Kingston DDR4 2667 MHz, and the primary storage is SSD 256 GB.

2.8 Literature Survey of Paper 8

[8] H. Li, Z. Lin, X. Shen, J. Brandt and G. Hua, "A convolutional neural network cascade for face detection," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5325-5334, doi: 10.1109/CVPR.2015.7299170.

Proposed Work:

1] In this work, we present a CNN cascade for fast face detection.

2] The proposed face detector is robust to large visual variations.

3] The proposed detector is very fast, achieving 14 FPS for typical VGA images on CPU and can be accelerated to 100 FPS on GP.

Out - Comes :

1] Detector evaluates the input image at low resolution to quickly reject non-face regions and carefully process the challenging regions at higher resolution for acdetector evaluates the input image at low resolution to quickly reject non-face regions and carefully process the challenging regions at higher resolution for accuracy.

2] The public face detection benchmark FDDB, the proposed detector outperforms the state-of-the-art methods.

Design Methodology :

1] Datasheet used :-AFW DATASHEET

2] Detector evaluates input image .

3] Precision -recall curve of the faster version of detector

4] Calibration nets are introduced in the cascade to accelerate detection and improve bounding box quality.

CHAPTER 3

PROBLEM STATEMENT AND AIM

3.1 Problem statement

In the background of the COVID-19 pandemic, institutions such as the academy suffer a great deal from practically closed globally if the current situation is not going to change. COVID-19 also known as Serious Acute Respiratory Syndrome Corona virus-2 is an infectious disease that is released from an infected sick person who speaks, sneezes, or coughs by respiratory droplets. This spreads quickly through close contact with anyone infected, or by touching objects or surfaces affected with a virus.

There's still currently no vaccine available to protect against COVID-19 and preventing exposure to the virus seems to be the only method to safeguard ourselves. Wearing a facemask that covers the nose and mouth in a public setting and often washing hands or the use of at least 70% alcohol-based sanitizers is one way to avoid being exposed to the virus. Amid the advancement of technology, Deep Learning has proven its effectiveness in recognition and classification through image processing. The research study uses deep learning techniques in distinguishing facial recognition and recognize if the person is wearing a facemask or not.

The system develops a Raspberry Pi and using pi cam, python, opencv based real-time facemask recognition will captures or detect the facial image if the person detected is not wearing a facemask. This study or project will beneficial in combating the spread of the virus and avoiding contact with the virus.

3.2 Project Aim:

“Design and Development of Face Mask Detection System”

To achieve the proposed Aims, the Objectives are-

A] Need Assessment study for Face mask Detection.

B] Design and Development of Software and Simulation of face mask detection

C] Development Of Hardware and Implementation of model.

D] System Evaluation of Face Mask Detection

CHAPTER 4

DESIGN METHODOLOGY

4.1 Proposed System Architecture

This system will help identify people on image/ video stream wearing a facemask with the help of Deep Learning and Computer Vision algorithms by using various libraries such as OpenCV, Keras, TensorFlow etc. The images are downloaded from various open source websites and are differentiated as “mask” and “no mask”. The images that we downloaded were of different sizes and different resolutions.

Resizing the input image :

- Applying the color filtering (RGB) over the channels (Our model MobileNetV2 supports 2D 3 channel image)
- Scaling / Normalizing images using the standard mean of PyTorch build in weights
- Center cropping the image with the pixel value of 224x224x3
- Finally Converting them into tensors (Similar to NumPy array)

We trained the model using tensor-flow retrain which captures the essential differentiating features between the classes of images. The differentiating features are saved in the form of a graph. It is trained once and reused to classify the input images into categories for which it is trained. Later, this trained graph is used by the Image classification algorithm for authentication.

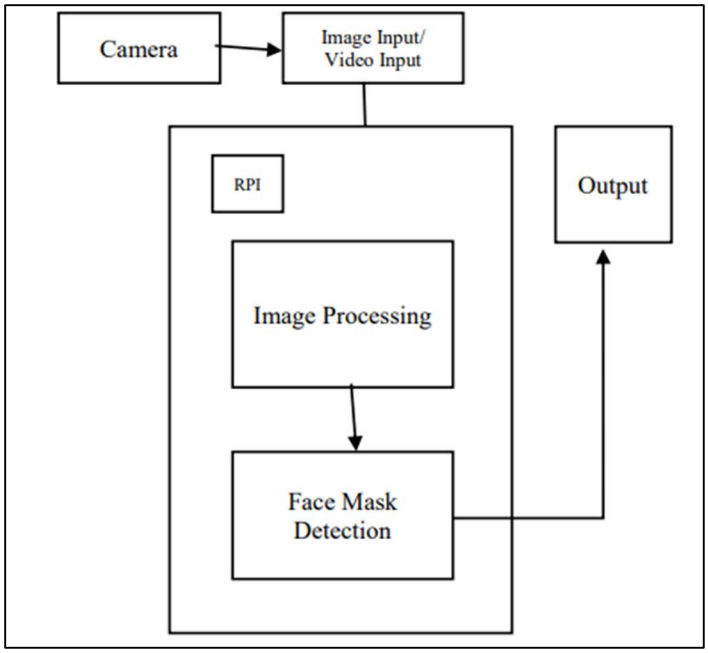

Figure 4.1.1 Face mask Detection Model

Here we have connected raspberry pi 3B+ with 5 mp pi cam module, HDMI Cable, keyboard, Mouse, RJ45 Cable.

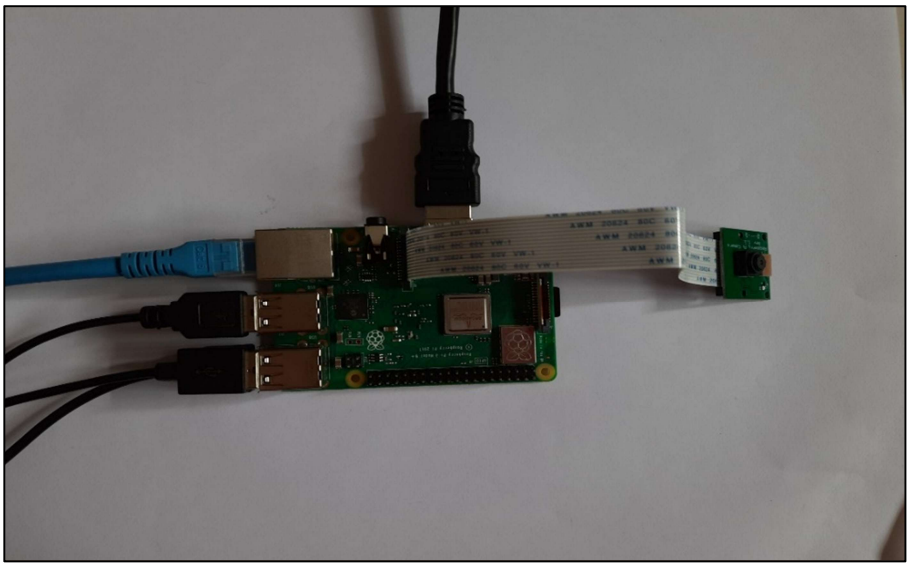

Figure 4.1.2 : System Overview

Here we have simulated Face mask detection System in proteus software. We connect Raspberry pi 3 with LM35[for measuring temperature], LM016L[displaying detection of mask, temperature],L293D, motor, compim, MCP3208.

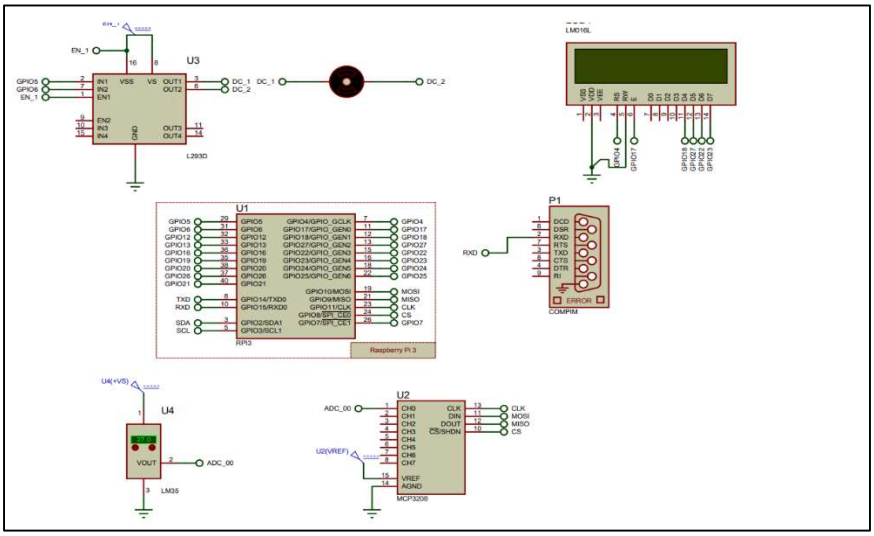

Figure 4.1.3 : Face mask detection simulation in proteus

4.2 Algorithm

Convolution Neural Networks (CNN) Algorithm In this paper, a deep learning algorithm is used to identify face mask recognition and Convolution Neural Networks (CNN) classification. A CNN is a form of artificial neural network that is specifically built to interpret pixel input and is mainly used for image recognition and analysis, in which each layer applies to a different set of filters. Around 100's to 1000's of filters is combined to give a final result and then the obtained output is sent to the next layer in this neural network. Evaluation of the proposed framework is done by the face mask detection algorithm using the TensorFlow software library The Mask detector model is trained by using Keras and TensorFlow. The steps involved in the algorithm is given below

STEP 1: DATASET COLLECTION

STEP 2: PRE-PROCESSING

STEP 3: SPLITTING

STEP 4: TRAINING STEP

STEP 5: TESTING/EVALUATION

4.3 System Flowchart

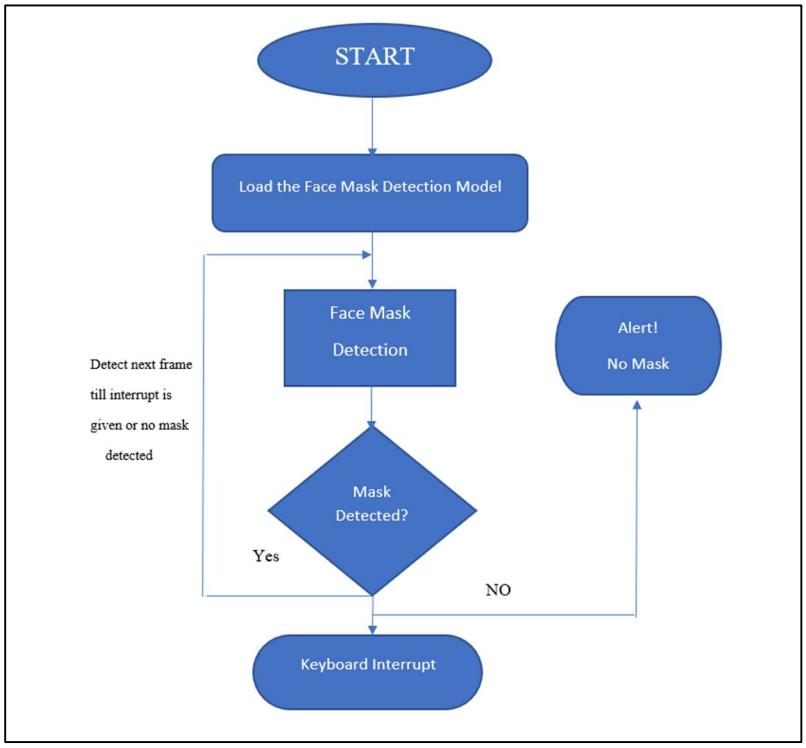

Figure 4.3 flowchart of face mask detection system

4.4 Technical Specifications: Hardware Component used

4.4.1]Raspberry Pi 3B+

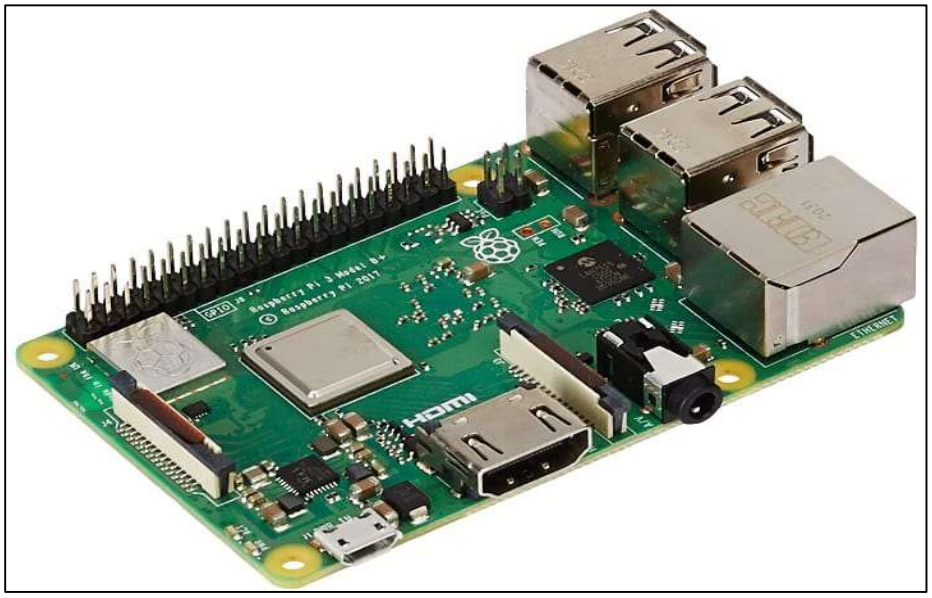

Figure 4.4.1: Raspberry Pi 3B+ Board

The Raspberry Pi is a low-cost tiny computer that connects to a computer monitor or television and operates with a regular keyboard and mouse as shown in Fig 1[8]. It is a handy little gadget that focuses on teaching people of all ages about scripting languages like Scratch and Python. It can perform all the functions of a desktop computer, such as internet surfing and viewing greater-definition clip, worksheets, and playing games. It has been used in several digital devices, including tweeting birdhouses, music machines, and detectors, as well as weather stations and infrared cameras since it is capable of interacting with the outside environment.It has a 1.2-GHz quad-core chipset BCM2387 with a GPU support of a dual-core and a video core multimedia co-processor and the GPU, which includes dual core multimedia co-processor, including a Bluetooth 4,1 (Bluetooth and Bluetooth Classic). With Bluetooth Low Energy (BLE) and BCM43143 Wi-Fi, the Raspberry Pi 3 offers an up-grade towards a new main processor and improved networking. Furthermore, the power management of Raspberry 3 has been improved, with an upgraded power supply with 3.5 Amps that can handle more powerful external USB devices. The built-in USB ports of Raspberry Pi 3 provide sufficient connectivity to link the mouse or anything else to the RPi.Most Raspberry Pi system chips can be overclocked to 800 MHz, and some can be overclocked to 1000 MHz. It is reported that the Raspberry Pi 2 can be similarly overclocked, even reaching 1500 MHz in extreme cases (without all safety features and overvoltage restrictions). On Linux distributions, you can use the program command to run "sudo raspiconfig" to perform boot overclocking without breaking the

warranty. In certain instances, the Pi will automatically deactivate overclocking when the chip temperature reaches 85°C to cancel the automated settings on overclock and overclocking (that will cancel warranty). A cooler can be used to prevent overheating of Raspberry pi.

Specifications & Features

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

PIN SPECIFICATIONS –

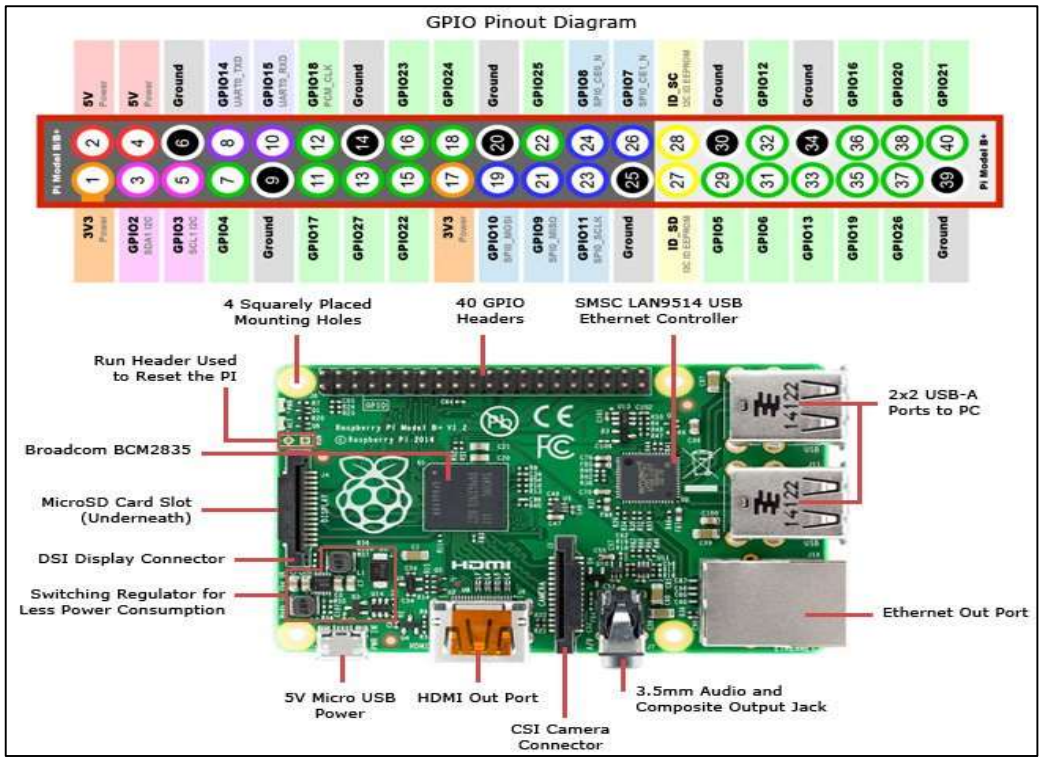

Figure 4.4.1.2: PIN SPECIFICATIONS of Raspberry Pi 3B+

4.4.2 R pi's Cam (Raspberry Pi Camera)

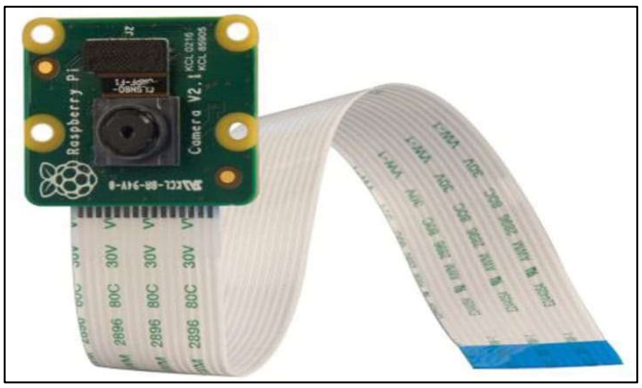

Figure 4.4.2: Pi cam

The **Pi camera module** is a portable light weight camera that supports Raspberry Pi. It communicates with Pi using the MIPI camera serial interface protocol. It is normally used in image processing, machine learning or in surveillance projects. It is commonly used in surveillance drones since the payload of camera is very less. Apart from these modules Pi can also use normal USB webcams that are used along with computer.

The board itself is tiny, at around 25mm x 23mm x 8mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short flexible ribbon cable. The camera connects to the BCM2835 processor on the Pi via the CSI bus, a higher bandwidth link which carries pixel data from the camera back to the processor. This bus travels along the ribbon cable that attaches the camera board to the Pi.

PiCam Features

- 5MP colour camera module without microphone for Raspberry Pi
- Supports both Raspberry Pi Model A and Model B
- MIPI Camera serial interface
- Omnivision 5647 Camera Module
- Resolution: 2592 * 1944
- Supports: 1080p, 720p and 480p
- Light weight and portable (3g only)

4.5 Software used

4.5.1.VISUAL STUDIO CODE:

Visual Studio Code is the part of visual studio family which is developed by Microsoft in November 2015. It is based-on Electron framework which is used for Node.js (node java script). It is written in TypeScript, JavaScript and CSS /15/16/.

Features: Features of Visual Studio Code are as follows:

1. It is open-source and Freeware text editor for private and commercial purposes 2 It is cross-platform source code editor debugger.
2. It supports many different programming languages while only proper installation of the extension is required for React JS, Java, JavaScript, C++, C#, Python etc.
3. GitHub is built-in
4. Products availability as it explains the definition and show opening and closing of the brackets etc.
5. It provided unique customizations 6. Provides Visual Studio Keymap extension for using Key binding
6. It provides in the portable mode that means it keeps data and settings in the same location of installation is possible even on a USB drive
7. It is available in many different language services
8. It is also available in Remote Development mode /15/16/.

4.5.2 PuTTy

PuTTY is an open-source software that provides both serial console and software transfer for network files. . It supports a wide range of network protocols, such as SCP, Telnet, SSH, rlogin, and a crude socket connection which can also be connected to a serial port. PuTTy is a terminal emulator which enables users to access the Raspberry Pi command-line interface from any laptop or desktop device. SSH (secure shell) is used for this, which opens a terminal window on the laptop or device that can be used to send commands to the Raspberry Pi and retrieve data from it before sending it to the computer. Putty itself has the main Ikey file format, known as ppk. Raspberry Pis are commonly used as stand-alone, lightweight network computers. Raspberry Pi is wired to the same local network as the remote computer. On the Raspberry Pi, SSH is enabled which is supported by default in some Raspberry Pi distributions, but it can be configured again later using the Raspi-config tools.

Fig 4.5.2 : Screenshot Of Putty Software

4.5.3 Machine learning (ML)

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. Some implementations of machine learning use data and neural networks in a way that mimics the working of a biological brain. In its application across business problems, machine learning is also referred to as predictive analytics.

4.5.4 Deep learning

Deep Learning Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

4.5.5 PYTHON

Python is an interpreter, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. We have used this software to write our facial detection program and all the necessary below said libraries are installed.

4.5.6 OPENCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. We have used functions available in this library to implement our facial recognition part.

4.5.7 TensorFlow

TensorFlow is a machine learning software library that is open source and free. It was created to perform large numerical computations without regard for deep learning. This TensorFlow can be used for a variety of activities, but it is primarily focused on deep neural network inference and training. TensorFlow also supports traditional machine learning. Google's TensorFlow is a Python library that allows for quick numerical computation. Deep learning models are either generated directly using TensorFlow, which is also a base library, or they are created to simplify the process by using wrapper libraries built on top of TensorFlow. TensorFlow enables the creation of dataflow graphs and structures to determine how the data flows through the graph by receiving inputs as a multi-dimensional tensor array. It allows building a flow chart for these inputs which is carried out on the one end and is performed on the other.

4.5.8 KERAS

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.

4.5.9 VNC Viewer

Virtual Network Computing (VNC) is a graphical desktop sharing application that lets us monitor the desktop interface of one machine with another computer or mobile device remotely. The VNC viewer transmits to the VNC server with a mouse, keyboard, or touch case, receiving updates back on the display. Working directly on the Raspberry Pi is not always convenient. You may also want to include a remote control from another device to work on it. VNC uses Real VNC, which is used with the Raspberry OS. It comprises VNC Viewer, which allows users to remotely access a Raspberry with desktop, and a VNC server enables to monitor the Raspberry Pi remotely. It must be enabled first before using the VNC server. The VNC server provides the users with wireless monitoring to the Raspberry graphical desktop, which enables communication. However, the VNC server can be used to access the graphic remote if the Raspberry is headless and doesn't have a graphic screen.

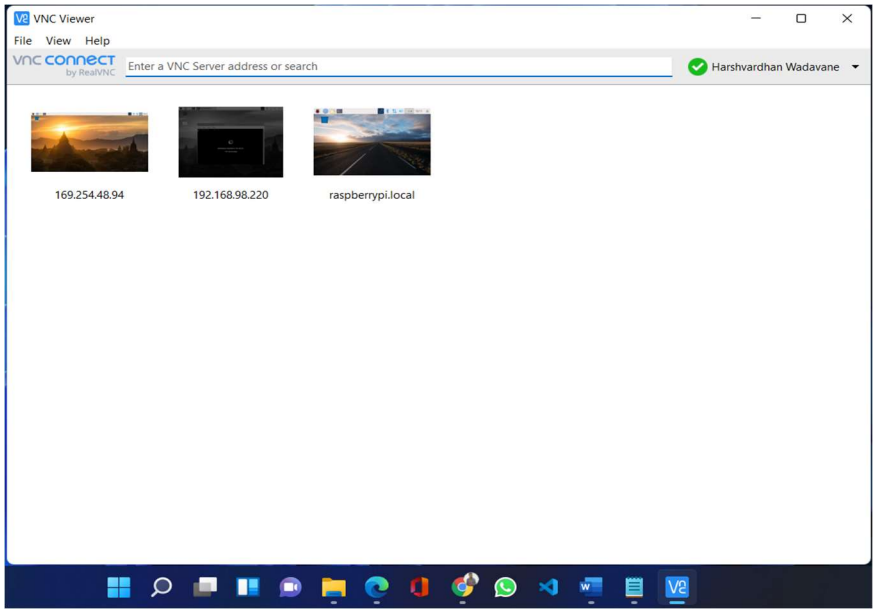

Figure 4.5.9 : Screenshot Of VNC Viewer

4.5.10 Raspberry Pi OS

Raspberry Pi OS (formerly Raspbian) is a Debian-based operating system for Raspberry Pi. Since 2015, it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry Pi family of compact single-board computers. The first version of Raspbian was created by Mike Thompson and Peter Green as an independent project. The initial build was completed in June 2012.

Raspberry Pi OS is highly optimized for the Raspberry Pi line of compact single-board computers with ARM CPUs. It runs on every Raspberry Pi except the Pico microcontroller. Raspberry Pi OS uses a modified LXDE as its desktop environment with the Openbox stacking window manager, along with a unique theme. The distribution is shipped with a copy of the algebra program Wolfram Mathematica and a version of Minecraft called Minecraft: Pi Edition, as well as a lightweight version of the Chromium web browser

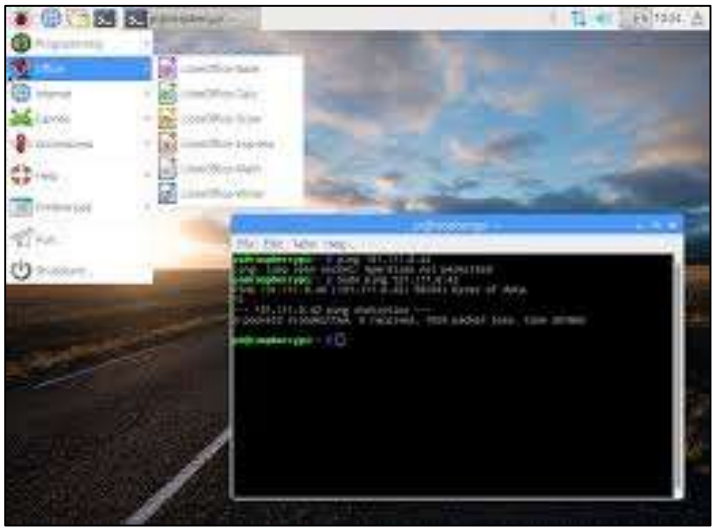

Figure 4.5.10 : Raspberry pi OS

4.6 Project Structure:

In this section, the deployment of this project is discussed. The whole project structure and different components of the project will be explained.

4.6.1 Dataset Collection

The images used for training and testing the model were obtained from the internet. The dataset used in this project was designed by Prajna Bhandary. This dataset contains 3,674 photos divided into two classes: 1915 images with masks and 1759 images without masks . To create this dataset, they took regular photographs of people's faces and then used a custom-designed computer vision Python script to apply face masks to the pictures, yielding an artificial dataset. Facial landmarks allow the users to instantly infer the position of facial components such as the eyes, nose, eyebrows, mouth, and jawline. Then, using facial landmarks, the dataset of faces wearing masks can be created. To determine the bounding box region of a face in an image, start with an image of an individual who's not wearing a face mask and then apply face detection. It can capture the face Region of Interest (ROI) after determining where the face is now in the picture, and then utilize facial landmarks to detect the position of mouth, eyes, nose, and other features. Initially, an image of a mask is required, which will be put to the face automatically utilizing facial landmarks (particularly, the regions around the mouth and chin) to determine where the mask should be placed. After that, the mask is scaled and twisted before being fitted to the face, and the process is repeated for each of the input images, yielding an artificial face mask dataset

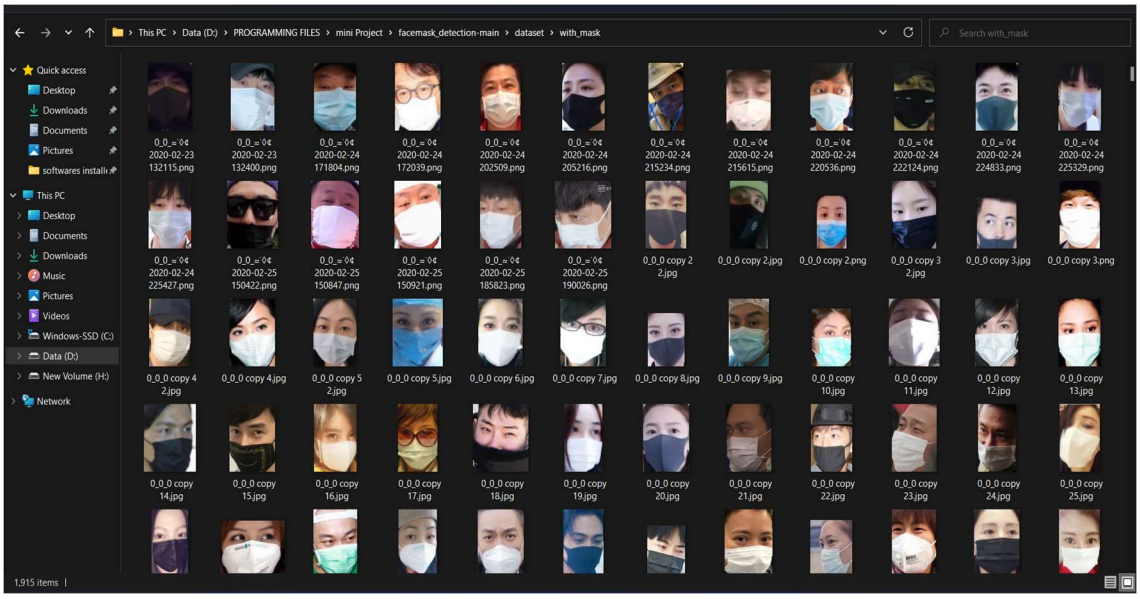

Figure 4.6.1.a : Screenshot of Folder Dataset with mask

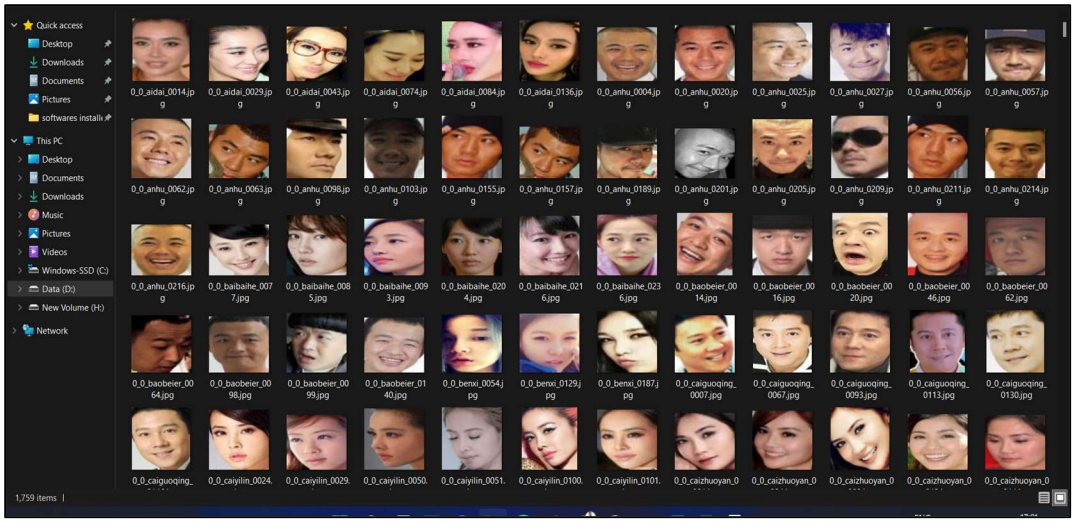

Figure 4.6.1.b: Screenshot of Folder Dataset without mask

4.6.2Train mask Detector:

In this section In VS code we have initialize the initial learning rate, number of epochs to train for and batch size . And then grab the list of images in our dataset directory, then initialize the list of data (i.e., images) and class images. And then construct the training image generator for data augmentation And then we load the MobileNetV2 network, ensuring the head FC layer sets are left off And then we construct the head of the model that will be placed on top of the the base model place the head FC model on top of the base model. then we serialize the model to disk and then we plot the training loss and accuracy.

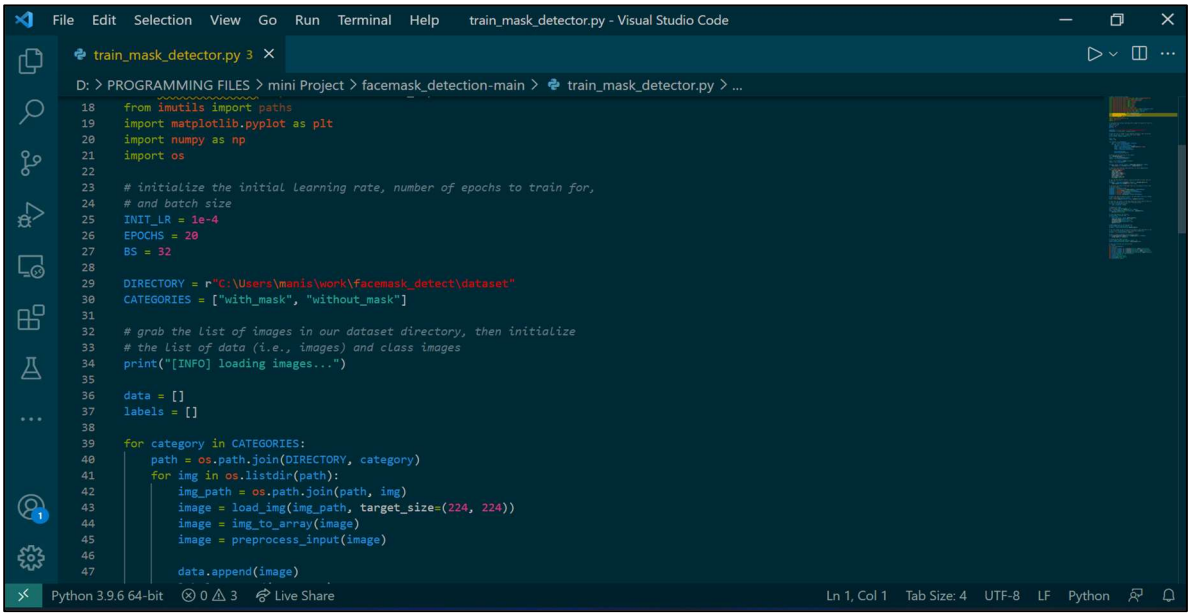

Figure 4.6.2.a: Screenshot Of Train mask Detector.py in VS code

Figure 4.6.2.b: Graph Of Training Loss Of Accuracy

In this section we will discuss about detect.py python file which play important role while execution of this project. First we have created function detect_and_predict_mask. In that it will grab the dimensions of the frame and then construct a blob from it,pass the blob through the network and obtain the face detections extract the confidence (i.e., probability) associated with the detection, extract the face ROI, convert it from BGR to RGB channel.then we load our serialized face detector model from disk, load the face mask detector model from disk.

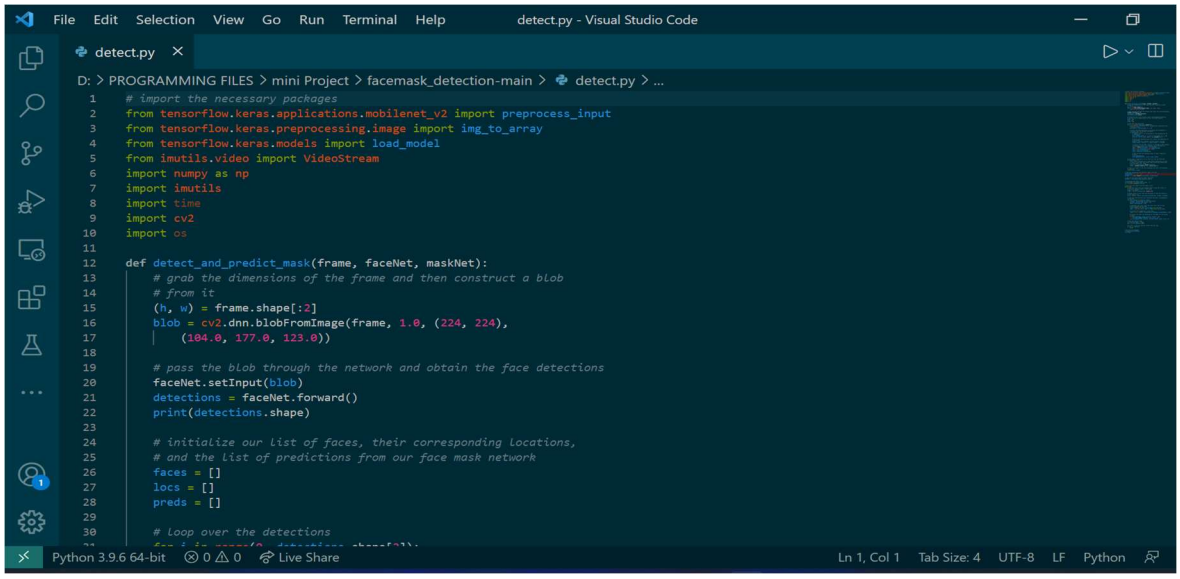

Figure 4.6.2.c : Screenshot Of detect.py In VS code

CHAPTER 5

RESULT AND DISCUSSION

1.1 RESULT AND DISCUSSION

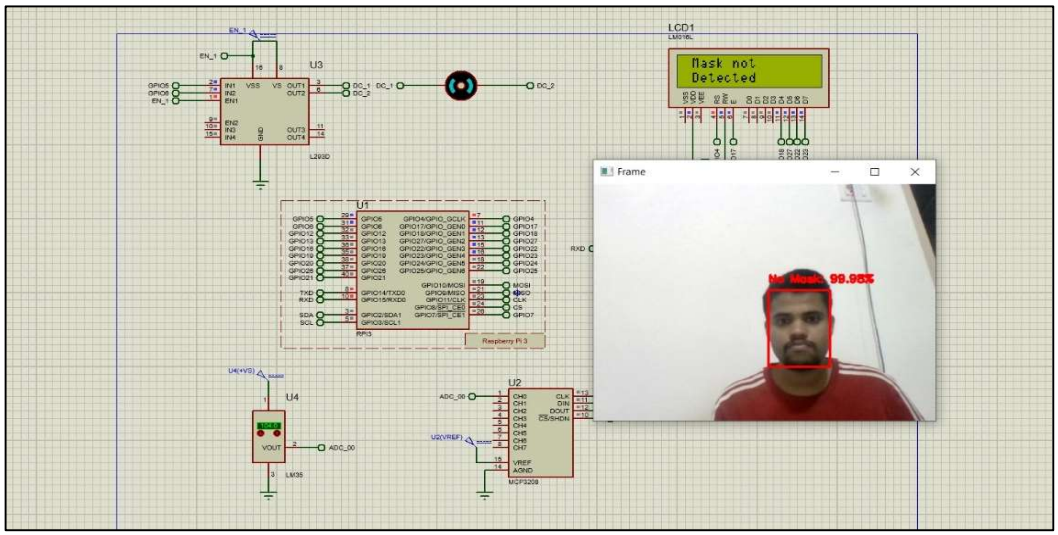

Figure5.1.1: Screenshot of No mask Detected in Proteus Simulation

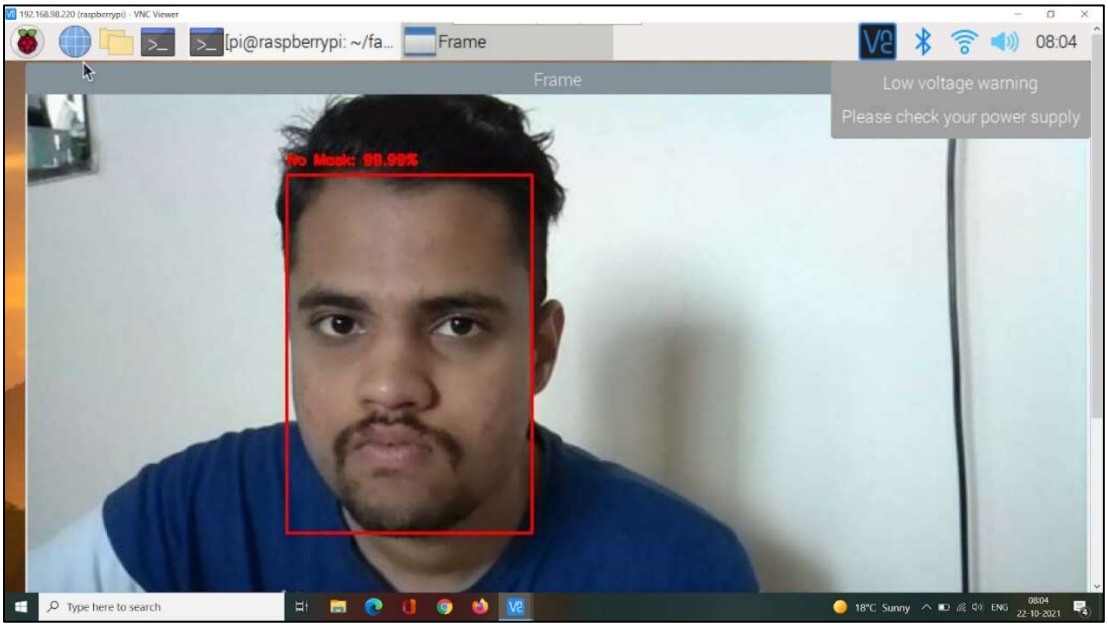

Figure 5.1.2: Screenshot of No mask Detected in Hardware simulation

In both images indicate person has not wear mask. Hence Red rectangle is produced over the face with alerting message ”No Mask Detected ”. If he wear mask, red rectangle will convert into green color rectangle with message “Face Mask Detected

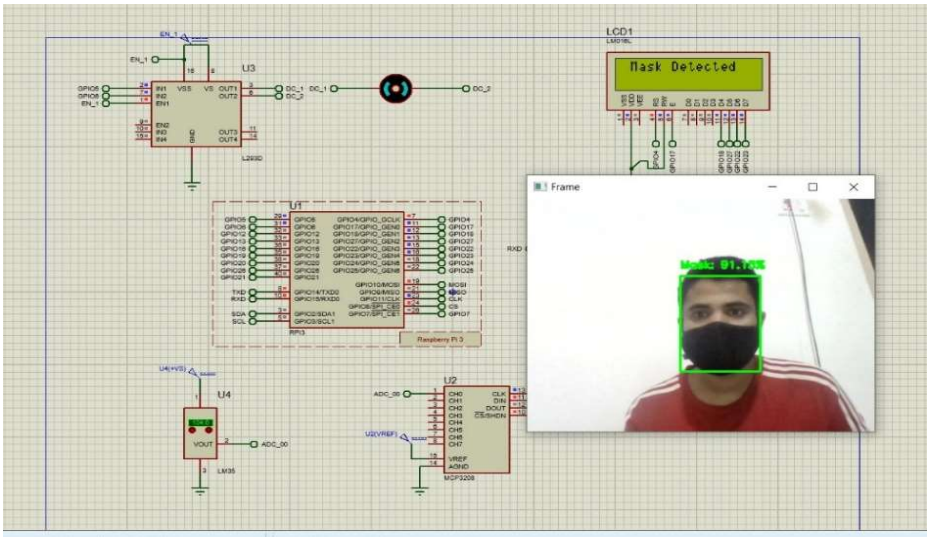

Figure 5.1.3: Face Mask Detected in Proteus Software

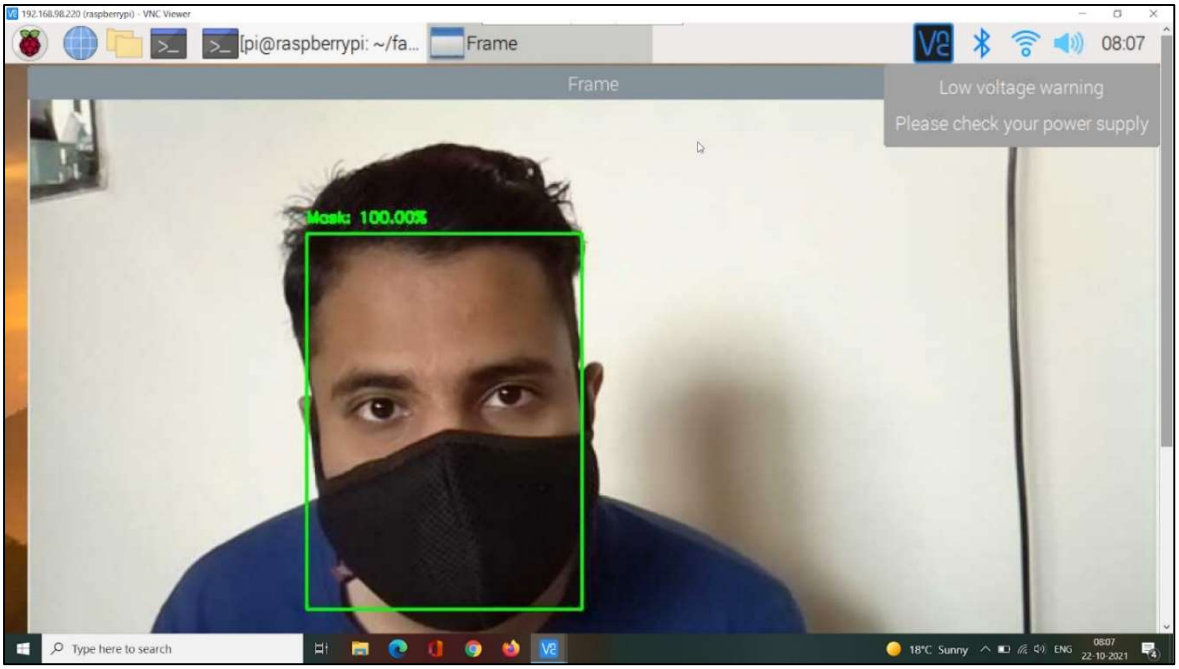

Figure 5.1.4: Face Mask Detected in hardware simulation

In this images person has wear mask, Hence system is detecting that face mask detected with percentages. Also in proteus software , system is also measure temperature using lm35

if and only if he has wear mask. If it is below certain condition it will open gate. If temperature is above certain condition gate will automatically close.

5.2Advantages:

- Intelligent Alerts.
- Facial Recognition.
- Camera Agnostic.
- Easy Implementation.
- Staff Friendly.
- No New Hardware Needed.

5.3 Disadvantages:

Lack of control over their personal information." Critics of mask recognition also think that this new technology could be prone to some of the same pitfalls as facial recognition. Many of the training datasets uses for facial recognition are dominating by light-skinned individuals.

5.4 Application :

- ✓ At the start-up entry/exit point. Relieving sentries from extra tasks.
- ✓ PointsPoints within offices are located properly.
- ✓ Bureaux.
- ✓ Measurement/Institutions.
- ✓ Shopping Center
- ✓ Juice Shops
- ✓ Walking Square
- ✓ Hall. . Audience
- ✓ Seminar rooms.
- ✓ Any location where the mask is required.

CHAPTER 6

CONCLUSION

In this project we have successfully implemented a working prototype of Face Mask detection system. This project can be used in places with large gatherings such as schools, colleges,offices, shopping malls etc.

The system first will detects whether the person is wearing a facemask and sends the data to the microcontroller. With the help of this project an atomized solution is achieved hence there's no need for any human to monitor COVID-19 protocols. Also this System can detect body temperature if we use lm35 sensor. Also we can extend its application like IOT based smart entering System if we use both lm35 and motor with our project.

The accuracy of facemask detection can be achieved by training the module with a larger image dataset. Raspberry Pi 4B or 3B+ has almost the necessary computational power for detecting facemask from image/video stream but with future Raspberry Pi releases, the process can be done with ease. In conclusion, Face Mask can help us to reduce the large gathering of people in one place without masks, reducing the risk of getting infected.

REFERENCES

[1] S. Joshi, S. S. Joshi, G. Kanahasabai, R. Kapil and S. Gupta, "Deep Learning Framework to Detect Face Masks from Video Footage," 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, pp. 435-440, doi: 10.1109/CICN49253.2020.9242625.

[2] X. Fan, M. Jiang and H. Yan, "A Deep Learning Based Light-Weight Face Mask Detector With Residual Context Attention and Gaussian Heatmap to Fight Against COVID-19," in IEEE Access, vol. 9, pp. 96964-96974, 2021, doi: 10.1109/ACCESS.2021.3095191.

[3] J. Zhang, F. Han, Y. Chun and W. Chen, "A Novel Detection Framework About Conditions of Wearing Face Mask for Helping Control the Spread of COVID-19," in IEEE Access, vol. 9, pp. 42975-42984, 2021, doi: 10.1109/ACCESS.2021.3066538.

[4] T. M. Hoang, G. P. Nam, J. Cho and I. -J. Kim, "DEFace: Deep Efficient Face Network for Small Scale Variations," in IEEE Access, vol. 8, pp. 142423-142433, 2020, doi: 10.1109/ACCESS.2020.3012660.

[5] L. Aziz, M. S. B. Haji Salam, U. U. Sheikh and S. Ayub, "Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review," in IEEE Access, vol. 8, pp. 170461-170495, 2020, doi: 10.1109/ACCESS.2020.3021508.

[6] B Varshini , HR Yogesh , Syed Danish Pasha , Maaz Suhail , V Madhumitha , Archana Sasi , IoT-Enabled Smart Doors for Monitoring Body Temperature and Face Mask Detection, Global Transitions Proceedings (2021), doi: https://doi.org/10.1016/j.gltp.2021.08.071

[7] S. V. Militante and N. V. Dionisio, "Real-Time Facemask Recognition with Alarm System using Deep Learning," 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC), 2020, pp. 106-110, doi: 10.1109/ICSGRC49013.2020.9232610.

[8] H. Li, Z. Lin, X. Shen, J. Brandt and G. Hua, "A convolutional neural network cascade for face detection," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5325-5334, doi: 10.1109/CVPR.2015.7299170.