

FACE MASK DETECTION AND SOCIAL DISTANCE MONITORING SYSTEM

Submitted in partial fulfilment of the requirements for the award of Bachelor of
Technology degree in Biomedical Engineering.

By

Kowsalya V



**DEPARTMENT OF BIOMEDICAL ENGINEERING
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Kowsalya V (37240046)** who have done the project work as a team who carried out the project entitled "**Face Mask Detection and Social Distance Monitoring System**" under my supervision from **September 2020** to **March 2021**.

Ms. SINDU DIVAKARAN, M.E., (Ph.D.,)
Internal Guide

Dr.T. SUDHAKAR., M.Sc., Ph.D.
Head of the Department

Submitted for Viva voce Examination held on 09.04.2021

Internal Examiner

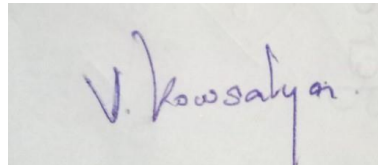
External Examiner

DECLARATION

I, **KOWSALYA V** (Reg.No.**37240046**) hereby declare that the Project report entitled “**Face Mask Detection and Social Distance Monitoring System**” done by us under the guidance of **Ms. SINDU DIVAKARAN**, M.E., (Ph.D.,) is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology in Biomedical Engineering.

DATE: 09.04.2021

PLACE: CHENNAI

A rectangular box containing a handwritten signature in blue ink. The signature appears to be 'V. Kowsalya' with a small dot at the end.

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to and **Dr. Premkumar M.Sc., Ph.D., Dr.T. Sudhakar., M.Sc., Ph.D., Head of the Department, Dept. Biomedical Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of` gratitude to our Project Guide **Ms. Sindu Divakaran, M.E., (Ph. D)., Department of Biomedical Engineering** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all **Teaching and Non-teaching** staff members of the **Department of Biomedical Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Changes in the lifestyle of everyone around the world is 'No entry without mask'. In those changes, wearing a mask has played a vital role, in every individual's life. face mask detection is a challenging task, it has receiving more and more attention due to the era of spreading of corona virus diseases. For the COVID-19 prevention detection of Face mask is very important issue in security purpose, this project can be used in Airports, Hospitals, Offices and Educational Departments etc.as a digitalized scanning tool. The method of detecting people's face and divided them into two classes namely the people mask and no masks is done with the help of image processing(python) and deep learning(Amrit Kumar Bhadani el.at 2020). We introduce the method of faster R-CNN to classify the images. A face mask detection dataset consists of with mask and without mask images, we are going to use OpenCV to real time face mask detection from an image via our webcam, by using a mobile network V2, our goal is to identify whether the person in image/ video stream is wearing a face mask or not with the help of computer vision and deep learning. We have done a comparison between the R-CNN and mobile network V2 to find the accuracy values. Social distancing of 2 meter has been recommended as one of the best solutions to stop the spread of COVID – 19(Gisella Ann Alcoran Alvarez el.at 2020). This done with the help of Arduino uno using temperature and ultrasonic sensors. By using this method, we can monitor the temperature and distance between the people. This system is helpful to the society.

Keywords: Deep learning, Open CV, Arduino Uno, Ultrasonic sensors, Mobile network V2, image processing.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	v
	LIST OF ABBREVIATIONS	vii
1	INTRODUCTION	1
1.1	Outline of The Project	1
1.2	Computer Vision	2
1.3	Deep Learning	2
1.4	Machine Learning	2
1.5	Open CV	3
1.6	Tensor flow	3
1.7	Keras	3
2	LITERATURE SURVEY	4
3	AIM AND SCOPE	8
3.1	Aim	8
3.2	Existing Method	9
3.2.1	Disadvantage of Raspberry Pi	9
3.3	Proposed Method	10
3.4	Scope	11

4	METHODOLOGY AND MATERIALS	12
4.1	CNN	12
4.2	Architecture diagram	12
4.2.1	Explanation of architecture diagram	13
4.3	R-CNN	14
4.3.1	Convolutional layer	14
4.3.2	Stride & Padding	14
4.3.3	Feature map	14
4.3.4	RPN	15
4.3.5	ROI pooling layer	15
4.3.6	Multi class classification	16
4.3.7	Bounding box regression	16
4.3.8	InceptionV2network	17
4.3.9	Mobilenetv2	17
4.4	Dataset with/without mask	18
4.4.1	Explanation of dataset	19
4.4.2	Explanation of flow diagram	20
4.4.3	Python	21
4.5	Block diagram – monitoring	22
4.5.1	Arduino uno	23
4.5.2	Temperature sensor	25
4.5.3	Ultrasonic sensor	25
4.5.4	LCD	26
4.5.5	Buzzer	27

5	RESULT & DISCUSSION	28
5.1	Result	28
5.1.1	Graphical Representation of loss & accuracy	30
5.2	Discussion	33
5.2.1	Tabular column of the network	32
5.2.2	SDMTD -Prototype model	36
6	SUMMARY & CONCLUSION	37
6.1	Summary	37
6.2	Conclusion	37
	REFERENCES	38
	APPENDIX	41
	Hardware program – Arduino	41
	Software program – Python	43

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	Block Diagram FMD	12
4.2	Block diagram faster R-CNN	13
4.3	Block diagram RPN	15
4.4	Inception V2 network	18
4.5	Dataset with/without mask	19
4.6	Flow diagram	20
4.7	Software image	21
4.8	Block diagram [monitoring]	22
4.9	Arduino uno	24
4.10	Temperature sensor	24
4.11	ultrasonic sensor	25
4.12	LCD	26
4.13	Buzzer	27
5.1.1	screenshots images without mask	28
5.1.2	screenshots images with mask	29
5.1.3	screenshots images without mask	29
5.1.4	screenshots images with mask	29

5.1.5	Complication screening fmd	30
5.1.6	Model accuracy graph fmd	30
5.1.7	Complication screening fmd	31
5.1.8	Face mask classifier in metrices	31
5.1.9	Model accuracy graph fmd	34
5.2.1	Prototype model	36

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CNN	Convolutional Neural Network
DWT	Discreate Wavelet Transform
FMD	Face Mask Detection
LBP	Local Binary Pattern
PCA	Principal Component Analysis
R-CNN	Regional Convolutional Network
RGB	Red Green Blue
ROI	Region Pooling Interest
SDMTD	Social distance Monitoring temperature & Distance.

CHAPTER 1

INTRODUCTION

1.1 OUTLINE OF THE PROJECT

As suggested medical practitioners, it is necessity to wear a mask whenever you are at public places or places with high probability of people gathering, to prevent spread of virus. Before COVID – 19 people used to wear mask to protect their health from air pollution. Scientists proofed that wearing a face masks works on impending Covid-19 is that the last epidemic virus that hit the human health within the last century. In 2020, the fast spreading of Covid-19 has forced the who to declare Covid-19 as international pandemic. Artificial Intelligence based on machine learning and deep learning will facilitate to fight Covid-19 in several ways. The provision of healthcare needs finding for emerging technology such as artificial intelligence, big data and machine learning to tackle and predict new diseases. In order to better understand infection rates and to trace quickly detect infections. People are forced by laws to wear face mask in public in many countries, these rules and laws were developed to prevent the spread of infection and keep the person from getting any airborne infectious germs. At the point when somebody coughs, talks, sneezes they could discharge germs into the air that many infect others nearby in many areas. However, the process of monitoring the large groups of people is becoming more difficult. The monitoring process involves the detection anyone who is not wearing a face mask. Here we introduce a face mask detection with the help of image processing techniques and deep learning method. The method is used for deep learning and classical machine learning techniques with the help of the OpenCV, tensor flow and keras (Vinitha el.at 2020) For feature extractions we introduce the Faster R-CNN are the suitable algorithm that achieved the highest accuracy and consumed the least time in the process of training and detection (Mingjie Jiang el.at 2020). Social distancing of 2 meter has been recommend by World health organization (WHO) is one of the best solutions to stop the spreading of COVID -19. Prevention demands the wearing mask and maintaining the 2-meter social distancing between the individuals as the virus can spread through saliva and through human contact Simply, we can monitor the people by using the temperature sensor and ultrasonic sensor in every public entrance.

1.2 COMPUTER VISION

Image processing is a sub set of computer vision. An Image processing and deep learning we use the algorithms we use computer vision to perform the emulation of vision at human scale (Othman N.A el.at 2017). In this project we have include for processing and analyzing understanding image in general a very high dimensional data from the real world in order to produce a numerical and symbolic information or it is technology of science and machine that see it obtain information from images. It is a digital image or videos used in computer vision having an automatic extraction, analysis and understanding of useful information from a single image and also it involves the theoretics and algorithmic basic to achieve an automatic visual understanding.

1.3 DEEP LEARNING

Deep learning is a subset of artificial intelligences and machine learning and uses multi layered artificial neural networks to deliver state of the art and accuracy value, in this we have used for detection and training method. It differs from traditional machine learning techniques it that they can automatically learn representations from data such as images, video or text (N. Ud Din el.at 2020). A machine learning algorithm is the multiple layers to progressively extract higher – level feature from the raw input. In image processing and deep learning techniques may identify the concepts relevant to a human such as a digital image.

1.4 MACHINE LEARNING

The machine learning algorithm also compare the output with the correct, to find the errors to modify the model accordingly. Machine learning algorithm can be applying and learned the new data in order to give the predict for future events, the system is able to analysis and train the dataset to make the predications and accuracy and output values, the system is able provides targets and gives an input.

1.5 OPENCV

In deep learning, and image processing OpenCV plays a significant role in real time operations, that is extremely vital in today's system. It is wont to a method pictures, videos and even a live streams ASCII text file laptop vision library could be an immense assortment of algorithms for laptop vision its basic target real time image processing (S. Kumar et.al 2018).

1.6 TENSOR FLOW

It is associate degree open- supply it may bey computer code library used for machine learning and deep learning it may be used across a variety of tasks however it's a specific concentrate on coaching and interference of deep neural network, tensor flow may be a symbolic science library supported dataflow and differentiable programming, Tensor flow may be a mathematical computation library for coaching and building your machine learning and deep learning model an easy to use high level arthropod genus.

1.7 KERAS

Keras is able to define and predict and train the machine learning models and to make predications. Keras is a neural network API. It is library written specifically in python. In addition, it works with other libraries and packages such as tensor flow which makes deep learning easier. It developed the fast-prototyping model for the quick experiment.

CHAPTER 2

LITERATURE REVIEW

Amrit Kumar Bhadani et.al (2020) reported that “face mask detection using image processing” plays a major role in face mask wearing condition and identification method. This paper is ready to classify the two categories of face mask wearing and unmasked faces. These developed by using a YOLOV3 algorithm for face detection, YOLOV3 uses darknet-53 because the backbone, which will automatically remove a mask covering face area and regenerate the image by building the missing hole, this paper presented a system for detecting the presence or absence of a mask or not. They used a deep learning real time face emotion classification and recognition, according to him the face mask detection is speed and efficiency and it gives the 80% of accuracy.

Arti Mahore et.al (2018) ‘the detection of 3D mask in 2D face recognition system’ plays a major role, here the detection based on anti-spoofing it follows the detection and approaches and types such as hardware, software and user collaboration. In hardware method uses texture – base analysis the input RGB image is covered luminance and chrominance parts, DWT is processed these channel effectively, for feature extraction process is carried out by using local binary pattern and support vector machine classifier is analysis the images of real or fake image these will recognize the face and verification of real mask, the image of face mask detection will gives feature point extraction, quantity detection and face mask database and it will gives the result.

Sabbir Ejaz Md et.al (2019) “Implementation of principle analysis on masked and non – masked face recognition”, here analyzed a masked and non – masked the data set used is Olivetti and oracle Research Laboratory (ORL) face database. The steps are used in this work includes facial image acquisition and facial feature extraction using PCA and Eigen vector calculation. As a result, it gives high recognition rate of face without masks. For facial recognition the author describes first we have to detect the face from image, after detecting face, we can recognize the person it is easy for human to detect face but harder for a system to detect face, for this reason author used deep learning to train the system for algorithm to

detect face from an image. For this the author used the mobile V2 network for classification and it will give a very low accuracy and result.

Toshnial Meenpal et.al (2019) reported that “facial mask detection using Semantic segmentation”, here the author classified the classes and labels are named as face or non- face. The convolutional neural network VGG – 16 architecture followed by fully convolutional neural network is used for segmentation. As a result, it recognizes multiple faces. This method is useful for frontal faces as well as non-frontal faces. As a result, it is focused on removal of erroneous prediction. Many new algorithms are being advised using a convolutional architecture. Furthermore, proposed model has also frontal faces. Along with this it is also able to detect multiple facial masks in a single frame. Experiments were performed on multi parsing human dataset obtaining mean pixel level accuracy, this classification is done on by using a mobile network V2 this will give a very low accuracy and results.

Jagadeeswari et.al (2020) the “performance evaluation of intelligent of face mask detection system using various deep learning classifiers” for the face mask detection the author can be analyses with mobile Net V2, ResNet 50, VGG 16, ADAM, SGD. These are the classifiers used for it. For each classifier followed by 3 optimizer and evaluate the performance. The optimizers are used here such as ADAM, ADAGRAD, SGD (Stochastic Gradient Descent). As a result, ADAM optimizer performance is very good and also observed that Mobile Net V2 classifier has best result with high accuracy, the author also compared Residual Network with convolutional neural network architecture which was designed to enable to hundreds or thousands of convolutional layers it is an artificial neural network using an extremely deep CNN composed of 152 layers therefore it gives a very high accuracy and results.

Mingjie Jiang et.al (2020) reported that “the, Retinal face mask detector” here introduces and plays vital role in RESNET and MOBILENET the dataset contained of 7959 images. Resnet is considered as standard backbone, the detection network includes a backbone a neck and head modules, as a result the ResNet accuracy is very much higher than the mobile net. The proposed system is able to

detect faces and showed that the wear mask detection performance was greatly improved with higher accuracy and robustness, Residual Network is a convolutional neural network architecture which was designed to enable to hundreds or thousands of convolutional layers it is an artificial neural network using an extremely deep CNN composed of 152 layers therefore it gives a very high accuracy and results.

Ejaz M.s et.al (2019) the "Component analysis on masked and non-masked face recognition", wearing a mask for face recognition the convolutional neural network is used for the better performance, ConvNet is a class of neural networks that specializes in processing data. It is a deep learning algorithm which can take in an input image, assign importance to various aspects in the image and be able to differentiate one from the other, CNN were designed to map image to an output variable. They have proven so effective that they are the method of any type of prediction problem involving image data as an input the important features without any human supervision, it can learn the key features for each class by itself, CNN is considered to be more powerful than RNN. CNN is a neural network that can use the internal memory to process and it gives the very high accuracy and results.

Vinitha v1 et.al (2020) the author describes the "Covid-19 face mask detection with deep learning and computer vision" it is the aim to introduce the model integration between deep learning and classical machine learning techniques with OpenCV, tensor flow and keras they have used deep transfer learning for the feature extraction and combined it with three classical machine learning algorithm, we introduce a comparison between the them to find the most suitable algorithm achieved the highest accuracy and consumed the least time in the process of training and detect. By using an open CV method, we can classify the real time images of face mask detection for this we use the convolutional network for images and classifying the layer, feature extraction images this will done by mobile network V2 contains the convolution layer with 32 flitters that is connect 53 layers deep it will load millions of images database and trained these are the method we used for classification and get an accuracy value for the result.

Amit chavda, et.al [2020] reported that the multi – stage CNN architecture for face mask detection it is a technology in fields like machine learning and deep learning have been solution for the face mask detection in various areas. Modern computer vision algorithms are approaching for human – level it has proven to be a revolutionary aspect of modern technology in this paper it proposes a two stage CNN architecture first stage we classify the human images, while second stage uses lightweight image classifier to classify the faces detected in the first either it is mask or no mask faces and drawn bounding boxes a round them along with the detected class name, this algorithm was further extended to videos as well. This system can then be integrated with an image or video capturing devices like a CCTV camera, to track safety violations promote the use of face mask, and ensure a safe working environment. RCNN and faster RCNN has been achieved higher accuracy and better results than most single stage detectors.

Kaihan Lin1, et.al [2020] reported that the face mask detection and segmentation based on improved mask R-CNN using deep convolutional neural network have been plays a major role applied for the face detection recently. To overcome this draw back we present a face detection and segmentation method based on improved mask R-CNN which incorporates for face detection and segmentation in one frame work, this includes many images as dataset for the information. We use RESNET to extract features, RPN is used to generate ROIS to extract spatial location. The proposed system is able to detect faces and showed that the wear mask detection performance was greatly improved with higher accuracy and robustness, Residual Network is a convolutional neural network architecture which was designed to enable to hundreds or thousands of convolutional layers it is an artificial neural network using an extremely deep CNN composed of 152 layers therefore it gives a very high accuracy and results.

Preeti Nagrath et.al [2020] reported that the A real time DNN – based face mask detection system here the author is used CNN for the methodology. The proposed system is able to detect faces and showed that the wear mask detection performance was greatly improved with higher accuracy and robustness, Residual Network is a convolutional neural network architecture which was designed to enable to hundreds or thousands of convolutional layers it is an artificial neural network using an extremely deep CNN composed of 152 layers therefore it gives a very high accuracy and results.

CHAPTER 3

AIM & SCOPE

3.1 AIM

The aim of this project to introduce the face mask detection by using a convolutional neural network on computer vision, the model integration between deep learning and classical machine learning techniques by open cv method for monitoring process involves the detection of anyone who is not wearing a face mask (Arthi Mahore et.al 2018). To check for the Individuals and crowds wearing masks in public, use digital screens to remind visitors to wear masks.

To develop a model to detect whether a person is wearing a face mask or not with the help of your webcam.

A major challenge to prevent the COVID – 19 infection is social distancing in public area here we have come up with an automated social distancing prototype model with the help of Arduino uno microcontroller, ultrasonic and temperature sensors, A buzzer alarms when the detected the body temperature and distance is above it will alert the person. Arduino uno runs the sensor and buzzer.

3.2 EXISTING SYSTEM

In the existing system, used microcontroller and sensor in embedding system. Sometimes sensor not sensed properly, it has the limitations like;

- 1) It's very costly depending on the system used, the number of detectors purchased.
- 2) Time consuming is more.
- 3) Lack of qualified professional

The main limitation is that if the image size is smaller than the pixel size, then it cannot be applied efficiently because then one pixel can contain two or more images.

Existing system used for the face mask detection as input parameter to detect the faces of the people. If the people face is closure, the ratio deteriorates from the standard ratio, the people is alerted with the help of a buzzer (Sandler M et.al

2018). For our system, a Pi camera is used to capture the images of the people's face and the entire system is incorporated using Raspberry-Pi.

3.2.1 Disadvantages of Raspberry pi

Even though Raspberry Pi can perform different tasks, there are some limitations due to its hardware. Because of its processor, it cannot run X86 operating systems. Users must not use normal computer standards to judge Raspberry Pi. It can work as a personal computer, but cannot replace it. Graphical processor is missing, missing eMMC internal storage. Since the raspberry pi does n't have any internal storage it requires a micro-SD card to work as an internal requires a micro-SD to work as an internal storage. Even though the leaving of the raspberry pi on all the time it won't harm it at all the raspberry pi is used to designed an embedded device, and run in an enclosed space. Unless you have it something your device should be fine, you are actually far more likely to damage the device by constantly unplugging it. It is a mini computer vulnerable to attacks, computer acknowledges, users of Raspberry pi are due to the security issues is exposed due to the operating system. It does not have any hard disk associated with it for permanent storage pop package on top of the soc, so it's not removable there is no real time monitoring system unable to do any complex multitasking not compatible with other operating system such as windows. Instead of Using the raspberry pi in face mask detection it consumes more time and high cost and the accuracy values are very low so we used the proposed method of deep learning techniques.

3.3 PROPOSED METHOD

In recent years, deep learning has created a significant improvement in face detection and classification tasks for hierarchical feature extraction and feature learning framework. Machine learning, especially deep machine learning architectures have become important matrix for the computer to be intelligent. Deep learning, which is also known as deep structured learning, hierarchical learning or deep machine learning is a convolutional neural network (CNN) based system with multiple hidden layers of units between the input and output layers. Deep learning has successfully used for classification of images (Li, R. wang 2020). Deep learning is a subset of artificial intelligences and machine learning

and uses multi layered artificial neural networks to deliver state of the art and accuracy value, in this we have used for detection and training method. In machine learning techniques it can detect automatically learn and can representations from data such as images, video or text. Deep learning algorithms are used for the multiple layers to progressively extract higher level to the raw input. In image processing it is able to identify the concepts in the digital images.

This paper proposes a deep CNN framework to detect and to label the images specifically face mask detection using a Faster R-CNN model in convolutional neural network. The development of Faster R-CNN model by adding some extra layers to the CNN of Fast R-CNN (Kaihan Lin et al. 2017). architecture which performs regression simultaneously. Convolutional layers extract the features from an image and the generation of region proposals from the convolutional feature map with more than one regional proposal to detect the particular images. Here we have referred so many papers and studied about face mask detection. The techniques are used for mask detection such as video analytic, image semantic segmentation, from finger prints, we use DWT (Discrete Wavelet transform) and LBP (Local Binary Pattern) (Jagadeeswari.C et al. 2020). All of these techniques are help in analyzed for checking a person wear a mask or not and also identify the face recognition of a person.

3.3 SCOPE

This application can be used in any working environment like any public place, station, corporate environment, streets, shopping malls, examination centers, etc., where accuracy and precision are highly desired to serve the purpose. Highly desired to solve the purpose. These techniques can be used in smart city innovation and it would boost up the development process in many developing countries. our analysis of the current circumstances presents a chance to be more ready for the next crisis, or to evaluate the effects of huge scope social change.

CHAPTER 4

METHODOLOGY AND MATERIAL

4.1 CONVOLUTIONAL NEURAL NETWORK(CNN)

A Convolutional neural **CNN** Convolutional Neural Network are designed to process data through multiple layers of arrays. This type of neural networks is used in application like image recognition of face recognition (Gayatri Deora et.al 2016). The neural network in deep learning and other CNN takes input as a two-dimensional array and it operates directly on the images focusing for feature extraction which other neural network focus on network, also known as CNN or Convnet is a class of neural networks that specializes in processing data that has a grid like topology such as an image.

4.2 ARCHITECTURE DIAGRAM

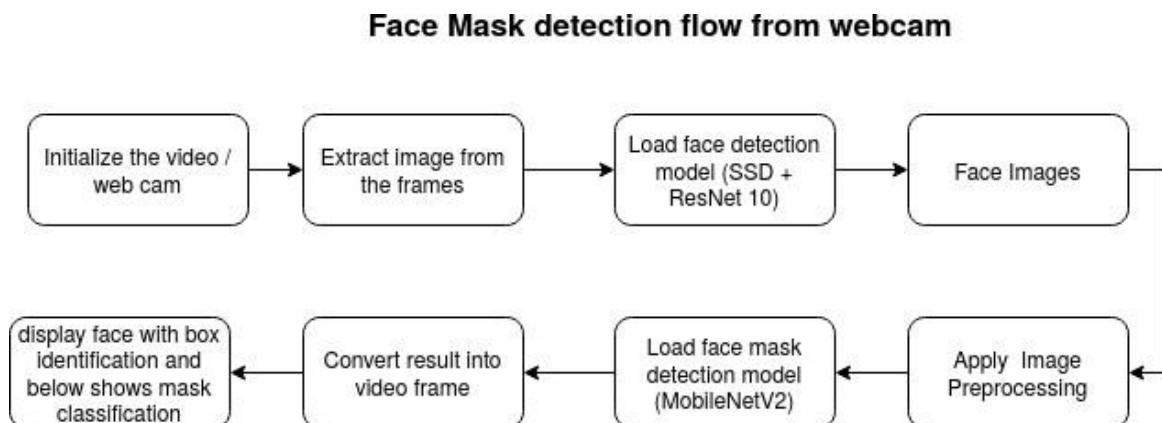


Fig 4.1 Block Diagram of Face mask detection

4.2.1 Explanation of architecture diagram

- As you can see in figure 4.1, we have predefined steps in order to run the face mask detector.
- First of all, we gather our dataset and divide it into two folders. One is with mask and another one is without mask.
- This a way typical way through which the AI technology works. Neural networks can learn through the dataset and start making decisions by themselves.
- Then we have given our dataset to the pytorch' CNN model. This model is serialized the given dataset like in which property it should fall into whether the person is wearing a mask or not.
- After that, we train our model on the given dataset. Once the model is trained, we test it with testing images and videos.

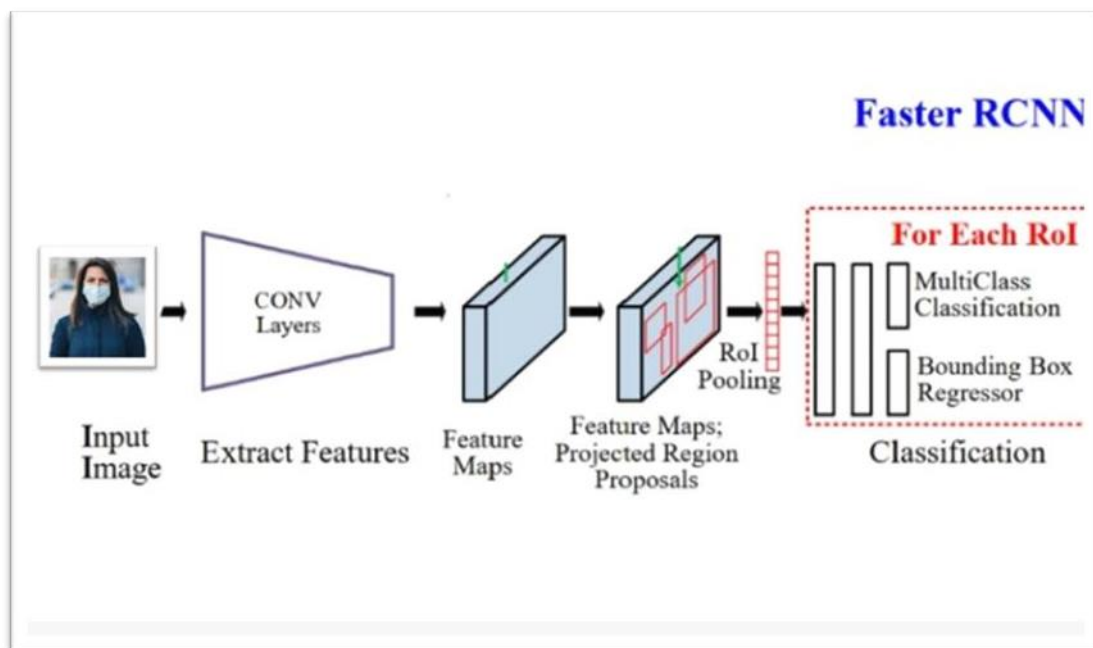


Fig 4.2 Block Diagram Faster R-CNN

4.3 REGION BASED CONVOLUTIONAL NEURAL NETWORK

convolutional A Faster R-CNN object detection network consists of two parts: a fully deep neural network named as RPN is to propose regions of interests and a Fast RCNN detector to classify the regions (Kaihan Lin et.al 2017).

The Faster R-CNN consists of important layers such as:

- ❖ Convolutional Layer
- ❖ Feature Map
- ❖ Regional Proposal Network
- ❖ ROI Pooling Layer

4.3.1 Convolutional layer:

Convolutional layer is the main building block of CNN where set of filters are being used which is useful for detecting the specific feature maps from the input image. It is denoted as a matrix ($N \times N \times 3$) where $N \times N$ denotes the size of the input image. The filter is convolved across the width and height of the input image. There are various filters which are used for the detection of many features that are convolved on the input image and a set of activation functions are computed which is given as the input to the next layer in the CNN.

4.3.2 Stride and padding:

The number of pixels shifts over the input image is known as stride. When the stride is one, then the filter is moved through one pixel at a time. The result after applying stride and filter is the extraction of feature maps from the image. After striding, the output dimension of the image is reduced. If the same dimension has to be retained, then padding should be done before applying filter. Padding is the process of appending zero to the input image.

4.3.3 Feature Map:

The feature map for faster R-CNN is the output to the filter and is applied to the previous layer. Each filter will draw across the entire previous layer, moved one pixel at a time. Each position will result and activation of the output and collected

give as a feature map. There are 6 convolutional kernels and each is used to generate a feature map based on input image. It can use small filters or large filters to compute the output image.

4.3.4 Regional proposal network:

The region proposal network (RPN) within the quicker region-based convolutional neural network (Faster R-CNN) is employed to scale back the procedure necessities of the illation method. The RPN quickly and expeditiously scans each location like as to access whether or not any process must be done out in a very given region. It will that by computing K bounding box proposals every with scores representing likelihood of object or not at every location. In quicker R-CNN They Used $K=$ nine representing three scales and three side ratios. Every regressor within the RPN solely computes four offset values w-width-length (X, Y)- Centre to the corresponding reference anchor box. The RPN uses a 3×3 window that slides over a high- level convolutional feature map, the effective size of that tiny window is really 177×177 once reprojected back to the input layer, that the RPN is really employing a heap of context once creating the proposals This 3×3 window is resampled to a 256 dimensional vector before feeding into 2 totally connected layers, a box regression layer computes the box offsets and also the box classification layer computes the arrogance scores that are associated with method of pictures(Kaihan Lin et.al 2017).

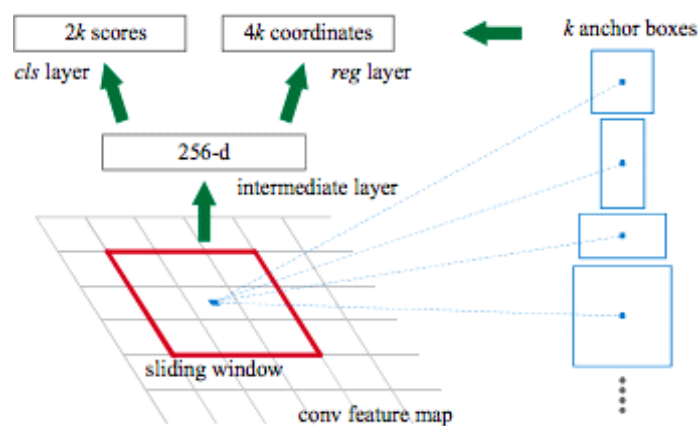


Fig.4.3 Block Diagram Regional Proposal Network

4.3.5 ROI pooling layer: ROI pooling layer produces the fixed-size feature maps from non-uniform inputs by using exploitation pooling on the inputs. The amount of output channels is upto to the amount of input channels for this layer. The most two operations within the pooling layer are:

- * Average pooling
- * Maximum pooling

The maximum pooling will take only the maximum value from the pool and the rest are dropped out whereas in average pooling, the average of all the pool will be taken without any drop out. ROI pooling layer takes two inputs:

- A feature map obtained from a Convolutional Neural Network after multiple convolutions and pooling layers. N proposals or Region of Interests from Region proposal network. Each proposal has five values, the first one indicating the index and the rest of the four are proposal coordinates. Generally, it represents the top-left and bottom-right corner of the proposal.
- ROI pooling takes every ROI from the input and takes a section of input feature map which corresponds to that ROI and converts that feature-map section into a fixed dimension map. The output fixed dimension of the ROI pooling for every ROI depends on the layer parameters.

4.3.6 Multi Classification:

Classification suggests that categorizing information and forming teams supported the similarities. In an exceedingly dataset, the options play a significant role in classifying our information. There are several eventualities that during which within there are a multiple class to which points belong, however a given belong to multiple classes.

4.3.7 Bounding-box regression:

Bounding-box regression may be technique to refine or predict localization boxes in image detection approaches bounding- box regressors area unit trained to regress from either region proposals or fasted anchor boxes to clos bounding boxes of a pre- defined target image categories. The model works by 1st ripping the input images into a grid of cells, wherever every cell is to blame for predicting a bounding box if the Centre of a bunding box if the Centre of a

bounding box falls among it. Every grid cell predicts a bounding box incentre coordinate and therefore the breadth and height of a images.

4.3.8 Inception V2 network:

Inception V2 Module are utilized in convolutional neural networks to permit for additional economical computation and deeper networks through a spatial property reduction with stacked 1*1 convolution. Originating layer may be combination of all 3 layers like 1*1 convolutional layer, 3*3 convolutional layer, 5*5 convolutional layer within their output filter concatenated into one output vector forming the input of subsequent layers.

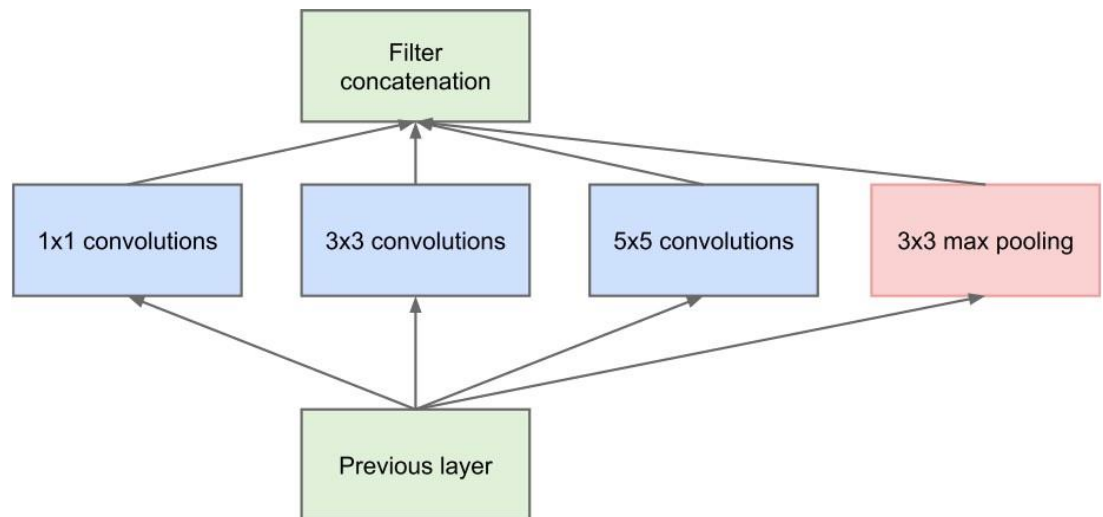


Fig 4.4 Block Diagram Inception V2 network

4.3.9 Mobile network V2: The MobileNet V2 is used for the model for image classification. It is pre – trained on the image net dataset, A typical MobileNetV2 architecture has multiple layers below (MS. Ejaz el.at 2019) In python, we will use the model library in TensorFlow to make a MobileNetV2 model, instead of defining/ building our own model. The weights of each layer indicate the padding, strides, kernel size, input channels and output channels. Select MobileNetV2 because the algorithm for building models which will be deployed on mobile device. A customized fully connected layer was developed that contains for consecutive layers on top of the MobileNetV2 model (Xin M et.al 2019).

The layers are:

1. Average pooling layer with 7x7 weights
2. Linear layer with Reactivation function
3. Dropout layer
4. Linear layer with SoftMax activation function with the results of 2 values.

The final layer SoftMax function gives two result each probability represents the classification of the “mask” or “non-mask”.

4.4 Dataset with mask:



Fig.4.5 Dataset with mask images

4.4. Dataset without mask:

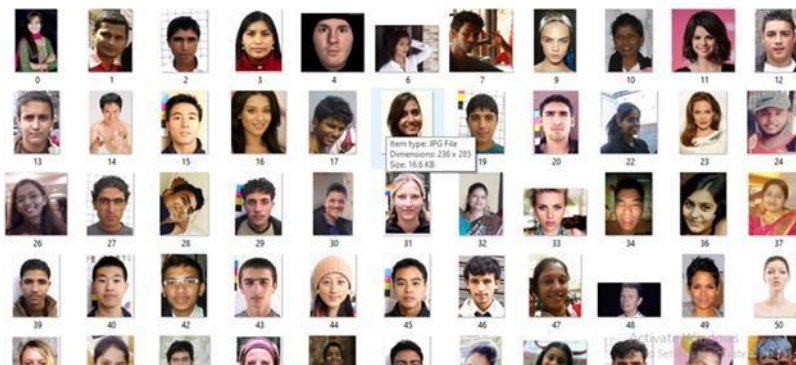


Fig.4.5 Dataset Without mask Images

4.4.1 Explanation of dataset:

- we used RMFD (Real- Time masked face- dataset) dataset to achieve the objective of this face mask detector.
- The dataset consists of total 1,295 images which are divided into 2 categories:
 - *With Mask - Containing 646 images
 - * Without Mask – Containing 649 images
- So, by Having this amount of dataset we tried to train the neural network model to detect whether a person is wearing a mask or not.
- Here the image containing masked and unmasked people, we have taken this image for making the neural model to understand the difference between people with masks and without masks.
- The figure 4.7 masked and unmasked Dataset as you can see, the above figure 4.7 shows the images used for training. we even used multiple images.

flow diagram

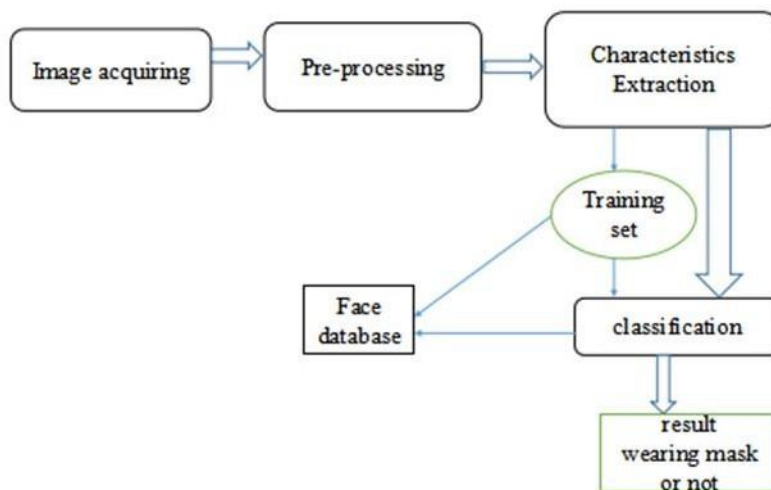


Fig 4.6 Flow diagram

4.4.2 Explanation of flow diagram:

1. Image Acquisition – The face mask detection images are collected from the Kaggle website.

2. Labelling an Images – Collected images are labelled using 'labelling' software which is a python based graphical image annotation tool. Labelling uses (an open-source widget toolkit) to create a graphical user interface (Kumar. S et.al 2018).

3. Pre-processing – Resize of an image means changing the image size without changing the amount of data in that image. Resizing without resampling changes the images physical size without changing the pixel dimensions in the image. No data is added to or removed from the image while resizing.

4. Contrast Enhancement - Image enhancement is that the method of rising the image quality knowledge content of original data before process. Distinction enhancement makes the image options a lot of clearly by creating optimum use of the colours (Sultana et.al 2019). Distinction manipulations involve dynamic the vary of values in a picture so as to extend distinction.

4.4.3 Python:



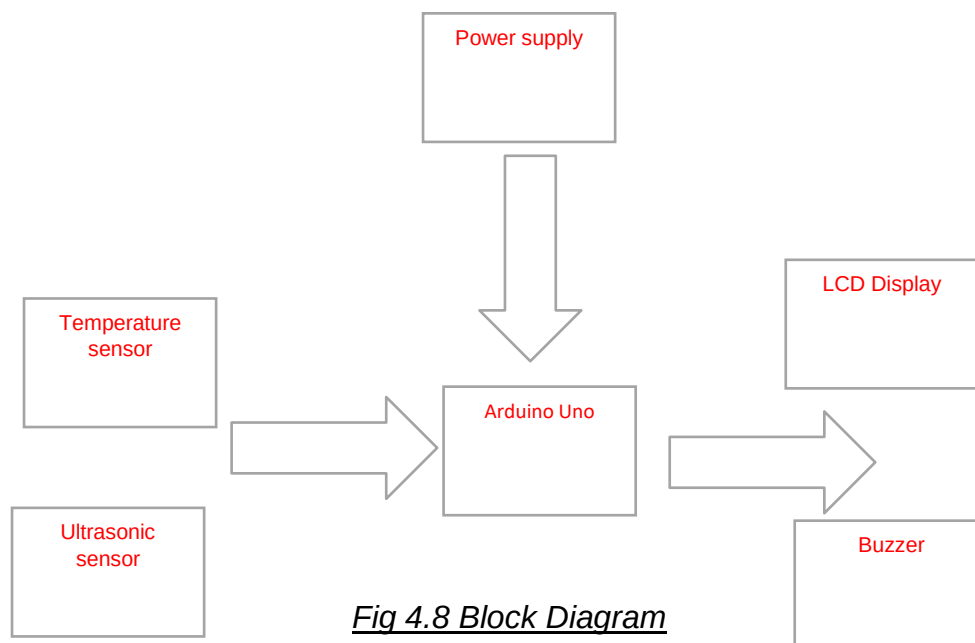
Fig 4.7 software image

Python is associated with object-oriented programming language. Its identity is designed for speedy prototyping of complicated applications. It interfaces to several system calls and libraries and is portable to C or C++, several giant corporations use the python programming language and programming is widely utilized in AI, language generation, neural networks and alternative advanced fields of computing python has deep specialized in code readability and this category can teach you python from basics (Toshnall Meenpal et.al 2019).

Python Programming Characteristics

- It provides rich data types and easier to read syntax than any other programming languages.
- It is a platform independent scripted language with full access to operating system API's
- Compared to other programming languages, it allows more run-time flexibility
- Libraries in Python are cross-platform compatible with Linux, Macintosh, and Windows
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- There are numerous libraries available in Python for scientific and numeric computing. Also, the language is heavily used in machine learning, data mining and deep learning.

4.5 Block diagram (Monitoring):



4.5.1 Arduino uno:

The Arduino Uno may be a microcontroller board grounded on the ATmega328 (datasheet). It contains of 14 digital input/outputs pins (out of that vi will be utilized as PWM outputs), vi analog inputs, a16 megacycle ceramic resonator, a facilitation for USB property, an influence jack, associate ICSP header, and a push. Its styles comprise of assistances that supports the microcontroller in each doable manner so as to induce to figure with it one must merely connect it to a laptop with a USB cable or power it with associate Ac- to- DC adapter or battery. It is accessible in 2 completely different versions particularly Arduino Uno and Genuino that may well be the variations is discovered with relation to the region.

Memory:

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library). Input and Output Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode (), digital Write (), and digital Read () functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 ohms.

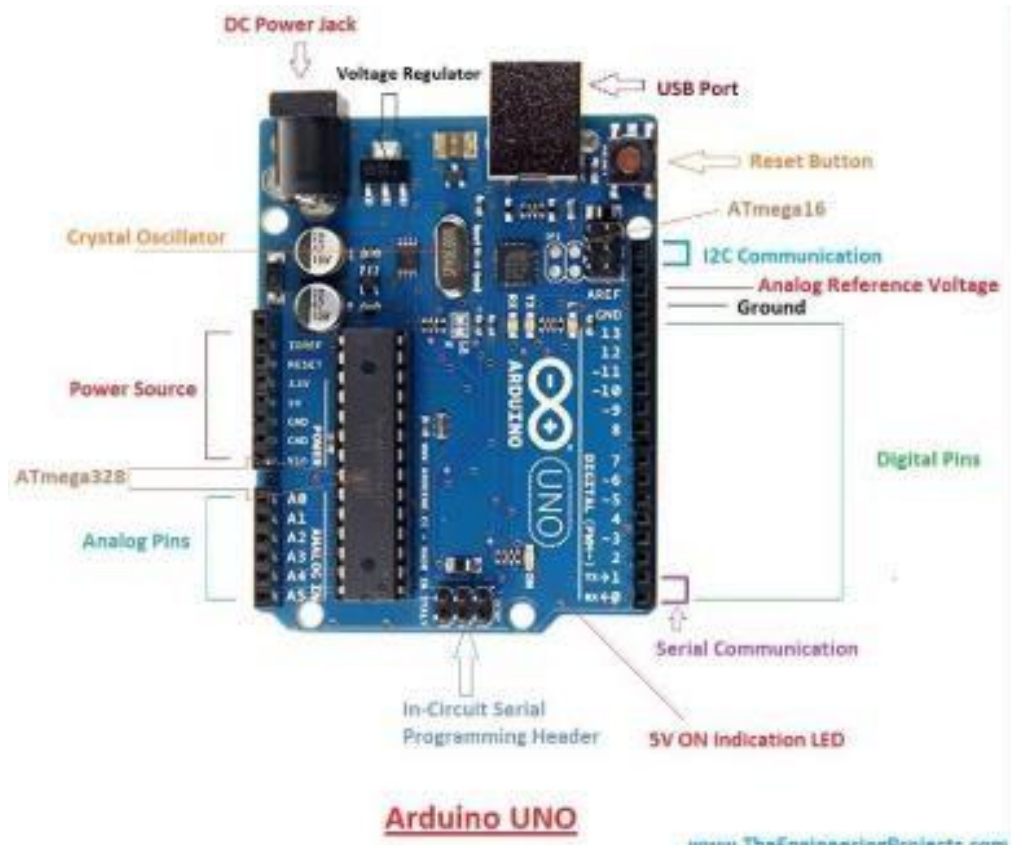


Fig 4.9 Arduino Uno

4.5.2 Temperature Sensor:

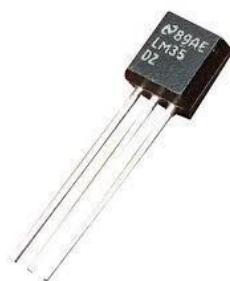


Fig 4.10 Temperature Sensor

A Temperature sensing element is associate device that measures temperature of its surroundings and converts the knowledge [input file, computer file] into electronic data to record, monitor, or signal temperature changes. There are many alternative styles of temperature sensors. Some temperature sensors need direct contact with the object that's being monitored (contact temperature sensors), whereas others indirectly live the temperature of associate object (non- contact temperature sensors). Non- contact temperature sensors are sometimes infrared (IR) sensors. They remotely find the IR energy by associate object and an indication to a tag electronic circuit that determines the object's temperature (Gisella Ann Alcoran Alvarez el.at 2020).

4.5.3 Ultrasonic sensor:



Fig 4.10 Ultrasonic sensor

An unhearable detector is associate degree instrument that measures the gap to associate degree object exploitation unhearable sound waves, associate degree unhearable detector uses an electrical device to send associate degree device receive unhearable pulses that relay back data regarding an object's proximity high frequency sound waves replicate from boundaries to supply vibrates at a frequency higher than the vary of human hearing. Transducer's area unit the microphones accustomed receive and send the unhearable sound. we use for the unhearable sensors, like several others, use one electrical device to send a pulse and to receive the echo. The detector determines the gap to a tonic pulse. ultrasonics area unit thus wide used, they will be faithfully enforced in grain bin sensing, drove applications and water level sensing and sensing cars at your native drive- thru edifice or bank.

4.5.4 LCD Display:

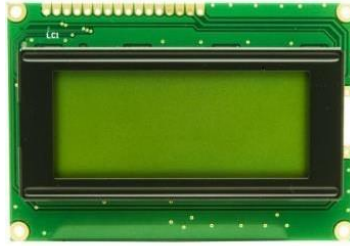


Fig 4.11 LCD

They use constant basic technology, except that capricious pictures square measures created of an oversized range of little pixels, whereas different displays have larger components. LCDs will either be usually on (positive) or off (negative), looking on the polarizer arrangement, for instance, a personality positive digital display with a backlight can have black writing on a background that's the color of the backlight, and a personality negative digital display can have a black background with the letters being of constant color because the backlight, optical filters square measure other to white on blue LCDs to offer them their charteristics look. The principle behind the LCD's is that once associate in electrical current is applied to the liquid molecules, this cause the angle of sunshine that is passing through the molecule of the polarized glass and conjointly cause a modification within the angle of the highest polarizing filter. As a result, a touch lightweight is allowed to pass the polarized glass through a specific space of the digital display. Thus, that specific space can become dark compared to different. The digital display works on the principle of block lightweight, whereas constructing the LCD's a mirrored mirror is organized at the rear. The entire region of the digital display should be coarctated by a standard conductor and higher than it ought to be the liquid matter.

4.5.5 Buzzer:

A buzzer or electronic device is associated degree audio device, which can be mechanical, mechanical devices, or electricity. Typical uses of buzzers and beepers embody alarm devices, timers and confirmation of user input like click or keystroke.



Fig.4.12 Buzzer

CHAPTER 5

RESULT AND DISCUSSION

5.1 RESULT

We have used around 1,295 images they are divided into training set and testing set these images consists of with mask and without mask. Training testing spilt is 80%, 20% respectively. Then these images are trained to the R-CNN model, we have used different convolution, max pool and flatten, dropout, fully connected layers. We have used adam as optimization function and cross entropy as loss function network was trained in 10 epochs and achieved 80% accuracy. This model can able detect single person in an image into two classes with mask indicates green colour and without mask indicates red colour effectively around 80% accurately. we have used cap = cv2. image capture (0) to capture images from web cam and this cascade classifier for object detection from the live web cam in the below fig 5.1.1,5.1.2 for detecting the face mask.



Fig 5.1.1 screenshots of without mask images



Fig 5.1.2 screenshots of with mask images

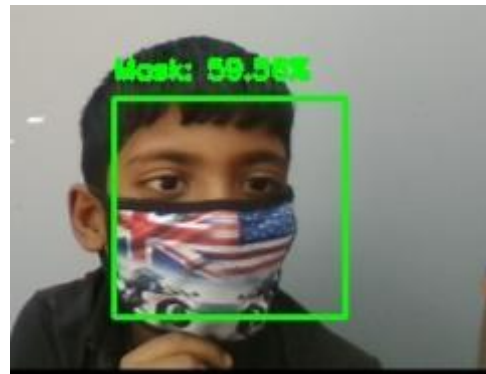


Fig 5.1.3 screenshots of with mask images



Fig 5.1.4 screenshots of without mask images

As, we identify the faces a pre- trained Model provided by the OpenCV method the frame work was used. The model was trained using the web images. OpenCV provides 2 Models for this detector Floating – point 16 version of the original caffe implementation. 2.8 bit quantized 16 version using tensor flow the caffe model in this face mask detector, there has been a lot of discussion around deep learning based approaches for person detection. This encouraged us to come up with our own algorithm to solve this problem. Our work on face mask detection comprises of data collection to tackle the variance in kinds of face masks in the above figures 5.1.3, 5.1.4 detection of face masks by mobileNetV2.The face mask detection model is combination of face detection model to identify the existing faces from camera feeds and then running those faces through a mask detection model.

5.1.1 Graphical Representation of Loss and Accuracy Values

These are trained by RPN by using the RPN the time is very less to find the accuracy values when compared to MobileNetV2.

```
Found 1295 images belonging to 2 classes.
Found 313 images belonging to 2 classes.

[13]: history = model.fit_generator(train_generator,
                                   epochs=3,
                                   validation_data=validation_generator,
                                   callbacks=[checkpoint])

Epoch 1/3
20/130 [====>.....] - ETA: 1:27 - loss: 0.9135 - acc: 0.4750
C:\Users\ARCHANA\Anaconda3\envs\tens\lib\site-packages\PIL\Image.py:963: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
  "Palette images with Transparency expressed in bytes should be "
129/130 [=====>.] - ETA: 0s - loss: 0.6982 - acc: 0.5946Epoch 1/3
130/130 [=====] - 113s 870ms/step - loss: 0.6973 - acc: 0.5961 - val_loss: 0.4058 - val_acc: 0.8211
Epoch 2/3
129/130 [=====>.] - ETA: 1s - loss: 0.4653 - acc: 0.7969Epoch 1/3
130/130 [=====] - 139s 1s/step - loss: 0.4631 - acc: 0.7977 - val_loss: 0.2198 - val_acc: 0.9201
Epoch 3/3
129/130 [=====>.] - ETA: 0s - loss: 0.4461 - acc: 0.8054Epoch 1/3
130/130 [=====] - 118s 909ms/step - loss: 0.4479 - acc: 0.8046 - val_loss: 0.2262 - val_acc: 0.9169
```

FIG 5.1.5 compilation screen for training script of facemask detection.

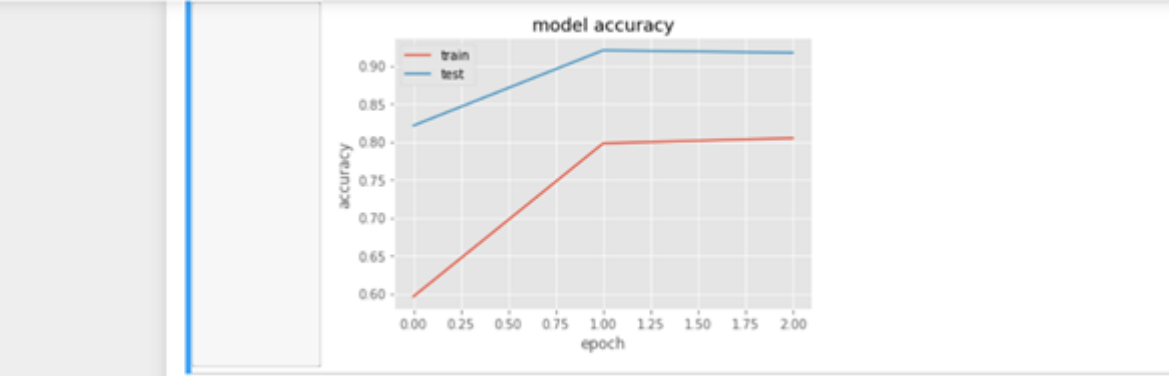


FIG 5.1.6 Model accuracy graph of facemask detection.

The graph shown in the above figure 5.2.2 overcome the accuracy results, we can see there are little signs of overfitting, with less the training loss. We have achieved around 80% accuracy in this face mask detection with very little training loss. There is no exaggeration in mentioning that this face mask detector can effectively detect the persons without masks anywhere and anytime.

These are trained by MobileNetV2 by using this network the time consuming is more when compared to RPN this will gives the accuracy values is 85%.

```

23/24 [=====>..] - ETA: 7s - loss: 0.7328 - acc: 0.6036 Epoch 1/10
199/24 [=====]
==] - 17s 84ms/sample - loss: 0.6385 - acc: 0.6935
24/24 [=====] - 204s 9s/step - loss: 0.7249 - acc: 0.6097 - val_loss: 0.6025 - val_acc: 0.6935
Epoch 2/10
23/24 [=====>..] - ETA: 7s - loss: 0.4893 - acc: 0.7737 Epoch 1/10
199/24 [=====]
==] - 15s 77ms/sample - loss: 0.4801 - acc: 0.7990
24/24 [=====] - 202s 8s/step - loss: 0.4864 - acc: 0.7792 - val_loss: 0.4742 - val_acc: 0.7990
Epoch 3/10
23/24 [=====>..] - ETA: 8s - loss: 0.3935 - acc: 0.8368 Epoch 1/10
199/24 [=====]
==] - 15s 77ms/sample - loss: 0.4059 - acc: 0.8291
24/24 [=====] - 210s 9s/step - loss: 0.3894 - acc: 0.8397 - val_loss: 0.4100 - val_acc: 0.8291
Epoch 4/10
23/24 [=====>..] - ETA: 8s - loss: 0.3146 - acc: 0.8764 Epoch 1/10
199/24 [=====]
==] - 18s 90ms/sample - loss: 0.3500 - acc: 0.8392
24/24 [=====] - 221s 9s/step - loss: 0.3133 - acc: 0.8763 - val_loss: 0.3679 - val_acc: 0.8392
Epoch 5/10
23/24 [=====>..] - ETA: 8s - loss: 0.2550 - acc: 0.9053 Epoch 1/10
199/24 [=====]
==] - 16s 79ms/sample - loss: 0.3750 - acc: 0.8040
24/24 [=====] - 222s 9s/step - loss: 0.2559 - acc: 0.9054 - val_loss: 0.4134 - val_acc: 0.8040
Epoch 6/10
23/24 [=====>..] - ETA: 8s - loss: 0.2730 - acc: 0.8820 Epoch 1/10
199/24 [=====]
==] - 16s 78ms/sample - loss: 0.2159 - acc: 0.8945
24/24 [=====] - 218s 9s/step - loss: 0.2681 - acc: 0.8844 - val_loss: 0.2513 - val_acc: 0.8945
Epoch 7/10
23/24 [=====>..] - ETA: 8s - loss: 0.2362 - acc: 0.9127 Epoch 1/10
199/24 [=====]

```

FIG 5.1.7 compilation screen for training script of facemask detection.

```

24/24 [=====] - 222s 9s/step - loss: 0.1815 - acc: 0.9362 - val_loss: 0.3563 - val_acc:
Epoch 10/10
23/24 [=====>..] - ETA: 8s - loss: 0.1756 - acc: 0.9321 Epoch 1/10
199/24 [=====]
==] - 19s 94ms/sample - loss: 0.2266 - acc: 0.8442
24/24 [=====] - 227s 9s/step - loss: 0.1765 - acc: 0.9310 - val_loss: 0.2997 - val_acc:
[INFO] evaluating network...
      precision    recall  f1-score   support

 withmask         1.00      0.68      0.81         97
 withoutmask       0.77      1.00      0.87        102

   accuracy
 macro avg       0.88      0.84      0.84         199
 weighted avg    0.88      0.84      0.84         199

[INFO] saving mask detector model...

```

FIG 5.1.8 Face Mask Classifier Performance Metrics

Confusion Matrix is commonly used for a summarization of prediction results on a classification problem. The number of correct and incorrect predications is summarized with counting values and each value broken down for each class. It shows the classification model is confused when it makes predictions, at this point in here it gives us insight not only into the errors being made by a classifier but also show the types of errors that are being made. All the experimental trails have been conducted on a laptop equipped by an intel i7-8750H processor (4.1 GHz), 16 GB of RAM with 1050ti max-Q with 4GB of VRAM. The Jupiter notebook software equipped with python 3.8 kernel was selected in this research for the development and implementation of the different experimental trails.

The metrics selected for evaluation of MobileNetV2 model is explained below.

Accuracy

$$= \text{Tp} + \text{Tn} / (\text{Tp} + \text{Fp} + \text{Fn} + \text{Tn})$$

$$\text{Precision} = \text{Tp} / (\text{Tp} + \text{Fn})$$

$$\text{Recall} = \text{Tp} / (\text{Tp} + \text{Fn})$$

f1 score

$$= 2 * \text{Recall} * \text{Precision} / (\text{Recall} + \text{Precision})$$

Where Tp = True positive,

Tn = True negative,

Fp = False positive,

Fn = False negative.

In the above formulas True positive values refer to images which were labelled true and after predication by model gave true result. Likewise, for True negative refers to images which were labelled true but after predication resulted in false result. False positive refers to images which were labelled false and after predication resulted in false hence false positives. False negative refers to images which are labelled false and after prediction resulted in true hence false negatives. The accuracy was a good measure to start with because the classes were balanced. Precision gave the measure of positive predicated values. Recall gave

the ability to classifier to find all positive samples and f1 score gave the measures of test accuracy. These evaluation metrics were chosen because of their ability to give best results in balanced dataset.

From the above face mask classifier performance in metrics are:

(1,1,125,7) (1,1,145,7) (1,1,155,7) (1,1,143,7) (1,1,119,7) (1,1,129,7) (1,1,130,7) (1,1,115,7) by using these metrics we have calculated the face mask classifier.

$$\text{Precision} = T_p / (T_p + T_n) = 1 / (1 + 129) = 1 / 130 = 0.77$$

$$= T_p / (T_p + T_n) = 1 / (1 + 130) = 1 / 131 = 0.76$$

$$\text{Recall} = T_p / (T_p + T_n) = 1 / (1 + 145) = 1 / 146 = 0.68$$

$$= T_p / (T_p + T_n) = 1 / (1 + 115) = 1 / 116 = 0.86$$

$$\text{f1 score} = 2 * \text{Recall} * \text{Precision} / (\text{Recall} + \text{Precision})$$

$$= 2 * 0.68 * 0.77 / (0.68 + 0.77)$$

$$= 0.72$$

$$= 2 * 0.86 * 0.76 / (0.86 + 0.76)$$

$$= 0.80 \text{ approximately.}$$

$$\text{support} = 97 + 102 = 199.$$

For the face mask classifier for f1 80% is got very accuracy.

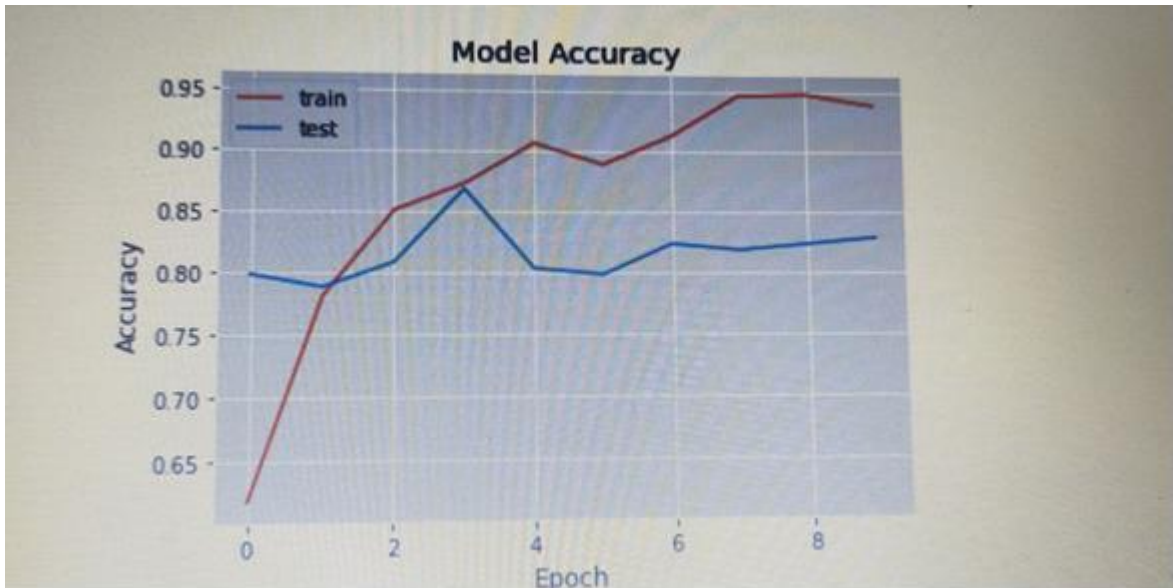


FIG 5.1.9 Model accuracy graph of facemask detection.

This graph shows in the above figure 5.2.6 the accuracy results, we can see there are little signs of overfitting, with training of 95% and tested here is 85% the training loss. We have achieved around 80% accuracy in this face mask detection with very little training loss. There is no exaggeration in mentioning that this face mask detector can effectively detect the persons without masks anywhere and anytime.

5.2 DISCUSSION

In this paper, we have implemented a Faster R-CNN model to detect whether a person or group of persons in an image wearing mask or not. we have trained the model in a such a way that it will work in varying dimensions and light conditions very effectively. This model can be used by any regulating authorities to control spreading of covid through proper monitoring and punishing the public who are not wearing masks. It can be further extended by integrating with app and recognizing the person and integrating his address to send fine to the citizens who are violating rules of wearing mask. After successfully implemented the coding the screenshots are placed in the above figure.

- In the first step we have collected the images of with mask and without mask, the data set consists of more than 1000 images
- Second step labelling the images of with mask and without mask.
- Third step data pre-processing
- fourth step model of the images
- Fifth step trained by the network
- plot the graph
- Classification of the images
- Select all and run the program code
- It will show the live camera
- Finally, the output of the face mask detection. Combining all the elements of our architecture, we tend to get correct mask observation system. Mobile NetV2, classifier employed in this system. The resultant system performance and has the potential to detect face mask in image with multiple face over a large vary angles.

5.2.1 TABULAR COLUMN OF THE NETWORK

Network	Accuracy	Loss
RPN	80%	20%
MobileNetV2	82%	15%

Loss is a number indicating how bad the model's predication was on a single example. If the Model predication is perfect, the loss is zero; otherwise, the loss is greater. The goal of training a model is to find a set of weights and biases that have loss, on average, across all examples. While Accuracy refers to how close a measurement is to the true value of what is being measured.

5.2.2 Social distancing monitoring with temperature and distance Prototype Model

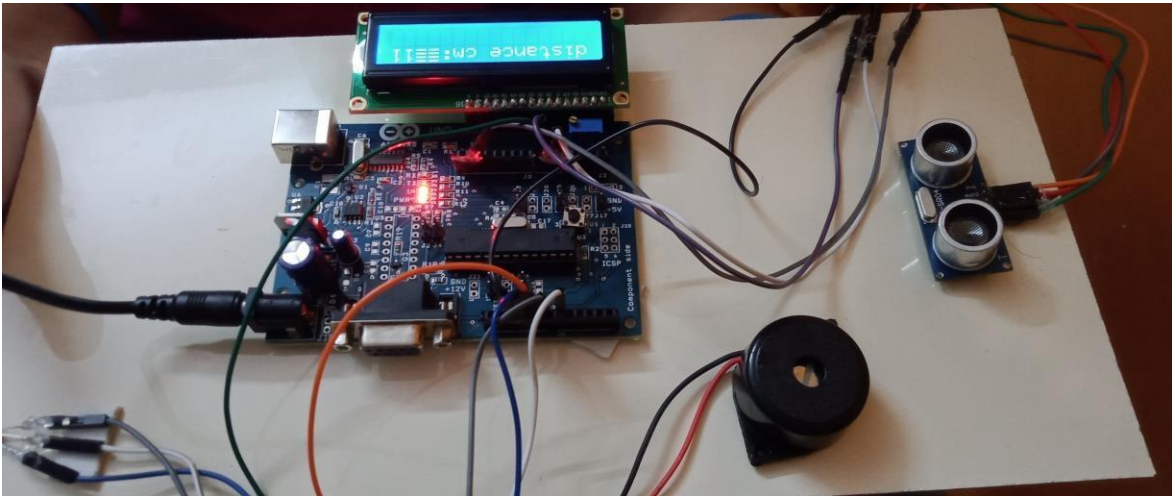


Fig 5.2.1 Prototype Model

This the prototype model used to measure the temperature and social distance. The Arduino uno is connected to the temperature, distance and buzzer. For temperature normal room temperature is measured, using an ultrasonic sensor the distance is measured when the distance is not 2metre the buzzer will give some sound. Therefore, this prototype model is very much to monitor the social distance.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

The model so developed is related to the detection of face masks being put by the people, are real time images. we trained the model by using the OpenCV, keras, Tensor flow an accuracy is achieved. Model training the classifier was subjected to images and live video stream. The faces were recognized in the images and videos and the face are extracted, the face mask classifier was applied and required results were obtained. As the model so developed is complicated in structure and gives instantaneous results and hence it can be used in the various. For monitoring the social distance, we used temperature and ultrasonic sensors. The models were tested with images and real – time video streams for this the accuracy are 80%, furthermore the proposed method achieves state-of-art it results on a face mask dataset.

6.2 CONCLUSION

As the technology are blooming with emerging trends the availability so we have novel face mask detector which can possibly contribute to public healthcare, the architecture consists of Mobile Net V2 as the backbone it can be used for high and low computation, But the facial mask detection and non-masked face detection accuracy provided high variations. The model was tested with image and real-time video stream, the accuracy of model is achieved and the optimization of the model is continuous process. This specific model could be used as case of edge analytics. The models were tested with images and real – time video streams for this the accuracy are 80%, furthermore the proposed method achieves state-of-art it results on a face mask dataset. By the development of face mask detection, we can detect if the person is wearing a mask or not and monitoring the social distance by using sensors, we can allow people, this system is great help to the society.

REFERENCE

1. Arthi Mahore, Meenakshi Tripathi, "Detection of 3d mask in 2D face recognition system using DWT and LBP: A Review", published in IEEE JCCIS: IEEE 3rd International conference on communication and information system (2018).
2. Gayathri Deora, Ramakrishna Godhula and Dr. Vishwas Udpikar "Study of Masked face Detection Approach in video Analytics", (CASP): IEEE conference on Advances in Signal Processing (2016).
3. Jagadeeswari.C, Uday Theja M.d, "Performance Evaluation of Intelligent face mask Detection system with various Deep Learning Classifiers: A Review", published in IJAST: International Journal of Advanced Science and Technology, (2020) Vol29, pp. 3074-3082.
4. He K, Zhang X, Ren S, and Sun J, " Deep residual learning for image recognition," IEEE Conference on Computer vision and Pattern Recognition (CVPR), (2016), pp. 770-778.
5. Li K, Ding G and Wang H " L-fcn A lightweight fully convolutional network for biomedical semantic segmentation," in IEEE(ICBIBM) International Conference on Bioinformatics and Biomedicine (2018) pp.2363-2367.
6. Kaihan Lin, Xianyoung Liu Huimin" Face detection and segmentation Based on improved Mask R-CNN", (2017).
7. Wang R, Li J, Fei J, " face detection based on YOLOV3: A Recent Trends", In ICCD Intelligent Computing, Communication and Devices, Singapore, (2017) pp. 277-284, doi: 10.1007/978-981-13- 9406-5_34.

8. Sandler M, Howard A, Zhu M, Zhmoginov M and Chen L, "MobileNetV2: Inverted Residuals and Linear Bottlenecks." IEEE/CVF Conference on computer Visions and pattern Recognition, (2018) pp. 4510-4520, doi. 10.1109/CVPR.2018.00474.

9. Ejaz M.S, Islam M.R, Sifatullah M, Sarker A "Implementation of principal component analysis on masked and non-masked face recognition: A Review", published in ICASERT: 1st International Conference on advances in science Engineering and Robotics Technology (2019) pp.15.

10. Sabbir Ejaz M.d, Sifatullah M.d, Rabiul Islam M.d and Ananya Sarker "Implementation of principle component analysis on masked and non-masked face recognition" IEEE(ICASE): IEEE 1st International conference on advances in science, engineering (2019).

11. Mingjie jiang, Xinqi Fan and Hang Yan "Retinal face mask detector" Hong Kong University (2020).

12. Nietro- Rodriguez, Mucidness M, Brea system for "Medical mask detection in the operating room through facial attributes pattern Recognition Image" (2015) pp.138-145.

13. Othman N.A, Aydin I," A new IOT combined face detection of people by using computer" vision for security application, A process published in IEEE (CICN): IEEE International conference on computational Intelligence and communication Networks (2017) pp1-5.

14. Ud Din N, Javed K, Bae S,"YIA novel GAN- based network for Unmasking of masked face: IEEE Access, 8" (2020) pp. 4427644287.

15. Kumar S, Negi A, Singh J.N, and Verma H, "A Deep learning for brain tumor mri images semantic segmentation using fcn: A Review", published on (ICCCA) 4th international conference on computing communication and Automation (2018) pp 1-4.
16. Sultana F, Sufian A, and Dutta P." A review of objection detection models based on convolutional neural network." (2019) arXiv preprint arXiv:1905.01614.
- 17.Toshnall Meenpal, Ashuthosh Balakrishnan and Amit Verma. "Facial Mask detection using semantic segmentation" IEEE (ICCCS) (2019) IEEE 4th international conference on computing communication and security.
18. Wei BU, et.al, "A cascade framework for Masked face Detection", IEEE (2017) 8th international conference on CTS & RAM Ningbo.
19. Jia X, "Image recognition method based on deep learning," (2017) Chinese control and Decision conference (CCDC), Chongqing.
20. Xin, M., Wang, Y. "Research on image classification model based on deep convolutional neural network", (2019) J image video proc. 40.

APPENDIX

HARDWARE PROGRAM - ARDUINO

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
int trigger=5;
int echo=6;
int buzzer=7;
int temp=A0;
void setup() {
    // put your setup code here, to run once:
    pinMode(7,OUTPUT);
    pinMode(temp,INPUT);
    pinMode(5,OUTPUT);
    pinMode(6,INPUT);
    Serial.begin(9600);
    lcd.begin(16,2);
}
void loop()
{
    long distance;
    long duration;
    digitalWrite(5,1);
    delayMicroseconds(10);
    digitalWrite(5,0);
    duration=pulseIn(6,HIGH);
    //Serial.println(duration);
    distance=duration*0.034/2;
    lcd.clear();

    //lcd.println("object detected");
```

```

lcd.setCursor(0,0);
lcd.println("distance cm:");
lcd.println(distance);
delay(2000);
Serial.println("object detected");
Serial.println("distance in cm:");
Serial.println(distance);
delay(2000);
if(distance<10)
{
//lcd.clear();
//lcd.setCursor(0,0);
//lcd.print("WEAR THE MASK");
//delay(1000);
digitalWrite(7,HIGH);
delay(1000);
}
else
{
  lcd.clear();
  digitalWrite(7,LOW);
}
int temperature=analogRead(temp);
temperature=temperature*0.488;
//lcd.clear();
lcd.setCursor(0,1);
lcd.println(temperature);
lcd.println("TEMPERATURE");
Serial.println("TEMPERATURE");
Serial.println(temperature);
delay(1000);
}

```

SOFTWARE PROGRAM - PYTHON

```
import os
import zip file
import random
import shutil
import tensor flow as tf
from tensor flow. keras. Optimizers import RMSprop
from tensor flow. keras. preprocessing. Image import imageDataGenerator
#import pillow as PIL
from tensor flow. keras. Layers import Conv2D, Input, Zeropadding2D
from tensor flow. keras. Models import Model, load model
import imutils
import numpy as np
Import matplotlib. pyplot as plt
#import matplotlib. image as mpimg

model=tf. keras. models. Sequential ([
    tf. keras. layers. Conv2D (100, (3,3), activation='relu', input
shape=150,150,3)),
    tf. keras. layers. MaxPooling2D (2,2),
    tf. keras. layers. MaxPooling2D (2,2),
    tf. keras. layers. Conv2D (100, (3,3), activation='relu'),
    tf. keras. layers. Conv2D (100, (3,3), activation='relu'),

    tf. keras. layers. Maxpooling2D (2,2),
    tf. keras. layers. Flatten (),
    tf. keras. layers. Dense (50, activation='relu'),
    tf. keras. layers. Dense (2, activation='SoftMax')

model. Compile (optimizer='adam', loss='binary cross entropy', metrics=['acc'])
```

```

train_datagen = ImageDataGenerator (rescale=1.0/255,
                                    rotation_range=40,
                                    width_shift_range=0.2,
                                    height_shift_range=0.2,
                                    shear_range=0.2,
                                    zoom_range=0.2,
                                    horizontal_flip=True,
                                    file_mode='nearest')
                                    file_mode='nearest')

train_generator = train_datagen.flow_from_directory (TRAINING_DIR,
import matplotlib.pyplot as plt
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show ()

face_classifier=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

labels_dict={0:'no mask',1:'no mask'} # Show the image

color_dict={0:(0,0,255),1:(0,255,0)}

webcam = cv2.VideoCapture(0) #Use camera 0

Classifier = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Show the image

cv2.imshow('LIVE', im)

# if Esc key is press then break out of the loop

webcam.release ()

```