VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELAGAVI – 590 018, KARNATAKA, INDIA

A MINI PROJECT REPORT
ON

"FACE MASK DETECTION USING MACHINE LEARNING"

Submitted in partial fulfillment of the requirements for the award of
BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By

Name	USN
GOKULNATH S	4VP18EC022
KARTHIK PRASAD	4VP18EC026
MADHAVA RAJ B	4VP18EC035
SANJAN GOVIND K	4VP18EC055

Under the Guidance of:
Prof. SRIKANTH RAO S.K.
Head of Department
Department of Electronics and Communication Engineering
Vivekananda College of Engineering & Technology

Prof. RAJANI RAI B
Assistant Professor,
Department of Electronics and Communication Engineering
Vivekananda College of Engineering & Technology

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
VIVEKANANDA COLLEGE OF ENGINEERING & TECHNOLOGY
[A Unit of Vivekananda Vidyavardhaka Sangha, Puttur (R)]
Affiliated to Visvesvaraya Technological University and Approved by AICTE New Delhi & Govt. of Karnataka
Nehru Nagara, Puttur – 574203, DK, Karnataka, India

VIVEKANANDA COLLEGE OF ENGINEERING & TECHNOLOGY
[A Unit of Vivekananda Vidyavardhaka Sangha, Puttur (R)]
Affiliated to Visvesvaraya Technological University and Approved by AICTE New Delhi & Govt. of Karnataka
Nehru Nagara, Puttur – 574203, DK, Karnataka, India

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the project work entitled **"FACE MASK DETECTION USING MACHINE LEARNING"** is carried out by **Mr. Gokulnath S, Mr. Karthik Prasad, Mr. Madhava Raj B, and Mr.Sanjan Govind K** bearing USNs respectively **4VP18EC022, 4VP18EC026, 4VP18EC035 and 4VP18EC055** bonafide students of **Vivekananda College of Engineering & Technology**, in partial fulfillment for the award of **Bachelor of Engineering** in **Electronics and Communication Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2020-21. It is certified that all corrections/ suggestions indicated for Internal Assessment have been incorporated in the report.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of the Guide
Prof. Rajani Rai B
Assistant Professor
VCET, Puttur

Signature of the Guide and HOD
Prof. Srikanth Rao SK
HOD, Dept. of ECE
VCET, Puttur

DECLARATION

We, **Gokulnath S (4VP18EC022),Karthik Prasad (4VP18EC026), Madhava Raj B (4VP18EC035), and Sanjan Govind K (4VP18EC055)** students of Sixth semester B. E. in Electronics & Communication Engineering, **Vivekananda College of Engineering & Technology**, Puttur, hereby declare that the Mini project work entitled **"Face Mask Detection Using Machine Learning"** has been carried out and duly executed by me at VCET, Puttur, under the guidance of , **Prof. Srikanth Rao SK**, Head of the Department, Department of Electronics & Communication Engineering, Vivekananda College of Engineering & Technology, Puttur, and **Prof. Rajani Rai B**, Assistant/Associate Professor, Department of Electronics & Communication Engineering, Vivekananda College of Engineering & Technology, Puttur and submitted in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Electronics & Communication** by **Visvesvaraya Technological University**, Belagavi during the academic year 2020-21.

GOKULNATH S	**4VP18EC022**
KARTHIK PRASAD	**4VP18EC026**
MADHAVA RAJ B	**4VP18EC035**
SANJAN GOVIND K	**4VP18EC055**

Date:

Place: VCET, Puttur

ABSTRACT

COVID-19 has affected the world badly. Studies have proved that wearing a face mask is one of the precautions to reduce the risk of viral transmission. And many public places as well as public service providers require customers to use the service and place only if they wear mask correctly. So, it is not possible to manually track the customer, whether they have the mask or not. That's why this technology holds the key here. In this paper, we propose Face Mask Detection using Machine Learning which is one of the high accuracy and efficient face mask detector. This proposed system mainly uses the techniques of Image pre-processing, Face detection and Face mask classifier. Our system is capable of detecting masked and unmasked faces and can be integrated with webcam cameras. This system will help to tack safety violations, promote the use of face masks and it ensure a safe working environment.

ACKNOWLEDGEMENT

We take this opportunity to express our deep heartfelt gratitude to all those people who have helped us in the successful completion of the project.

First and foremost, we would like to express our sincere gratitude towards our guides **Mr. Srikanth Rao SK** and **Mrs. Rajani Rai B** for providing excellent guidance, encouragement and inspiration throughout the project work. Without his invaluable guidance, this work would never have been a successful one.

We would like to express our sincere gratitude to the Coordinator, **Mr. Naveena C** for providing excellent guidance and support.

We would like to express our sincere gratitude to our Head of the Department of Electronics and Communication Engineering, **Prof. Srikanth Rao SK** for his guidance and inspiration.

We would like to thank our Principal **Dr. Mahesh Prasanna K** for providing all the facilities and a proper environment to work in the college campus.

We are thankful to the Management for providing the platform to carry out our project work.

We are thankful to all the teaching and non-teaching staff members of Electronics and Communication Engineering Department for their help and needed support rendered throughout the project.

TABLE OF CONTENTS

LIST OF FIGURES

Figure No.	Title	Page No.

List of Tables

Chapter-1

INTRODUCTION

1.1 PROJECT BACKGROUND

In 2020, COVID-19 being a pandemic disease had rapid spread of virus over the world creating a red alert in global health, humanity and everyday lifestyle of humans and daily lives had deep impact. All around the globe, illness on respiratory such as Severe Acute Respiratory Syndrome (SARS), and Middle East Respiratory Syndrome (MERS) are spreading, on December 2019 in Wuhan, China a new critical respiratory illness has arisen and has infected millions of peoples and lost millions of helpless lives in more than 200 countries according to world health organization (WHO) and declared global pandemic. The spread of the virus is through close contact with the people and in overcrowded public areas . The guidelines listed by the WHO, primary precaution should be taken to prevent the spread of virus is to wear facemask and maintain social distance. An efficient based computer vision approach aims on real-time application which monitors the individuals in public whether wearing facemask or not and safe social distancing by implementing raspberry pi4 module to the monitor and spot violation through public cameras. Modern deep learning algorithms are mixed with techniques of geometry to build robust models which are able to cover aspects such as, detection, tracking, and validation . This paper addresses building a machine learning model to detect facemask.

The trend of sporting face mask publically is rising because of the Covid-19 epidemic everywhere in the world. Because Covid-19 people wont to wear mask to shield their health from air pollution. Whereas other are selfconscious concerning their looks, they hide their emotions from the general public by activity their faces. Somebody treated the wearing face masksworks on hindering Covid-19 transmission. 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. Quite 5 million cases were infected by Covid-19 in not up to half dozen month across 188 countries. The virus spreads through shut contact and in packed and overcrowded areas. The corona virus epidemic has given rise to a unprecedented degree of worldwide scientific cooperation. Computer science supported machine learning and deep learning will facilitate to fight Covid-19 in several ways. Machine learning a valuate huge quantities of knowledge to forecast the distribution of Covid-19 to function early warning mechanism for potential pandemics, and classify vulnerable population. Folks are forced by laws to wear face masks publically many countries. These rules and law we have a tendency yore developed as associate degree action to the exponential growth in cases an deaths in several areas.

Face mask detection involves in detecting the location of the face and then determining whether it has a mask on it or not. The issue is proximately cognate to general object detection to detect the classes of objects. Face identification categorically deals with distinguishing a specific group of entities i.e. Face. It has numerous applications, such as autonomous driving, education, surveillance, and so on. This paper presents a simplified approach to serve the above purpose using the basic Machine Learning (ML) packages such as TensorFlow, Keras, OpenCV and Scikit-Learn.

1.2 SOME IMPORTANT TERMINOLOGIES

MobileNetV2: MobileNetV2 is that the latest technology of mobile visual recognition, including classification, object detection and semantic segmentation. The classifier uses deep intelligent separable convolution, its purpose is to significantly reduce the complexity cost and model size of the network, so it's suitable for mobile devices, or devices with low computing power. In MobileNetV2, another best module introduced is that the reverse residual structure. The non-linearity within the narrow layer is removed. Maintain because the backbone of feature extraction, MobileNetV2 achieves the simplest performance in object detection and semantic segmentation. For MobileNetV2 classifier, ADAM optimizer has been applied to see performance.

ADAM : Adam, a stochastic optimization algorithm supported step the target function is predicated on an adaptive estimation of low-order moments. this manner it's computationally efficient and may be executed almost without memory. It’s the diagonal of the gradient is rescaled unchanged, which is extremely suitable for the subsequent problems large in terms of knowledge and/or parameters. Hyper parameters are intuitive explain that they typically don't require much adjustment.

Machine learning: Machine learning is a method of teaching prediction based on some data. It is a branch of artificial intelligence. Which numerically improves on data over as more data as add in algorithm the performance of the system is improved.

These are the three types of machine learning :

Supervised learning: Its supervised learning we have several data points or samples described using predictive variables or features and the target variable our data represented in table structure. Game supervised learning is build a model its able to predict the target variable.

Unsupervised learning: is a machine learning task the uncovering hidden patterns from unlabeled data.

Reinforcement learning (RL): in which machine or software agents interact with an environment reinforcement learning agents are able to automatically figure out how to optimize their behavior given a system of reward and punishments reinforcement learning draws inspiration from behavioral psychology.

Computer Vision: It is a filed that include processing analyzing and understanding image in general high dimensional data from the real world in order to produce numerical and symbolic information or it is a technology of science and machine that see it obtain information from images.

Deep Learning: Deep learning is a powerful set of techniques for learning using neural network. Neural network are beautiful biologically inspired programing paradigm which enables a computer to learn from data.

CNN(Convolutional Neural Network): They 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. The primary difference between CNN and other ordinary neural network is that CNN takes input as a two dimensional array and operates directly on the images rather than focusing on feature extraction which other neural network focus on. The dominant approach of CNN includes solutions for problems of recognition. Top companies like Google and facebook have invested in research and developments towards recognition projects to get activites done with greater speed.

A convolutional neural network uses three basic ideas:

- **Local respective fields**
- **Convolution**
- **Polling**

CNN plays a significant part in computer vision related examples in recognizing patterns, on account of its less computation cost and also the ability of spatial extraction. CNN utilizes convolution portions to combine with the primary images in order to remove top-level features. The commencement network that is proposed in permits the network to get familiar with the mix of kernels. Planning to build a good Convolutional Neural

Network architecture actually remains as a primary inquiry. To prepare a lot further neural network, K. He et al. proposed Residual Network (ResNet) that can take in personality planning from the past layer. As article locators are generally conveyed on portable or any embedded device, where the computing assets are extremely restricted, Mobile Network (MobileNet) is proposed. This utilizes profundity shrewd convolution to remove highlights and channelised convolutions to change channel numbers, so that the computational expense of the MobileNet is a lot lower compared to networks utilizing standard convolutions. In below figure we have shown a Schematic Diagram for Basic Convolution Neural Network.

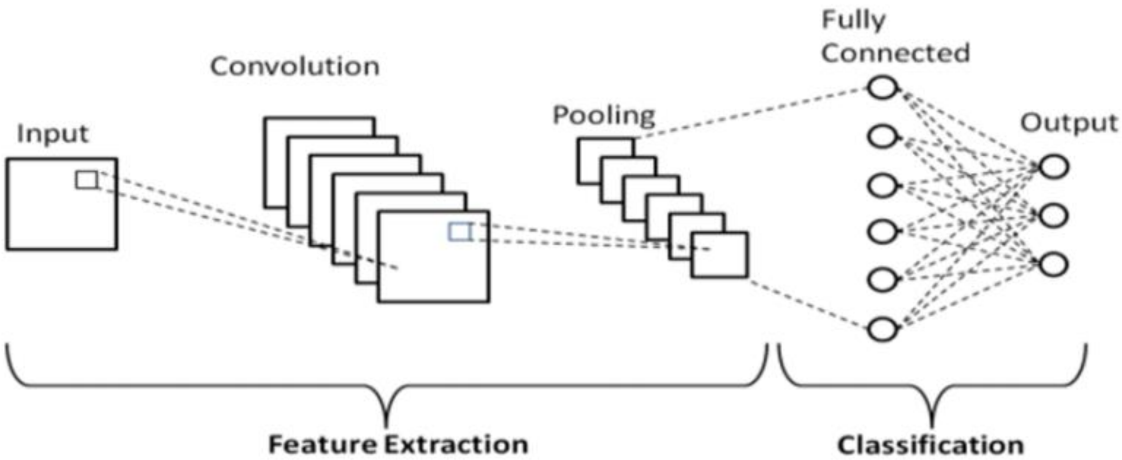


Fig1.1 Schematic Diagram for Basic Convolution Neural Network

Machine Learning packages used in current model:

OpenCV: OpenCV (Open Source Computer Vision Library) is a collection of algorithms for computer vision. It basics focus on real time image processing it is free for commercial and research use under a BSD license.

Tensor Flow: Tensor Flow is a mathematical computation library for training and building your machine learning and deep learning model with a simple to use high level **API**s.

Keras: Keras is a neural network API. It is library written specifically in python. In addition, It works with other libraries and packages such as tensorflow which makes deep learning easier. Keras was developed to allow for quick experimentation and for fast prototyping.

1.3 STUDIES ON OVERLAY

The facemask detection model has become a most required and essential model during COVID-19 pandemic. Since manually monitoring whether people are wearing facemask in public and crowded areas, research paper had built real-time automated model integrated with surveillance cameras in public which detects whether people are wearing facemask and maintaining social distance in public areas and report to the respective authorities using computer vision and implementing raspberry pi4. Global health crises have occurred due to coronavirus, facial masks form a basic prevention from the virus, hence, hybrid model is built using classical and deep machine learning consisting of two components. The first component is for feature extraction is by using Resnet50 and the second is for classification processing of mask is using ensemble, support vector machine (SVM) algorithms and decision tree. Uses three datasets after investigation. Simulated Masked Face Dataset (SMFD) is the first dataset , the second dataset is Labeled Faces in the Wild (LFW) and third one is the Real-World Masked Face Dataset (RMFD). SVM learning algorithm achieved 99.49% accuracy in SMFD. RMFD achieved 99.64% of accuracy, LFW achieved 100% of testing accuracy. Healthcare system is under crisis. List of precautionary measures is being taken care in order to reduce the spread of viruses in which wearing facemasks is one of them. A system is created to find people not wearing facemasks in smart cities using Closed-Circuit-Television (CCTV) cameras. The trained system achieved 98.7% accuracy in differentiating people with mask and without mask. An efficient and high-accuracy detector called RetinaFaceMask detector is built to spot whether people are wearing facemasks. The framework is a one-stage detector with a new background attention module to concentrate on face mask identification and a pyramid network feature to combine high-level semantic data with several feature maps. An approach to the algorithm for the elimination of new background attention module artifacts to delete projections of high union intersections and poor confidences. Results of RetinaFaceMask achieves state-of-the-art results on facemask dataset with 2.3% and 1.5% higher than the standard result and mask detection precision, with 11.0% and 5.9% higher than the standard results. An architecture is built to find whether people are wearing facemask in live streaming videos and even with human face images using Single Shot Detector (SSD) serves the purpose of object detection. Concepts of transfer learning in neural networks used in finding presence or absence of facemask in video streams and in images. Experimental findings indicate that the model performs well with 100% accuracy and 99% precision and recall, respectively. Simplified approach towards detecting facial masks even in motion using basic machine learning packages such as Tensorlow, Keras, OpenCV, and Scikit-learn. The method attains accuracy up to 95.77% and 94.58% respectively on two different datasets. An automated process for finding whether individuals wear facemask in public. The model is built by fine-tuning the pre-trained state-of-the-art deep learning models called InceptionV3. Simulated Face Mask Dataset (SMFD) dataset is used to train the dataset. Here, on the public face dataset mask is put and then it's simulated.

This is used to better training and testing of the model. The image augmentation technique is used for improved training and testing of the model for restricted data usability The model reaches 99.9% precision during training and 100% accuracy during testing. An efficient real-time system approach towards a computer vision based to detect both violation of wearing facemask and social distancing in public areas using the convergence of advanced deep learning algorithms with geometric techniques resulted in model creation which is robust in nature and covers the aspects of validation, detecting, tracking. Rasberry-pi4 is implanted in public surveillance cameras and the built robust model was allowed to run on raspberry-pi4. The proposed model uses lightweight neural network MobileNetV2 to analyse Real-Time Streaming Protocol (RSTP) video stream using OpenCV and transfer learning techniques with Single Shot Detector (SSD) used to achieve resource limitation and accuracy recognition in monitoring real-time video surveillance cameras in public areas to spot out violation of wearing facemask and maintaining social distancing. Hybrid models are built in detection of facemask using classical machine learning and deep learning. The system has two components, one component is used for extraction of features using Resnet50 and the other component is used to classify facemask using Support Vector Machine (SVM), ensemble algorithm and decision tree. The Real World Masked Face Dataset (RMFD), the Labelled Faces in the Wild (LFW) and Simulated Masked Face Dataset (SMFD) are the three datasets used for experiment of considered three algorithms. SVM algorithm resulted more efficient than the other three algorithms with 99.64% accuracy in RMFD, 99.49% accuracy in SMFD and reached 100% accuracy in LFW. A system is proposed to find absence of facial mask with the people in public areas in smart cities by monitoring with Closed-Circuit-Television (CCTV) cameras. A person without a facial mask is detected and notified to higher authorities through the city network. The architecture is developed using Convolutional Neural Network (CNN) which helps for feature extraction from the dataset images as well the images captured by cameras in real-time. There is 98% of accuracy result by the trained architecture.

1.4 COMPARISON WITH DIFFERENT ANN

Artificial Neural Network: We are using different architecture and models of ANN were used for face mask detection. ANN can be used in face mask detection because these models can simulate the way neurons work in human brain. We compared between different neural network for face mask recognition system and lastly we use those model which have better accuracy.

Retinal Connected Analysis Network(RCNN): We presented face mask detection system based on RCNN that examine small windows of an image to check each window contain face with or without mask. First, a preprocessing step, adapted from, is applied to window of the image. Then window is passed through the a neural network, which decides whether the window contain face with or without mask. I used the two training dataset of image. In first dataset with mask image collected by us consist of more than 1800 images. The second dataset without mask consist of more than 1800 images. The recognition face with mask and without mask equal to 80% accuracy rate.

Principal Component Analysis with ANN: We are using PCA with class specific linear projection to detect or recognized face with or without mask in a real time video stream.

The system steps to search for face with or without mask in an image:

1. Select every 20x20 region of input image
2. Use intensity values of its pixel as 400 inputs to ANN
3. If the value is above 0.5 the region represent a face
4. Repeat steps several times, each times on a resized version of the original input image to search for face at different scale.

Convolutional Neural Network in this planned method, the mask detection model is constructed victimization the successive API of the Keras library. This permits us to make the new layers for our model step by step. The assorted layers used for our CNN model is represented below. The 1st layer is that the Conv2D layer with one hundred filters and therefor the filter size or the kernel size of 3x3. During this first step, the activation operate used is the “ReLu”. This ReLu function stands for the corrected linear measure which is able to output the input directly if is positive, otherwise, it’ll output zero. The input size is also initialized as 150X150X3 for all the photographs to be trained and tested victimization this model. In the second layer, the MaxPooling2D is employed with the poll size of 2x2. The next layer is once more a Conv2D layer with another one hundred filters of constant filter size 3X3 and {also the} activation operate used is that the ‘ReLu’. This Conv2D layer is followed by a MaxPooling3=2D layer with size 2x2. In consecutive step, we have a tendency to use the flatten (the layers into one 1D layer. After the flatten layer, we use the dropout (0.5) layer to forestall t form overfitting. Finally, towards the end, we have a tendency to use the dense layer with fifty units and therefore the activation operate as ReLu. The last layer of our model are going to be another dense layer with solely 2 units and the activation function used will be the ‘softmax’ function. The softmax function outputs a vector which is able to represent the chance distribution of every of the input units. Here, two inputs units are used. The softmax function will output a vector with two probability distribution values.

Fast Neural Network: A FNN approach to reduce the computational time for locating human faces with or without mask. Each image divided into small sub images and the each one is tested separately using a fast ANN. The experimental result of comparison with conventional Neural Network showed high speed achieved when applying FNN.

Table 1.1 Comparison of Neural Network Method

Comparison of Neural Network Method

Table1:

S.no	Methodlogy	Recognition rate(%)
1	RCNN	90.45
2	PCA with ANN	95.67
3	CNN	95.22
4	FNN	94

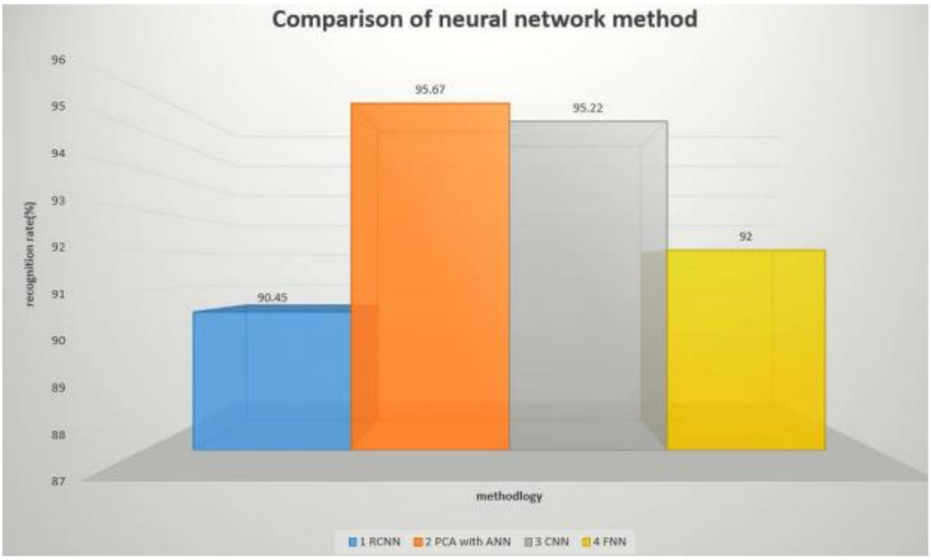


Fig1.2 Comparison of Neural Network Method

After the comparison the two best methodology is CNN and PCA with ANN both recognition rate are approximately same then we are using the CNN for final face detection model.

CHAPTER-2

LITERATURE SURVEY

Generally, most of the projects specialize in face construction identity recognition when wearing mask. During this projects, the focus is on recognizing the people that wearing mask, or not help in decreasing the transmission and spreading of covid-19. The scientist has proven that wearing a mask help in minimizing the spreading rate of Covid-19. In [1], the authors developed a face mask wearing condition identification method. They were ready to classify three categories of face mask-wearing. The categories are face mask- wearing, incorrect face mask-wearing and no face mask-wearing. Saber et al [2], have applied the principal component analysis on masked and unmasked face recognition to acknowledge the person. Also, PCA was utilized in [3]. The author proposed a way that's used for removing glasses from human frontal faces. In [4], the authors used the YOLOv3 algorithm for face detection. YOLOv3 uses Darknet-53 because the backbone. Nizam et al [5] proposed a completely unique GAN-based network, which will automatically remove mask covering the face area and regenerate the image by building the missing hole. In [6], the authors presented a system for detecting the presence or absence of a compulsory medical mask within the OR. The general is to attenuate the false positive face detection as possible without missing mask detection so as to trigger alarms just for medical staff who don't wear a surgical mask. Shaik et al[7] used deep learning real-time face emotion classification and recognition. They used VGG-16 to classify seven countenance. Under the present Covid-19 lock-in time, this technique is effective in preventing spread in may use cases. Here are some use cases which will benefit from system.

Chapter-3

Proposed System for the Face Mask Detection

The proposed system focuses on how to identify a person wearing a mask on the image/video stream. Help with computer vision and deep learning algorithm by using OpenCV, tensor flow, Keras library.

3.1 METHODOLOGY

» Data gathering

» Training the model

» Face mask detection

In the first phase, We will collect the images with or without the mask. This dataset consists of 3600+ images belonging to two classes. In the second phase, We will train the recognizer for detecting people with or without masks, and in the last phase, We will use the trainer data to classify each face as with a mask or without a mask.

Here in Figure below we have described an architecture that shows how our system functions automatically to prevent the spread of COVID19.

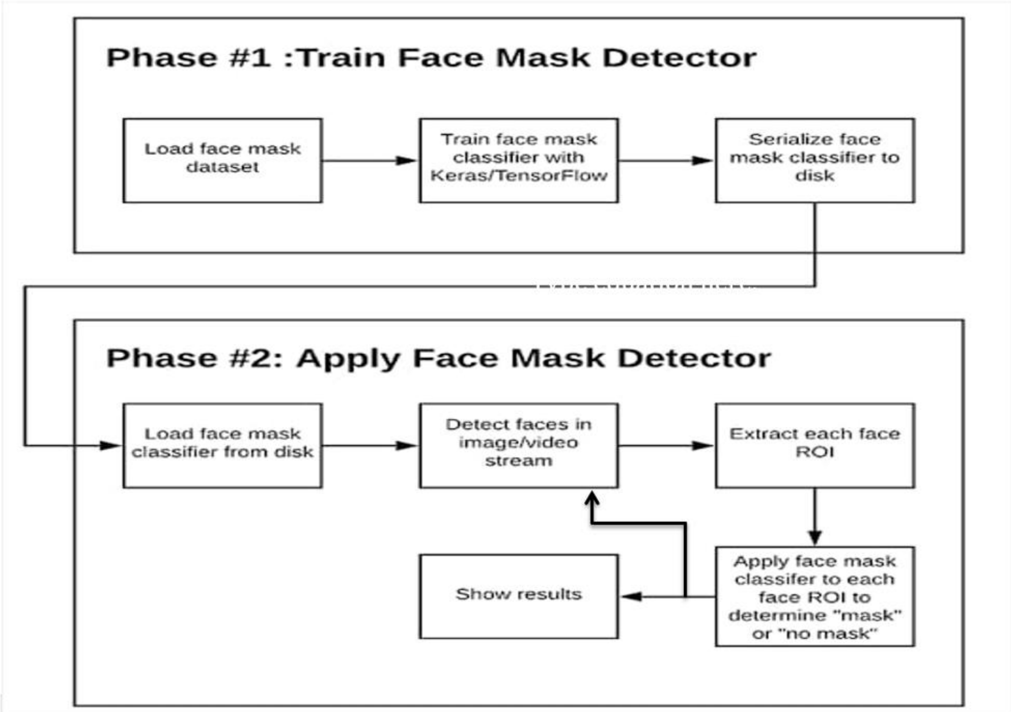


Fig 3.1 Overall System Architecture

Our system uses the TensorFlow and Keras algorithm to detect whether an individual is wearing a face mask along with the Convolutional Neural network model. Here we first train the system with the Dataset from Kaggle and train it with Keras and TensorFlow, once the training is done then we will load face mask classifier from the disk, here faces are detected from a real time video stream. This process also involves use of MobileNet in order to train a huge collection of images and classification of high-quality images. Here image dataset is loaded from Keras and then the images are converted into an array, later MobileNet is used to preprocess input image and to append image to the data list. In the proposed system the main contribution includes person face identification and face mask detection. These both are done in Real Time with the help of MobileNet and OpenCV. A square box is been displayed on every person's face with the color of red and green where red indicates the person is not wearing a mask and green indicates a person is wearing a mask. We have used a face cropped dataset from Kaggle of about 3918 images of persons with masks and without masks. These images are used in order to train the model that classifies into two categories: that is, faces with masks and faces without masks. These datasets are then converted into arrays in order to create a Deep Learning Model. The result of the person from the video displays a person with a square bound box. This system monitors continuously, and whenever a person is identified without a mask then the person's face is been captured and then it is sent to the higher authorities, also to that person. Due to the outbreak of novel Corona Virus this proposed model can be implemented in public at real-time for monitoring the people wearing face masks. Our model can be used for monitoring automatically in public places that would help for those who monitor people physically/manually, that is the reason we picked this architecture. Our system can be used in airports, schools, railway stations, shopping malls, offices and other public areas to make sure that in-public people are wearing masks.

3.2 STEPS FOR FACEMASK DETECTION

STEP 1: Collection of Data and pre-processing:

The proposed system used face cropped data containing images with different angles and different poses of face with and without masks that are labelled and is used to train our model. The real time automated face mask detection has been done by MobileNet and OpenCV. The dataset consists of 3600+ images that are used to train our proposed model. The data are divided into two different categories: Faces with mask and without mask. Faces with masks includes mask with hand, with masks and other objects that cover the face, that provides us an advantage to improve variants of the dataset.

a)Data Collection: The dataset pictures for covert and unmasked faces were collected from images dataset offered by Kaggle, a open-source online community of data scientists and machine learning practitioners and few within the public domain the masked i obtained from the factitious generated by me through the picture redaction tool and few from collected from the public domain. Within the dataset, consist of 1800+ with masked face and 1800+ are while not masked face. The data set is collected for the training the face mask detection model.

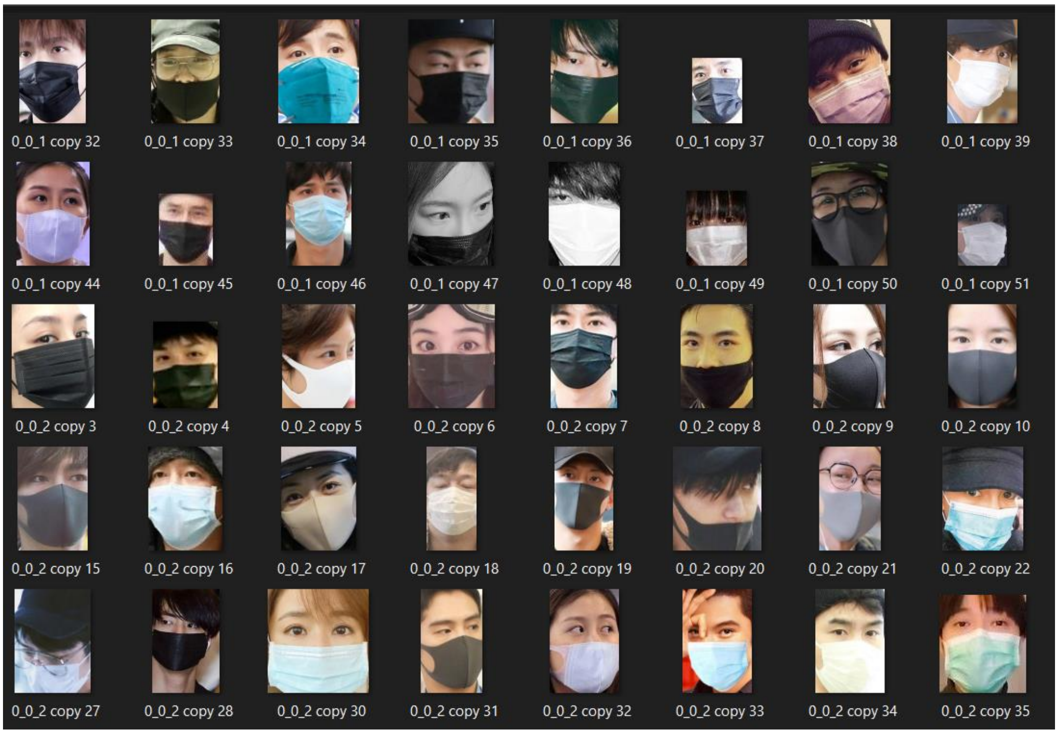


Fig 3.2 With mask image dataset

Fig 3.3 Without mask image dataset

b)Data Pre-processing: Data preprocessing involves conversion of data from a given format to much more user friendly, desired and meaningful format. It can be in any form like tables, images, videos, graphs, etc. These organized information fit in with an information model or composition and captures relationship between different entities. The proposed method deals with image and video data using Numpy and OpenCV.

c) Data Visualization: Data visualization is the process of transforming abstract data to meaningful representations using knowledge communication and insight discovery through encodings. It is helpful to study a particular pattern in the dataset. The total number of images in the dataset is visualized in both categories – ‘with mask’ and ‘without mask’. The statement categories=os.listdir(data path) categorizes the list of directories in the specified data path. The variable categories now looks like: [‘with mask’, ‘without mask’] . Then to find the number of labels, we need to distinguish those categories using labels=[i for i in range(len(categories))]. It sets the labels as: [0, 1]. Now, each category is mapped to its respective label using label dict=dict(zip(categories,labels)) which at first returns an iterator of tuples in the form of zip object where the items in each passed iterator is paired together consequently. The mapped variable label dict looks like: {‘with mask’: 0, ‘without mask’: 1}

d) Conversion of RGB image to Gray image: Modern descriptor-based image recognition systems regularly work on grayscale images, without elaborating the method used to convert from color-to-grayscale. This is because the color-to-grayscale method is of little consequence when using robust descriptors. Introducing non-essential information could increase the size of training data required to achieve good performance. As grayscale rationalizes the algorithm and diminishes the computational requisites, it is utilized for extracting descriptors instead of working on color images instantaneously .

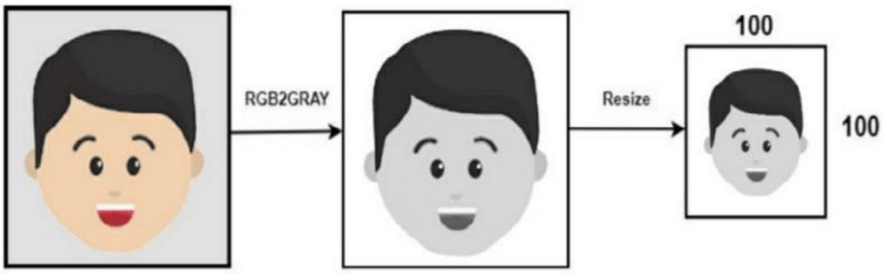


Fig 3.4 Conversion of a RGB image to a Gray Scale image of 100 x 100 size

We use the function cv2.cvtColor(input image, flag) for changing the color space. Here flag determines the type of conversion. In this case, the flag cv2.COLOR BGR2GRAY is used for gray conversion. Deep CNNs require a fixed-size input image. Therefore we need a fixed common size for all the images in the dataset. Using cv2.resize() the gray scale image is resized into 100 x 100.

e) Image Reshaping: The input during relegation of an image is a three-dimensional tensor, where each channel has a prominent unique pixel. All the images must have identically tantamount size corresponding to 3D feature tensor. However, neither images are customarily coextensive nor their corresponding feature tensors. Most CNNs can only accept fine-tuned images. This engenders several problems throughout data collection and implementation of model. However, reconfiguring the input images before augmenting them into the network can help to surmount this constraint. The images are normalized to converge the pixel range between 0 and 1. Then they are converted to 4 dimensional arrays using data=np.reshape(data,(data.shape[0], img size,img size,1)) where 1 indicates the Grayscale image. As, the final layer of the neural network has 2 outputs – with mask and without mask i.e. it has categorical representation, the data is converted to categorical labels.

STEP 2: Splitting the Data

In this step, we divide the data into training set, and the testing set which will contain the image on which the CNN model will be trained and test set and the images on which the model will be tested. In this case, we use split_size=0.8, which means that 80% of the total images will enter the training set, and the remaining 20% of the images will enter the test set.

STEP 3: Building and Training the Model

In the proposed system, the custom dataset is loaded and the algorithm is being trained based on labelled images. In this step the images are resized and its been converted into numpy array format. This model uses MobileNet performing as a backbone and train the model using TenserFlow. Parameters with a learning rate (initial) of INIT_LR = 1e-4, batch size BS = 32 and the number of epoch EPOCHS = 20. For the model webcam is used for face mask detection and once the person is found we mark the person with the square bounded box.

a) Building the model using CNN architecture: CNN has become ascendant in miscellaneous computer vision tasks. The current method makes use of Sequential CNN. The First Convolution layer is followed by Rectified Linear Unit (ReLU) and MaxPooling layers. The Convolution layer learns from 200 filters. Kernel size is set to 3 x 3 which specifies the height and width of the 2D convolution window. As the model should be aware of the shape of the input expected, the first layer in the model needs to be provided with information about input shape. Following layers can perform instinctive shape reckoning. In this case, input shape is specified as data.shape[1:] which returns the dimensions of the data array from index 1. Default padding is "valid" where the spatial dimensions are sanctioned to truncate and the input volume is non-zero padded. The activation parameter to the Conv2D class is set as "relu". It represents an approximately linear function that possesses all the assets of linear models that can easily be optimized with gradient-descent methods. Considering the performance and generalization in deep learning, it is better compared to other activation functions. Max Pooling is used to reduce the spatial dimensions of the output volume. Pool size is set to 3 x 3 and the resulting output has a shape (number of rows or columns) of: shape of output = (input shape - pool size + 1) / strides), where strides has default value (1,1). The second Convolution layer has 100 filters and Kernel size is set to 3 x 3. It is followed by ReLu and MaxPooling layers. To insert the data into CNN, the long vector of input is passed through a Flatten layer which transforms matrix of features into a vector that can be fed into a fully connected neural network classifier. To reduce overfitting a Dropout layer with a 50% chance of setting inputs to zero is added to the model. Then a Dense layer of 64 neurons with a

ReLu activation function is added. The final layer (Dense) with two outputs for two categories uses the Softmax activation function.

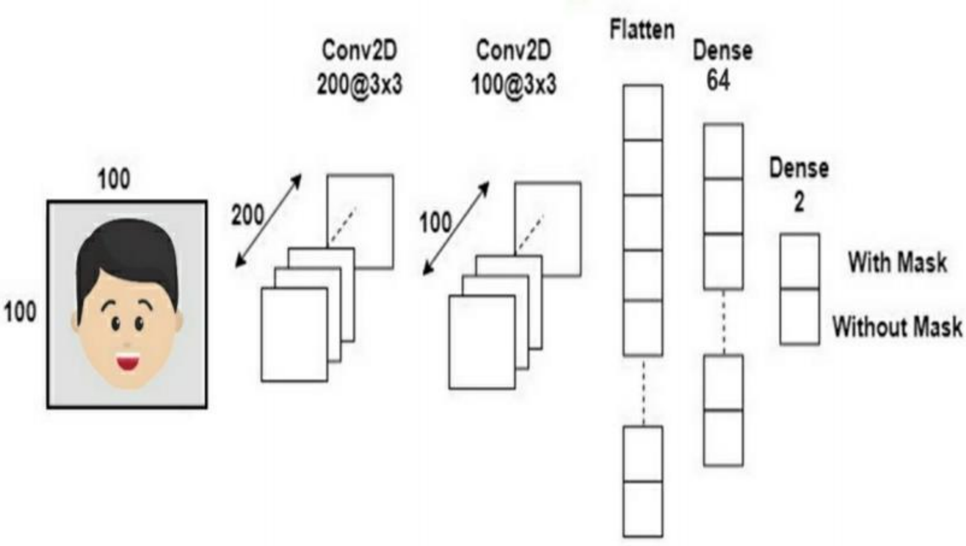
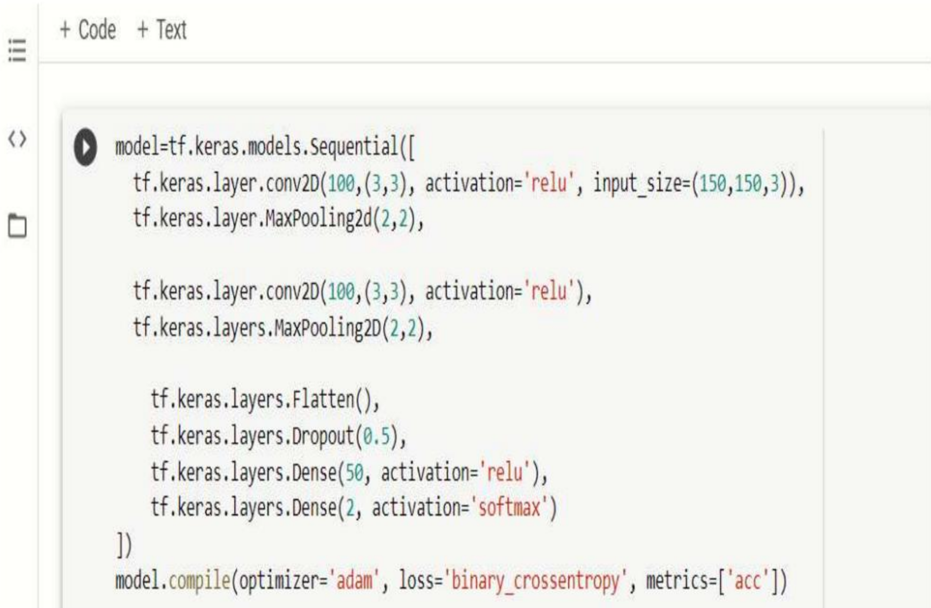


Fig 3.5 Convolutional Neural Network architecture

b)Training the model: The learning process needs to be configured first with the compile method [13]. Here "adam" optimizer is used. categorical crossentropy which is also known as multiclass log loss is used as a loss function (the objective that the model tries to minimize). As the problem is a classification problem, metrics is set to "accuracy".



```
+ Code  + Text

model=tf.keras.models.Sequential([
  tf.keras.layer.conv2D(100,(3,3), activation='relu', input_size=(150,150,3)),
  tf.keras.layer.MaxPooling2d(2,2),

  tf.keras.layer.conv2D(100,(3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(50, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
```

Fig 3.6 Training the model

Here, we use “ADAM” optimizer and binary crossentropy as our loss function because there are only two types. In addition, you can even use MobileNetV2 to get better accuracy.

STEP 4: Labeling the Information: After building the model, we label the result with two probabilities. [“0” is “without_mask” , “1” is “with_mask”]. We also set the color of the bounding rectangle using RGB values. [“RED” stands for “without_mask” “GREEN” stand for “with_mask”].

STEP 5: Importing the Face Detection Program: From now on, we plan to use it detect whether we are wearing a mask through the PC’s webcam. For this, first of all, we need to implement face detection model.

STEP 6: Detecting the Faces with and without Mask: In the last step, we use the Open CV library to run an infinite loop to use our webcam, where the cascade classifier is used to detect faces. The code webcam=cv2.videocapture(0) indicates the usage of the webcam. The model will predict the likelihood of each of the two categories (without_mask, with_mask]). Based on a higher probability, tags will be selected and displayed around our face.

Chapter-4

RESULTS

4.1EXPERIMENTAL RESULT:

The experimental result of system performance is evaluated with the MobileNetV2 classifier ADAM optimizes.

```
Epoch 7/20
91/91 [==============================] - 36s 392ms/step - loss: 0.0674 - acc: 0.9733 - val_loss: 0.2987 - val_acc: 0.9312
Epoch 8/20
91/91 [==============================] - 36s 392ms/step - loss: 0.0716 - acc: 0.9715 - val_loss: 0.1256 - val_acc: 0.9638
Epoch 9/20
91/91 [==============================] - 36s 392ms/step - loss: 0.0903 - acc: 0.9623 - val_loss: 0.3195 - val_acc: 0.9130
Epoch 10/20
91/91 [==============================] - 35s 380ms/step - loss: 0.0790 - acc: 0.9733 - val_loss: 0.1342 - val_acc: 0.9674
Epoch 11/20
91/91 [==============================] - 36s 396ms/step - loss: 0.1043 - acc: 0.9669 - val_loss: 0.1361 - val_acc: 0.9493
Epoch 12/20
91/91 [==============================] - 37s 402ms/step - loss: 0.0939 - acc: 0.9632 - val_loss: 0.1233 - val_acc: 0.9710
Epoch 13/20
91/91 [==============================] - 34s 373ms/step - loss: 0.0906 - acc: 0.9577 - val_loss: 0.1429 - val_acc: 0.9674
Epoch 14/20
91/91 [==============================] - 36s 398ms/step - loss: 0.0892 - acc: 0.9660 - val_loss: 0.0872 - val_acc: 0.9783
Epoch 15/20
91/91 [==============================] - 36s 392ms/step - loss: 0.0950 - acc: 0.9596 - val_loss: 0.2265 - val_acc: 0.9239
Epoch 16/20
91/91 [==============================] - 35s 384ms/step - loss: 0.0897 - acc: 0.9688 - val_loss: 0.1134 - val_acc: 0.9746
Epoch 17/20
91/91 [==============================] - 34s 370ms/step - loss: 0.0854 - acc: 0.9651 - val_loss: 0.1678 - val_acc: 0.9493
Epoch 18/20
91/91 [==============================] - 35s 389ms/step - loss: 0.0794 - acc: 0.9678 - val_loss: 0.0981 - val_acc: 0.9710
Epoch 19/20
91/91 [==============================] - 36s 393ms/step - loss: 0.0762 - acc: 0.9707 - val_loss: 0.1470 - val_acc: 0.9601
Epoch 20/20
91/91 [==============================] - 34s 376ms/step - loss: 0.0711 - acc: 0.9751 - val_loss: 0.1081 - val_acc: 0.9746
```

Fig 4.1 compilation screen for training script of facemask detection

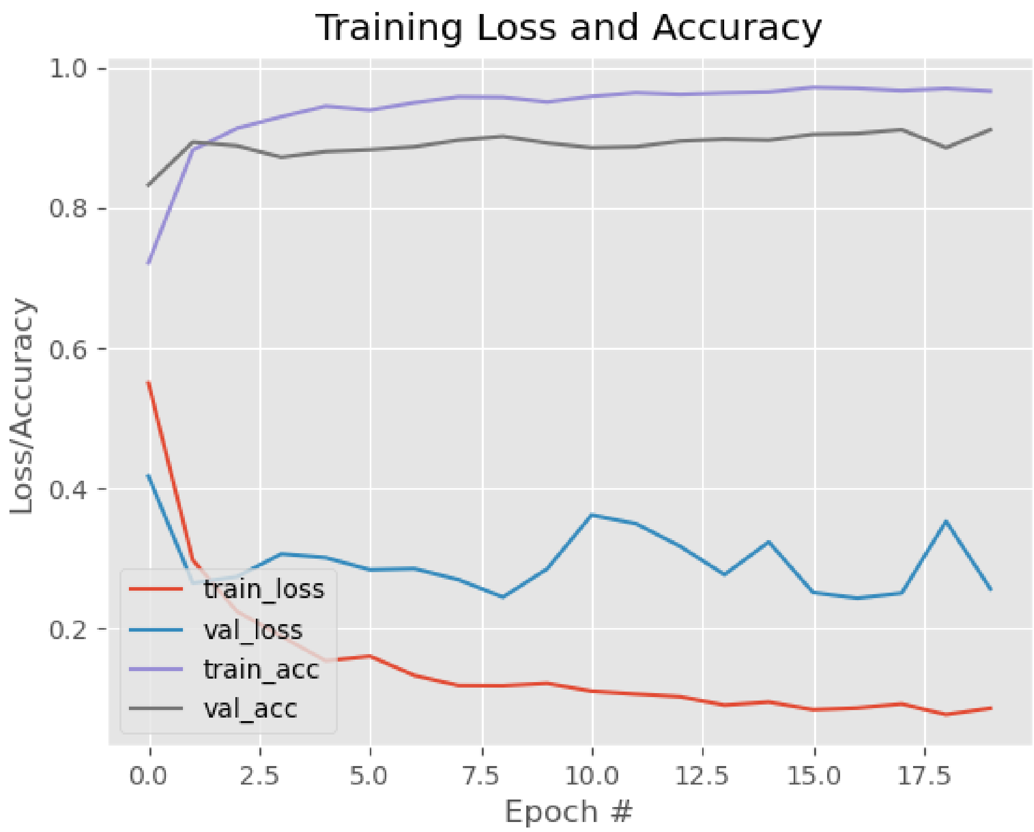


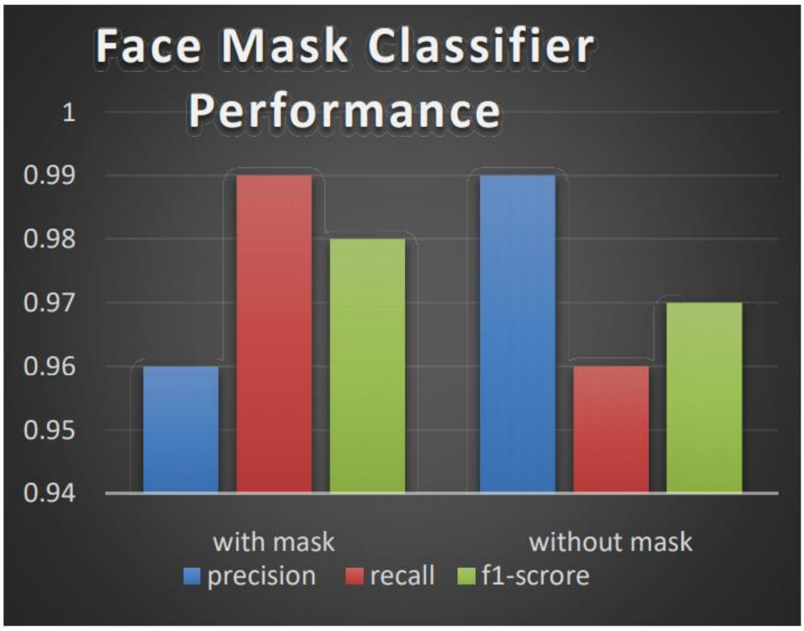
Fig 4.2 Training Model accuracy and loss curves

Face Mask Classifier Performance Metrics:

```
]: predict=model.predict(test_X,batch_size=BS)
   predict=np.argmax(predict,axis=1)
   print(classification_report(test_Y.argmax(axis=1),predict,target_names=lb.classes_))

              precision    recall  f1-score   support

   with_mask       0.96      0.99      0.98       138
without_mask       0.99      0.96      0.97       138

    accuracy                           0.97       276
   macro avg       0.98      0.97      0.97       276
weighted avg       0.98      0.97      0.97       276
```

Fig 4.3 Face Mask Classifier Performance Metrics

Table 4.1 Performance Metrics Histogram graph



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.

4.2 Face mask detect from real time video stream:

First we run following command in the Command Prompt to start the Face Mask detector:

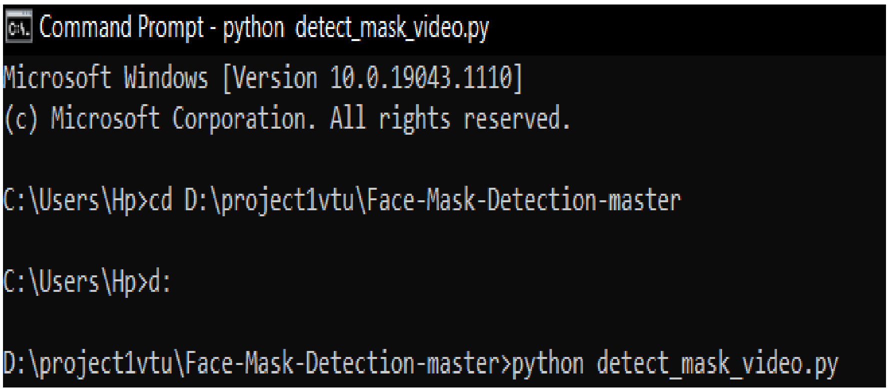


Fig 4.4 Command in the Command Prompt to start the Face Mask Detector

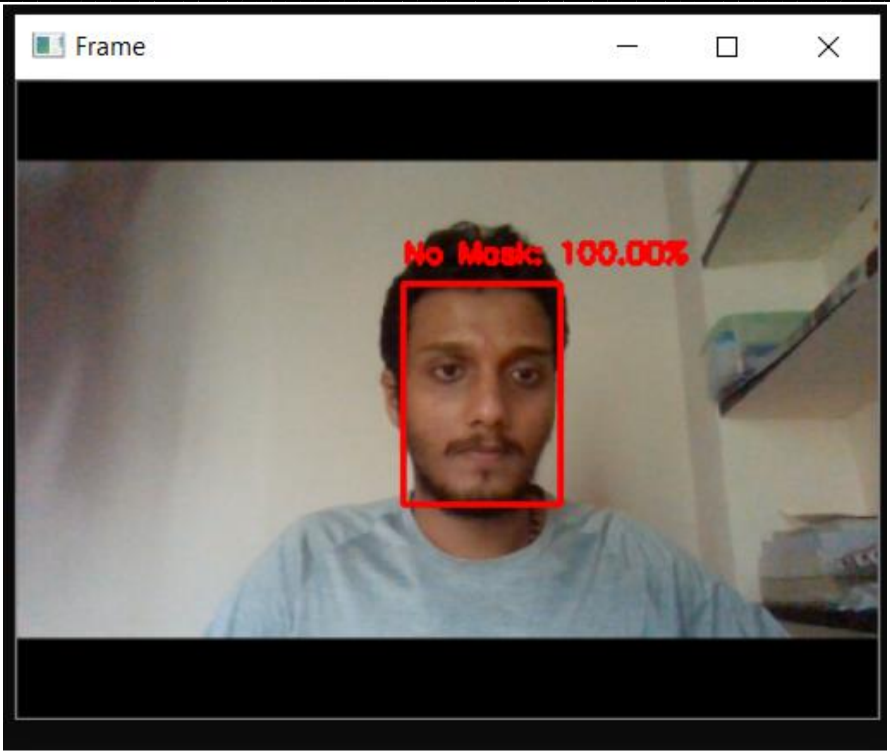


Fig 4.5 Output when no mask

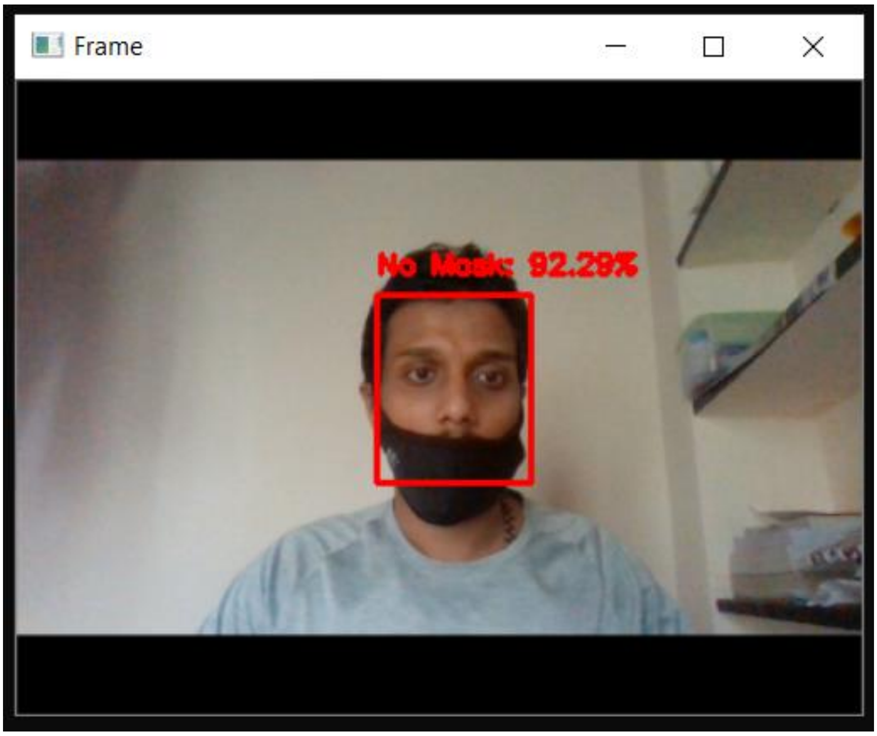


Fig 4.6 Output when mask weared improperly

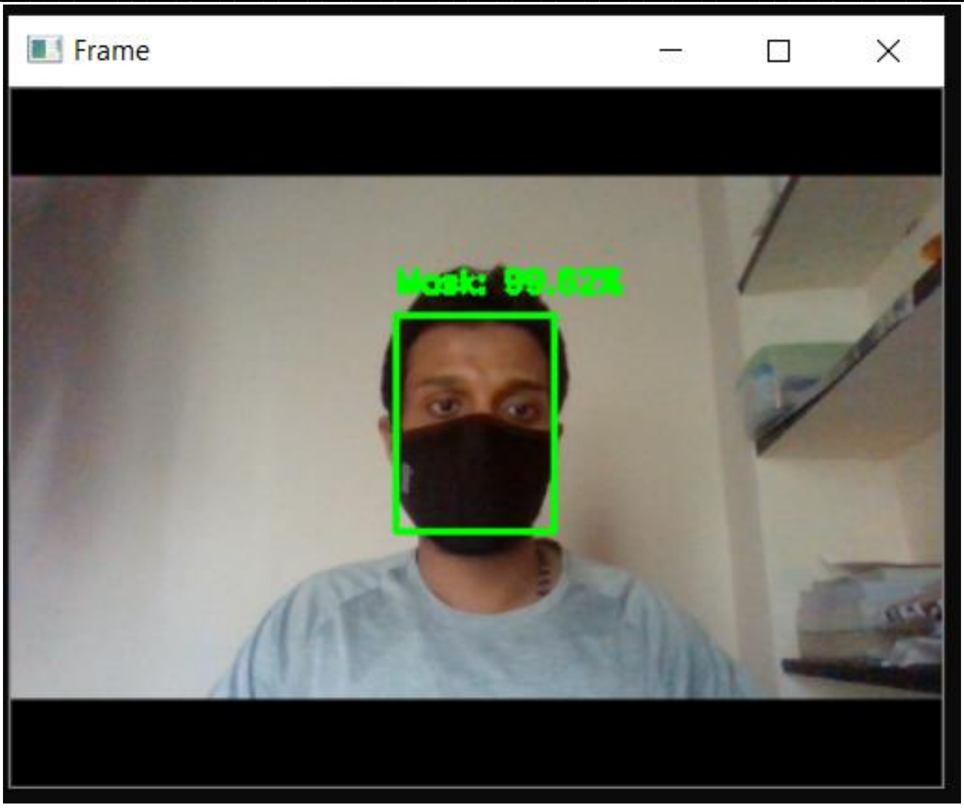


Fig 4.7 Output when Mask weared properly

4.3 IMPLEMENTING THE MODEL

Our system uses a custom dataset with the input video taken from any camera device. The system feeds with a real time video in public places which automatically monitors and detects whether or not people are wearing face masks. Whenever a person is found without wearing a mask then his/her photo is captured, then it is been sent to the higher officials/authorities as well the victim so that they can take any further actions.

Chapter-5

CONCLUSION AND SCOPE FOR FUTURE WORK

5.1 CONCLUSION

In our Project we have proposed a system that automatically identifies whether or not a person is wearing a face mask and notify the higher authorities if not wearing a mask. This proposed system uses Computer Vision and MobileNet to help the public ensure that they are wearing face masks and to keep away from the spread of COVID-19 virus. . The our face mask detection is trained on CNN model and we are used Oven CV, Tensor Flow, Keras and python to detect whether person is wearing a mask or not . Our research also helps police or higher authorities that makes it easier to identify whether a person is wearing a mask, if not then they will be also having the victim's photo by which they can take further actions. The proposed system can be implemented in places like railway stations, shopping malls, offices, schools, airports, etc. The accuracy of model is achieved and, the optimization of the model is continuous process. This specific model could be used as use case of edge analytics.

5.2 FUTURE WORK

There are many more different cases in which this model can be integrated for the safety of the public:

- Identify a person if he is doing any crime by wearing face mask.
- Identify what type of mask is the person wearing.
- Coughing and Sneezing Detection.
- Temperature Screening
- The model can be further improved to detect if the mask is virus prone or not ie. the type of the mask is surgical, N95 or not.

REFERENCES

[1] B. QIN and D. Li, Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19, May 2020, doi: 10.21203/rs.3.rs-28668/v1.

[2] M.S. Ejaz, M.R. Islam, M. Sifatullah, A. Sarker ,Implementation of principal component analysis on masked and non-masked face recognition 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (2019), pp. 15, doi: 10.1109/ICASERT.2019.8934543.

[3] Jeong-Seon Park, You Hwa Oh, Sang ChulAhn, and Seong Whan Lee, Glasses removal from facial image using recursive error compensation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 805–811, doi: 10.1109/TPAMI.2005.103.

[4] C. Li, R. Wang, J. Li, L. Fei, Face detection based on YOLOv3, in:: Recent Trends in Intelligent Computing, Communication and Devices, Singapore, 2020, pp. 277–284, doi: 10.1007/978-981-13- 9406-5_34.

[5] N. Ud Din, K. Javed, S. Bae, J. YiA novel GAN-based network for unmasking of masked face IEEE Access, 8 (2020), pp. 4427644287, doi: 10.1109/ACCESS.2020.2977386.

[6] A. Nieto-Rodríguez, M. Mucientes, V.M. BreaSystem for medical mask detection in the operating room through facial attributes Pattern Recogn. Image Anal. Cham (2015), pp. 138- 145, doi: 10.1007/978-3-319-19390-8_16.

[7] S. A. Hussain, A.S.A.A. Balushi, A real time face emotion classification and recognition using deep learning model, J. Phys.: Conf. Ser. 1432 (2020) 012087, doi: 10.1088/1742- 6596/1432/1/012087.

[8] MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications - Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam.