Face Mask Detection

This project report is submitted to Yeshwantrao Chavan College of Engineering

(An Autonomous Institution Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)

In partial fulfilment of the requirement For the award of the degree

of

Bachelor of Engineering in Electronics & Telecommunication Engineering

by

Aakib Sayyad

Ritik Pusdekar

Mohit Zade

Tejas Ubale

Rohit Samarth

Under the guidance of

Prof N.A. Pande

DEPARTMENT OF Electronics & Telecommunication Engineering

Nagar Yuwak Shikshan Sanstha's

YESHWANTRAO CHAVAN COLLEGE OF ENGINEERING,

(An autonomous institution affiliated to Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur)

NAGPUR – 441 110

2021-2022

CERTIFICATE OF APPROVAL

This is to Certify that the project report entitled "**Face Mask Detection**" has been successfully completed by **Aakib Sayyad, Ritik Pusdekar, Mohit Zade**, **Tejas Ubale and Rohit Samarth** under the guidance of **Prof. N A Pande** in recognition to the partial fulfilment for the award of the degree of Bachelor of Engineering in Electronics & Telecommunication Engineering, **Yeshwantrao Chavan College of Engineering (An Autonomous Institution Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)**

Signature	**Signature**
Prof. N.A Pande	Dr. M. S. Narlawar
(Guide)	**(HOD, ET Dept)**

Declaration

We hereby declare that

a. The work contained in this project has been done by us under the guidance of my supervisor(s).

b. The work has not been submitted to any other Institute for any degree or diploma.

c. We have followed the guidelines provided by the Institute in preparing the project report.

d. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary

Signature & Name of the Students

1. Ritik Pusdekar
2. Mohit Zade
3. Tejas Ubale
4. Rohit Samarth
5. Aakib Sayyad

Acknowledgement:

We sincerely acknowledge with a deep sense of gratitude to our project guide Prof. N.A Pande for the guidance and encouragement she gave us for the preparation of this project. Without her support, the project would have been difficult.

We are highly obliged to Dr. U. P. Waghe (Principal), Dr. M. S. Narlawar, H.O.D (Department of Electronics & Telecommunication Engineering), for their noble, spontaneous, and timely help that carried us throughout our endeavor and finally made a grand success.

We also thank the staff of our electronics department for all the cooperation and friendly treatment given to us during the project. We are also thankful to our colleagues and all those who have extended the necessary help during the course of work.

CONTENTS

LIST OF FIGURES

Abstract:

The COVID-19 pandemic is causing a worldwide emergency in healthcare. This virus mainly spreads through droplets which emerge from a person infected with corona-virus and poses a risk to others. The risk of transmission is highest in public places. One of the best ways to stay safe from getting infected is wearing a face mask in open territories as indicated by the World Health Organization (WHO). In this project, we propose a method which employs TensorFlow and OpenCV to detect face masks on people. A bounding box drawn over the face of the person describes whether the person is wearing a mask or not. If a person's face is stored in the database, it detects the name of the person who is not wearing a face mask and an email will be sent to that person warning them that they are not wearing a mask so that they can take precautions.

In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras and OpenCV. The proposed method detects the face from the image correctly and then identifies if it has a mask on it or not. As a surveillance task performer, it can also detect a face along with a mask in motion. The method attains accuracy up to 95.77% and 94.58% respectively on two different datasets. We explore optimized values of parameters using the Sequential Convolutional Neural Network model to detect the presence of masks correctly without causing over-fitting.

Chapter 1: Introduction

According to the World Health Organization (WHO)'s official Situation Report – 205, coronavirus disease 2019 (COVID-19) has globally infected over 20 million people causing over 0.7million deaths . Individuals with COVID19 have had a wide scope of symptoms reported – going from mellow manifestations to serious illness. Respiratory problems like shortness of breath or difficulty in breathing is one of them. Elder people having lung disease can possess serious complications from COVID-19 illness as they appear to be at higher risk.

To curb certain respiratory viral ailments, including COVID-19, wearing a clinical mask is very necessary. The public should be aware of whether to put on the mask for source control or aversion of COVID-19. Potential points of interest of the utilization of masks lie in reducing vulnerability of risk from a noxious individual during the "pre-symptomatic" period and stigmatization of discrete persons putting on masks to restrain the spread of virus. WHO stresses on prioritizing medical masks and respirators for health care assistants. Therefore, face mask detection has become a crucial task in present global society.

Chapter 2: Problem Definition:

COVID-19 pandemic has rapidly affected our dayto-day life disrupting world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services.

Therefore, face mask detection has become a crucial task to help global society.

Our Goal is to propose a model through which we can determine if a person is wearing the mask or not. We aim to increase the accuracy of the model and provide assistance.

Chapter 3: Literature Review

[1] Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV

Year of published: 2020

Authors: Arjya Das,Mohammad Wasif Ansari,Rohini Basak

This paper presents a simplified approach to achieve this purpose using Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn. The proposed method detects the face from the image correctly and then identifies if it has a mask on it or not. As a surveillance task performer, it can also detect a face along with a mask in motion.

[2] An Approach to Face Detection and Recognition

Year of published: 2016

Authors: Divya Meena,Ravi Sharan

Face detection has attracted immense attention because it has many applications in computer vision communication and automatic control system. Research into face detection, expression recognition, face tracking, pose estimation is required. By giving a single image, the challenge is to detect the face from that image. Face detection is a challenging task because faces are not rigid and it changes in size, shape, color etc. In this paper, viola Jones algorithm is used to detect face and principal component analysis for face recognition

[3] Face Mask Detection Using OpenCV

Year of published: 2021

Authors: Harish Adusumalli ,D Kalyani

The COVID-19 pandemic is causing a worldwide emergency in healthcare. This virus mainly spreads through droplets which emerge from a person infected with coronavirus and poses a risk to others. The risk of transmission is highest in public places. One of the best ways to stay safe from getting infected is wearing a face mask in open territories as indicated by the World HealthOrganization (WHO). In this project, we propose a method which employs TensorFlow and OpenCV to detect face masks on people. A bounding box drawn over the face of the person describes whether the person is wearing a mask or not.

[4] Advice on the use of masks in the context of COVID-19

Year of published: 2020

Authors: WHO

This document provides advice on the use of masks in communities, during home care, and in health care settings in areas that have reported cases of COVID-19. It is intended for individuals in the community, public health and infection prevention and control (IPC) professionals, health care managers, health care workers (HCWs), and community health workers.

[5] Face Mask Detection using Transfer Learning of InceptionV3

Year of published: 2020

Authors: G. Jignesh Chowdary, Narinder Singh Punn ,Sanjay Kumar Sonbhadra and Sonali Agarwal

In this paper, a transfer learning model is proposed to automate the process of identifying the people who are not wearing mask. The proposed model is built by fine-tuning the pre-trained state-of-the-art deep learning model, InceptionV3. The proposed model is trained and tested on the Simulated Masked Face Dataset (SMFD). Image augmentation technique is adopted to address the limited availability of data for better training and testing of the model. The model outperformed the other recently proposed approaches by achieving an accuracy of 99.9% during training and 100% during testing.

Chapter 4 : Construction and Working of Model:

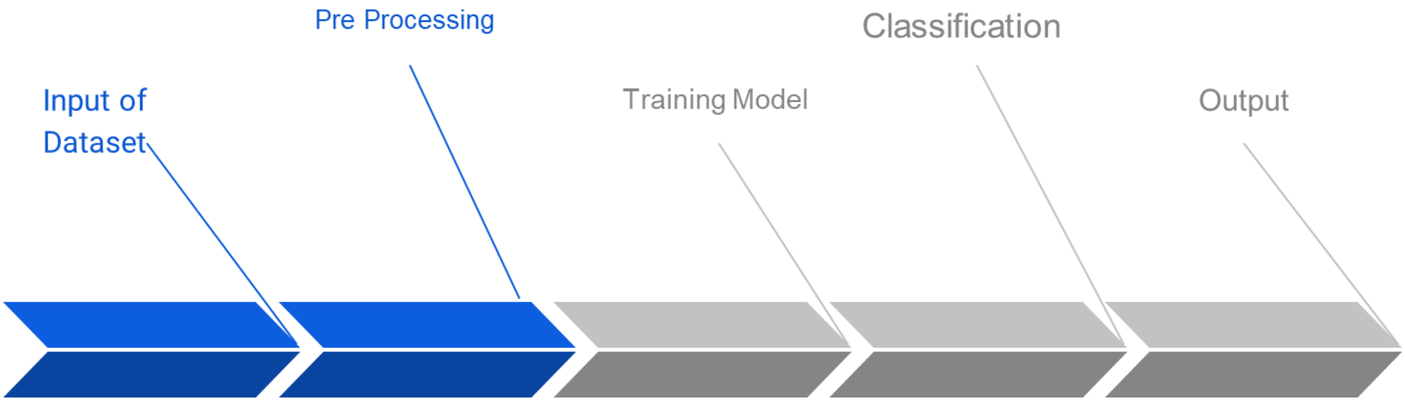

Figure 1. Flowchart of proposed model

4.1 Dataset:

The dataset was collected from Kaggle Repository and it contains 3833 images, out of the total images 1918 images are without mask images and 1915 are with mask images.

Figure 2. Images with Mask

Figure 3 Images Without Mask

4.2. Pre processing:

In the proposed model read the images in the dataset using OpenCV. This list of images in the dataset directory, then initialize the list of data(i.e. images) and class images. After the images are successfully loaded they are converted into an array. This data is then appended in data and labels.

When the data is loaded and converted into an array, One hot encoding is performed for this label binarizer is used. Label Binarizer is a function to perform the transform operation of LabelBinarizer with fixed classes. It helps to convert multi-class labels to binary labels.

4.3 Training model and Classification:

Now the data is successfully loaded in the model, this data is now split into train and test using sklearn. The train split is passed through the Image Data Generator.ImageDataGenerator class provides a quick and easy way to augment your images. It provides a host of different augmentation techniques like standardization, rotation, shifts, flips, brightness change, and many more.

Inorder to train the model MobilenetV2 is used. This model is a convolutional neural network architecture that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers.The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. As a whole, the architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers.

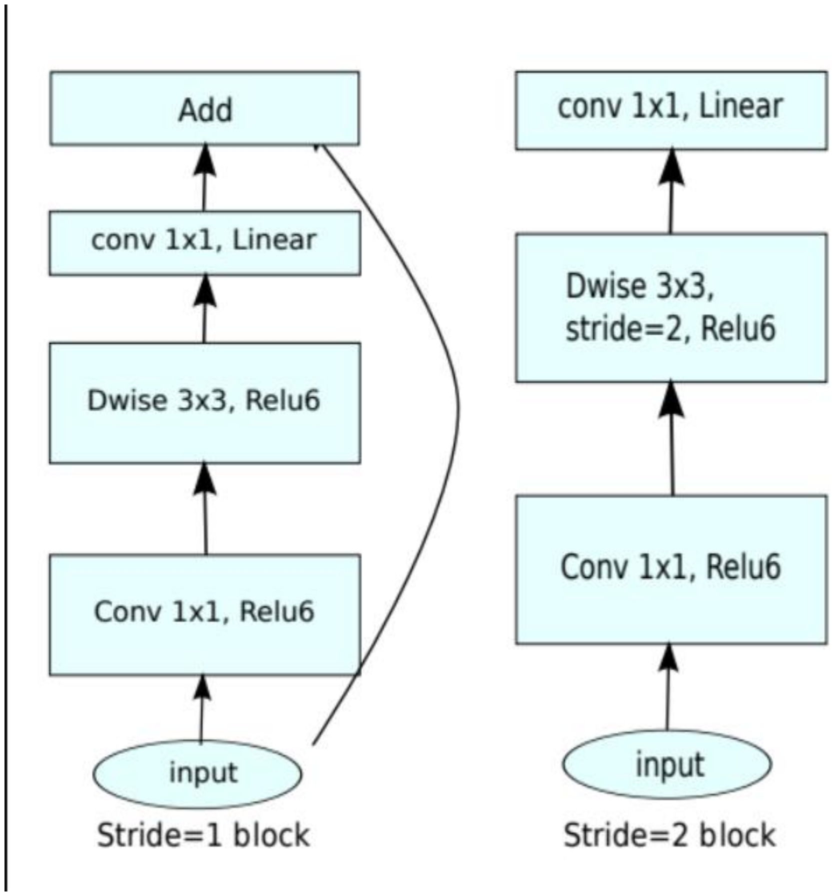

Figure 4. MobilenetV2 CNN model

4.4 Prediction:

Inorder to predict the are using imutil, it will help to start the video stream through which we will be able to detect the mask. If the person is wearing a mask in the video it will be shown in green box and if the person is not wearing a mask it will show in red box.

Chapter 5: Details of

5.1 Tools used

5.1.3 TensorFlow:

TensorFlow, an interface for expressing machine learning algorithms, is utilized for implementing ML systems into fabrication over a bunch of areas of computer science, including sentiment analysis, voice recognition, geographic information extraction, computer vision, text summarization, information retrieval, computational drug discovery and flaw detection to pursue research . In the proposed model, the whole Sequential CNN architecture (consists of several layers) uses TensorFlow at backend. It is also used to reshape the data (image) in the data processing.

5.1.2: Keras:

Keras gives fundamental reflections and building units for creation and transportation of ML arrangements with high iteration velocity. It takes full advantage of the scalability and cross-platform capabilities of TensorFlow. The core data structures of Keras are layers and models. All the layers used in the CNN model are implemented using Keras. Along with the conversion of the class vector to the binary class matrix in data processing, it helps to compile the overall model.

5.1.3 OpenCV:

OpenCV (Open Source Computer Vision Library), an open source computer vision and ML software library, is utilized to differentiate and recognize faces, recognize objects, group movements in recordings, trace progressive modules, follow eye gesture, track camera actions, expel red eyes from pictures taken utilizing flash, find comparative pictures from an image database, perceive landscape and set up markers to overlay it with increased reality and so forth [20]. The proposed method makes use of these features of OpenCV in resizing and color conversion of data images

5.1.4 CNN model:

In neural networks, Convolutional neural networks (ConvNets or CNNs) is one of the main categories to do image recognition, images classification. Object detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications take an input image, process it and classify it under certain categories (as in this paper in normal and abnormal) Computers see an input image as an array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d(h = Height, w = Width, d = Dimension). Eg., An image of a 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

CNN model works in following steps:

- Provide input image into convolution layer
- parameters, apply filters with strides, padding if required. Perform convolution on the image and apply ReLU activation to the matrix.
- Perform pooling to reduce dimensionality size
- Add as many convolutional layers until satisfied
- Flatten the output and feed into a fully connected layer (FC Layer)
- Output the class using an activation function (Logistic Regression with cost functions) and classifies images.

5.2 Processes Followed:

5.2.1 Convolutional layer:

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

- An image matrix (volume) of dimension **$(h \times w \times d)$**
- A filter **$(f_h \times f_w \times d)$**
- Outputs a volume dimension **$(h - f_h + 1) \times (w - f_w + 1) \times 1$**

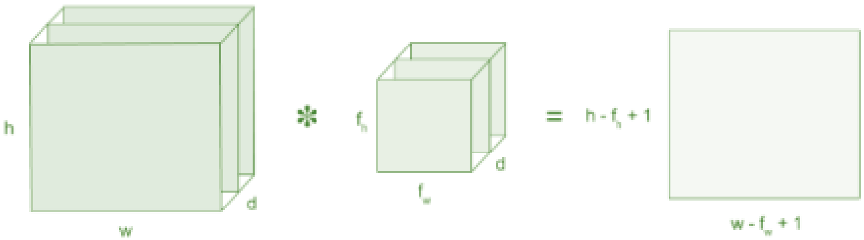

Figure 5 Convolutional layer in CNN model

5.2.2 MaxPool layer:

The pooling operation involves sliding a two-dimensional filter over each channel of feature map and summarising the features lying within the region covered by the filter.
For a feature map having dimensions **$n_h \times n_w \times n_c$**, the dimensions of output obtained after a pooling layer is

$(n_h - f + 1) / s \times (nw - f + 1)/s \times nc$

Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

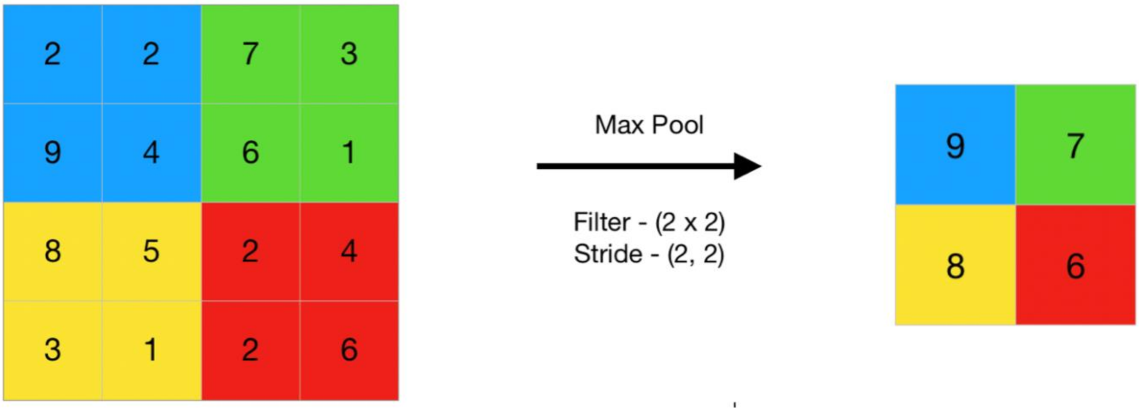

Figure 6 MaxPool in CNN model

5.2.3 Batch Normalization:

Batch normalization, it is a process to make neural networks faster and more stable through adding extra layers in a deep neural network. The new layer performs the standardizing and normalizing operations on the input of a layer coming from a previous layer.

Batch normalization takes place in following steps:

- Normalization of the Input

Normalization is the process of transforming the data to have a mean zero and standard deviation one. In this step we have our batch input from layer h, first, we need to calculate the mean of this hidden activation.

$$\mu = \frac{1}{m} \sum h_i$$

Here, m is the number of neurons at layer h.Once we have meant at our end, the next step is to calculate the standard deviation of the hidden activations.

$$\sigma = \left[\frac{1}{m} \sum (h_i - \mu)^2 \right]^{1/2}$$

Further, as we have the mean and the standard deviation ready. We will normalize the hidden activations using these values. For this, we will subtract the mean from each input and divide the whole value with the sum of standard deviation and the smoothing term (ε).

- Rescaling of Offsetting

In the final operation, the re-scaling and offsetting of the input take place. Here two components of the BN algorithm come into the picture, γ(gamma) and β (beta). These parameters are used for re-scaling (γ) and shifting(β) of the vector containing values from the previous operations.

$$h_i = \gamma \, h_{i(norm)} + \beta$$

These two are learnable parameters, during the training neural network ensures the optimal values of γ and β are used. That will enable the accurate normalization of each batch.

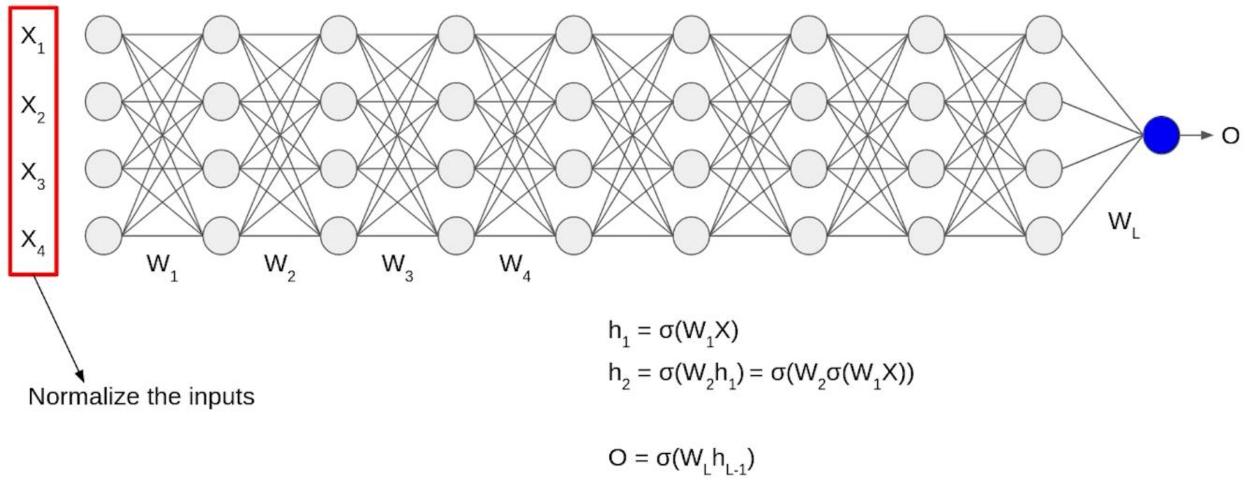

Figure 7 Batch Normalization in CNN model

5.2.4 Dense Layer:

It is a fully connected layer. Each node in this layer is connected to the previous layer i.e densely connected. This layer is used at the final stage of CNN to perform classification

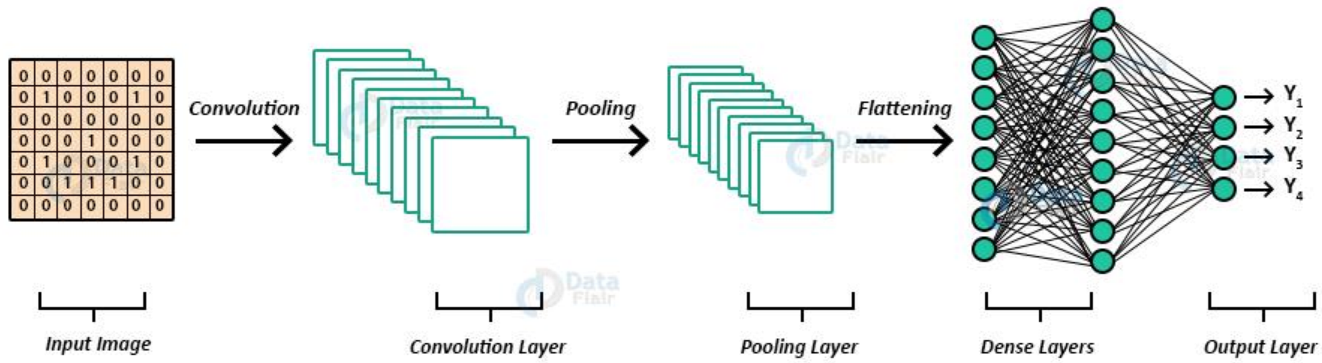

Figure 8 Dense layer in CNN model

Chapter 6: Results

6.1 Model Output:

These images are further passed through the training model, using this images the model is trained and after completing the training the model the testing and validation accuracy is measured.

The learning rate in the model is set as 0.0001 and the decay steps and decay rate is set at 10000 and 0.96 respectively.The amount that the weights are updated during training is referred to as the step size or the “learning rate”.

The model is runned for 20 epochs cycle after completing the 20 epocs the model training accuracy is measured at 98.02% and model validation accuracy is given as 99.5%, model training loss and model validation loss obtained is 2.3% and 1.2% respectively.

The black and purple lines depict the training accuracy and validation accuracy which should be higher than the training loss and validation loss which are depicted by the blue and red line in the graph shown below. Validation accuracy is the accuracy of the trained model to predict the validation dataset correctly; higher validation accuracy means that trained model is performing in the best possible way and it will predict the accurately in real world.

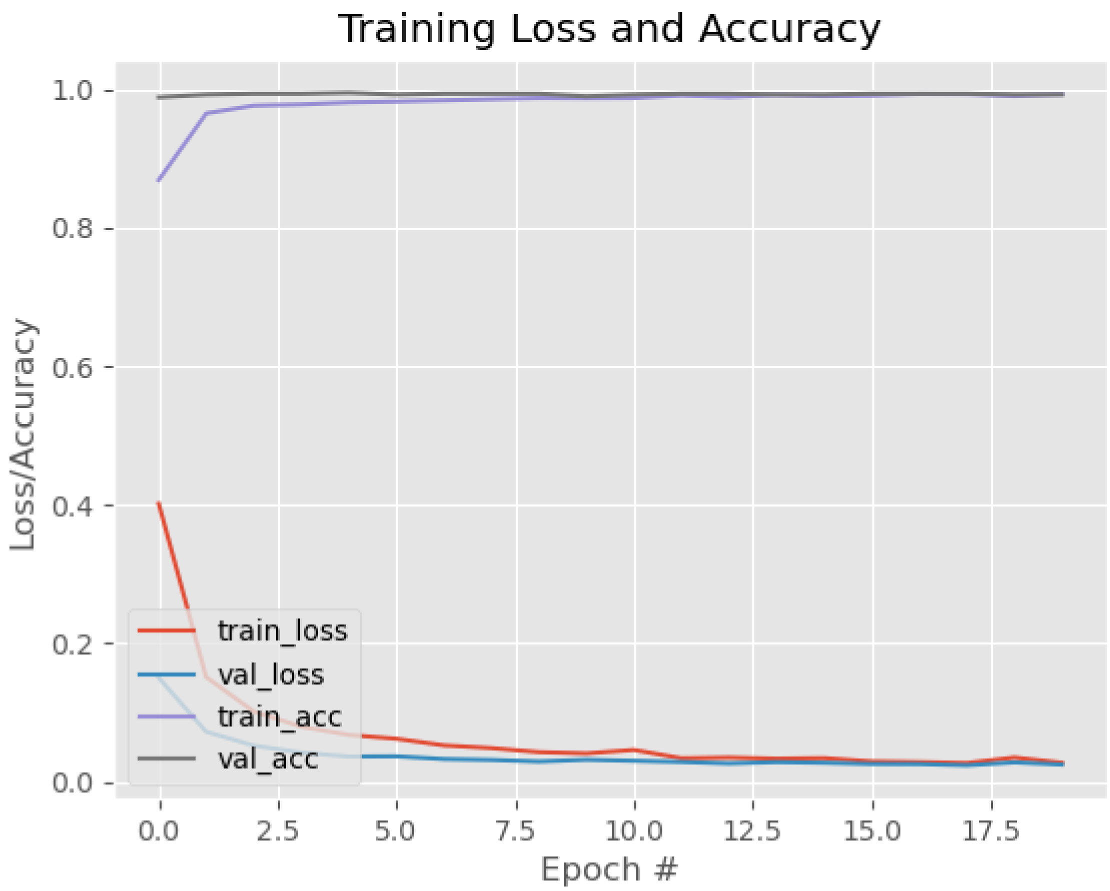

Figure 9 Training Loss and Accuracy

6.2 Prediction Output:

6.2.1 Without Mask:

After the model is trained with the help of system camera we will classify the image captured as mask and unmask

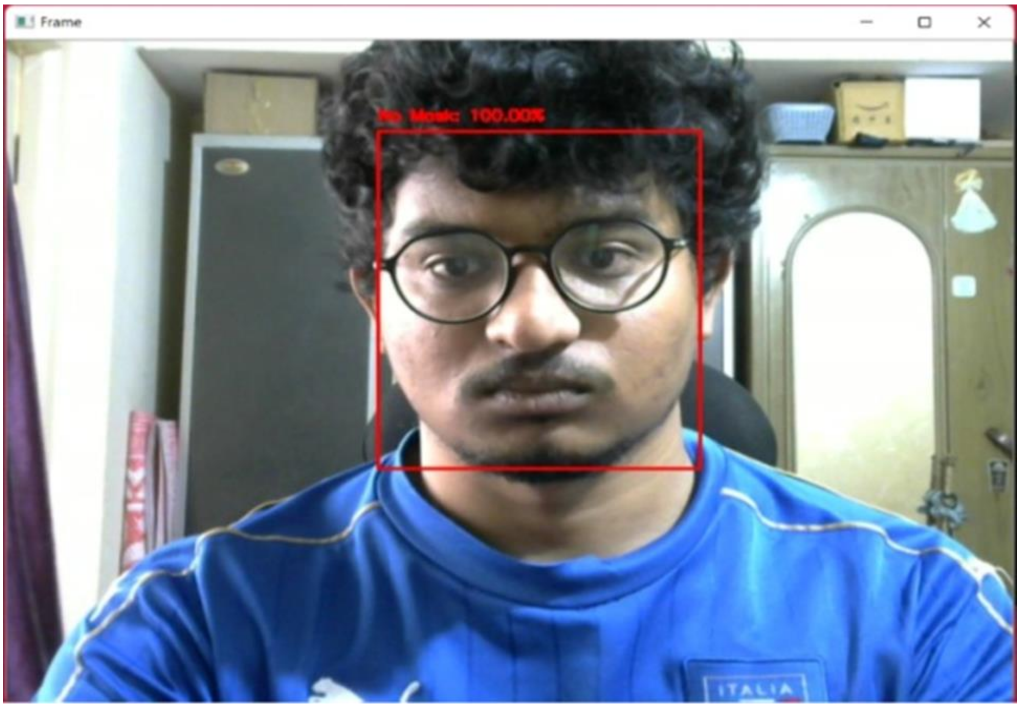

Figure 10 Without Mask (Single Person)

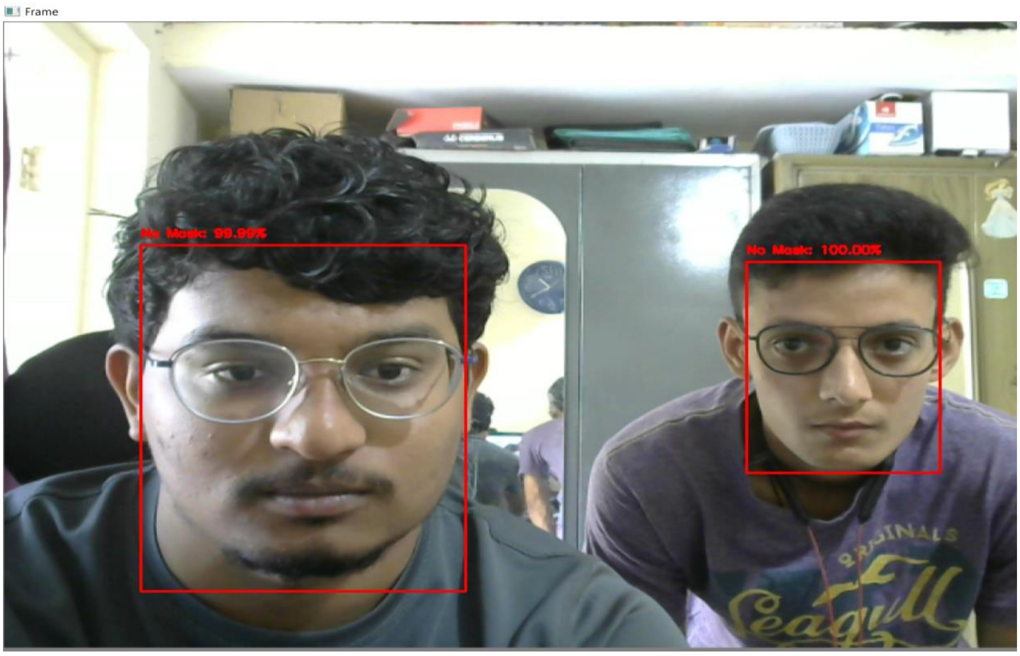

Figure 11 Without Mask (Multiple Person)

6.2.2 With Mask:

When a person Identifies the person not wearing Mask And those details not in the database it tries Match faces in the database. A bounding box drawn over the face of the person describes weather the person is wearing a mask

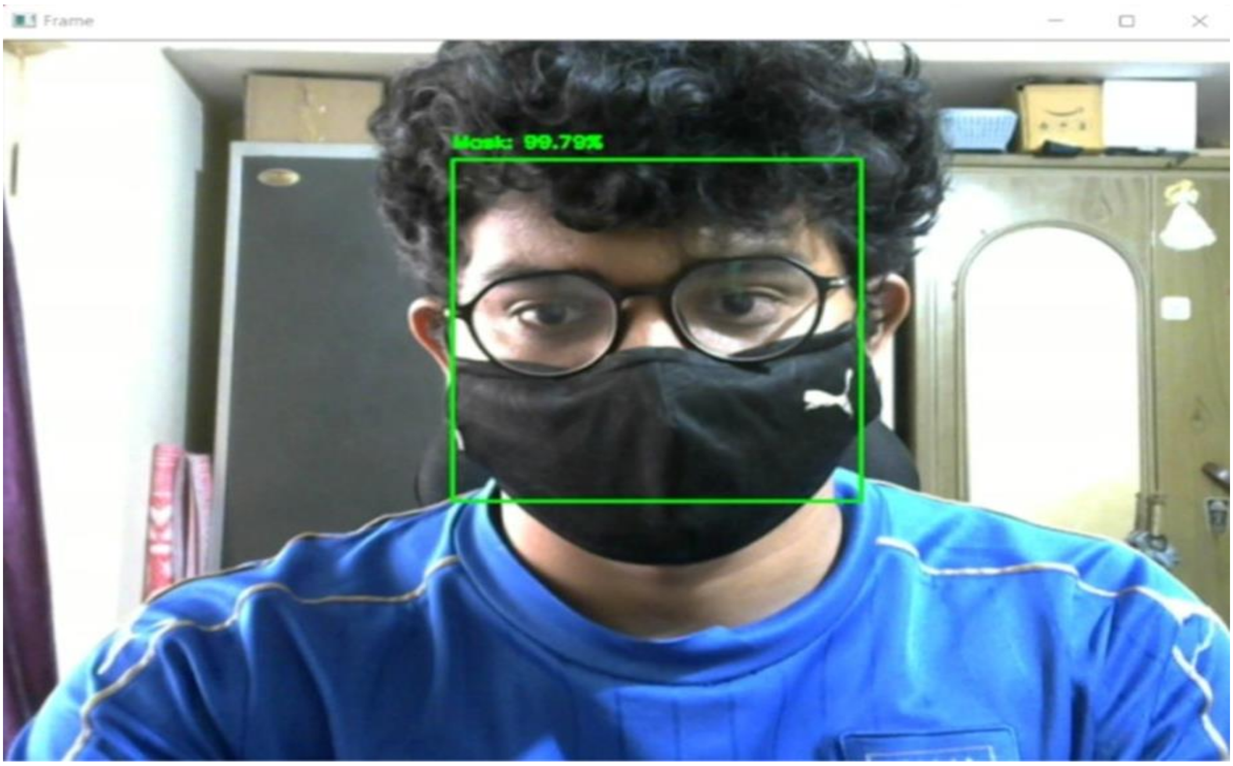

Figure 12 With Mask (Single Person)

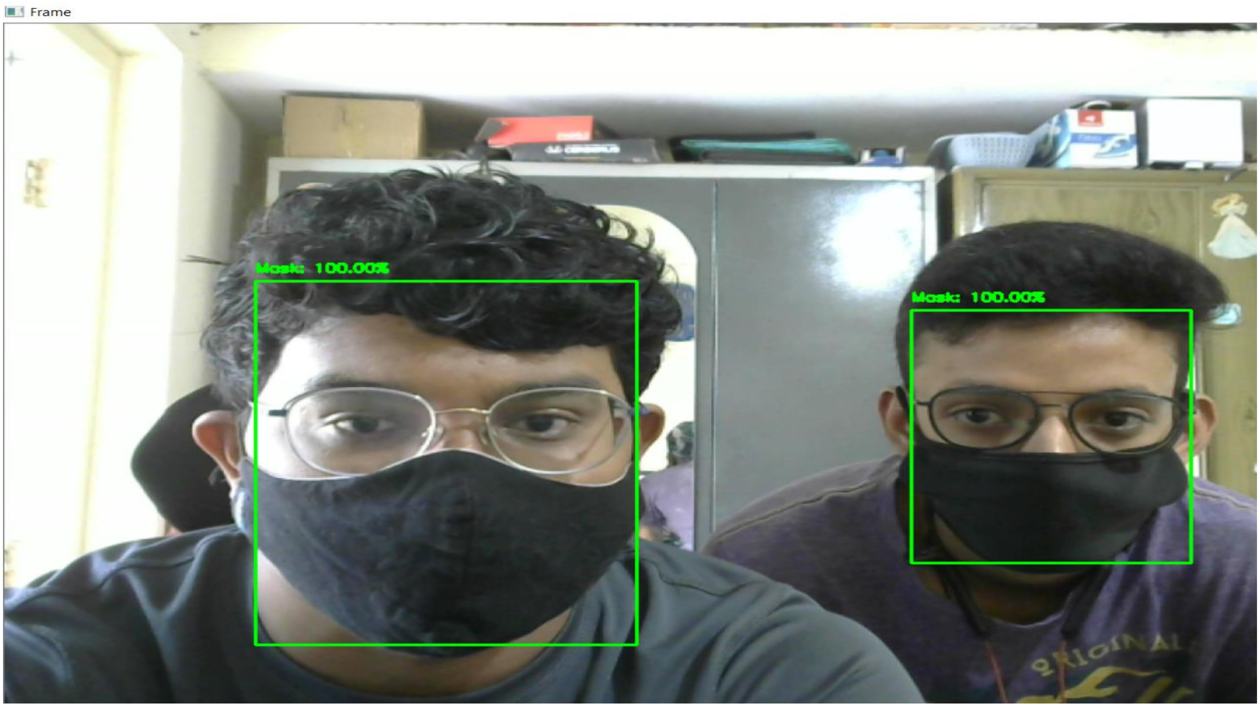

Figure 13 With Mask (Multiple Person)

6.2.3 One Person with Mask One without Mask:

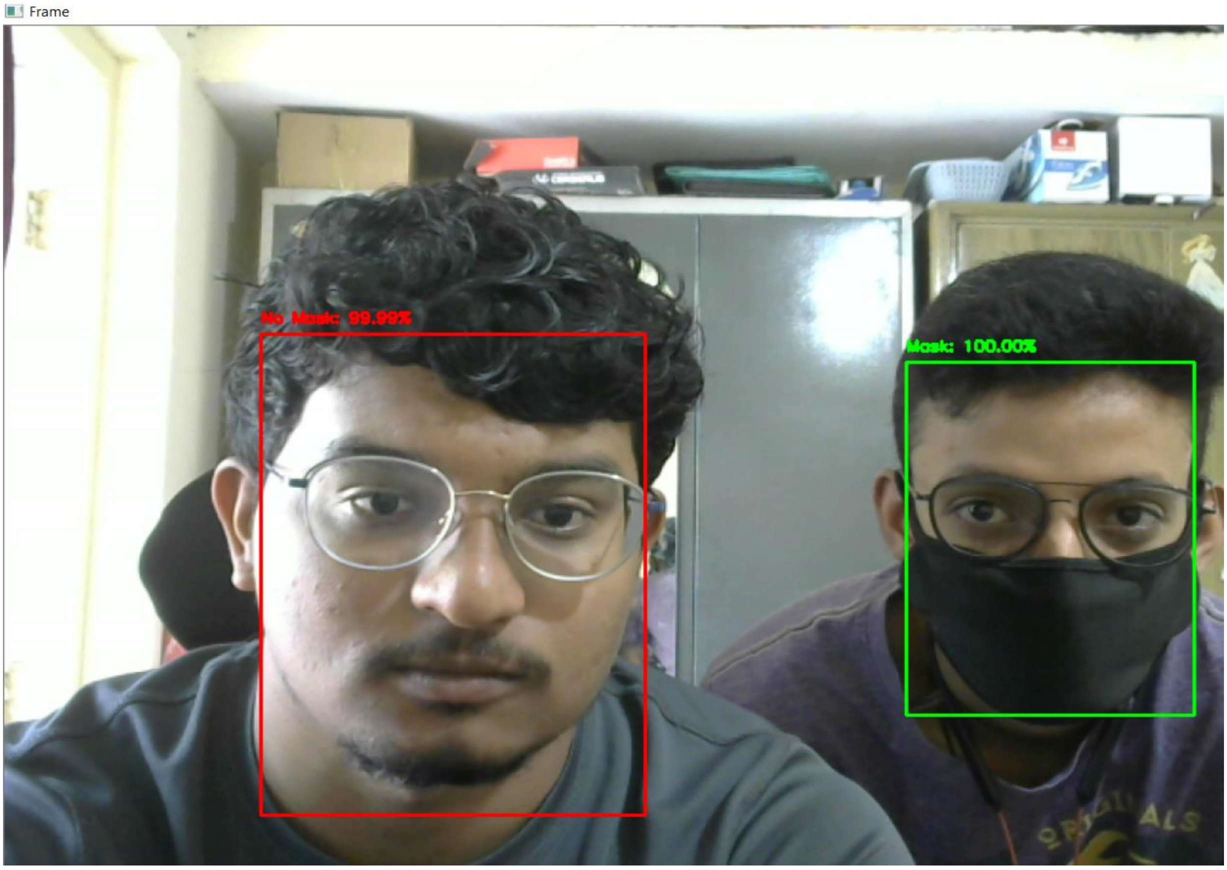

Figure 14 One Person with Mask One without Mask

Chapter 7: Conclusion and Future Scope

7.1 Scope:

The proposed system can be enhanced to detect multiple faces available on the screen using multi colour channels.

7.2 Conclusion:

With the increasing number of COVID cases all over the world, a system to replace humans to check masks on the faces of people is greatly needed. This system satisfies that need. This system can be employed in public places like railway stations and malls. It will be of great help in companies and huge establishments where there will be a lot of workers. This system will be of great help because it is easy to evaluate whether the person is wearing a mask or if he is not wearing a mask.

Chapter 8: Literature cited

[1] Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV.Arjya Das,Mohammad Wasif Ansari,Rohini Basak.2020.

[2] An Approach to Face Detection and Recognition.Divya Meena,Ravi Sharan.2017

[3] Face Mask Detection Using OpenCV Harish Adusumalli, D Kalyani, R.Krishna.2021

[4] Advice on the use of masks in the context of COVID-19.WHO.2019

[5] Face Mask Detection using Transfer Learning of InceptionV3 Year of publication: 2020 Authors: G. Jignesh Chowdary, Narinder Singh Punn ,Sanjay Kumar Sonbhadra and Sonali Agarwal

Chapter 9: Appendix

A. Research Papers Referred

Face Mask Detection Using OpenCV

Harish Adusumalli
Department of CSE
K L Education Foundation, Green Fields,Vaddeswaram, Guntur Dist, Andhra Pradesh, India - 522502
adusumalliharish48@gmail.com

D. Kalyani
Department of CSE
K L Education Foundation, Green Fields,Vaddeswaram, Guntur Dist, Andhra Pradesh, India - 522502
kalyani.digala@gmail.com

R.Krishna Sri
Department of CSE
K L Education Foundation, Green Fields,Vaddeswaram, Guntur Dist, Andhra Pradesh, India - 522502
ravurikrishnasri21@gmail.com

M.Pratapteja
Department of CSE
K L Education Foundation, Green Fields,Vaddeswaram, Guntur Dist, Andhra Pradesh, India - 522502
pratapteja11@gmail.com

P V R D Prasada Rao
Department of CSE
K L Education Foundation, Green Fields,Vaddeswaram, Guntur Dist, Andhra Pradesh, India – 522502
pvrdprasad@kluniversity.in

***Abstract*— The COVID-19 pandemic is causing a worldwide emergency in healthcare. This virus mainly spreads through droplets which emerge from a person infected with coronavirus and poses a risk to others. The risk of transmission is highest in public places. One of the best ways to stay safe from getting infected is wearing a face mask in open territories as indicated by the World Health Organization (WHO). In this project, we propose a method which employs TensorFlow and OpenCV to detect face masks on people. A bounding box drawn over the face of the person describes weather the person is wearing a mask or not. If a person's face is stored in the database, it detects the name of the person who is not wearing face mask and an email will be sent to that person warning them that they are not wearing a mask so that they can take precautions.**

Keywords - COVID-19, Tensorflow, OpenCV, Face Mask, Image Processing, Computer Vision

I. Introduction

COVID-19 had a massive impact on human lives. The pandemic lead to the loss of millions and affected the lives of billions of people. Its negative impact was felt by almost all commercial establishments, education, economy, religion, transport, tourism, employment, entertainment, food security and other industries. According to WHO(World Health Organization), 55.6 million people were infected with Coronavirus and 1.34 million people died because of it as of November 2020. This stands next to black death which almost took the lives of 60 percent of population in Europe in the 14th century. After the person gets infected, it takes almost fourteen days for the virus to grow in the body of its host and affect them and in the meantime, it spreads to almost everyone who is in contact with that person. So, it is extremely hard to keep the track of the spread of COVID-19.

COVID-19 mainly spreads through droplets produced as a result of coughing or sneezing by an infected person. This transfers the virus to any person who is in direct close contact (within one-meter distance) with the person suffering from coronavirus. Because of this, the virus spreads rapidly among the masses. With the nationwide lockdowns being lifted, it has become even harder to track and control the virus. Face masks are an effective method to control the spread of virus. It had been found that wearing face masks is 96% effective to stop the spread of virus. The governments, all over the world, have imposed strict rules the everyone should wear masks while they go out. But still, some people may not wear masks and it is hard to check weather everyone is wearing mask or not. In such cases, computer vision will be of great help.

There are no efficient face mask detection applications to detect weather the person is wearing face mask or not. This increases the demand for an efficient system for detecting face masks on people for transportation means, densely populated areas, residential districts, large-scale manufacturers and other enterprises to ensure safety. This project uses machine learning classification using OpenCV and Tensorflow to detect facemasks on people.

MACHINE LEARNING CLASSIFIERS:

These are used to predict the class/target/labels/categories of a given data points. Classification belongs to the category of supervised learning in which the targets are provided with input data. They are used in many applications like medical diagnosis, spam detection, target marketing etc. They use a mapping function (f) from input variables (X) to discrete output variables(Y).

OPENCV:

OpenCV is an open-source library which is primarily used for Computer Vision Applications. This contains many functions and algorithms for Motion tracking, Facial recognition, Object Detection, Segmentation and recognition and many other applications. Images and real time video streams can be manipulated to suit different needs using this library.

TENSORFLOW:

It is an open-source machine learning framework to build and train neural networks. It has a collection of tools, libraries and community resources which helps in easy building of deployment of ML powered applications. This is developed and maintained by Google and was released in 2015.

II. RELATED WORK

In [1] the authors used PCA (Principal Component Analysis) method to identify faces with masks, which is essential in the field of security. This is one of the few works which concentrated on detection of human faces where they are wearing masks. They found that the accuracy in human face detection decreases by 70% when a face mask is present.

In [2] the authors have developed a method to identify how a person is wearing the face mask. They were able to classify three categories of facemask-wearing condition namely correct facemask-wearing, incorrect facemask- wearing, and no facemask-wearing. This method achieved over 98% accuracy in detection.

In [3], the researchers proposed a method for the identification of faces using Generalized Intersection over Union (GIoU) based on Mask R-CNN. They proposed this method to reduce the background noise by correctly identifying the face instead of bounding box which adds noise to the face features and reduces the accuracy of detection.

Nicolae-Cătălin Ristea, Radu Tudor Ionescu [4] proposed a novel data augmentation approach for mask detection from speech.Original and translated utterances were changed over into spectrograms were given as inputs to a bunch of ResNet neural organizations with different depths. In [5] the authors have employed a GAN-based network using two discriminators for the removal of face mask from a face and reconstruct the face without the face mask using the CelebA dataset.

III. METHODOLOGY

Dataset Collection: The dataset was collected from Kaggle Repository and was split into training and testing data after its analysis.

Training a model to detect face masks: A default OpenCV module was used to obtain faces followed by training a Keras model to identify face mask.

Detecting the person not wearing a mask: A open CV model was trained to detect the names of the people who are not wearing masks by referring the database.

Sending the e-mail: The system was designed to send an e-mail to the person not wearing a mask using smtplib.

IV. WORKING

This project makes the use of OpenCV,Caffe-based face detector, Keras, TensorFlow and MobileNetV2 for the detection of face mask on humans. The dataset which is being used contains 3835 images out of which 1916 images have people with masks in them and 1919 people without masks in them.

First a base model is generated. This is done my using Keras and MobileNetV2. First a base model is generated and a head model is generated on top of that. The head model consists of a network with 128 layers, an activation function of "Relu" and a dropout of 0.5 followed by another network with 2 layers and an activation function "softmax". All these three layers combined, will give out model which will be trained.

The generated model is then trained with the labeled dataset by splitting it into two portions. One portion contains 75 percent images and it is used for training. The remaining portion contains the remaining 25 percent of images and is used for testing the model accuracy. After the model is trained, it can be used for detection of facemask on human faces.

The trained model is loaded and image which contains human faces with or without masks or a continuous video stream with humans is given as input. The image or a frame of the video, in case the input is a video stream, is first sent to the default face detector module for the detection of human faces. This is done by resizing the image or the video frame first, followed by detecting the blob in it. This detected blob is sent to the face detector model which outputs only the cropped face of a person without the background. This face is given as the input to the model which we trained earlier. This outputs weather there is a mask or not.

Another model is trained with the faces of humans. The images used for the training of the model are provided with the name and email address of that person as the labels of those images. This is done by using Open CV. When an input image is given to the CV model, it detects the face of a person and asks the user to provide the name and email address of that person which will be stored in the database. The output of the first model is given as the input to this model. This face will be compared with the persons present in the database. And if his face matches, then a bounding box will be drawn over his face with his name on it and an email and Sms will be sent to him that he is not wearing a mask. Else, only the words "Mask" will be present below the bounding box if the person is wearing a mask and "No Mask" if the person is not wearing one.

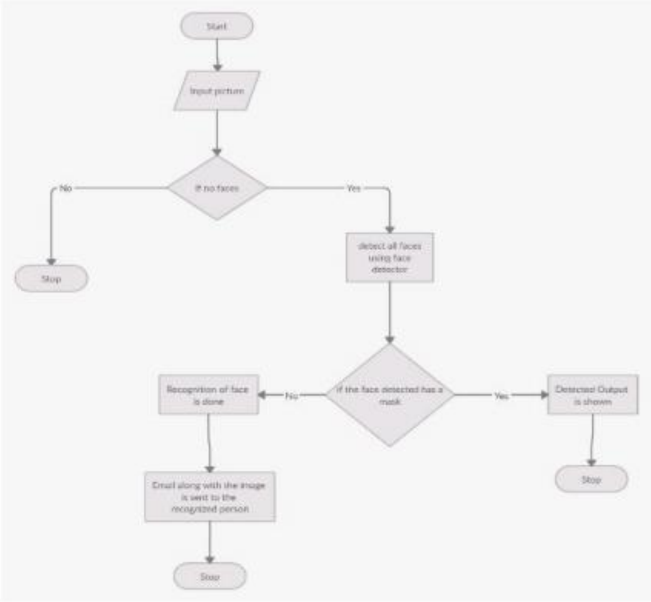

Figure 4.1: Flowchart of the Project

Figure 5.2:When Person Wearing Mask.A Bounding Box Drawn Over the Face of the Person Describes The Person Wearing Mask.

V.RESULTS

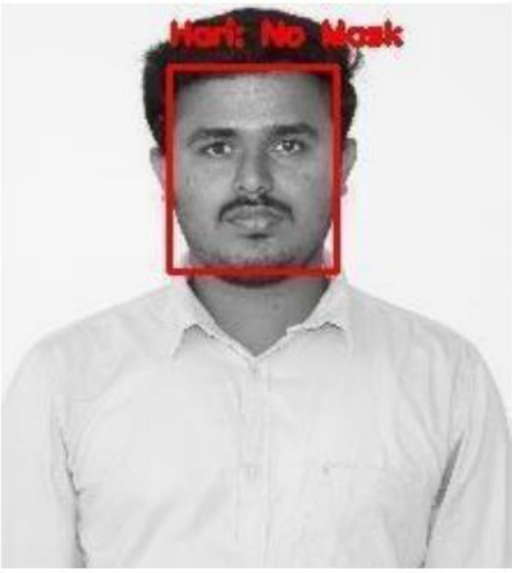

Figure 5.1: When The Person Not Wearing the Mask . A bounding box drawn over the face of the person describes weather the person is wearing a mask or not. If a person's face is stored in the database, it detects the name of the person who is not wearing face mask.

Fig5.4: when a person Identifies the not wearing Mask And those details not in the database it try's Match faces in the database. A bounding box drawn over the face of the person describes weather the person is wearing a mask.

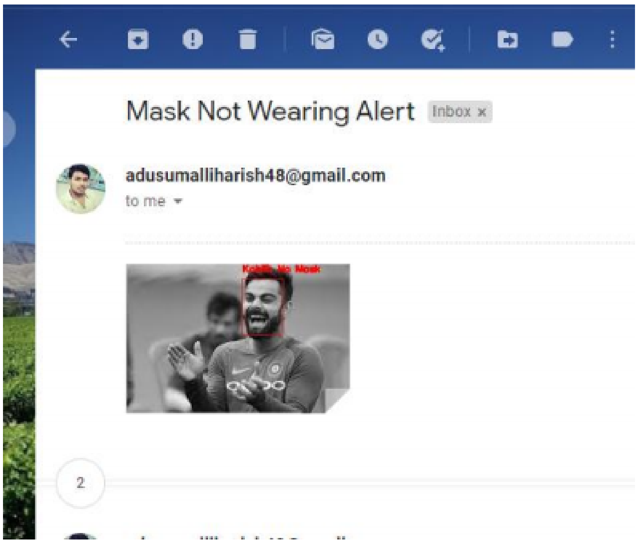

Figure 5.4: A bounding box drawn over the face of the person describes weather the person is wearing a mask or not. If a person's face is stored in the database, it detects the name of the person who is not wearing face mask and an email will be sent to that person warning them that they are not wearing a mask so that they can take precautions.

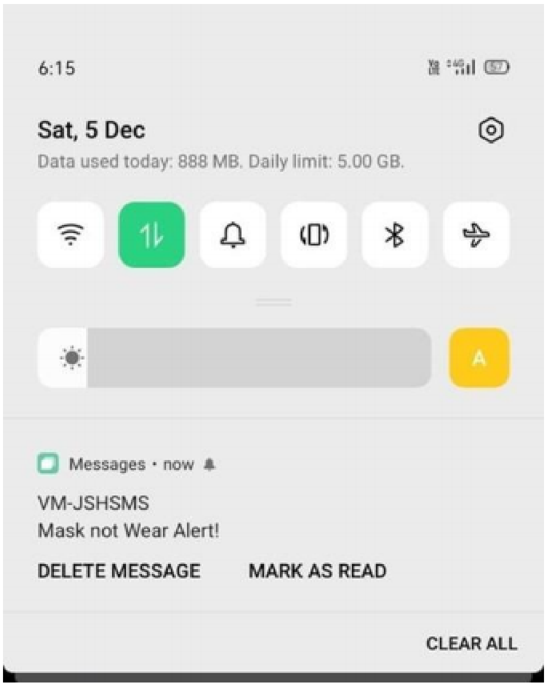

Figure 5.5:A bounding box drawn over the face of the person describes weather the person is wearing a mask or not. If a person's face is stored in the database, it detects the name of the person who is not wearing face mask and SMS will be sent to that person warning them that they are not wearing a mask so that they can take precautions.

VI. CONCLUSION

With the increasing number of COVID cases all over the world, a system to replace humans to check masks on the faces of people is greatly needed. This system satisfies that need. This system can be employed in public places like railway stations and malls. It will be of a great help in companies and huge establishments where there will be a lot of workers. This system will be of a great help there because it is easy to obtain and store the data of the employees working in that Company and will very easy find the people who are not wearing the mask and a mail will sent to that respective person to take Precautions not wearing mask.

Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV

Arjya Das
Department of Information Technology
Jadavpur University
Kolkata, India
arjyadas1999@gmail.com

Mohammad Wasif Ansari
Department of Information Technology
Jadavpur University
Kolkata, India
razamoeezraza@gmail.com

Rohini Basak
Department of Information Technology
Jadavpur University
Kolkata, India
visitrohinihere@gmail.com

***Abstract*—COVID-19 pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn. The proposed method detects the face from the image correctly and then identifies if it has a mask on it or not. As a surveillance task performer, it can also detect a face along with a mask in motion. The method attains accuracy up to 95.77% and 94.58% respectively on two different datasets. We explore optimized values of parameters using the Sequential Convolutional Neural Network model to detect the presence of masks correctly without causing over-fitting.**

Keywords*—*Coronavirus, Covid-19, Machine Learning, Face Mask Detection, Convolutional Neural Network, TensorFlow

I. Introduction

According to the World Health Organization (WHO)'s official Situation Report – 205, coronavirus disease 2019 (COVID-19) has globally infected over 20 million people causing over 0.7million deaths [1]. Individuals with COVID-19 have had a wide scope of symptoms reported – going from mellow manifestations to serious illness. Respiratory problems like shortness of breath or difficulty in breathing is one of them. Elder people having lung disease can possess serious complications from COVID-19 illness as they appear to be at higher risk [2]. Some common human coronaviruses that infect public around the world are 229E, HKU1, OC43, and NL63. Before debilitating individuals, viruses like 2019-nCoV, SARS-CoV, and MERS-CoV infect animals and evolve to human coronaviruses [3]. Persons having respiratory problems can expose anyone (who is in close contact with them) to infective beads. Surroundings of a tainted individual can cause contact transmission as droplets carrying virus may withal arrive on his adjacent surfaces [4].

To curb certain respiratory viral ailments, including COVID-19, wearing a clinical mask is very necessary. The public should be aware of whether to put on the mask for source control or aversion of COVID-19. Potential points of interest of the utilization of masks lie in reducing vulnerability of risk from a noxious individual during the "pre-symptomatic" period and stigmatization of discrete persons putting on masks to restraint the spread of virus. WHO stresses on prioritizing medical masks and respirators for health care assistants[4]. Therefore, face mask detection has become a crucial task in present global society.

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 [5]. 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.

The rest of the paper is organized as follows: Section II explores related work associated with face mask detection. Section III discusses the nature of the used dataset. Section IV presents the details of the packages incorporated to build the proposed model. Section V gives an overview of our method. Experimental results and analysis are reported in section VI. Section VII concludes and draws the line towards future works.

II. Related Work

In face detection method, a face is detected from an image that has several attributes in it. According to [21], research into face detection requires expression recognition, face tracking, and pose estimation. Given a solitary image, the challenge is to identify the face from the picture. Face detection is a difficult errand because the faces change in size, shape, color, etc and they are not immutable. It becomes a laborious job for opaque image impeded by some other thing not confronting camera, and so forth. Authors in [22] think occlusive face detection comes with two major challenges: 1) unavailability of sizably voluminous datasets containing both masked and unmasked faces, and 2) exclusion of facial expression in the covered area. Utilizing the locally linear embedding (LLE) algorithm and the dictionaries trained on an immensely colossal pool of masked faces, synthesized mundane faces, several mislaid expressions can be recuperated and the ascendancy of facial cues can be mitigated to great extent. According to the work reported in [11], convolutional neural network (CNNs) in computer vision comes with a strict constraint regarding the size of the input image. The prevalent practice reconfigures the images before fitting them into the network to surmount the inhibition.

Here the main challenge of the task is to detect the face from the image correctly and then identify if it has a mask on it or not. In order to perform surveillance tasks, the proposed method should also detect a face along with a mask in motion.

III. Dataset

Two datasets have been used for experimenting the current method. Dataset 1 [16] consists of 1376 images in which 690 images with people wearing face masks and the rest 686 images with people who do not wear face masks. Fig. 1 mostly contains front face pose with single face in the frame and with same type of mask having white color only.

Fig. 1. Samples from Dataset 1 including faces without masks and with masks

Dataset 2 from Kaggle [17] consists of 853 images and its countenances are clarified either with a mask or without a mask. In fig. 2 some face collections are head turn, tilt and slant with multiple faces in the frame and different types of masks having different colors as well.

Fig. 2. Samples from Dataset 2 including faces without masks and with masks

IV. Incorporated Packages

A. *TensorFlow*

TensorFlow, an interface for expressing machine learning algorithms, is utilized for implementing ML systems into fabrication over a bunch of areas of computer science, including sentiment analysis, voice recognition, geographic information extraction, computer vision, text summarization, information retrieval, computational drug discovery and flaw detection to pursue research [18]. In the proposed model, the whole Sequential CNN architecture (consists of several layers) uses TensorFlow at backend. It is also used to reshape the data (image) in the data processing.

B. *Keras*

Keras gives fundamental reflections and building units for creation and transportation of ML arrangements with high iteration velocity. It takes full advantage of the scalability and cross-platform capabilities of TensorFlow. The core data structures of Keras are layers and models [19]. All the layers used in the CNN model are implemented using Keras. Along with the conversion of the class vector to the binary class matrix in data processing, it helps to compile the overall model.

C. *OpenCV*

OpenCV (Open Source Computer Vision Library), an open-source computer vision and ML software library, is utilized to differentiate and recognize faces, recognize objects, group movements in recordings, trace progressive modules, follow eye gesture, track camera actions, expel red eyes from pictures taken utilizing flash, find comparative pictures from an image database, perceive landscape and set up markers to overlay it with increased reality and so forth [20]. The proposed method makes use of these features of OpenCV in resizing and color conversion of data images.

V. The Proposed Method

The proposed method consists of a cascade classifier and a pre-trained CNN which contains two 2D convolution layers connected to layers of dense neurons. The algorithm for face mask detection is as follows:

Algorithm 1: Face Mask Detection

```
Input: Dataset including faces with and without masks
Output: Categorized image depicting the presence of face mask
for each image in the dataset do
    Visualize the image in two categories and label them
    Convert the RGB image to Gray-scale image
    Resize the gray-scale image into 100 x 100
    Normalize the image and convert it into 4 dimensional array
end
for building the CNN model do
    Add a Convolution layer of 200 filters
    Add the second Convolution layer of 100 filters
    Insert a Flatten layer to the network classifier
    Add a Dense layer of 64 neurons
    Add the final Dense layer with 2 outputs for 2 categories
end
Split the data and train the model
```

A. Data 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 [6]. The proposed method deals with image and video data using Numpy and OpenCV.

a) 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 [7].

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}

b) 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 nonessential 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 [8].

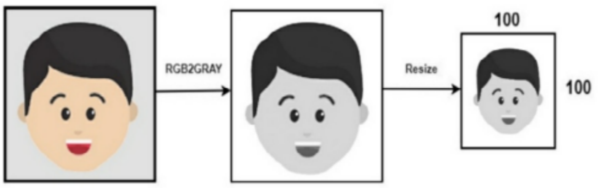

Fig. 3. Conversion of a RGB image to a Gray Scale image of 100x100 size

We use the function *cv2.cvtColor(input_image, flag)* for changing the color space. Here flag determines the type of conversion [9]. 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.

c) 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 [10]. 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. [11].

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.

B. Training of Model

a) Building the model using CNN architecture: CNN has become ascendant in miscellaneous computer vision tasks [12]. 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 [13]. 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 [14]. 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) [15].

As shown in fig, 4, 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.

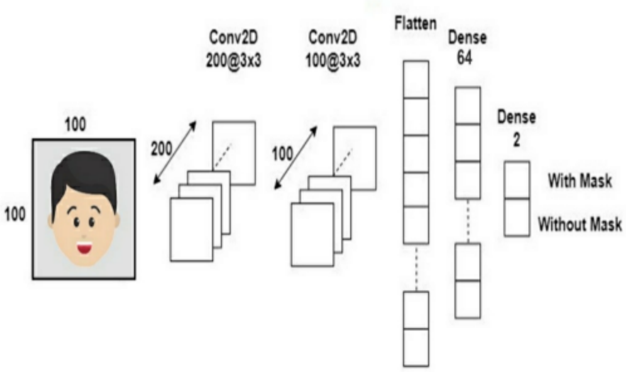

Fig. 4. Convolutional Neural Network architecture

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".

b) Splitting the data and training the CNN model: After setting the blueprint to analyze the data, the model needs to be trained using a specific dataset and then to be tested against a different dataset. A proper model and optimized *train_test_split* help to produce accurate results while making a prediction. The test_size is set to 0.1 i.e. 90% data of the dataset undergoes training and the rest 10% goes for testing purposes. The validation loss is monitored using *ModelCheckpoint*. Next, the images in the training set and the test set are fitted to the Sequential model. Here, 20% of the training data is used as validation data. The model is trained for 20 epochs (iterations) which maintains a trade-off between accuracy and chances of overfitting. Fig. 5 depicts visual representation of the proposed model.

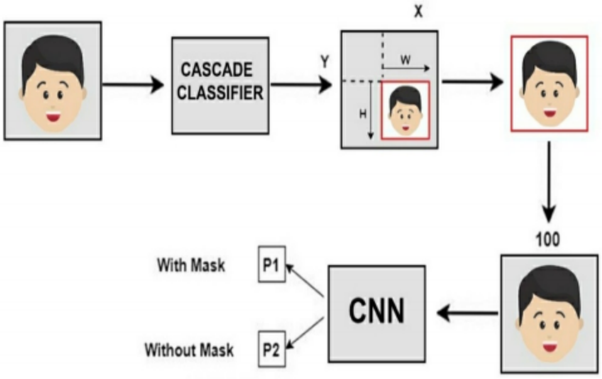

Fig. 5. Overview of the Model

VI. Result and Analysis

The model is trained, validated and tested upon two datasets. Corresponding to dataset 1, the method attains accuracy up to 95.77% (shown in fig. 7). Fig. 6 depicts how this optimized accuracy mitigates the cost of error. Dataset 2 is more versatile than dataset 1 as it has multiple faces in the frame and different types of masks having different colors as well. Therefore, the model attains an accuracy of 94.58% on dataset 2 as shown in Fig. 9. Fig. 8 depicts the contrast between training and validation loss corresponding to dataset 2. One of the main reasons behind achieving this accuracy lies in *MaxPooling*. It provides rudimentary translation invariance to the internal representation along with the reduction in the number of parameters the model has to learn. This sample-based discretization process down-samples the input representation consisting of image, by reducing its dimensionality. Number of neurons has the optimized value of 64 which is not too high. A much higher number of neurons and filters can lead to worse performance. The optimized filter values and pool_size help to filter out the main portion (face) of the image to detect the existence of mask correctly without causing over-fitting.

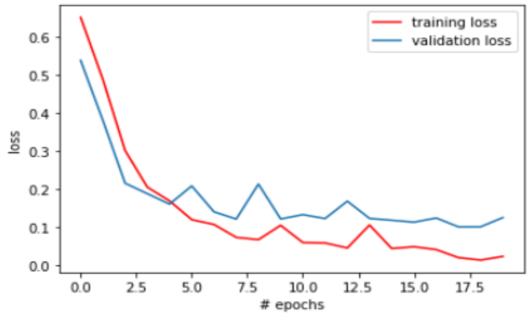

Fig. 6. # epochs vs loss corresponding to dataset 1

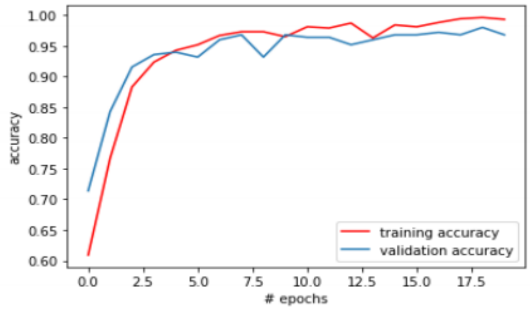

Fig. 7. # epochs vs accuracy corresponding to dataset 1

The system can efficiently detect partially occluded faces either with a mask or hair or hand. It considers the occlusion degree of four regions – nose, mouth, chin and eye to differentiate between annotated mask or face covered by hand. Therefore, a mask covering the face fully including nose and chin will only be treated as "with mask" by the model.

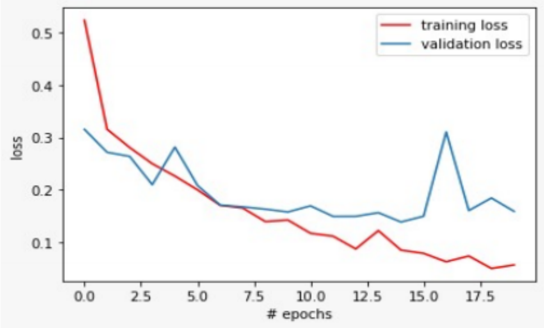

Fig. 8. # epochs vs loss corresponding to dataset 2

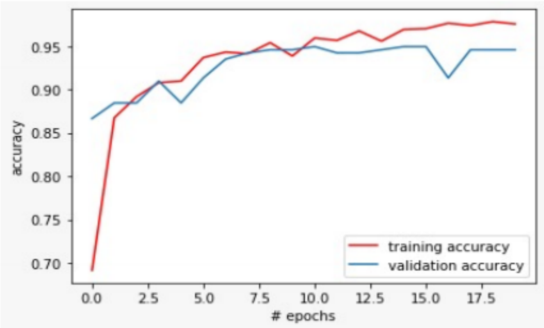

Fig. 9. # epochs vs accuracy corresponding to dataset 2

The main challenges faced by the method mainly comprise of varying angles and lack of clarity. Indistinct moving faces in the video stream make it more difficult. However, following the trajectories of several frames of the video helps to create a better decision – "with mask" or "without mask".

VII. Conclusions

In this paper, we briefly explained the motivation of the work at first. Then, we illustrated the learning and performance task of the model. Using basic ML tools and simplified techniques the method has achieved reasonably high accuracy. It can be used for a variety of applications.Wearing a mask may be obligatory in the near future, considering the Covid-19 crisis. Many public service providers will ask the customers to wear masks correctly to avail of their services. The deployed model will contribute immensely to the public health care system. In future it can be extended to detect if a person is wearing the mask properly or not. The model can be further improved to detect if the mask is virus prone or not i.e. the type of the mask is surgical, N95 or not.

References

[1] W.H.O., "Coronavirus disease 2019 (covid-19): situation report, 205". 2020

[2] "Coronavirus Disease 2019 (COVID-19) – Symptoms", Centers for Disease Control and Prevention, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html. 2020.

[3] "Coronavirus — Human Coronavirus Types — CDC", Cdc.gov, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/types.html. 2020.

[4] W.H.O., "Advice on the use of masks in the context of COVID-19: interim guidance", 2020.

[5] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask detector", arXiv.org, 2020. [Online]. Available: https://arxiv.org/abs/2005.03950. 2020.

[6] B. Suvarnamukhi and M. Seshashayee, "Big Data Concepts and Techniques in Data Processing", International Journal of Computer Sciences and Engineering, vol. 6, no. 10, pp. 712-714, 2018. Available: 10.26438/ijcse/v6i10.712714.

[7] F. Hohman, M. Kahng, R. Pienta and D. H. Chau, "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers," in IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 8, pp. 2674-2693, 1 Aug. 2019, doi: 10.1109/TVCG.2018.2843369.

[8] C. Kanan and G. Cottrell, "Color-to-Grayscale: Does the Method Matter in Image Recognition?", PLoS ONE, vol. 7, no. 1, p. e29740, 2012. Available: 10.1371/journal.pone.0029740.

[9] Opencv-python-tutroals.readthedocs.io. 2020. Changing Colorspaces — Opencv-Python Tutorials 1 Documentation. [online] Available at:https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_colorspaces/py_colorspaces.html. 2020.

[10] M. Hashemi, "Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation", Journal of Big Data, vol. 6, no. 1, 2019. Available: 10.1186/s40537-019-0263-7 . 2020.

[11] S. Ghosh, N. Das and M. Nasipuri, "Reshaping inputs for convolutional neural network: Some common and uncommon methods", Pattern Recognition, vol. 93, pp. 79-94, 2019. Available: 10.1016/j.patcog.2019.04.009.

[12] R. Yamashita, M. Nishio, R. Do and K. Togashi, "Convolutional neural networks: an overview and application in radiology", Insights into Imaging, vol. 9, no. 4, pp. 611-629, 2018. Available: 10.1007/s13244-018-0639-9.

[13] "Guide to the Sequential model - Keras Documentation", Faroit.com, 2020. [Online]. Available: https://faroit.com/keras-docs/1.0.1/getting-started/sequential-model-guide/. 2020.

[14] Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S., 2020. Activation Functions: Comparison Of Trends In Practice And Research For Deep Learning. [online] arXiv.org. Available at: https://arxiv.org/abs/1811.03378. 2020.

[15] K. Team, "Keras documentation: MaxPooling2D layer", Keras.io, 2020. [Online]. Available: https://keras.io/api/layers/pooling_layers/max_pooling2d/. 2020.

[16] "prajnasb/observations", GitHub, 2020. [Online]. Available: https://github.com/prajnasb/observations/tree/master/experiements/data. 2020.

[17] "Face Mask Detection", Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/andrewmvd/face-mask-detection. 2020.

[18] "TensorFlow White Papers", TensorFlow, 2020. [Online]. Available: https://www.tensorflow.org/about/bib. 2020.

[19] K. Team, "Keras documentation: About Keras", Keras.io, 2020. [Online]. Available: https://keras.io/about. 2020.

[20] "OpenCV", Opencv.org, 2020. [Online]. Available: https://opencv.org/. 2020.

[21] D. Meena and R. Sharan, "An approach to face detection and recognition," 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, 2016, pp. 1-6, doi: 10.1109/ICRAIE.2016.7939462.

[22] S. Ge, J. Li, Q. Ye and Z. Luo, "Detecting Masked Faces in the Wild with LLE-CNNs," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 426-434, doi: 10.1109/CVPR.2017.53.

B. CO and PO Mappings

Yeshwantrao Chavan College of Engineering
Hingna Road, Wanadongri, NAGPUR - 441 110
Department of Electronics & Telecommunication Engineering
CO-PO MAPPING (Session 2021-22)

				PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
VIII	ET2451 : Major Project	**1**	Design and analyze application based electronic systems.	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
		2	Implement core / multidisciplinary / industry based electronics projects in cost effective manner.	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
		3	Communicate technical details effectively	**3**				**3**		**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

C. Project group information

Sr. No.	Name	Registration number	Email Id
1.	Ritik Pusdekar	18050070	ritikpusdekar@gmail.com
2.	Aakib Sayyad	18050078	aaqibsayyad@gmail.com
3.	Mohit Zade	18050013	mohitzade29@gmail.com
4.	Tejas Ubale	18050008	tejasubale18@gmail.com
5.	Rohit Samarth	18050080	rohitsamarth.199@gmail.com

Chapter 10: Relevance of Project

COVID-19 had a massive impact on human lives. The pandemic led to the loss of millions and affected the lives of billions of people. Its negative impact was felt by almost all commercial establishments, education, economy, religion, transport, tourism, employment, entertainment, food security and other industries. According to WHO(World Health Organization), 55.6 million people were infected with Coronavirus and 1.34 million people died because of it. After the person gets infected, it takes almost fourteen days for the virus to grow in the body of its host and affect them and in the meantime, it spreads to almost everyone who is in contact with that person. So, it is extremely hard to keep track of the spread of COVID-19. COVID-19 mainly spreads through droplets produced as a result of coughing or sneezing by an infected person. This transfers the virus to any person who is in direct close contact (within one-meter distance) with the person suffering from coronavirus. Because of this, the virus spreads rapidly among the masses. With the nationwide lockdowns being lifted, it has become even harder to track and control the virus.

Face masks are an effective method to control the spread of virus. It has been found that wearing face masks is 96% effective to stop the spread of virus. The governments, all over the world, have imposed strict rules that everyone should wear masks while they go out. But still, some people may not wear masks and it is hard to check whether everyone is wearing a mask or not. There are no efficient face mask detection applications to detect whether the person is wearing a face mask or not. This increases the demand for an efficient system for detecting face masks on people for transportation means, densely populated areas, residential districts, large-scale manufacturers and other enterprises to ensure safety. This project uses machine learning classification using OpenCV and Tensorflow to detect facemasks on people.

During this period it was important for companies and institutions to check if people are following the mandate and wearing masks. So the system we have made will help these companies and institutions in keeping track of people wearing masks and people not wearing masks. The project can be useful in two ways:

1. For companies to check if the customer availing their services is wearing a mask or not
2. For institutions to keep track of people who are not wearing a mask

By making sure that people are wearing masks while availing services or while walking around a crowded place is one of the best ways to reduce transmission of Covid-19 virus. The project aims to help in the compulsion of wearing masks in dire conditions such as a pandemic or an epidemic right now or even in the future when another virus strikes us.