A Mini Project Report

on

Face Mask Detector

Submitted in partial fulfilment of the requirements
of the degree

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

by

Krutika Bhagane

Nikhil Bhosale

Abhishek Gambre

Anish Ghogare

Supervisor

Mrs. Prajakta Pote

Department of Computer Engineering

Pillai Hoc College of Engineering And Technology, Rasayani

University Of Mumbai

(AY 2021-22)

CERTIFICATE

This is to certify that the Mini Project entitled "**Face Mask Detector"** is a bona fide work of **Krutika Bhagane, Nikhil Bhosale, Abhishek Gambre, Anish Ghogare** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering".**

(Mrs. Prajakta Pote)
Supervisor

(Prof. Rohini Bhosale)
Head of Department

(Dr.J.W. Bakal)
Principal

MINI PORJECT APPROVAL

This Mini Project entitled "**Face Mask Detector"** is a bona fide work of **Krutika Bhagane, Nikhil Bhosale, Abhishek Gambre, Anish Ghogare** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1. ___________________

(Internal Examiner Name & Sign)

2.____________________

(External Examiner name & Sign)

Date: __/__/____

Place: __________

LIST OF FIGURES

CONTENTS

ABSTRACT

In the recent times, the Coronaviruses that are a big family of different viruses have become very common, contagious and dangerous to the whole human kind. It spreads human to human by exhaling the infection breath, which leaves droplets of the virus on different surface which is then inhaled by other person and catches the infection too. So, it has become very important to protect ourselves and the people around us from this situation. We can take precautions such as social distancing, washing hands every two hours, using sanitizer, maintaining social distance and the most important wearing a mask. Public use of wearing a mask has become very common everywhere in the whole world now. From that the most affected and devastating condition is of India due to its extreme population in small area. This project proposes a method to detect the face mask is put on or not for offices, or any other work place with a lot of people coming to work. We have used convolutional neural network for the same. The model is trained on a real-world dataset and tested with live video streaming with a good accuracy. Further the accuracy of the model with different hyper parameters and multiple people at different distance and location of the frame is done.

Keywords— Face Mask Detection, Convolutional Neural Network, MobileNetV2, Corona virus Precaution.

ACKNOWLEDGEMENT

It is a privilege for us to have been associated with **Mrs. Prajakta Pote**, our guide, during this project work we have greatly benefited by her valuable suggestions and ideas. It is with great pleasure that we express our deep sense of gratitude to them for their valuable guidance, constant encouragement and patience throughout this work.

We are also indebted to **Prof. Rohini Bhosale** (Head of the Department of Computer Engineering) for extending the help to academic literature. We express our gratitude to **Dr. Bakal** (Principal) and **Mrs. Archana Arudkar** (Project coordinator) for their constant encouragement, cooperation, and support. We take This opportunity to thanks all our classmates for their company during the couse work and for the useful discussion we had with them.

____

Krutika Bhagane

____

Nikhil Bhosale

____

Abhishek Gambre

____

Anish Ghogare

Date:__/__/____

1. INTRODUCTION

The spread of COVID-19 is increasingly worrying for everyone in the world. This virus can be affected from human to human through the droplets and airborne. According to the instruction from WHO, to reduce the spread of COVID-19, every people need to wear face mask, do social distancing, evade the crowd area and also always maintain the immune system. Therefore, to protect each other, every person should wear the face mask properly when they are in outdoor. However, most of selfish people won't wear the face mask properly with so many reasons. Public use of face masks has been common in China and other nations in the world since the beginning of the new coronavirus disease outbreak. We now know from recent studies that a significant portion of individuals with coronavirus lack symptoms ("asymptomatic") and that even those who eventually develop symptoms ("pre-symptomatic") can transmit the virus to others before showing symptoms, according to the advisory published by the health Centre. "This means that the virus can spread between people interacting in close proximity — for example, speaking, coughing, or sneezing — even if those people are not exhibiting symptoms". The recent information also gives trace of a new strain of corona virus, the mutant corona virus which, in which the virus has changed its structure and become mutant. The new strain is not even able to detect using the RT-PCR test we use now. So, it is inevitable for the people of an overpopulated country like India to wear masks and let the work go on. Nobody can keep an eye one very person coming in the work space is wearing a mask or not. So the need of Face mask detection arose. The model in the project uses the Convolutional Neural Network. It is a deep neural network model used for analysing any visual imagery. It takes the image data as input, captures all the data, and send to the layers of neurons. It has a fully connected layer, which processes the final output that represents the predicational out the image. The Convolutional neural network model used here is the MobileNetV2 architecture. MobileNet model is a network model using depth wise separable convolution as its basic unit. Its depth wise separable convolution has two layers: depth wise convolution and point convolution. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depth wise 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. Figure1showsthe framework of MobileNetV2 which is used in the model discussed in this project.

2. LITERATURE SURVEY

Object detection is one of the trending topics in the field of image processing and computer vision. Ranging from small scale personal applications to large scale industrial applications, object detection and recognition is employed in a wide range of industries. Some examples include image retrieval, security and intelligence, OCR, medical imaging and agricultural monitoring. In object detection, an image is read and one or more objects in that image are categorized. The location of those objects is also specified by a boundary called the bounding box. Traditionally, researchers used pattern recognition to predict faces based on prior face models. A break through face detection technology then was developed named as Viola Jones detector that was an optimized technique of using Haar, digital image features used in object recognition. However, it failed because it did not perform well on faces in dark areas and non-frontal faces. Since then, researchers are eager to develop new algorithms based on deep learning to improve the models. Deep learning allows us to learn features with end-to-end manner and removing the need to use prior knowledge for forming feature extractors. There are various methods of object detection based on deep learning which are divided into two categories: one stage and two stage object detectors.

Like object detection, face detection adopts the same architectures as one-stage and two-stage detectors, but in order to improve face detection accuracy, more face-like features are being added. However, there is occasional research focusing on face mask detection. Some already existing facemask detectors have been modelled using OpenCV, Pytorch Lightning, MobileNet, RetinaNet and Support Vector Machines. Here, we will be discussing two projects. One project used Real World Masked Face Dataset (RMFD) which contains 5,000 masked faces of 525 people and 90,000 normal faces. These images are 250 x 250 in dimensions and cover all races and ethnicities and are unbalanced. This project took 100 x 100 images as input, and therefore, transformed each sample image when querying it, by resizing it to 100x100. Moreover, this project uses PyTorch then they convert images to Tensors, which is the base data type that PyTorch can work with. RMFD is im-balanced (5,000 masked faces vs 90,000 non-masked faces). Therefore, the ratio of the samples in train/validation while splitting the dataset was kept equal using the train test split function of sklearn. Moreover, to deal with unbalanced data, they passed this information to the loss function to avoid unproportioned step sizes of the optimizer. They did this by assigning a weight to each class, according to its representability in the dataset. They assigned more weight to classes with a small number of samples so that the network will be penalized more if it makes mistakes predicting the label of these classes. While classes with large numbers of samples, they assigned to them a smaller weight. This makes their network training agnostic to the proportion of classes.

To load the data efficiently this project used the data loader. For instance, in this project, they used the PyTorch lighting, and to load them for training and validation they divided data into 32 batches and assigned the works of loading to the 4 number of workers, and this procedure allowed them to perform multi-process data loading. Like most of the projects, this project also used Adam optimizer. If any Model has a high rate of learning, it learns faster, but it bounces a lot to reach the global minima and may diverge from the global minima. However, a small learning rate may take considerably lower time to train, but it reaches to the global minima. If the loss of the model declines quickly for any learning rate, then that learning rate would be the best learning rate. However, it seems that this project considered the 0.00001 learning rate would be the best for their model so that it could work efficiently. To train the model they defined a model checkpointing callback where they wanted to save the best accuracy and the lowest loss. They tried to train the model for 10 epochs and after finding optimal epoch, they saved the model for 8 epochs to test on the real data. To get rid of the problem of occlusions of the face which causes trouble face detectors to detect masks in the images, they used a built-in OpenCV deep learning face detection model. For instance, the Haar-Cascade model could be used but the problem of the Haar-Cascade model is that the detection frame is a rectangle, not a square. That is why, without capturing the portion of the background, the face frame can fit the entirety of the face, which can interfere with the face mask model predictions. In the second project, a dataset was created by Prajna Bhandary using a Py Image Search reader. This dataset consists of 1,376 images belonging to all races and is balanced. There are 690 images with masks and 686 without masks. Firstly, it took normal images of faces and then created a customized computer vision Python script to add face masks to them. Thereby, it created a real-world applicable artificial dataset. This method used the facial landmarks which allow them to detect the different parts of the faces such as eyes, eyebrows, nose, mouth, jawline etc. To use the facial landmarks, it takes a picture of a person who is not wearing a mask, and, then, it detects the portion of that person's face. After knowing the location of the face in the image, it extracted the face Region of Interest (ROI). After localizing facial landmarks, a picture of a mask is placed into the face. In this project, embedded devices are used for deployment that could reduce the cost of manufacturing. MobileNetV2 architecture is used as it is a highly efficient architecture to apply on embedded devices with limited computational capacity such as Google Coral, NVIDIA Jetson Nano. This project performed well, however, if a large portion of the face is occluded by the mask, this model could not detect whether a person is wearing a mask or not. The dataset used to train the face detector did not have images of people wearing face masks as a result, if the large portion of faces is occluded, the face detector would probably fail to detect properly. To get rid of this problem, they should gather actual images of people wearing masks rather than artificially generated images.

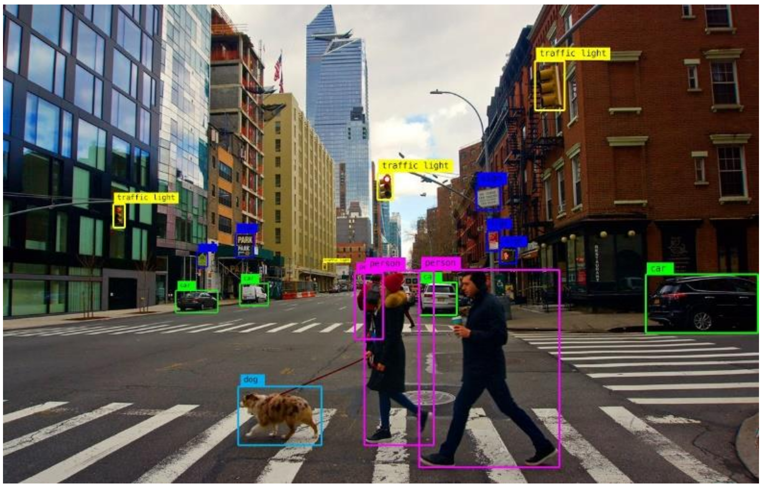


Fig1: Tracking of Objects

Table 1: Summary of Literature survey

Sr.No	Title	Date of Publishment	Advantages
1	Deep learning and control algorithms of direct perception for autonomous driving	2019	Used in Machine learning.
2	Masked face recognition data sets and application National natural science foundation of China	2020	Load of data sets of mask and without mask people's images.
3	Face Mask Detector	July 2020	Detecting of face mask
4	Object detection with deep learning IEEE transactions on neural networks and learning systems	2019	Detection of objects by IEEE transactions.

3. EXISTING SYSTEM

A pre-trained MobileNet with a global pooling block for face mask detection. The pre-prepared MobileNet takes a shading picture and creates a multi-dimensional component map. The worldwide pooling block that has been used in the proposed model changes the element map into an element vector of 64 highlights. At long last, the SoftMax layer performs paired order utilizing the 64 highlights. We have assessed our proposed model on two openly accessible datasets. Our proposed model has accomplished 99% and 100% exactness on DS1 and DS2 separately. The worldwide pooling block that has been utilized in the proposed model dodges overfitting the model. Further, the proposed model beats existing models in the quantity of boundaries just as preparing time. But this model cannot detect face mask for multiple faces at a time. In project, utilizes a proficient and strong item location calculation to naturally identify the appearances with veils or without covers, making the plague avoidance work cleverer. In particular, they gathered a broad data set of 9886 pictures of individuals with and without face covers and physically named them, at that point use multi-scale preparing and picture mistake techniques to improve YOLOv3, an article recognition calculation, to consequently distinguish whether a face is wearing a veil. Our analysis results show that the mean Average Precision (map) of the improved YOLOv3 calculation model came to 86.3%. This work can viably and naturally distinguish whether individuals are wearing veils, which decreases the pressing factor of conveying HR for checking covers openly puts and has high functional application esteem.

Fig2: Face-Mask Detection

4. PROPOSED SYSTEM

The model proposed here is designed and modelled using python libraries namely TensorFlow, Keras and OpenCV. The model we used is the MobileNetV2 of convolutional neural network. The method of using MobileNetV2 is called using Transfer Learning. Transfer learning is using some pre trained model to train your present model and get the prediction which saves time and makes using training the different models easy. We tune the model with the hyper parameters: learning rate, number of epochs and batch size. The model is trained with a dataset of images with two class, with mask and without mask. The dataset has 993 images of with mask class and 1918 images of without mask class.

(i) Training the model with the taken dataset.

(ii) Deploying the model

In the project we have developed a model using the above-mentioned libraries. We have tested the model for different conditions with different hyper parameters, for which the results are mentioned in the next section. First, we feed the dataset in the model, run the training program, which trains the model on the given dataset. Then we run the detection program, which turns on the video stream, captures the frames continuously from the video stream with an anchor box using object detection process. This is passed through the MobileNetV2 model layers which classifies the image as with or without mask. If the person is wearing a mask, a green anchor box is displayed and red if not wearing a mask with the accuracy for the same tagged on the anchor box. Figure2shows the flow of the Face Mask Detection model used in this project.

The face mask recognition system uses AI technology to detect the person with or without a mask. It can be connected with any surveillance system installed at your premise. The 3 authorities or admin can check the person through the system to confirm their identity. The system sends an alert message to the authorized person if someone has entered the premises without a face mask. The accuracy rate of detecting a person with a face mask is 95-97% depending on the digital capabilities. The data has been transferred and stored automatically in the system to enable reports whenever you want.

5. FLOWCHART

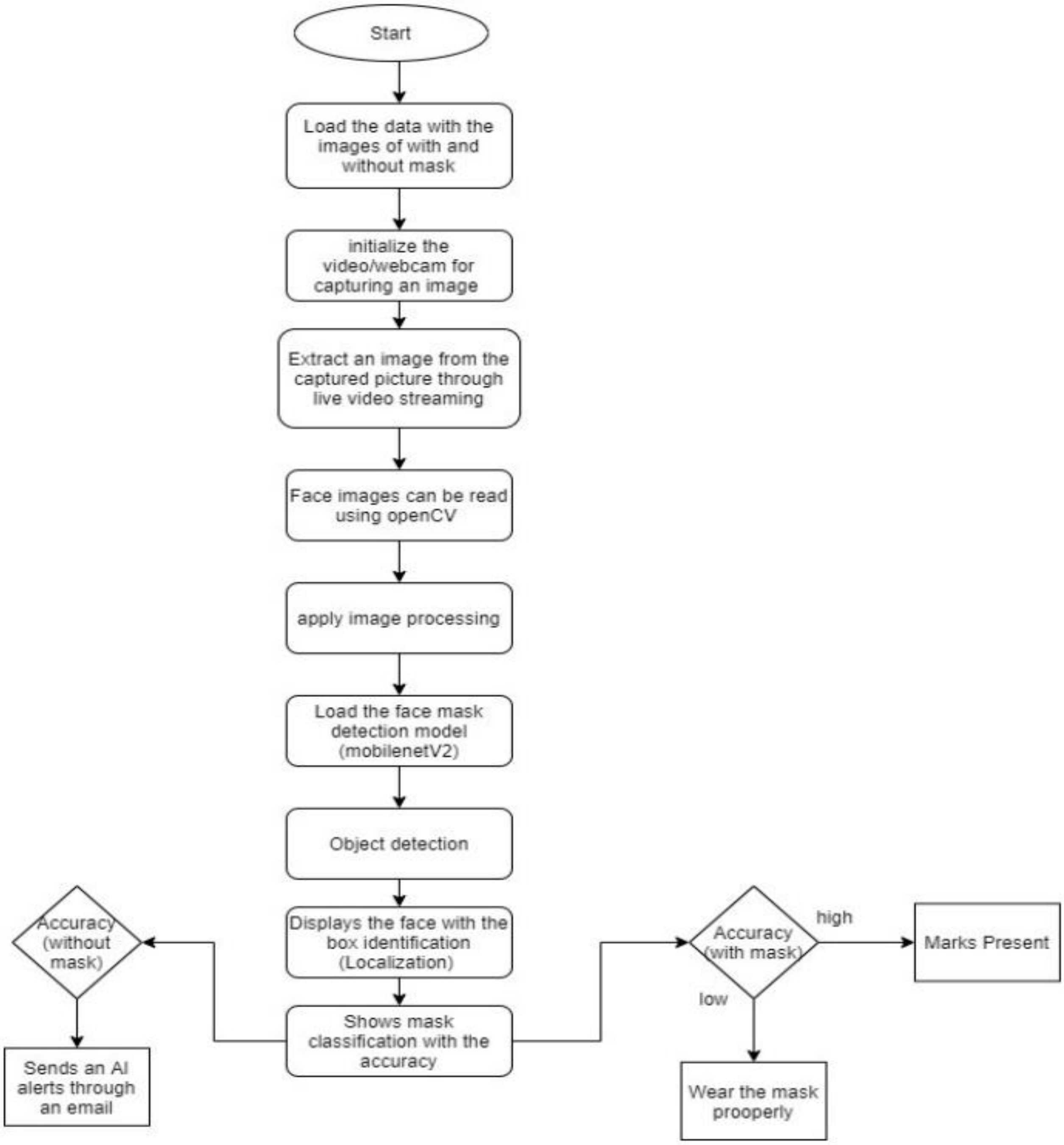


Fig3: Face Detection Model Flowchart.

6. METHODOLOGY

Dataset

The dataset which we have used consists of 3835 total images out of which 1916 are of masked faces and 1919 are of unmasked faces. All the images are actual images extracted from Bing Search API, Kaggle datasets and RMFD dataset. From all the three sources, the proportion of the images is equal. The images cover diverse races i.e Asian, Caucasian etc. The proportion of masked to unmasked faces determine that the dataset is balanced. We need to split our dataset into three parts: training dataset, test dataset and validation dataset. The purpose of splitting data is to avoid overfitting which is paying attention to minor details/noise which is not necessary and only optimizes the training dataset accuracy. We need a model that performs well on a dataset that it has never seen (test data), which is called generalization. The training set is the actual subset of the dataset that we use to train the model. The model observes and learns from this data and then optimizes its parameters. The validation dataset is used to select hyperparameters (learning rate, regularization parameters). When the model is performing well enough on our validation dataset, we can stop learning using a training dataset. The test set is the remaining subset of data used to provide an unbiased evaluation of a final model fit on the training dataset. Data is split as per a split ratio which is highly dependent on the type of model we are building and the dataset itself. If our dataset and model are such that a lot of training is required, then we use a larger chunk of the data just for training which is our case. If the model has a lot of hyperparameters that can be tuned, then we need to take a higher amount of validation dataset. Models with a smaller number of hyperparameters are easy to tune and update, and so we can take a smaller validation dataset. In our approach, we have dedicated 80% of the dataset as the training data and the remaining 20% as the testing data, which makes the split ratio as 0.8:0.2 of train to test set. Out of the training data, we have used 20% as a validation data set. Overall, 64% of the dataset is used for training, 16% for validation and 20% for testing.

Architecture

The working of the Single Shot Detector(SSD) algorithm relies on an input image with a specified bounding box against the objects. The methodology of predicting an object in an image depends upon very renowned convolution fashion. For each pixel of a given image, a set of default bounding boxes (usually 4) with different sizes and aspect ratios are evaluated. Moreover, for all the pixels, a confidence score for all possible objects are calculated with an additional label of ‘No Object’. This calculation is repeated for many different feature maps. In order to extract feature maps, we usually use the predefined trained techniques which are used for high quality classification problems. We call this part of the model a base model. For the SSD, we have VGG-16 network as our base model. At the training time, the bounding boxes evaluated are compared with the ground truth boxes and in the back propagation, the trainable parameters are altered as per requirement. We truncate the VGG-16 model just before the classification layer and add feature layers which keep on decreasing in size. At each feature space, we use a kernel to produce outcomes which depicts corresponding scores for each pixel whether there exists any object or not and the corresponding dimensions of the resulting bounding box. VGG-16 is a very dense network having 16 layers of convolution which are useful in extracting features to classify and detect objects. The reason for the selection is because the architecture 4 consists of stacks of convolutions with 3x3 kernel size which thoroughly extract numerous feature information along with max-pooling and ReLU to pass the information flow in the model and adding non linearity respectively from the given image. For additional nonlinearity, it uses 1x1 convolution blocks which does not change the spatial dimension of the input. Due to the small size filters striding over the image, there are many weight parameters which end up giving an improved performance. The block diagram shows the working functionality of SSD.

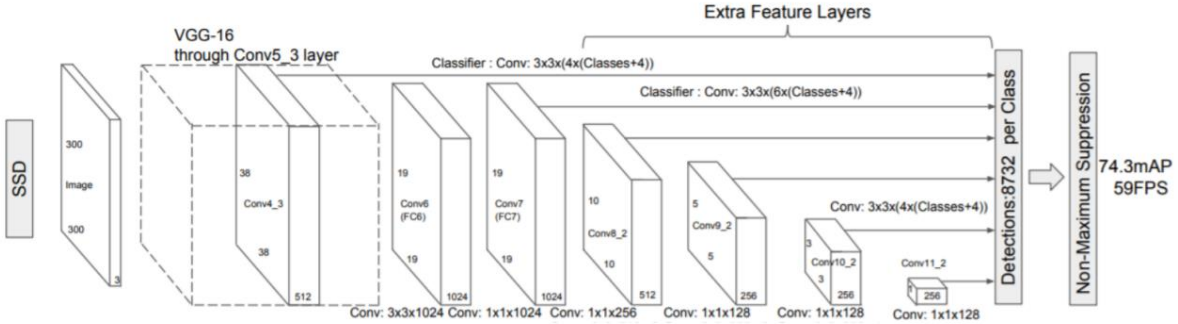


Fig4: Working Of SSD.

The problem can be solved in two parts: first detecting the presence of several faces in a given image or stream of video and then in the second part, detect the presence or absence of face mask on face. In order to detect the face, we have used the OpenCV library. The latest OpenCV includes a Deep Neural Network (DNN) module, which comes with a pre-trained face detection convolutional neural network (CNN). The new model enhances the face detection performance compared to the traditional models. Every face detected comes out with a level of confidence which is then compared with a threshold value to filter out the irrelevant detections. After we have the faces, we need to evaluate the bounding box around it and send it to the second part of the model to check if the face has a mask or not. The second part of the model is trained by us using a dataset consisting of images with mask and without mask. We have used Keras along with Tensorflow to train our model. First part of the training includes storing all labels of the images in a Numpy array and the corresponding images are also reshaped (224, 244, 3) for the base model. Image augmentation is a very useful technique because it increases our dataset with images with a whole new perspective. Before inputting, we performed the following image augmentations randomly: rotations up to 20 degrees, zooming in and out up to 15%, width or height shift up to 20%, up to 15 degrees shear angle in the counter clockwise direction, flip inputs horizontally and points outside the boundaries of the inputs are filled from the nearest available pixel of the input. For the image classification, it is now a common practice to use transfer learning which means using a model which has been pre-trained on millions of labels before and it has been tested that this method results in significant increase in accuracy. Obviously, the assumption here is that both the problems have sufficient similarity. It uses a well-structured and deep neural network that has been trained on a large amount of data set. Due to somewhat same nature of the problem, we can use the same weights which have the capability to extract features and later in the deep layers, convert those features to objects. The base model that we have used here is MobileNetV2 with the given ‘ImageNet’ weights. ImageNet is an image database that has been trained on hundreds of thousands of images hence it helps a lot in Image classification. The overall process flow diagram of the algorithm is shown below.

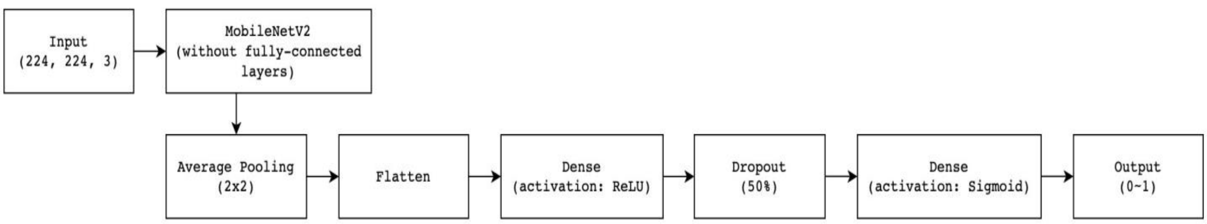


Fig5: Process Flow Diagram.

Training

At the training time, for each pixel, we compare the default bounding boxes having different sizes and aspect ratios with ground truth boxes and finally use Intersection over Union (IoU) method to select the best matching box. IoU evaluates how much part of our predicted box match with the ground reality. The values range from 0 to 1 and increasing values of IoU determine the accuracies in the prediction; the best value being the highest value of IoU. The equation and pictorial description of IoU is given as follow:

$$IoU(B_1, B_2) = \frac{B_1 \cap B_2}{B_1 \cup B_2}$$

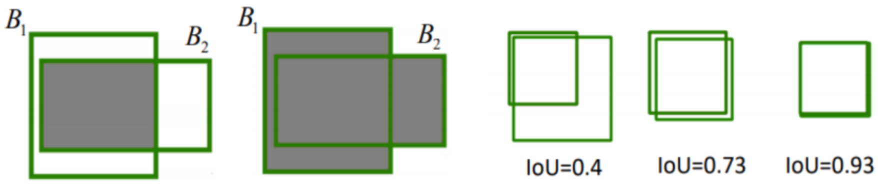


Fig6: Pictorial Representation of IoU.

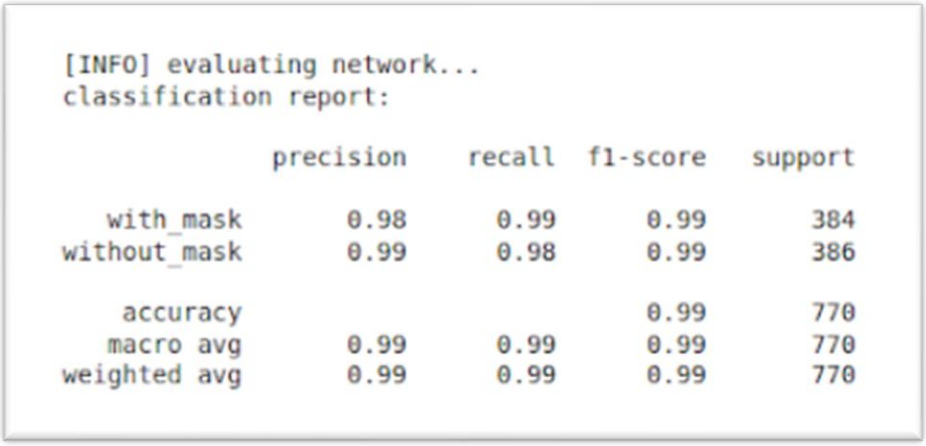
7. EVALUATION

Testing

We tried using three different base models for detecting 'mask' or 'no mask'. The exercise was done to find the best fit model in our scenario. The evaluation process consists of first looking at the classification report which gives us insight towards precision, recall and F1 score. The equations of these three metrics are as follows:
Using these three metrics, we can conclude which model is performing most efficiently. The second part consists of plotting the train loss, validation loss, train accuracy and validation accuracy which also proves helpful in choosing a final model.

Xception

The complete form of Xception is "Extreme Inception". Basically, the Xception architecture is designed with depth wise separable convolution layers with residual connections. These convolutional layers are linearly stacked with each other. This architecture is easy to define and easy to modify. If we compare it with Inceptions to get the following result:



[INFO] evaluating network...
classification report:

	precision	recall	f1-score	support
with_mask	0.98	0.99	0.99	384
without_mask	0.99	0.98	0.99	386
accuracy			0.99	770
macro avg	0.99	0.99	0.99	770
weighted avg	0.99	0.99	0.99	770

Fig7: Xceptions

MobileNetV2

MobileNetV2 is an architecture of bottleneck depth-separable convolution building of basic blocks with residuals. It has two types of blocks. The first one is a residual block with stride of 1. Second one is also residual block with stride 2 and it is for downsizing.

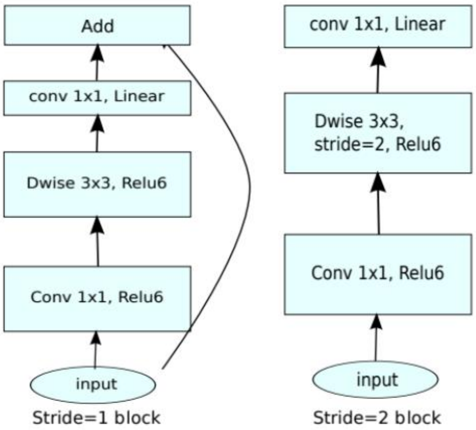


Fig8: MobileNetV2 Flowchart.

classification report:

	precision	recall	f1-score	support
with_mask	0.98	1.00	0.99	384
without_mask	1.00	0.98	0.99	386
accuracy			0.99	770
macro avg	0.99	0.99	0.99	770
weighted avg	0.99	0.99	0.99	770

Fig9: Classification report for MobileNetV2.

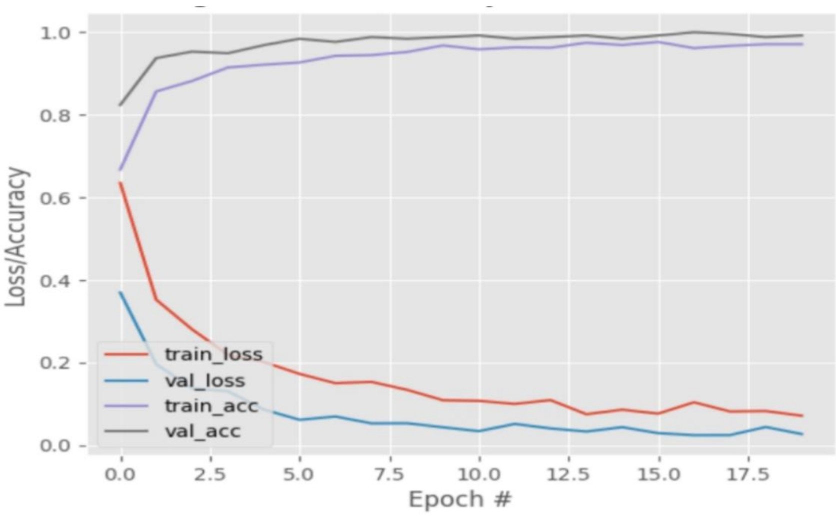


Fig10:Loss graph for MobileNetV2.

8. RESULT

We implemented our model on images containing one and more faces. We also implemented it on videos and live video streams by removing and wearing masks one by one. Some screenshots of the results are shown below:

Fig11:Person without mask.

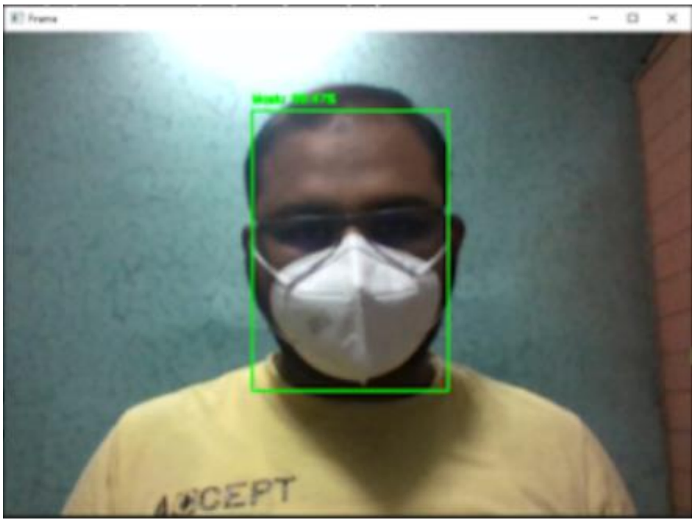


Fig12:Person with mask.

9. FUTURE WORK

More than fifty countries around the world have recently initiated wearing face masks compulsory. People have to cover their faces in public, supermarkets, public transports, offices, and stores. Retail companies often use software to count the number of people entering their stores. They may also like to measure impressions on digital displays and promotional screens. We are planning to improve our Face Mask Detection tool and release it as an open-source project. Our software can be equated to any existing USB, IP cameras, and CCTV cameras to detect people without a mask. This detection live video feed can be implemented in web and desktop applications so that the operator can see notice messages. Software operators can also get an image in case someone is not wearing a mask. Furthermore, an alarm system can also be implemented to sound a beep when someone without a mask enters the area. This software can also be connected to the entrance gates and only people wearing face masks can come in.

There were not many challenges faced but the two problems that were time consuming and made the tasks tedious are discussed as follows. One was the excessive data loading time in Google Collab Notebook while loading the dataset into it. Since the runtime restarting refreshes all the cells, the cell for dataset loading took most of the time while running. Secondly, the access problem in Google Collab Notebook: it did not allow the access of webcam which posed a hurdle in testing images and live video stream through Google Collab Notebook. Therefore, we had to run the code locally on the computer through which we tested the code on the live video stream.

10. CONCLUSION

By the development of face mask detection, we can detect if the person is wearing a face mask and allow their entry would be of great help to the society. The accuracy of the model will be achieved and the optimization of the model is a continuous process and so we are building a highly accurate solution. We can prevent people from Virus Transmission through this model.

To moderate the spread of the COVID-19 pandemic, measures should be taken. We have demonstrated a facemask detector using Convolutional Neural Network and move learning techniques in neural organizations. To train, validate and test the model, we utilized the dataset that consisted of 993 masked faces pictures and 1918 exposed faces pictures. These pictures were taken from different assets like Kaggle and RMFD datasets. The model was induced on pictures and live video transfers. To choose a base model, we assessed the measurements like precision, accuracy, and recall and chose MobileNetV2 architecture with the best exhibition having 99% precision and 99% recall. It is additionally computationally efficient using MobileNetV2 which makes it simpler to introduce the model to inserted frameworks. This face mask detector can be sent in numerous regions like shopping centres, air terminals and other substantial traffic places to screen people in general and to dodge the spread of the infection by checking who is following essential rules and who isn't.

11. REFRENCES

i. W. H. Organization, “Coronavirus disease (COVID-19) advice for the public: When and how to use masks,” 2020. https://www.who.int/emergencies/diseases/novel-coronavirus2019/advice-for-public/when-and-how-to-use-masks

ii. N. Leung, D. Chu, E. Shiu, K. Chan, J. McDevitt, B. Hau, H. Yen, Y. Li, D. Ip, J. Peiris, W. Seto, G. Leung, D. Milton, and B. Cowling, “Respiratory virus shedding in exhaled breath and efficacy of face masks,” Nature Medicine, Jan. 2020.

iii. S. Zhou, S. Lukula, C. Chiossone, R. Nims, D. Suchmann, and M. K. Ijaz, “Assessment of a respiratory face mask for capturing air pollutants and pathogens including human influenza and rhinoviruses,” Journal of Thoracic Disease, vol. 10, pp. 2059–2069, 03 2018.

iv. M. Sande, P. Teunis, and R. Sabel, “Professional and home-made face masks reduce exposure to respiratory infections among the general population,” PloS one, vol. 3, p. e2618, 02 2008.

v. A.G. Howard, M. Zhu, B. Chen et al., “Mobilenets:efficient convolutional neural networks for mobile vision applications,” 2017, https://arxiv.org/abs/1704.04861.

vi. Wei Wang, Yutao Li, Ting Zou, Xin Wang, JieyuYou,Yanhong Luo, "A Novel Image Classification Approach via Dense-MobileNet Models", Mobile InformationSystems, vol.2020,ArticleID 7602384, 8 pages, 2020. https://doi.org/10.1155/2020/7602384

vii. B. Venkateswarlu, J. Kakarla and S. Prakash, "Facemask detection using MobileNet and Global Pooling Block," 4 2020 IEEE 4th Conference on Information & Communication Technology (CICT), 2020, pp. 1-5, doi: 10.1109/CICT51604.2020.9312083.

viii. M. S. Ejaz and M. R. Islam, "Masked Face Recognition Using Convolutional Neural Network," 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019, pp. 1-6, doi: 10.1109/STI47673.2019.9068044