Face Mask Detection and Authentication

M.Sc. Data Science
Dhirubhai Ambani Institute of Information and Communication Technology
Gujarat, India (394210)

Dev Patel
202018055@daiict.ac.in

Smit Gandhi
202018057@daiict.ac.in

***Abstract* -- The world is struggling with Covid-19 pandemic and are so many essential equipment's needed to combat against Corona virus. One of such most essential is Face Mask and mask was not mandatory for everyone but as the day surpasses scientist and Doctors have recommended everyone to wear the mask. Therefore, to detect whether a person is wearing Face Mask or not, there are detection technique. Face Mask Detection is a project based on Artificial Intelligence. General view of this project is detect people with or without mask. In this project we use image data which is in form of with mask and without mask and all the photos are in form of artificial mask. In making of this project, we have two phases. Train model using Convolution or any pretrained model which detect face masks in images. Then, detect faces in video or images and get prediction from our trained model. We make four models in this project which are ResNet, AlexNet, GoogleNet and VGG. After making model we implement live face mask detection. If model will not detect face mask, then it will beep sound and alert people. ResNet gives accurate result also VGG gives good result but AlexNet and GoogleNet create confusion while detecting.**

***Index Terms* -- COVID-19: Detect with live Webcam: Faces with mask and without mask: Image Classification models:**

I. INTRODUCTION

THE world is currently under the onslaught of COVID-19. COVID-19 is an infectious disease caused by severe acute respiratory syndrome (SARS-CoV-2) [1]. People can become infected by coming into close social contact with the infected person through respiratory droplets during coughing, sneezing and/or talking. Moreover, the virus can also be spread by touching a surface or object that has the virus on it, and then by touching your mouth, nose, or eyes. For now, we can protect ourselves by avoiding getting exposed to the virus. According to WHO the best way to avoid spreading or being infected with the disease is to practice social distancing and wearing face covering when in public areas and private sectors. The two main prevention approaches are avoiding unnecessary contact and wearing face mask. Implementing these guidelines, seriously impacts the current security systems based on facial recognition that has already been put by several corporations and government organizations in place. Fingerprint or password-based security system, which involves contacting finger with sensor hence is not a good way to prevent the spread of disease making it unsafe. Face recognition-based security system however avoids unnecessary contact making it much safer than the former one. But such systems assume the that a picture of the entire face can be taken to perform recognition effectively. Widespread use of face masks thus renders the existing facial recognition systems in-efficient and they can make the entire infrastructure around facial recognition inoperable. Modern deep learning-based face recognition systems have proven superior accuracy. The accuracy of these systems depends on the nature of the available training images. Most of these systems assume access to un-occluded faces for recognition. This condition is fair when you can make sure that the system has access to the complete un-occluded face of the person being recognized. The system trained on such images learns to pay attention to important face features such as the eyes, nose, lips, face edges etc. But when these systems are presented a faced mask, the system fails to identify the person rendering the system unusable.

Our contributions are as follows

- We took artificial mask face dataset
- Train four models
- Apply all models on real time face mask detection.

In this project, initially we lode the data after that we pre-process on face mask data, data splitting, visualizing image and finally we start to train model. In this section we train four model such as ResNet model, AlexNet model, GoogleNet model and VGG model. In the training model we use direct models optimize it and schedule it. This process flow is for ReseNet, AlexNet

and GoogleNet model. This four models train on twenty epochs. There are five types of ReseNet models such as ResNet18, ResNet34, ResNet50, Resnet101, ResNet152 and here we use ResNet101 model. Also, for VGG model there are two types such as VGG16 and VGG19 and here we use VGG16 model. For all the model we derive confusion matrix so that we easily know that which model is best accurate among all models. At last, we visualizing all models. Form confusion matrix we can say that ResNet, best and VGG model is good. GoogleNet and AlexNet are not much accurate and generate confusion. Our last phase is detecting live face mask using webcam. In this implementation we use those four models for best accuracy and if model detect without mask, then it will beep and alert us by generating sound.

II. METHODOLOGY

This section discusses the different methods and models applied to image dataset predict whether face contain with mask or without mask. Fig. 1 shows the flowchart of our methodology which includes data collection, followed by data pre-processing, data visualization, implementation of different models and finally face mask detection via webcam.

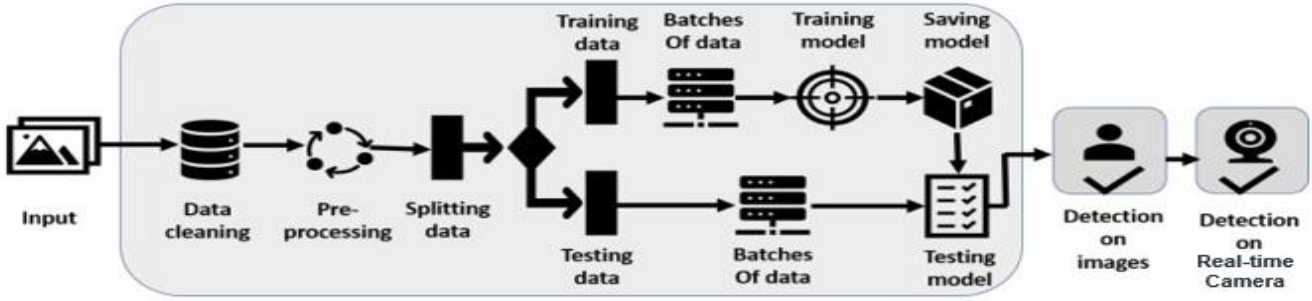

Fig 1: Flowchart of our Methodology

A. Data Collection

Face mask detection contain 690 with mask and 690 without mask face data. For test our model we take 97 with and without mask face dataset. Here for getting best accuracy and for fitting our model in nice way we take artificial face mask dataset. We got this dataset from [1], this research paper makes artificial face mask data. The process and batter explanation of our dataset are in fig 2.

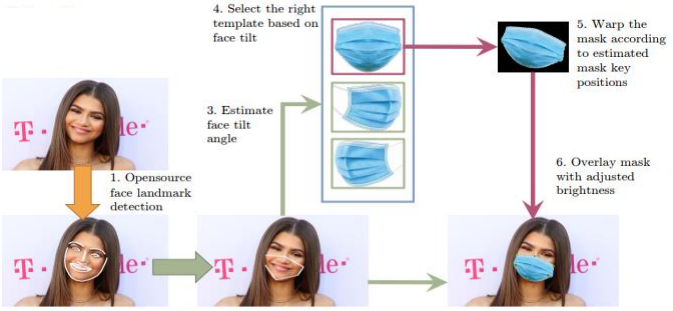

Fig 2: Dataset Description

B. Data Preprocessing

In the section of data pre-processing, after load our data. As we are going to be using a pre-trained model, we will need to ensure that our images are the same size and have the same normalization as those used to train the model - which we find on the torchvision model's page. We use the same data augmentation as always: randomly rotating, flipping horizontally and cropping. After that we split our data into train and test dataset and then we make data Loder which contain sixteen image of batch size with contain four number of workers at a time.

C. Model

After data preprocessing, we need to train our image data in image classification models such as ResNet, AlexNet, MobileNet V2, GoogleNet, VGG etc. So, for that we take four different models to train and test in testing dataset. Image Classification is a fundamental task that attempts to comprehend an entire image as a whole.

The goal is to classify the image by assigning it to a specific label. Typically, Image Classification refers to images in which only one object appears and is analyzed. In contrast, object detection involves both classification and localization tasks, and is used to analyze more realistic cases in which multiple objects may exist in an image.

1. ResNet Model

ResNet is a short name for a residual network, residual learning is based on deep convolutional neural networks have achieved the human level image cla ssification result. deep networks extract low, middle and high-level features and classifiers in an end-to-end multi-layer fashion, and the number of stacked layers can enrich the "levels" of features. The stacked layer is of crucial importance in deep networks.

When the deeper network starts to converge, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated (which might

be unsurprising) and then degrades rapidly. Such degradation is not caused by overfitting or by adding more layers to a deep network leads to higher training error. The deterioration of training accuracy shows that not all systems are easy to optimize.

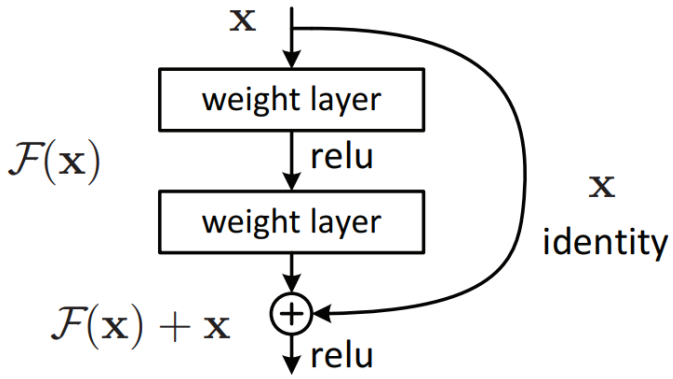

Fig 3: Residual learning: a building block

To overcome this problem, Microsoft introduced a deep residual learning framework. Instead of hoping every few stacked layers directly fit a desired underlying mapping, they explicitly let these layers fit a residual mapping. The formulation of F(x)+x can be realized by feedforward neural networks with shortcut connections. Shortcut connections are those skipping one or more layers shown in Fig. 3. The shortcut connections perform identity mapping, and their outputs are added to the outputs of the stacked layers. By using the residual network, there are many problems which can be solved such as:

- ResNets are easy to optimize, but the "plain" networks (that simply stack layers) show higher training error when the depth increases.
- ResNets can easily gain accuracy from greatly increased depth, producing results which are better than previous networks.

2. AlexNet Model

AlexNet is the name of a convolutional neural network which has had a large impact on the field of machine learning, specifically in the application of deep learning to machine vision. It famously won the 2012 ImageNet LSVRC-2012 competition by a large margin (15.3% VS 26.2% (second place) error rates). The network had a very similar architecture as LeNet by Yann LeCun et al but was deeper, with more filters per layer, and with stacked convolutional layers. It consisted of 11×11, 5×5,3×3, convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum. It attached ReLU activations after every convolutional and fully-connected layer. AlexNet was trained for 6 days simultaneously on two Nvidia Geforce GTX 580 GPUs which is the reason for why their network is split into two pipelines.

Architecture: In AlexNet's first layer, the convolution window shape is 11×11. Since most images in ImageNet are more than ten times higher and wider than the MNIST images, objects in ImageNet data tend to occupy more pixels. Consequently, a larger convolution window is needed to capture the object. The convolution window shape in the second layer is reduced to 5×5, followed by 3×3. In addition, after the first, second, and fifth convolutional layers, the network adds maximum pooling layers with a window shape of 3×3 and a stride of 2. Moreover, AlexNet has ten times more convolution channels than LeNet.

After the last convolutional layer there are two fully-connected layers with 4096 outputs. These two huge fully-connected layers produce model parameters of nearly 1 GB. Due to the limited memory in early GPUs, the original AlexNet used a dual data stream design, so that each of their two GPUs could be responsible for storing and computing only its half of the model. Fortunately, GPU memory is comparatively abundant now, so we rarely need to break up models across GPUs these days (our version of the AlexNet model deviates from the original paper in this aspect).

3. GoogleNet Model

Google Net (or Inception V1) was proposed by research at Google (with the collaboration of various universities) in 2014 in the research paper titled "Going Deeper with Convolutions". This architecture was the winner at the ILSVRC 2014 image classification challenge. It has provided a significant decrease in error rate as compared to previous winners AlexNet (Winner of ILSVRC 2012) and ZF-Net (Winner of ILSVRC 2013) and significantly less error rate than VGG (2014 runner up). This architecture uses techniques such as 1×1 convolution in the middle of the architecture and global average pooling.

Architecture: The overall architecture is 22 layers deep. The architecture was designed to keep computational efficiency in mind. The idea behind that the architecture can be run on individual devices even with low computational resources. The architecture also contains two auxiliary classifier layers connected to the output of Inception (4a) and Inception (4d) layers.

The architectural details of auxiliary classifiers as follows:

- An average pooling layer of filter size 5×5 and stride 3.
- A 1×1 convolution with 128 filters for dimension reduction and ReLU activation.

- A fully connected layer with 1025 outputs and ReLU activation
- Dropout Regularization with dropout ratio = 0.7
- A SoftMax classifier with 1000 classes output similar to the main SoftMax classifier.

This architecture takes image of size 224 x 224 with RGB color channels. All the convolutions inside this architecture uses Rectified Linear Units (ReLU) as their activation functions.

4. Vgg16 Model

Vgg16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. Vgg16 was trained for weeks and was using NVIDIA Titan Black GPU's.

Architecture: The input to cov1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filter, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e., the padding is 1-pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2-pixel window, with stride 2.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

All hidden layers are equipped with the rectification (ReLU) non-linearity. It is also noted that none of the networks (except for one) contain Local Response Normalization (LRN), such normalization does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time.

III. Experimental Results

This section presents the results of our train model on face dataset. We finding model accuracy, to check model capability we make confusion matrix of all model. Here we use four models so experimental results of our model are provided below.

A. ResNet Model

In ResNet Model, we generate twenty epochs for all models, so we get train loss and test loss with individual accuracy and the value of train loss is 0.0717, accuracy is 0.9711. The value of test loss is 0.0003, accuracy is 1 and also, we get best accuracy value for ResNet model is 1. This model took 7.48 hr. to complete all epochs. We make confusion matrix, so values of normalized confusion matrix is,

```
==========================================
Normalized confusion matrix:
[1. 0.]
[0. 1.]
==========================================
```

Fig 4: Normalized Confusion Matrix of ResNet Model

From the normalized matrix we can see here we got best accuracy value for ResNet model on both components is 1.

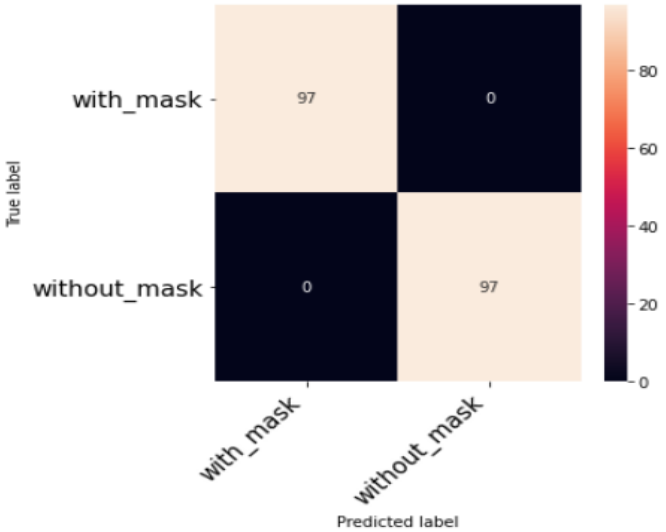

Fig 5: Confusion Matrix of ResNet Model

B. AlexNet Model

In AlexNet Model, the value of train loss is 0.1047, accuracy is 0.9474. The value of test loss is 0.0185, accuracy is 0.9948 and also, we get best accuracy value

for AlexNet model is 0.994845. This model took 38 min. to complete all epochs. Normalized confusion matrix is given by,

```
==========================================
Normalized confusion matrix:
[1. 0.]
[0.0103 0.9897]
==========================================
```

Fig 6: Normalized Confusion Matrix of AlexNet Model

From the normalized matrix we can see here we got best accuracy value for AlexNet model on both components is 1 and 0.9897.

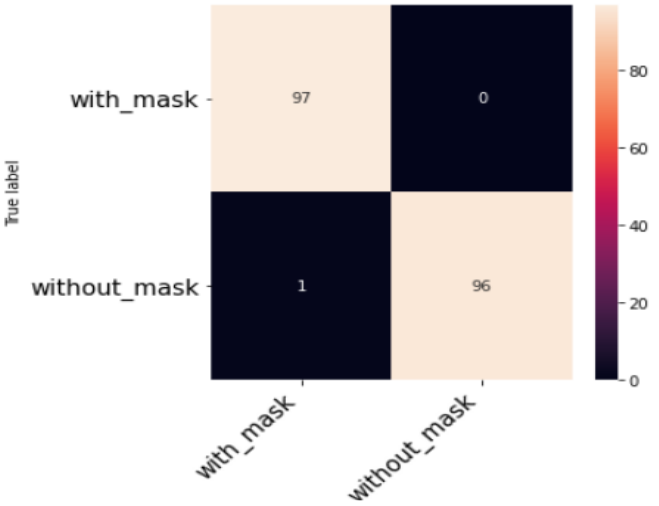

Fig 7: Confusion Matrix of AlexNet Model

C. *GoogleNet Model*

In GoogleNet model, we get train loss and test loss with individual accuracy and the value of train loss is 0.0691, accuracy is 0.9652. The value of test loss is 0.0062, accuracy is 0.9948 and also, we get best accuracy value for GoogleNet model is 1. This model took 2.1 hr. to complete all epochs. Value of normalized confusion matrix is,

```
==========================================
Normalized confusion matrix:
[1. 0.]
[0. 1.]
==========================================
```

Fig 8: Normalized Confusion Matrix of GoogleNet Model

From the normalized matrix we can see here we got best accuracy value for GoogleNet model on both components is 1.

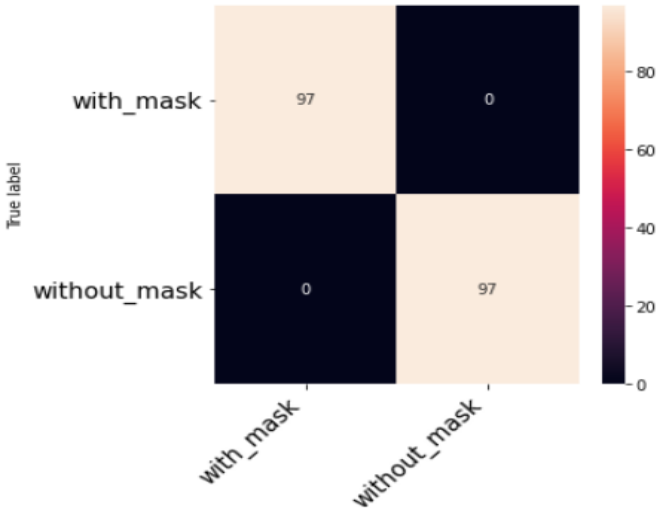

Fig 9: Confusion Matrix of GoogleNet Model

D. *VGG Model*

In VGG model, we train VGG16 and the value of train loss is 0.0918, accuracy is 0.9550. The value of test loss is 0.0002, accuracy is 1 and also, we get best accuracy value for VGG model is 1. This model took 10.2 hr. to complete all epochs. Normalized confusion matrix is given by,

```
==========================================
Normalized confusion matrix:
[1. 0.]
[0. 1.]
==========================================
```

Fig 10: Normalized Confusion Matrix of Vgg16 Model

From the normalized matrix we can see here we got best accuracy value for Vgg model on both components is 1.

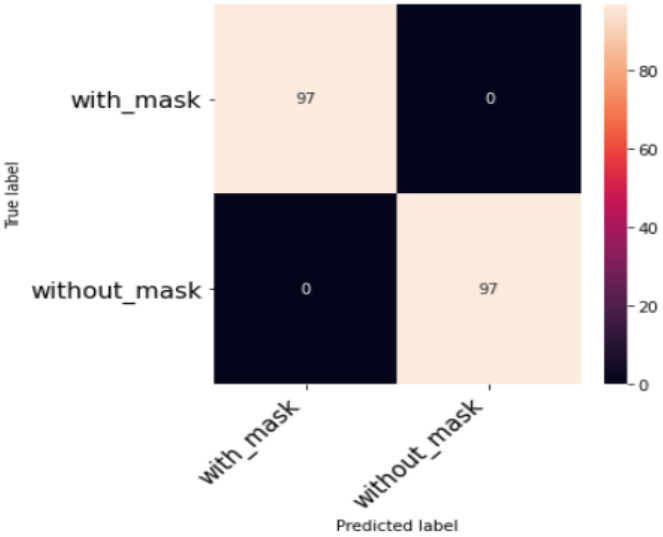

Fig 11: Confusion Matrix of Vgg16 Model

E. Model Compression

In this section, our motive is to compare model in two-way, first compression based on all trained model accuracy and second compression by using webcam detection.

1. Comparison throughout trained model

So, for comparison from trained model, we can say that all the accuracy is equivalent to 1. If we talk about best accuracy three of them is equal to 1 so ResNet, GoogleNet and Vgg16 is batter then AlexNet model, also if we consider others parameter from model training table ResNet model is best among others because train accuracy is higher (0.9711). As from train loss GoogleNet is best because it has very low loss (0.0691) and also from Epoch time parameter we can say that GoogleNet model is best.

	ResNet	VGG16	GoogleNet	AlexNet
Train loss	0.0717	0.0918	0.0691	0.1047
Test loss	0.0003	0.0002	0.0062	0.0185
Train Accuracy	0.9711	0.9550	0.9652	0.9474
Test Accuracy	1	1	0.9948	0.9948
Best Accuracy	1	1	1	0.9948
Epochs	20	20	20	20
Epochs Time	7.48 hr.	10.2 hr.	2.1 hr.	38 min.

Tab 1: Model Performance Table

2. Compression throughout Webcam Detection

In this part we present our model performance in a live detection. For that we load or model and connect with laptop or pc webcam and also implementing alert system. Here we represent four global position of wearing mask for detection by loading all our model.

✔	Indicate that On Detection Time Model Predict Right Prediction
✘	Indicate that On Detection Time Model Predict Wrong Prediction

Tab 2: Sign Indication Table

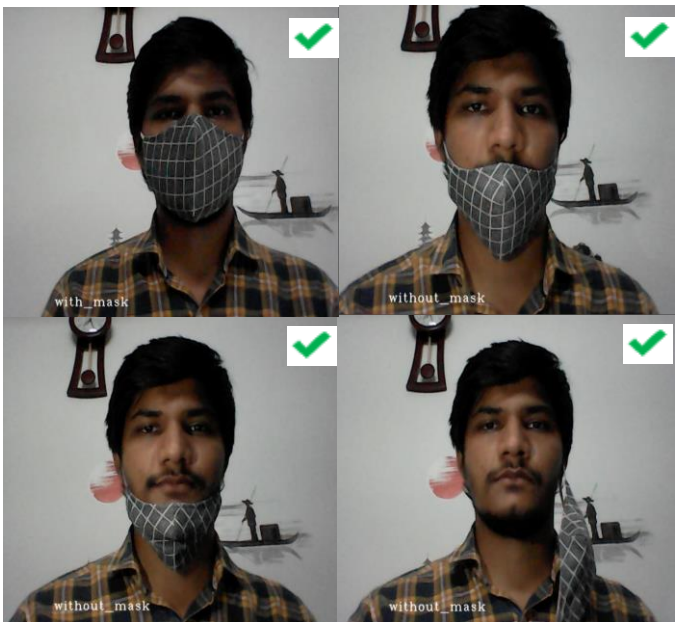

Fig 12: Webcam View by Using ResNet Model

In Fig. 12, give us live webcam view by taking four global position using ResNet model. In this result we found that ResNet model give us good performance among all model because this trained model also detect other different masks and give us right detection

Fig 13: Webcam View by Using AlexNet Model

In Fig. 13, give us live webcam view by using AlexNet Model. In this result we found that sometime AlexNet model give us wrong detection. As we can see in Fig. 13, first row of second image a person didn't wear mask properly and our model give us with mask.so at that position our model predicts wrong result. And also, AlexNet model detect only surgical mask.

*Photos which used in this research paper are authentic and co-author's photos.

Fig 14: Webcam View by Using GoogleNet Model

In Fig. 14, give us live webcam view by using GoogleNet Model. The results of this model are same as our AlexNet Model. This model also detects only surgical mask face and not detect any other different type of mask.

Fig 15: Webcam View by Using Vgg16 Model

In Fig. 15, give us live webcam view by using Vgg16 Model. In this result we found that Vgg16 Model give us good performance, right prediction, and detect on all position with right label. but this model has only one weakness, the model detects only for surgical mask and not for other.

Considering all of these, we can say that ResNet model is much batter among all model.

F. Limitations

The curb of face mask detection project is that it takes long time to train epochs in for VGG and ResNet model.

Also, model will detect only medical mask when detecting live webcam specially AlexNet, GoogleNet and Vgg model.

IV. Discussion

This section is for Summarize whole project. Initially we collect data and study for model. Before the train model we need some good quality data so required pre-process on data. After completing the pre-process, we used to do data splitting and visualize image. So now we are ready to train models. Our first is ResNet, second is AlexNet, third is GoogleNet and final one is Vgg model. In all those models training we use twenty (20) epochs. For that we find cross entropy loss and after that we optimize models. At last, in training model, we do visualize models. Among those models' best model is ResNet, Vgg, GoogleNet and AlexNet respectively. Now we detect real time face mask using webcam and for that we use those train models and get best accurate model which is ResNet model. Vgg is also good model But GoogleNet and AlexNet are create confusion, and gives wrong prediction.

As consideration of future work our project will implement in robotic form and help to detect face mask, The Face Mask Detection System can be utilized at office area to recognize if employees are keeping up safety standards at work. It screens employees without masks and sends them a suggestion to wear a cover. Also, we can use our model as metal detector in commercial areas, so that people will easily detect and aware about wear the mask.

V. References

[1] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei, H. Chen, Y. Miao,Z. Huang, and J. Liang, "Masked Face Recognition Dataset and Application", Computer Vision and Pattern Recognition, China, Mar. 20, 2020.

[2] Mohammad Marufur Rahman, "An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network", Vancouver, BC, Canada, Oct. 08,2020.

*Photos which used in this research paper are authentic and co-author's photos.

[3] Joo Er, M., Chen, W., Wu, S.: High-speed face recognition based on Discrete Cosine Transform and RBF Neural Networks. IEEE Transctions on neural networks 16(3), 679–691 (2005)

[4] Adnane Cabani, Karim Hammoudi, Halim Benhabiles, Mahmoud Melkemi, "Maskedface-Net A Dataset Of Correctly/Incorrectly Masked Face Images In The Context Of Covid-19", Aug 18, 2020.

[5] P. Viola and M. J. Jones, “Robust real-time face detection,” Int. J.Comput. Vision, vol. 57, no. 2, pp. 137–154, May 2004.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classificationwith deep convolutional neural networks,” in Advances in Neural Infor-mation Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, andK. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[7] Toshanlal Meenpal, Ashutosh Balakrishnan, Amit Verma, "Facial Mask Detection using Semantic Segmentation", 4th International Conference on Computing, Communications and Security (ICCCS), 2019.

[8] E. Sneha Sen, Khushboo Sawant, "Face mask detection for covid_19 pandemic using pytorch in deep learning", 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1070 012061.

[9] Ms. R. Suganthalakshmi A. Hafeeza, P. Abinaya, A.Ganga Devi, "Covid-19 Facemask Detection with Deep Learning and Computer Vision", Punalkulam, Pudukottai Dist, Mar 27, 2021.

[10] *A.Kumar, A.Kaur, M.Kumar,Face detection techniques: review, Artificial intelligence review,volume.52 no.2pp.927-928,2019.D.H.Lee,K.-L.CHEN,K.H Liou,C.Liu,andJ.Liu,Deep learning and control algorithms of direct perception for autonomous driving,2019.*