INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

COVID-19 Face Mask Detection using Deep Convolutional Neural Networks & Computer Vision

Premanand Ghadekar[1]*, **Gurdeep Singh**[2], **Joydeep Datta**[2], **Aryan Kumar Gupta**[3], **Divsehaj Singh Anand**[3], **Shreyas Khare**[3], **Preeti Oswal**[3], **Dheeraj Sharma**[3]

1 Department of Information Technology, Vishwakarma Institute of Technology, Pune, 411037, India
2 Corporate Venturing & Innovations Group, Tata Communications Ltd., Mumbai
3 Department of Information Technology, Vishwakarma Institute of Technology, Pune, 411037, India

OPEN ACCESS

Received: 01.06.2021

Accepted: 15.09.2021

Published: 15.11.2021

Citation: Ghadekar P, Singh G, Datta J, Gupta AK, Anand DS, Khare S, Oswal P, Sharma D (2021) COVID-19 Face Mask Detection using Deep Convolutional Neural Networks & Computer Vision. Indian Journal of Science and Technology 14(38): 2899-2915. https://doi.org/10.17485/IJST/v14i38.996

* **Corresponding author**.

premanand.ghadekar@vit.edu

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To propose a model which could classify in real-time if an individual is wearing a face mask or not wearing a face mask. A lightweight system that could be easily deployed and assist in surveillance. **Methods/Statistical analysis:** Analysis of the proposed model shows a limited number of research studies with regards to facial localizations. Several state-of-the-art methods were taken into considerations out of which the CNN architectural approach is analyzed in this study. Taking into consideration the use-case of deployments and structuring, a new Keras-based model is proposed that surpasses the achievement results of MobileNet-V2 and VGG-16 standard architectures. Effective facial localization is tackled with the MTCNN approach. **Findings:** The system has achieved a confidence score of 0.9914, an average weighted F1-score of 0.98, a precision value of 0.99. The proposed model has been compared with standard architectures of VGG16 and MobileNetV2 with regard to the accuracy, support values, precision, recall, and F1-score metrics. The proposed model performs better w.r.t traditional architectures. The average latency involved in prediction is 0.034 seconds making the average FPS 30 Frames per second. The compact architecture makes the model best for deployment in real-time scenarios. The system incorporates the concept of image localization with Multi-Task Cascaded Convolutional Neural Network (MTCNN) architecture. The analysis shows MTCNN is performing much better than Haar-Cascade in real-time facial prediction scenarios. **Novelty/Applications:** This compact architecture with minimal layers is easily deployable in edge devices. It can be used for mass screening at public places like railway stops, bus stops, streets, malls, entrances, schools, and many service-oriented business verticals requiring users to access the services as long as the mask has been worn correctly.

Keywords: COVID19; Deep learning; Computer Vision; Mask Detection; Deep Convolutional Neural Network; Image Localization

1 Introduction

The ongoing coronavirus outbreak has drastically affected human survival. As per the WHO[1] directives, individuals having respiratory issues or assisting others with such symptoms should always wear face masks. It is so important in COVID-19 pandemic situations, Leung et al.[2] showed that surgical face masks could lessen the spread of the virus. And it is also mandatory to wear Face masks in grocery stores, restaurants, and other public places. This has become the new Normal. As doctors suggest, COVID-19 spreading can be slowed down by strictly maintaining a safe distance from others and wearing a mask when talking or going outside. The pandemic has given rise to worldwide scientific cooperation, and a lot of research with the help of the latest technologies is going on to develop various solutions to fight the pandemic.

A large number of service-oriented business verticals require users to access the services as long as the mask has been worn correctly[3] for general health. This young system has a significant research gap and a limited case study on standard architectures for classifications. As a result, it is crucial to develop an effective face mask detection system, which is capable of detecting face masks for a wide range of applications. Face mask detection refers to detecting if a person is wearing a mask or not the face's location.

Existing systems are incapable of detecting facial masks with drastically low latency. The present systems developed are limited mostly to labs. The major reason behind their incapability is the adoption of large complex in-depth architectures. This makes the model less generalized and makes prediction challenging. Moreover, the present systems are incompatible for effective facial localizations, for better prediction. The major challenge the system faces are threshold resolutions required for predictions.

2 Literature review

Shashi Yadav et al.[4] in Deep Learning-based Safe Social Distancing and Face Mask Detection in Public Areas for COVID19. An efficient approach focused on detecting face masks in public places and social distancing. This computer vision-based system is a transfer learning implementation with Single Shot Detector (SSD) and lightweight neural network MobilenetV2. Hence, achieves a balance of resource limitations and recognition accuracy. This model was deployed on a raspberry pi4 based edge device. The accuracy achieved was between 85% and 95%. This approach gave an aspect of standard SSD architecture-based prediction. However, SSD was a traditional approach for localization, and the latency involved in it was also too high. Along with it pre-processing should also be taken into account.

Mohammad Marufur Rahman et al.[5] using Convolutional Neural Networks, and computer vision proposed a run-time facial mask detection system. Face masks are crucial in times of covid-19 pandemic. The proposed model does image pre-processing to convert the RGB image into a grayscale image. Deep learning architecture called CNN is used to identify whether people are wearing masks or not. As a result, training accuracy of 98.7% and testing accuracy of 0.98 units is obtained. This approach helped in dealing with the pre-processing aspect while training a model. However, there lies a scope of improvement in enhancing the images involved in the training phase. Several approaches like- normalization, edge detectors, sharpening can also be applied.

Walid Harir et al.[6] Efficient Masked Face Recognition Method during the COVID-19 Pandemic discussed how the masks that are our new normal had left earlier proposed face mask detection algorithms incapable of recognizing facial sentiments. Further, they proposed an approach that involves discarding the masked regions and extracting features using deep learning to efficiently detect facial masks. First, the masked face region was omitted, then a transfer learning approach was applied to extract significant features from the region of interest (eyes and forehead). The highest accuracy obtained was 91.3% respectively. This approach has analyzed the transfer learning pipeline. This pipeline helped in setting up the analysis phase of the proposed research where the developed model is compared against the existing architectures using the transfer learning toolkits.

Another research work in this domain is by Vinitha V et al.[7] Covid 19 face mask detection with deep learning and computer vision. The proposed system throws light on identifying faces of individuals wearing masks from video/image frames using image processing, OpenCV, Deep Learning frameworks like TensorFlow, Keras, PyTorch. A deep learning model is previously trained MobileNetV2 and is applied mask detector over images / live video stream. The accuracy for mask and no mask obtained was between 87% and 95%. This approach gave an intuition about the prediction accuracy in a real-time environment. The accuracy parameter was oscillating between 87% to 95% which presents a huge scope for improvement for dealing with images of low resolutions. The research throws light on covering AI edge cases of low-resolution images and is found to be working effectively with great accuracy.

Mingjie Jiang, Xinqi Fan & Hong Yan et al.[8] "Retina Mask: A face mask detector" proposed a single-stage detector, which comprises a feature pyramid network. Also presented a comparative study of Resnet and MobileNet. It was observed in both face and mask detection precision of ResNet with ImageNet pre-train is more than 10% as compared to that with MobileNet backbone. This approach has extensively utilized extraction of the facial region using eyelids which is creating a bottleneck for effective facial localization.

Sethi, S., Kathuria, M et al.[9]. "Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread" proposed a technique of using a group single and two stage detectors at the pre processing level. Such a technique significantly boosted the accuracy and also ameliorated the detection speed.

Singh, S., Ahuja et al. "Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment."[10]. The research discussed some cutting edge deep learning techniques which could improve the accuracy and precision of the model and aid the mask detection process by using bounding boxes which would help in detecting a masked face or a non-masked face in real time.

R. Suganthalakshmi, A. Hafeeza et al.[11]"Covid-19 Facemask Detection with Deep Learning and Computer Vision". The proposed research presented some techniques to optimize the mask detection pipeline and improve the accuracy of the model. The research also discussed some real life applications of the mask detection model.

Batagelj, B.; Peer et al.[12] "How to Correctly Detect Face-Masks for COVID-19 from Visual Information" proposed methods to evaluate the model by introducing a dataset for mask detection. They also presented a comparative of the existing research work and proposed some design methodologies which could be used to improve the performance of the model.

Hence, an architecture with low latency and more accuracy is required which is capable of effective facial localization with low latency, covering AI edge cases for low-resolution images, so that it can be deployed in public places, EDGE devices to detect in real-time whether people are wearing the face mask or not wearing a face mask. The presented study focuses on the drawbacks of previous research and provides a better solution for the same. The proposed model is integration between computer vision and some standard deep learning architectures. The proposed system uses MTCNN for feature extractions which makes it far more accurate for facial localization and combines it with architectures like VGG16, MobileNetV2, CNN for classification. A comparison between VGG16, MobileNetV2, and CNN is briefly introduced below in results and discussions. Significant contributions of this work are:

- Real-time Prediction to check if an individual is wearing a face mask or not wearing a face mask.
- Comparative study of CNN, VGG16, and MobileNet based on their performance analysis.
- Designing architecture with low latency of 0.034 seconds, high accuracy of 99.14%, and 30 Frames per second.
- Optimizing performance by incorporating concepts of image localization with MTCNN architecture.
- Covering AI edge cases of low-resolution images.

3 Methodology

A. Dataset Collection

The dataset consists of two categories, (a)With Mask and (b)Without Mask. Initially, total images without masks were around 2000, and 3500 belonged to the mask categories. As shown in Figure 1, the images in the dataset were biased. By using resampling techniques, the images in the 'without mask' category were increased up to 3300. As shown in Figure 2. The biases in the dataset were also reduced by adding images of different skin colors and features.

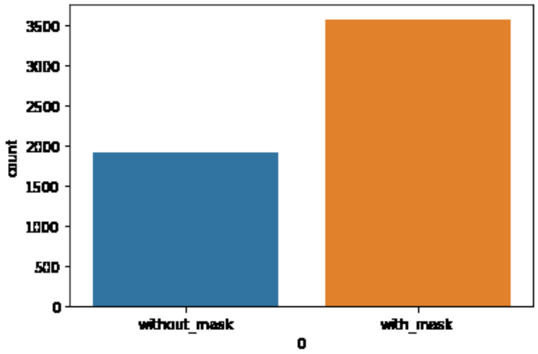

Fig 1. Biased Dataset

Hence, the dataset consists of around 7000 total images. Out of which, 30% is used for validation. After the training is done, the network parameters are to be used for testing.

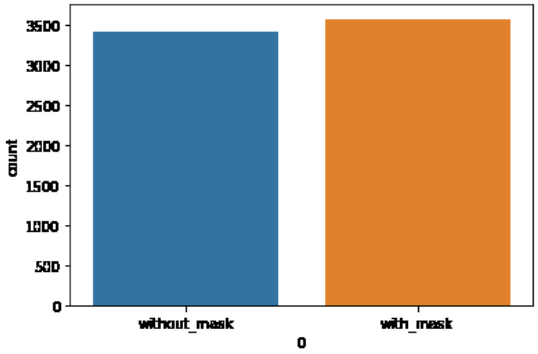

Fig 2. Unbiased Dataset

B. Data Preprocessing

B.1 Data Augmentation: A technique adopted commonly in image data classification, which generates several images for a single source image at different orientations and zoom levels. It helps to substantially increase the data required for predictions and classifications by adjusting the conditions of data biasedness.

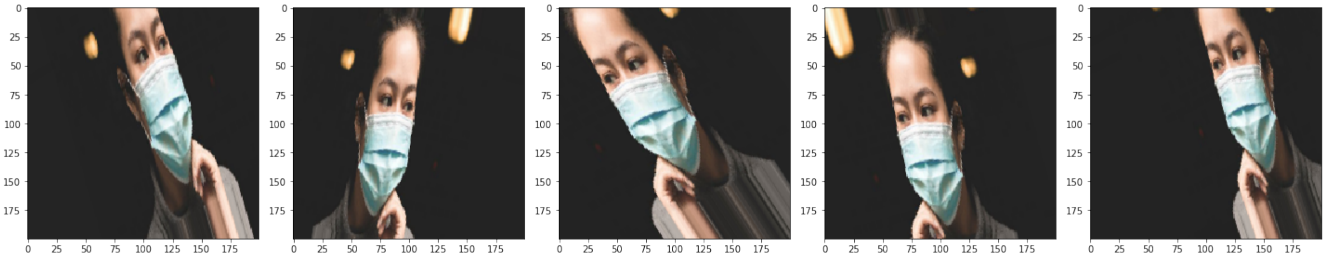

Fig 3. A sample of input data augmented for training phase

At first, the input image is rescaled to reduce the computational cost. Additionally, at different angles the rescaled image is rotated, followed by width shift, height Shift and Horizontal flip at the end.

B.2 Grayscale Conversion: A technique applied on several input images for better classification. It converts an input three-channel RGB image into one channel Grayscale Image. It helps in better feature selection and improved model understanding.

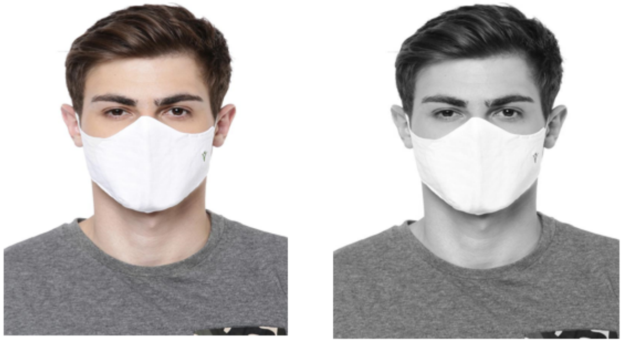

Fig 4. Images after Grayscale Conversion

The open source implementation is found in python's OpenCV library. The script utilised for conversion into grayscale is 'cv2.cvtColor'. It usually enhances the feature extraction pipeline and reduces the computational complexity by ranging down the three channels to one channel margins.

Figure 4 clearly highlights the outputs after grayscale conversion.

B.3 Data Normalization: A technique that involves bringing back the image pixel data values from the range of 0-255. The faster results and efficient computation is achieved by dividing the image pixels by the value of 255 i.e. maximum value of a grayscale pixel. As a result, the resultant image is a grayscale, normalized image with pixel values in between 0 to 1. It reduces the model biasedness by ranging down the pixels on a scale of 1 starting from 0.

It helps in better reading of the model as the system understands the model better within 0-1.

C. Overview of the proposed model

The proposed model takes 3500 images of people wearing masks and 3300 images of people without face masks as input. The first convolutional layer takes input in the form of images and as an activation function, the ReLu layer is added. The pooling layer is stacked up after the ReLu layer. The Max-Pooling Layer is added to reduce the spatial size of the representation. The proposed model has a second convolutional layer with the same sequence of layers viz. activation layer, ReLu layer, and a max-pooling layer. Further, for regularization, a dropout of 0.3 is applied, followed by a flatten layer to reshape the tensor. Again, a dropout layer was used to drop 100 neurons. Finally, the model is tuned by adding another dropout layer of 0.5, and two dense layers of 100 neurons and two neurons have been used.

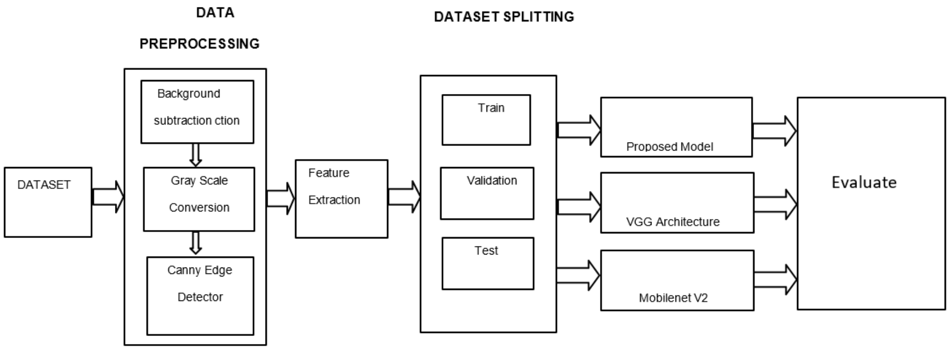

Fig 5. Project Flow Diagram

D. Overview of VGG-16

The proposed model adopts the standard VGG-19 architecture in this field of study. To improve the accuracy of the model, several hyper-parameter tuning approaches have been adopted, dropout to reduce the overfitting conditions and batch

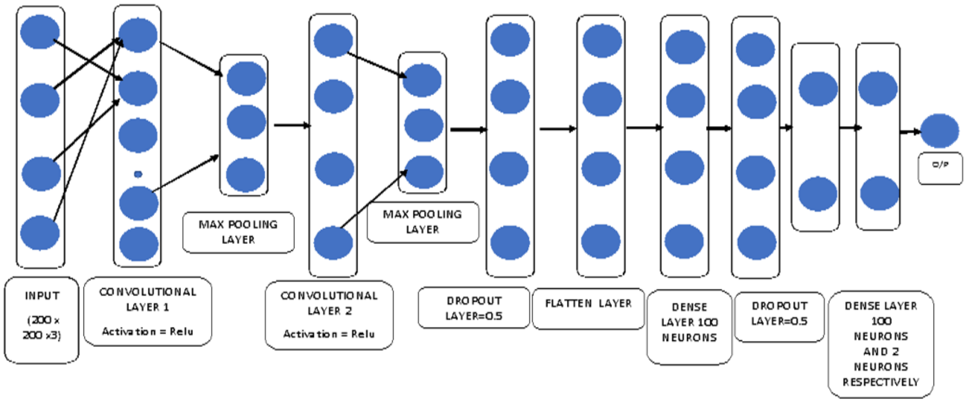

Fig 6. Proposed Model Architecture

normalization layers to improve the training time and accuracy parameters.

VGG is a 16 layer deep convolutional neural network. VGG stands for Visual Geometric Group from Oxford University. This architecture was developed by Simonyan and Zisserman [(13)] in 2014 and was second runner-up in the Visual Recognition Challenge. This architecture has been used widely for classifying images in deep CNN using transfer learning Srikanth et al. (2019) [(14)], classifying objects, etc. dataset [(15)]. The size of the input image to the network is 224x224.

After every max-pooling, the size is reduced by a factor of two. This formula is used to calculate the size of the output layer

$$[(N-f)/S]+1 \tag{1}$$

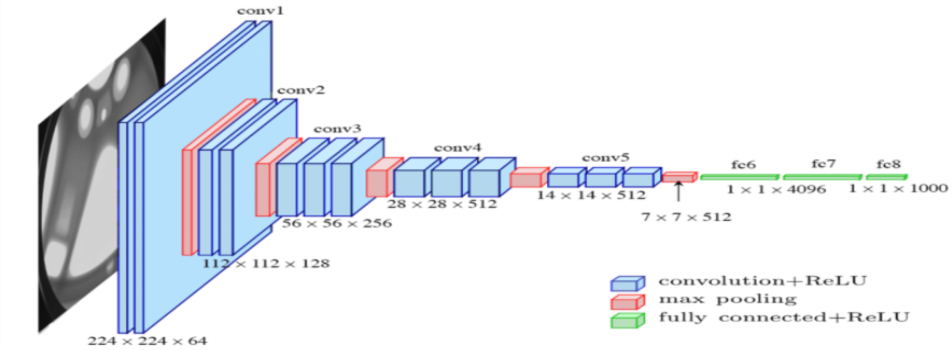

Fig 7. VGG16 Architecture–Image from ResearchGate

In Figure 8, the blue rectangles represent convolution layers with ReLU as the activation function. The green rectangles represent the fully connected layers. In VGG 16 there are two contiguous blocks of convolutional layers. Followed by that there is a max-pooling layer. The max-pooling layer follows all the 3 convolutional layers. Finally, three dense layers. There are 21 layers in total, viz. 13 convolutional layers, 3 fully connected layers, and 5 max-pooling layers. Total 16 layers have tunable parameters, which include 13 convolution layers and three fully connected layers. The output layer uses a SoftMax function which maps 1000 outputs per image in the ImageNet. Here, N stands for input size, f stands for kernel size, and S stands for stride.

E. Overview of Convolutional Neural Network

Convolutional Neural Network is a network of deep neural networks architecture conventionally proposed by Yann LeCun in 1989. It has been used extensively in many verticals like image classification (Pinto et al. 2017) [(16)], textual classification, feature extraction Scarpa et al. (2018) [(17)], Kuo and Huang (2018) [(18)]. A CNN-based architecture has been implemented in the proposed approach for extracting crucial image features and establishing correlations for predictions.

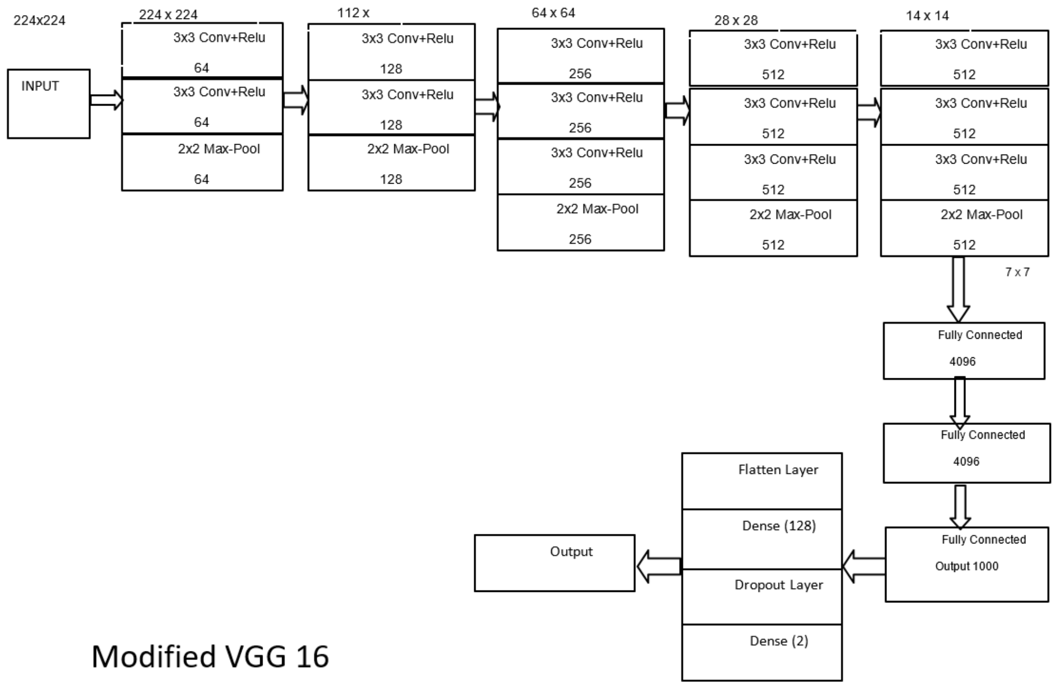

Fig 8. Modified VGG16 Architecture for Binary Classification

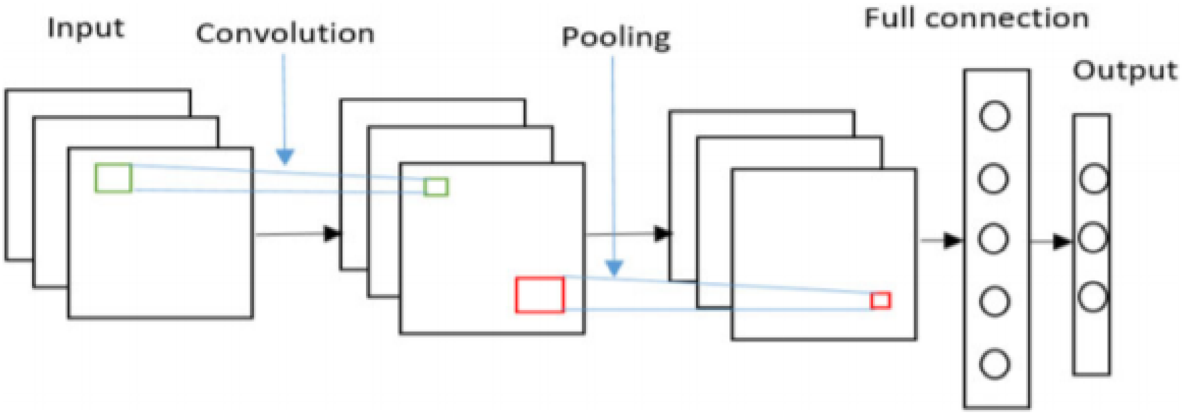

Fig 9. Convolutional Neural Network

In figure no 1, the standard architecture of CNN is shown.

$$Z_{abc}{}^{(n]} = \Sigma_{r=0}^{-1} \Sigma_{t=0}^{h^{(n-1]}-1} \Sigma_{s=0}^{h^{(n-1]}-1} {}^{(n-1]} \times X_{abcd} + v^{(n-1]}{}_{abc} \quad (2)$$

Max Pool is incorporated after CNN architectural layer, abc is the layers involved in the convolutional layers, srt are the lower limits of summations, which includes the filters.

E.1 Activation Functions

Activation functions are generally an output function that helps to determine the output of a neural network in each iteration. There are several activations already proposed in the field of deep learning. The one very relevant to the model is described below.

E.2 ReLu Layer

ReLu stands for Rectified Linear Unit Layer, which acts as an activation layer for CNN architecture models. For Image classification verticals, weights updating via gradient descent are quite negligible. On the contrary, ReLu provides significant weight updating that impacts the training lifecycle of the model significantly.

$$g(x) : max(0,x) \tag{3}$$

The aforementioned equation no. 5 clearly explains the working of the ReLu function. It eliminates all the negatively predicted outputs to 0 and the positively predicted values the same.

E.3 Pooling Layer

This layer maps the significant attribute to a newly formed feature by mapping the most relevant features from the input feature dimensions of this layer.

The Kernel returns the value which is maximum among all possible pixel values in the specified kernel. Hence it reduces the dimensions from 2*2 to 1 by selecting the maximum value from 4-pixel values.

$$maxpool\ = max(y1, y2, y3, y4, \ldots\ldots,\ yn) \tag{4}$$

The aforementioned equation no. 6 clearly explains the working of the max pool function. It extracts the highly significant pixel values.

The Kernel returns the average of all the pixel values in the window size described. Equation no.7 describes the working of the average pooling layer.

$$avg.\ pool = \frac{1}{n}(\textstyle\sum_{i=1}^{n}\ = y_i) \tag{5}$$

Where y_i stands for the individual pixel values.

E.4 Fully-Connected Layer

As the input image has been transformed into a multi-level perceptron, the output of the pooling layer is then flattened into a column vector. The output is then fed to the neural network which feeds it in a forwarding direction also known as the feed-forward neural network.

E.5 MTCNN VS HAAR-CASCADE

Detecting faces is one of the most critical aspects of the proposed research. Haar cascade, which is based on the Viola-Jones detection algorithm. The Haar Cascade classifier is applied to study the pixels in the image into squares, using the Haar wavelet technique. On the other hand, MTCNN or Multi-task Cascaded Convolutional Neural Network consists of convolutional layers stacked together in three stages. MTCNN covers all sides of the faces, whereas Haar Cascade considers only the frontal face. In the proposed work, MTCNN has been used to effectively localize all sides of the face and hence get better recall and precision values. Image localization, the task of drawing bounding boxes around the predicted object, is better done using MTCNN than Haar Cascade, which also strengthened the case to use MTCNN in the proposed model.

4 Results and Discussions

The proposed work has been compared with existing research work in the same domain and it outfromed them. Table 1 below shows a result of some previous notable research with the proposed research work.

The proposed model has been trained on Jupyter Notebook in a python-based environment for 20 epochs. It has also attained an accuracy of 99.14 % on the training set with 96.77 % accuracy on the Validation set. The prescribed model has also been evaluated on a Test set where it has yielded an accuracy of 98.32% with the loss of 0.121. Figure 10 shows the behavior of the model throughout the training journey against the validation set.

Figure 11 shows a curve of accuracy vs. epoch, which shows that accuracy increases as epochs increase. Hence, it can be inferred that the model learns well from the training data and also performs well on unseen data.

For a better understanding, the model is evaluated on several evaluating parameters, including precision, recall, F1-score, and support values. The metrics are calculated by using true and false positives, true and false negatives.

Table 1. Performance comparison w.r.t recent research work

Sr. No.	Research Paper	Accuracy Achieved
1.	Shashi Yadav et al.[4] Deep Learning-based Safe Social Distancing and Face Mask Detection in Public Areas for COVID19.	85%-95%
2.	Vinitha V(2020).COVID-19 Facemask Detection with Deep Learning and Computer Vision.International Research Journal of Engineering and Technology (IRJET)][7]	87%-95%
3.	Walid, Hariri. (2020). Efficient Masked Face Recognition Method during the COVID-19 Pandemic[6]	91.3%
4.	Rahman, Mohammad et.al.(2020).An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network[5]	98.7%
5.	Proposed Model	99.14%

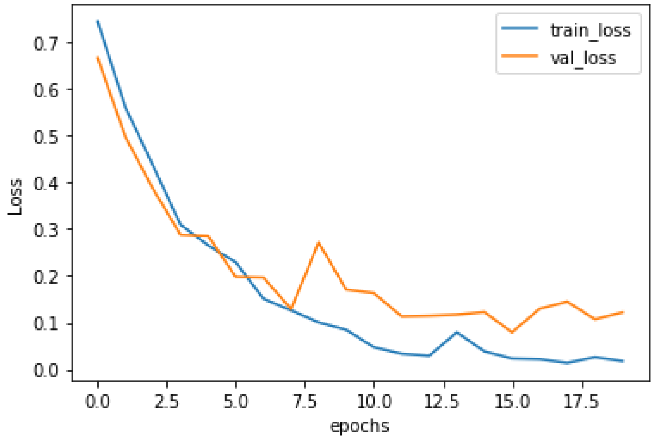

Fig 10. Loss vs. epochs plot for the proposed model

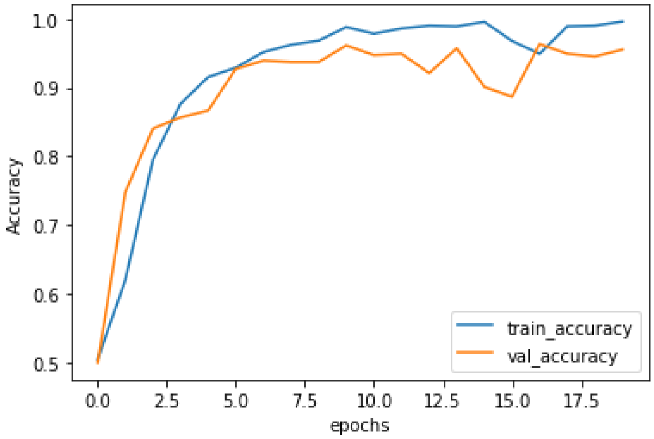

Fig 11. Accuracy vs. epoch for the proposed model

Table 2. Classification report for the proposed model

Class	Precision	Recall	f1-score	Support
No Mask	0.98	0.99	0.98	686
Mask	0.99	0.98	0.98	690
accuracy			0.98	1376
macro average	0.98	0.98	0.98	1376
weighted average	0.98	0.98	0.98	1376

Table 2 shows that the model learns effectively in its training phase, and when evaluated on testing phase, it predicts correct classes with a shallow error rate. The average weighted F1-score achieved was 0.98 for classification. The model learns effectively during its training phase and predicts accurately on validation sets.

Further, for getting more insights into the performance of the model, the error matrix is plotted. Figure 12 shows the confusion matrix for the proposed model.

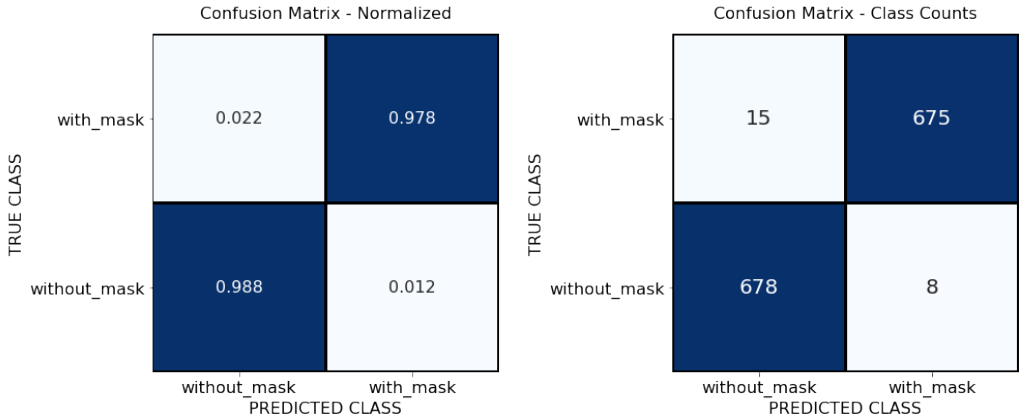

Fig 12. Confusion matrix

From the above confusion or error matrix, it can be stated that the proposed model performs well on unseen data also. The model predicts 23 false predictions on data of 1376 unseen images with an error rate of 1.67%.

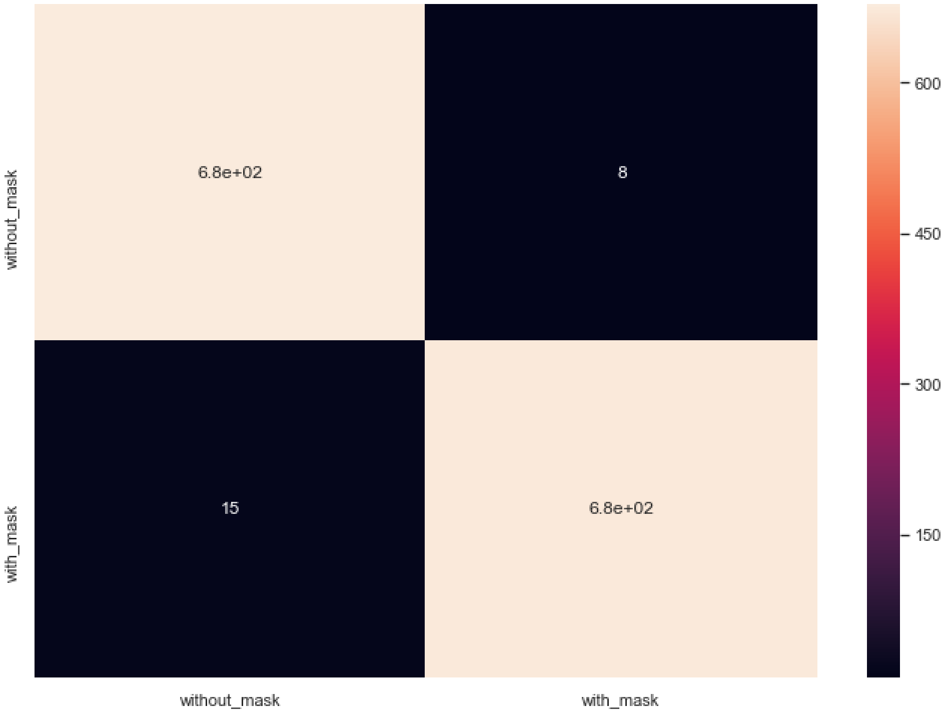

Fig 13. Heatmap for the proposed model

The above Figure 13 shows that the proposed model learns effectively throughout its training journey. Each of the individual classes is highly positively correlated with its class using the color-coding approach.

For further analysis, some standard architectures have been tested on the same data set. VGG-16 is a 16-layer deep convolutional-based neural network model as proposed by K. Simonyan and A. Zisserman[19] in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition."

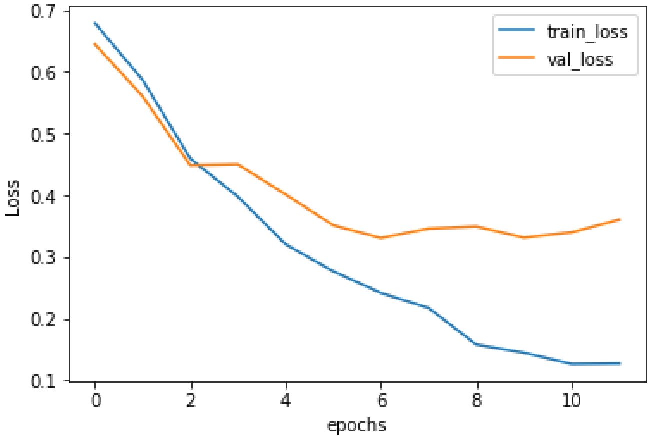

Fig 14. Loss vs. epochs plot for VGG model

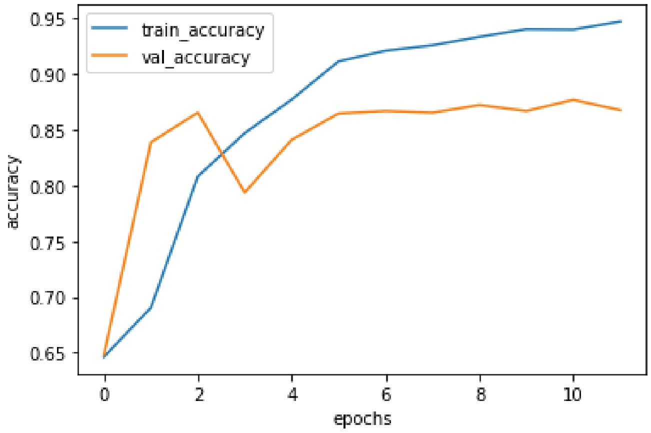

Fig 15. Accuracy vs. epochs plot for VGG model

Figure 14 discusses accuracy vs. epoch and loss vs. epoch curves of the VGG model. VGG16 is a 16-layer deep architecture that requires a considerable number of weight parameters. As a result, it increases the inference time. As shown in Figure 15 , it can be interpreted that accuracy of the VGG-Model increases while increasing epochs, but it fails to generalize effectively. It learns even fine noise details from the training set. Consequently, increases the difference between training and validation accuracy.

To get more insights about the performance of VGG16, a classification report is plotted, as shown in Table 3 .

As mentioned in Table 3, it can be interpreted that VGG16 classifies the images with no mask correctly with an accuracy of 90%, and images with the mask are classified correctly for about 94% time.

From the above matrices, it infers that VGG16 does not generalize the data effectively on unseen data or target data. It can be interpreted from Figure 16. Around 72 images with masks have been incorrectly classified as images without masks, and 40 images of the category without masks have been classified into the category of "with mask." Further, the test set accuracy of VGG16 can be increased by adding some regularization.

Although variants of VGG had pretty good accuracy, they were not efficient or 'light' enough for a real-world application. To be used at edge devices the architecture needs to be smaller and faster. MobileNetV2 chose to factorize a convolution into

Table 3. Classification report for VGG16

Class	Precision	Recall	f1-score	Support
No Mask	0.90	0.94	0.92	686
Mask	0.94	0.90	0.92	690
accuracy			0.92	1376
macro average	0.92	0.92	0.92	1376
weighted average	0.92	0.92	0.92	1376

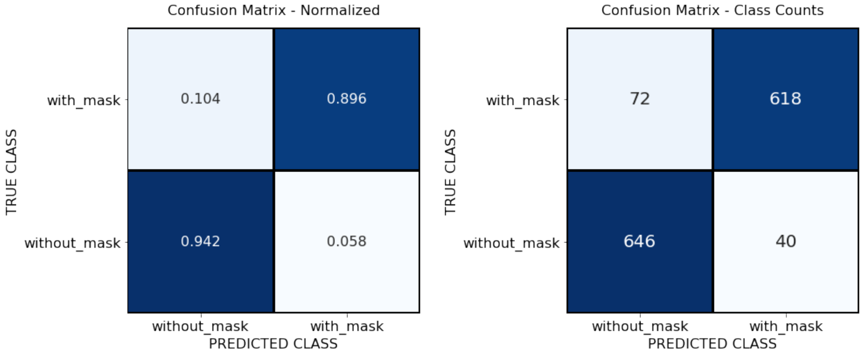

Fig 16. Confusion matrix for VGG16

two steps. Depth-wise separable convolutions, MobileNetV2 could reduce computation and the model size dramatically. Hence MobileNetV2 is also analyzed. The results of MobileNetV2 can be seen below.

Table 4. Classification report for MobileNetV2

Class	Precision	Recall	f1-score	Support
No Mask	0.93	0.14	0.24	686
Mask	0.54	0.99	0.70	690
accuracy			0.57	1376
macro average	0.73	0.57	0.47	1376
weighted average	0.73	0.57	0.47	1376

Table 4 shows that the MobileNetV2 model is not capable of efficient predictions. It gives average weighted F1-score metrics as 0.57, which is considered to be the lowest among all evaluated architectures in this field of study. The reason being that the resulting network, when modified as per binary classification, gets very complex. The model learns effectively on the training data. It learns even fine details of images so accurately that it ends up miss-classifying the predicted data. As a result, the model gets saturated while evaluating the validation set of data.

Above mentioned Figures 17 and 18 , shows Accuracy vs. epoch and Loss vs. epoch curves. It shows that the model learns effectively but fails to generalize effectively on test data. A primary reason being the over-learning of the trained parameters in the training phase of the model.

The above Figure 19 shows that the MobileNetV2 model is giving a considerable amount of false predictions. The total number of images of the category with the mask has been classified into without mask are 7 in number, while pictures of the category without the mask that are classified into with mask are total of 590 in number. It shows that MobileNetV2 predicts images with an error rate of 43.38%.

The proposed model in this field of study has been evaluated on several resolutions of images for an in-depth inference. The idea behind carrying out this analysis is to find the threshold resolution of images for predictions. The carried analysis is shown in Figure 19.

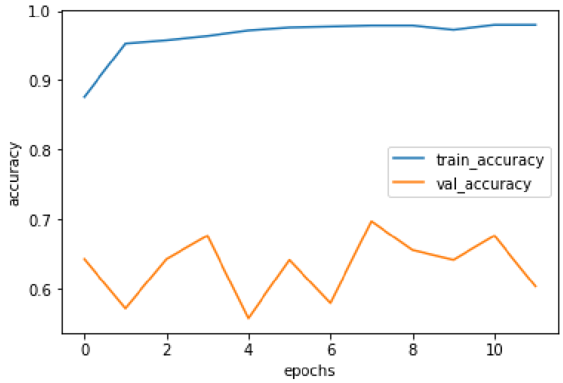

Fig 17. Accuracy vs. epoch plot for MobileNetV2

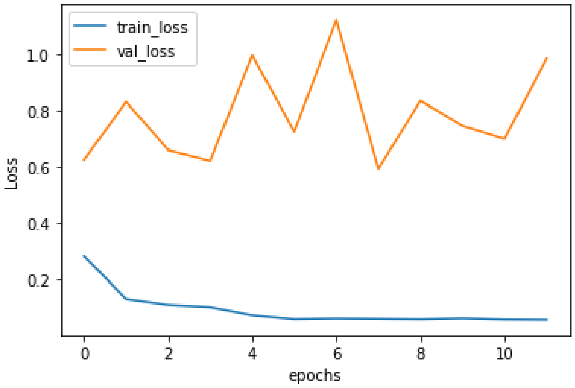

Fig 18. Loss vs. epoch plot for MobileNetV2

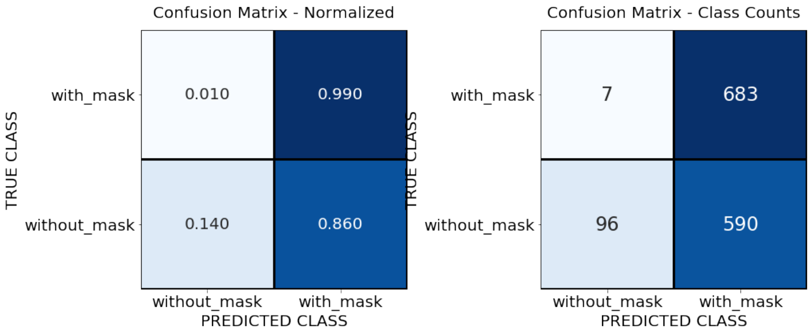

Fig 19. Confusion matrix for MobileNetV2

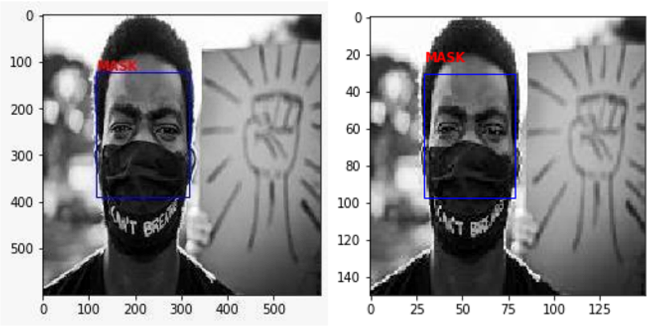

Fig 20. Resolution: 600 x 600 and 150 x 150

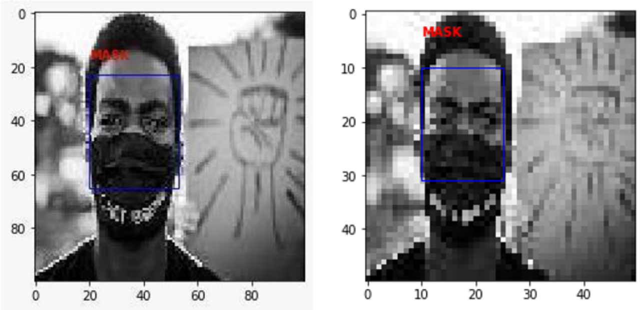

Fig 21. Resolution: 100 x 100 and 50 x 50

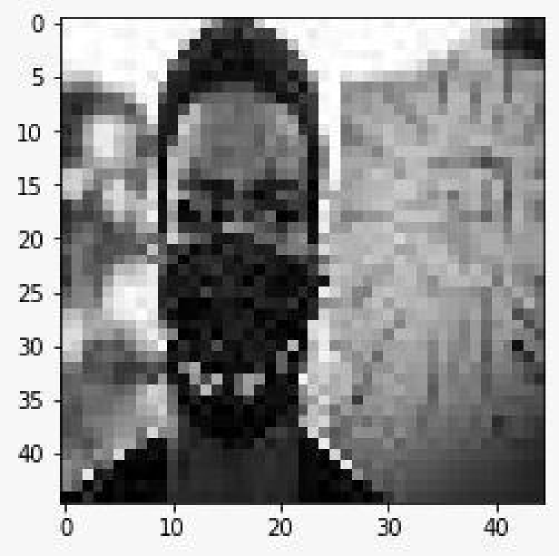

Fig 22. Resolution 45 x 45 is not detected

The above analysis shows that 50 x 50 is the threshold resolution for the proposed model. All images of resolutions 50 x 50 and above are correctly classified by the model into their respective classes making it deployable for real-time systems.

It is further essential to evaluate the model for bias conditions. It enables the interpretation of the behavior of the model for edge cases. The carried analysis is shown in Figure 23.

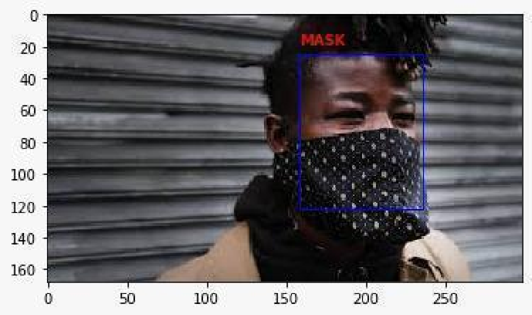

Fig 23. Black mask on a dark shaded person

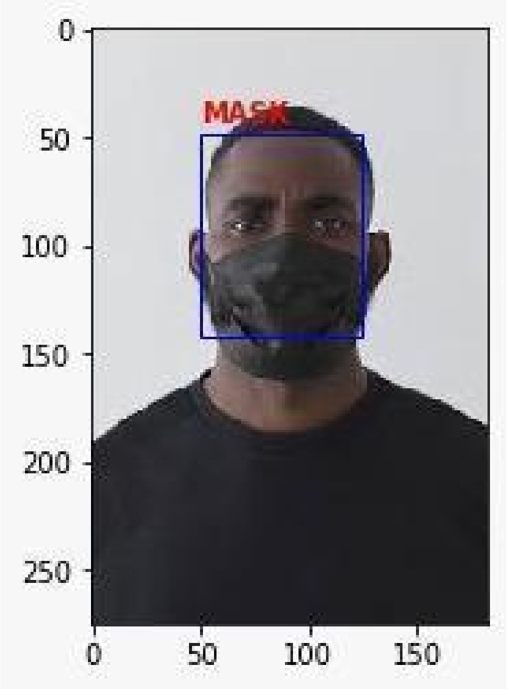

Fig 24. Black mask on a Dark shaded person

The above figure clearly shows that the model can generalize between a black mask and a similar shaded person. Even in the regions of low contrast differences, the model is found to work with high accuracy.

One of the crucial factors while interpreting models is the latency involved with it for predictions. The proposed model has been analyzed and evaluated for the time taken for predictions. A total of 10 different image frames have been fed to the model for inference time evaluation. The average time taken for the prediction of a single input image is shown in Table 5.

The above table shows that the proposed model takes on an average of 0.034 seconds for predicting an input image when fed to the model. Further, it was noted that the minimum time taken by a particular frame to the process was 0.033 seconds, which makes the best FPS achieved by the system is 30 frames per second.

5 Limitations and Future Scope

The proposed model developed in this field of study works well in classifying the images in real-time into mask or no-mask categories. However, the FPS and Inference time involved for the prediction can be optimized and the scope of improvement lies in this domain. Also, the model can be improved to work efficiently on diverse AI-based edge cases covering cases of 'skin-colored masks' on 'skin-colored faces'. The threshold resolution which the model needs for prediction is 50 x 50. This creates a challenge for other resolutions lower than the mentioned threshold. In the near future, these challenges can be improved

Table 5. Average Inference Time taken by the model for predictions

Sr. No.	Frame Sequence	Prediction Time(seconds)
1	Frame No. 1	0.040 sec
2.	Frame No. 2	0.037 sec
3.	Frame No. 3	0.032 sec
4.	Frame No. 4	0.025 sec
5.	Frame No. 5	0.034 sec
6.	Frame No. 6	0.040 sec
7.	Frame No. 7	0.035 sec
8.	Frame No. 8	0.027 sec
9.	Frame No. 9	0.037 sec
10.	Frame No. 10	0.036 sec
Average Prediction Time		0.034 sec

drastically by accommodating some state-of-the-art architectures for facial localization. Moreover, the performance can be further optimized by improving the FPS (Frames Per Second) and Inference time involved in predictions. The experiments can also be carried out on GPU-based cores supported by NVIDIA, to accelerate the computation and enhance parallel processing capabilities for the frames. The traditional processing pipelines can be further optimized by the rapids framework, in the upcoming research studies for lower inferences, better performance in terms of effective AI edge-cases detection, and effective localization.

6 Conclusion

Wearing a face mask has now become a new normal. In the proposed system, MTCNN was used instead of Haar Cascade for image localization as it gave better recall and precision values while detecting in real time if a person is wearing a face mask or not. An accuracy of 99.14 % on the training Set and 96.77 % on the validation set was achieved. The proposed model has also been evaluated on a test set where it gave an accuracy of 98.32% with a loss of 0.121. The average inference time taken by the model for predictions is 0.034 secs with an FPS of 30. The proposed model also outperformed compared to some standard architectures like VGG 16 and MobileNetV2, which gave an accuracy of 90% and an average F1 score metrics of 0.57, respectively. However, the proposed research can be further improved by using better image localization techniques and overall enhancing the FPS and making the model work with biases and also with low resolution image or video feeds. The model architecture can also be made lightweight and can be made to use as an edge device.

References

1) https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public. .
2) Leung NH, Chu DK, Shiu EY, Chan KH, Mcdevitt JJ, Hau BJ, et al. Respiratory virus shedding in exhaled breath and efficacy of face masks. *Nature Medicine*. 2020;26:676–680. Available from: https://doi.org/10.1038/s41591-020-0843-2.
3) Fang Y, Nie Y, Penny M. Transmission dynamics of the COVID-19 outbreak and effectiveness of government interventions: A data-driven analysis. *Journal of Medical Virology*. 2020;92(6):645–659. Available from: https://dx.doi.org/10.1002/jmv.25750.
4) Yadav S. Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence. *International Journal for Research in Applied Science and Engineering Technology*. 2020;8(7):1368–1375. Available from: https://dx.doi.org/10.22214/ijraset.2020.30560.
5) Rahman MM, Manik MMH, Islam MM, Mahmud S, Kim JH. An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network. *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. 2020.
6) Walid H. Efficient Masked Face Recognition Method during the COVID-19 Pandemic. *Research Square*. 2020. doi:10.21203/rs.3.rs-39289/v1.
7) Vinitha V. COVID-19 Facemask Detection with Deep Learning and Computer Vision. *International Research Journal of Engineering and Technology*. 2020;7(8). Available from: https://www.irjet.net/archives/V7/i8/IRJET-V7I8530.pdf.
8) Jiang M, Fan X. RetinaFacemask:A Face Mask Detector. 2020. Available from: https://arxiv.org/abs/2005.03950.
9) Sethi S, Kathuria M, Kaushik T. Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread. *Journal of Biomedical Informatics*. 2021;120:103848. Available from: https://dx.doi.org/10.1016/j.jbi.2021.103848.
10) Singh S, Ahuja U, Kumar M, Kumar K, Sachdeva M. Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. *Multimedia Tools and Applications*. 2021;80(13):19753–19768. Available from: https://dx.doi.org/10.1007/s11042-021-10711-8.
11) Suganthalakshmi R, Hafeeza A, Abinaya P, Devi A. Covid-19 Facemask Detection with Deep Learning and Computer Vision. *International Journal of Engineering Research & Technology(IJERT) ICRADL - 2021*. 2021;9(5).
12) Batagelj B, Peer P, Štruc V, Dobrišek S. How to Correctly Detect Face-Masks for COVID-19 from Visual Information? *Applied Sciences*. 2021;11(5):2070. Available from: https://dx.doi.org/10.3390/app11052070.

13) Simonyan K, Zisserman A. Very Deep Convolutional Networks For Large-Scale Image Recognition. 2015. Available from: https://arxiv.org/pdf/1409.1556.pdf.

14) Tammina S. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *International Journal of Scientific and Research Publications*. 2019;9(10):143–150. Available from: http://www.ijsrp.org/research-paper-1019.php?rp=P949194.

15) Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009. Available from: https://ieeexplore.ieee.org/document/5206848.

16) Rawat W, Wang Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*. 2017;29(9):2352–2449. Available from: https://dx.doi.org/10.1162/neco_a_00990.

17) Scarpa G, Gargiulo M, Mazza A, Gaetano R. A CNN-Based Fusion Method for Feature Extraction from Sentinel Data. *Remote Sensing*. 2018;10(2):236. Available from: https://www.mdpi.com/2072-4292/10/2/236.

18) Kuo PH, Huang CJ. An Electricity Price Forecasting Model by Hybrid Structured Deep Neural Networks. *Sustainability*. 2018;10(4):1280. Available from: https://dx.doi.org/10.3390/su10041280. doi:10.3390/su10041280.

19) Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science*. 2014. Available from: https://arxiv.org/abs/1409.1556.