




A real time face mask detection system using convolutional neural network

Hiten Goyal¹ · Karanveer Sidana¹ · Charanjeet Singh¹ · Abhilasha Jain¹  · Swati Jindal¹

Received: 25 August 2021 / Revised: 29 November 2021 / Accepted: 3 January 2022 /
Published online: 25 February 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

In current times, after the rapid expansion and spread of the COVID-19 outbreak globally, people have experienced severe disruption to their daily lives. One idea to manage the outbreak is to enforce people wear a face mask in public places. Therefore, automated and efficient face detection methods are essential for such enforcement. In this paper, a face mask detection model for static and real time videos has been presented which classifies the images as “with mask” and “without mask”. The model is trained and evaluated using the Kaggle data-set. The gathered data-set comprises approximately about 4,000 pictures and attained a performance accuracy rate of 98%. The proposed model is computationally efficient and precise as compared to DenseNet-121, MobileNet-V2, VGG-19, and Inception-V3. This work can be utilized as a digitized scanning tool in schools, hospitals, banks, and airports, and many other public or commercial locations.

Keywords OpenCV · Convolutional neural network (CNN) · COVID-19 · Deep learning · Real-time face mask detection

1 Introduction

As the COVID-19 (Coronavirus) pandemic continues to spread, most of the world’s population has suffered as a result. COVID-19 is a respiratory disorder that results in severe cases of pneumonia in affected individuals [14]. The disease is acquired via direct contact with an infected person, as well as through salivation beads, respiratory droplets, or nasal droplets released when the infected individual coughs, sneezes, or breathes out the virus into an airspace [16]. Globally, thousands of individuals die from the COVID-19 virus daily. A Coronavirus (COVID-19) report by the World Health Organization (WHO) reveals that, as of 22 November 2021, there were 258 million confirmed cases of COVID-19

✉ Abhilasha Jain
abhilasha_cse@mrsptu.ac.in

¹ Department of Computer Science and Engineering, Maharaja Ranjit Singh Punjab Technical University, Bathinda, India

cases and 5,148,221 deaths worldwide [24]. Therefore, people should wear face masks and keep a social distance to avoid viral spread of disease. An effective and efficient computer vision strategy intends to develop a real-time application that monitors individuals publicly, whether they are wearing face masks or not. The first stage to identify the existence of a mask on the face is identifying the face. This divides the whole procedure into two parts: face detection and mask detection on the face.

Computer vision and image processing have an extraordinary impact on the detection of the face mask. Face detection has a range of case applications, from face recognition to facial movements, where the latter is required to show the face with extremely high accuracy [10]. As machine learning algorithms progress rapidly, the threats posed by face mask detection technology still seem effectively handled. This innovation is becoming increasingly important as it is used to recognize faces in images and in real-time video feeds.

However, for the currently proposed models of face mask detection, face detection alone is a very tough task [1]. In creating more improved facial detectors, following the remarkable results of current face detectors, the analysis of events and video surveillance is always challenging.

Recent years have seen the rise of big data as well as a dramatic rise in the capabilities of computers that use parallel computing [9]. Target detection has become an important research area in computer vision and is also extensively utilized in the real world. For instance, traditional techniques for targeting, like face recognition, autonomous driving, and even target tracking, employ artificially extracted features; however, there are some issues that include incomplete feature extraction, and a weak recognition effect [17]. With the introduction of convolutional neural networks, significant advances have already been made in the field of image classification [15].

In research papers [21, 23], authors have used predefined standard models like VGG-16, Resnet, MobileNet, which require large memory and computational time. In this paper, an effort was made to customise the model in order to reduce memory size, computing time, and boost the accuracy of the model's findings. This paper presents a face mask detection system based on deep learning. The presented approach can be used with surveillance cameras to detect persons who do not wear face masks and hence restrict COVID-19 transmission.

The major contribution of the proposed work is given below:

1. Created a facemask detector implemented in three phases to assist in precisely detecting the presence of a mask in real-time using images and video streams.
2. The dataset is made available on GitHub that consists of masked and unmasked images. This data set can be used to create new face mask detectors and use them in a variety of applications.
3. The proposed model has used some popular deep learning methods to develop the classifier and gather photos of an individual wearing a mask and distinguish between face masks and non-face mask classes. This work is implemented in Python along with Open-CV and Keras. It requires less memory and computational time in comparison to other models discussed later Section 5 which makes it easy to deploy for surveillance.
4. The study offers future research recommendations based on the findings for developing reliable and effective AI models that can recognize faces in real-life situations.

The structure of the remaining paper is as follows: Section 2 presents the literature review in the field of face mask detection from various scholarly sources. Section 3 contains information about dataset used for training the purposed model. Section 4 comprises planning

and methodology followed for developing the model. Section 5 contains results and comparisons with other models based on various parameters. Finally, Section 6 concludes the paper, followed by the discussions about future work.

2 Related work

Over the past few years, object recognition algorithms employing deep learning models have become theoretically more competent when compared to shallow models in tackling complicated jobs [21]. One example is building a real-time system/model that is capable of detecting whether people have worn a mask or not in public areas.

Shaik and Ahlam [7] utilised real-time deep learning to classify and recognize emotions, and VGG-16 was used to categorise seven faces. This approach thrives under the current Covid-19 lock-down period for preventing the propagation of cases. Moreover, Ejaz et al. [4] used main component analysis to recognize individuals with a masked from those with an unmasked face.

In an attempt to track and enforce compliance to health protocols, one application of facial recognition was implemented by Li et al. [11] using CNN (Convolution Neural Network) to identify whether an individual was wearing a mask or not. They developed the HGL method to classify head poses using masks for faces, which made use of the analysis of colour texture in photographs and line portraits. The front accuracy was 93.64%, meanwhile the side-to-side accuracy was measured at 87.17%.

Qin and Li [20] constructed a face mask recognition design using the condition identification process. The problem was broken down into four parts in the paper: preprocessing the picture, cropping the facial regions, super-resolution operation, and predicting the end condition. The primary innovation during this study was that it used super-resolution to improve low-quality image performance. The proposed method by the author used SRCNet and detected face masks and the mask's position with an accuracy of 98.7%.

Nizam et al. [22] created a GAN-based system to remove any facemasks that are detected and synthesise missing facial components with finer details and the reconstruction of regions. The proposed GAN utilised two discriminators: the first took the structure of the face mask, meanwhile the second was capable of extracting the region obscured by the face mask. In the process of training the model, they employed two synthetic datasets.

In [12], the authors utilize the Darknet-53 (YOLOv3 algorithm) for facial detection.

Deep Learning is primarily a mixture of machine learning and artificial intelligence. It has largely proven to be more adaptable and produce more precise models than machine learning inspired by brain cell functionality [19].

The authors of [2] invented a mobile phone-based detection method. Three elements were retrieved from GLCMs of face masks' micro-images. The KNN algorithm was employed to perform a three-result recognition analysis, with an overall accuracy of 82.87%. To identify face masks in micro-photos, the algorithm utilized a grey level co-occurrence matrix. However, the model only worked on cellphones and therefore wouldn't be suitable for all applications.

The authors of [23] recommended the use of a pre-trained MobileNet with the global pooling block that can be used for facial recognition and detection. The pre-configured MobileNet creates a multi-dimensional component map from a shaded image. Over-fitting is not a problem in the model proposed since it uses an overall pooling block.

3 Dataset

The dataset used in this research was collected in various picture formats such as JPEG, PNG, and others. Figure 1 exhibits the sample of the dataset. It was a mixture of different open-source datasets and images, including Kaggle’s dataset for Face Mask Detection by Omkar Gaurav. As a result, there were different varieties of images with variations in size and resolution. All the photos were from open-source resources, out of which some resemble real-world scenarios, and others were artificially created to put a mask on the face.

Omkar Gaurav’s dataset gathered essential pictures of faces and applied facial landmarks to find the individual’s facial characteristics in the image. Major Facial landmarks include the eyes, nose, chin, lips, and eyebrows. This intelligently creates a dataset by forming a mask on a non-masked image.

Finally, the dataset was divided into two classes or labels. These were ‘with_mask’ and ‘without_mask’, and together, the images were curated, aggregating to around 4000 images. The data set can be downloaded from [GitHub](#).



Fig. 1 Preview of Face mask Dataset

4 Proposed methodology

In order to predict whether a person has put on a mask, the model requires learning from a well-curated dataset, as discussed later in this section. The model uses Convolution Neural Network layers (CNN) as its backbone architecture to create different layers. Along with this, libraries such as OpenCV, Keras, and Streamlit are also used. The proposed model is designed in three phases: Data pre-processing, CNN model training and Applying face mask detector as described in Fig. 2.

4.1 Data pre-processing

The accuracy of a model is dependent on the quality of the dataset. The initial data cleaning is done to eliminate the faulty pictures discovered in the dataset. The images are resized into a fixed size of 96 x 96, which helps to reduce the load on the machine while training and to

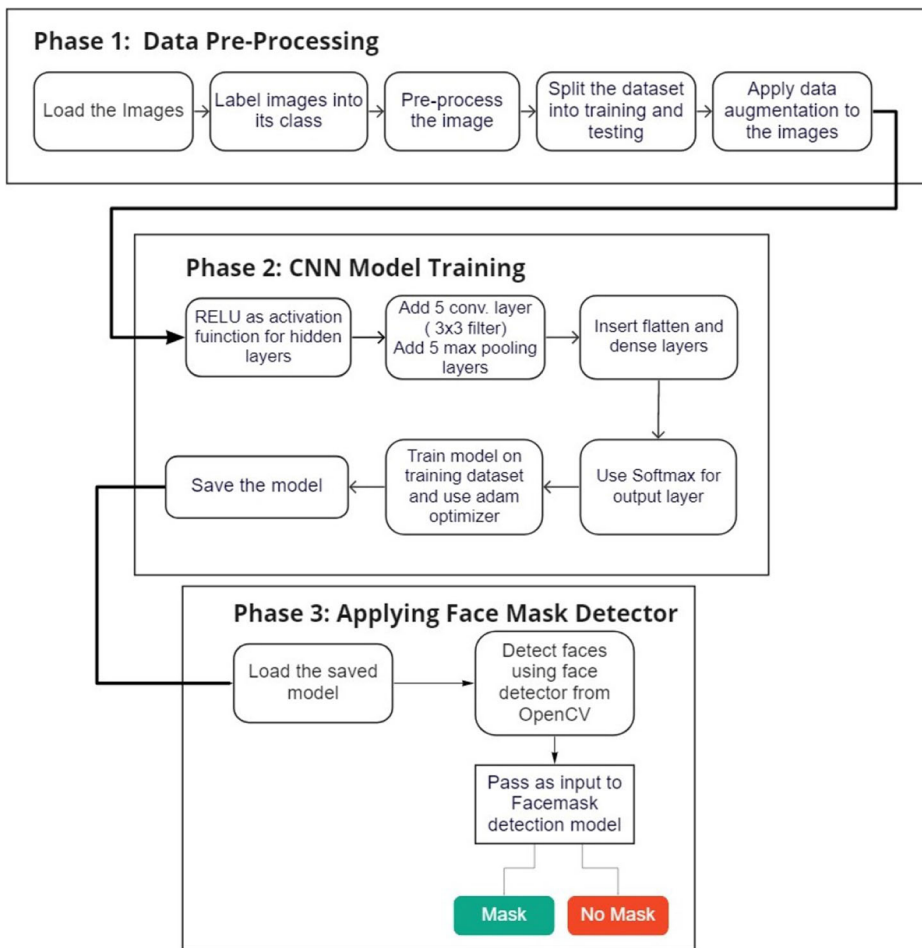


Fig. 2 The Proposed Architecture

provide optimum results. The images are then labelled as being with or without masks. The array of images are then transformed to a NumPy array for quicker computation. Along with that, the *preprocess_input* function from the MobileNetV2 is also used. Following that, the data augmentation technique is utilized to increase the quantity of training dataset and also improve it's quality. A function *ImageDataGenerator* is used with appropriate values of rotation, zoom, horizontal or vertical flip, to generate numerous versions of the same picture. The training samples has been increased to elude over-fitting. It enhances generalization and robustness of the trained model. The whole dataset is then divided into training data and test data in a ratio of 8:2 by randomly selecting images from the dataset. The *stratify* parameter is used to keep the same proportion of data as in the original dataset in both the training and testing datasets.

4.2 CNN model training

The model architecture adopted for the research is described in Table 1. The main components of the architecture are 2D convolutional layers (conv2D), pooling layer, activation functions and fully-connected layers. The proposed model comprises of a total of 5 *Conv2D* layers with padding 'same' and stride of 1. At each conv2D layer, feature map of 2D input data is extracted by "sliding input" across a filter or kernel and perform following operation:

$$C(Z) = (P \times Q)(x) = \left(\int_{-\infty}^{\infty} P(Z) \times Q(Z - x) dZ \right) \quad (1)$$

In the above, P represents the matrix of the input image, and Q is convolutional kernel giving C as output.

Pooling layers decrease the size of the feature map. Thus, the number of trainable parameters is reduced, resulting in rapid calculations without losing essential features. Two major kinds of pooling operations can be carried out: max pooling and average pooling. Max pooling implies making the most significant value present in the specific location where the

Table 1 Model Summary

Layer(type)	Output Shape	Structure
Conv2D	96 x 96	Filters = 16, Filter Size = 3 x 3, Stride = 1
MaxPooling	48 x 48	Filters = 16, Filter Size = 2 x 2, Stride = 1
Conv2D	48 x 48	Filters = 32, Filter Size = 3 x 3, Stride = 1
MaxPooling	24 x 24	Filters = 32, Filter Size = 2 x 2, Stride = 1
Conv2D	24 x 24	Filters = 64, Filter Size = 3 x 3, Stride = 1
MaxPooling	12 x 12	Filters = 64, Filter Size = 2 x 2, Stride = 1
Conv2D	12 x 12	Filters = 128, Filter Size = 3 x 3, Stride = 1
MaxPooling	6 x 6	Filters = 128, Filter Size = 2 x 2, Stride = 1
Conv2D	6 x 6	Filters = 256, Filter Size = 3 x 3, Stride = 1
MaxPooling	3 x 3	Filters = 256, Filter Size = 2 x 2, Stride = 1
Classification Layer	–	Fully-connected, Softmax

Total params: 2,818,658

Trainable params: 2,818,658

Non-trainable params: 0

kernel resides. On the other hand, average pooling computes the mean of every value in that region.

Activation functions are the nodes that are placed at the end or among neuronal networks (layers). They decide whether or not the neuron fires. Choice of activation function at hidden layers as well as at output layer is important as it controls the quality of model learning. The ReLU activation function is primarily used for hidden layers; whereas, Softmax is used for the output layer and calculates probability distribution from a real number vector. The latter is the preferred choice for multi-class classification problems. Regarding ReLU, it offers better performance and widespread depth learning compared to the function of sigmoid and tanh [18].

After all Convolutional layers have been implemented, the FC layers are applied. These layers help to classify pictures in both the multi class and binary categories. In these layers, the softmax activation function is the choice of preference to produce probabilistic results.

$$\text{ReLU} : f(x) = \max(0, x) \quad (2)$$

$$\text{Softmax} : (x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (3)$$

4.2.1 Training of face mask detection model

The classification of a supervised learning CNN model is done after its training to classify the trained images to their respective classes by learning important visual patterns. TensorFlow and Keras are the primary building blocks for the proposed model. In this study, 80% of the dataset contributes to the training set and the rest to the testing set. The input image is pre-processed and augmented using the steps described above. There is a total of 5 *Conv2D* layers with ReLU activation functions with a 3 x 3 filter and 5 Max-Pooling Layers with a filter size of 2 x 2. Flatten and Dense are used as the fully connected layers. The output layer uses softmax as its activation function. This results in 2,818,658 trainable parameters in this Convolutional Neural Network (see Table 1). Tables 2 and 3 demonstrate the training process that is implemented using SGD and Adam. It compares them on different parameters such as Accuracy, Loss, Validation Accuracy, and Validation Loss. From Table 1 and Table 2, it is observed that Adam Optimizer works better than SGD optimizer as it provides better results with an increase in epochs. Figures 3 and 4 shows Adam optimizer gives better performance than SGD optimizer in all recall levels. The hyper-parameters used in this model are described below in Table 4. Binary_crossentropy is used for calculation of the classification loss for the model. For a classification problem, it yields a value between 0 and 1 (probability value).

4.3 Applying face detection model

4.3.1 Face detection using open-CV

The face detection model used is pre-trained on 300x300 image size with 140,000 iterations. The DNN face detector uses a ResNet-10 architecture built upon the Single Shot Detector (SSD) framework as a base model. A Single Shot Detector applies a single shot to identify numerous objects in a picture using a multi-box. Therefore, it has a much quicker and more accurate object detecting system.

Table 2 Training Model on SGD Optimizer

Epochs	loss	accuracy	val_loss	val_accuracy
10	0.6913	0.5000	0.6888	0.5100
20	0.6902	0.5025	0.6872	0.5100
30	0.6890	0.5075	0.6857	0.5100
40	0.6882	0.5075	0.6841	0.5250
50	0.6873	0.5138	0.6826	0.5300
60	0.6858	0.5400	0.6808	0.5400
70	0.6851	0.5562	0.6789	0.5600
80	0.6836	0.5738	0.6769	0.5800
90	0.6823	0.5875	0.6747	0.6100
100	0.6807	0.5962	0.6722	0.6250

Another deep learning framework is Caffe, which is created and maintained by Berkeley AI Research (BAIR) and other partners for the community as a faster, more powerful, and more effective alternative to existing object detection techniques. To use the model, Caffe model files are required that can be downloaded from the OpenCV GitHub repository. The *deploy.prototxt* file describes the network architecture and *res10_300x300_ssd_iter_140000.caffemodel* is the model which has weights of the layers. The *cv2.dnn.readNet* function takes two parameters (“path/to/prototxtfile”, “path/to/caffemodelweights”). We receive the number of identified faces after using the face detection model, which is then provided as input to the face mask detection model. Faces can be detected using this face detection model in both static pictures and real-time video streams. Overall, this model is rapid and accurate and has minimal resource usage.

4.3.2 Application of face mask model

Now that the model is trained, it can be implemented for any image to detect the presence of mask. The given image is first fed to the face detection model to detect all faces within the image. Then, these faces are passed as an input to the CNN-based face mask detection model. The model would extract hidden patterns/features from the image and thus classify the images as either “Mask” or “No Mask”. Figure 5 describes the complete procedure.

Table 3 Training Model on Adam Optimizer

Epochs	loss	accuracy	val_loss	val_accuracy
10	0.2938	0.8612	0.2520	0.9050
20	0.2172	0.9125	0.2506	0.9100
30	0.1524	0.9375	0.1783	0.9300
40	0.1688	0.9350	0.2488	0.9300
50	0.1223	0.9600	0.2259	0.9450
60	0.1204	0.9575	0.3382	0.9150
70	0.0760	0.9800	0.2224	0.9400
80	0.0478	0.9825	0.2671	0.9450
90	0.0541	0.9825	0.2407	0.9500
100	0.0288	0.9937	0.2102	0.9500

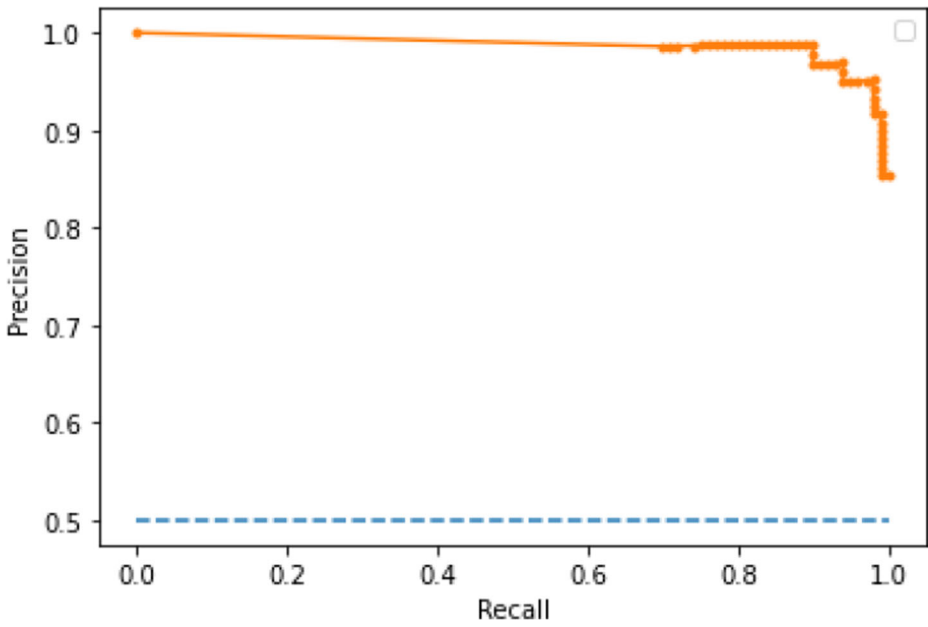


Fig. 3 Adam: Precision vs Recall

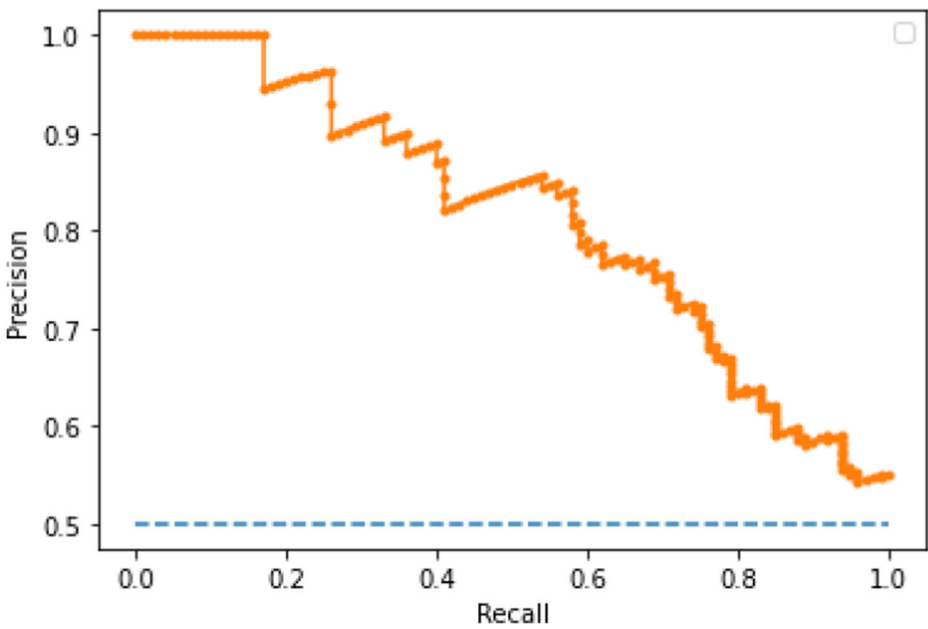


Fig. 4 SGD: Precision vs Recall

Table 4 Hyper Parameters used in Training

Parameter	Detail
Learning Rate	0.0005
Epochs	100
Batch Size	32
Optimizer	Adam
Loss Function	binary_crossentropy

4.3.3 Deployment

In the last step the CNN model is integrated into a web-based application that is hosted in order to be shared easily with other users and they can upload their image or live video feed to the model to recognize facial masks and then get the predicted result. Streamlit, an open-source python library, is used to design and construct a simple web app that allows users to submit a picture with a single click of a button and receive the outcome in a matter of seconds.

5 Results and analysis

The final results, as shown in Table 5, were achieved after multiple experiments using various hyper-parameter values such as learning-rate, epoch size, and batch size. Table 4 depicts the hyper-parameters used.

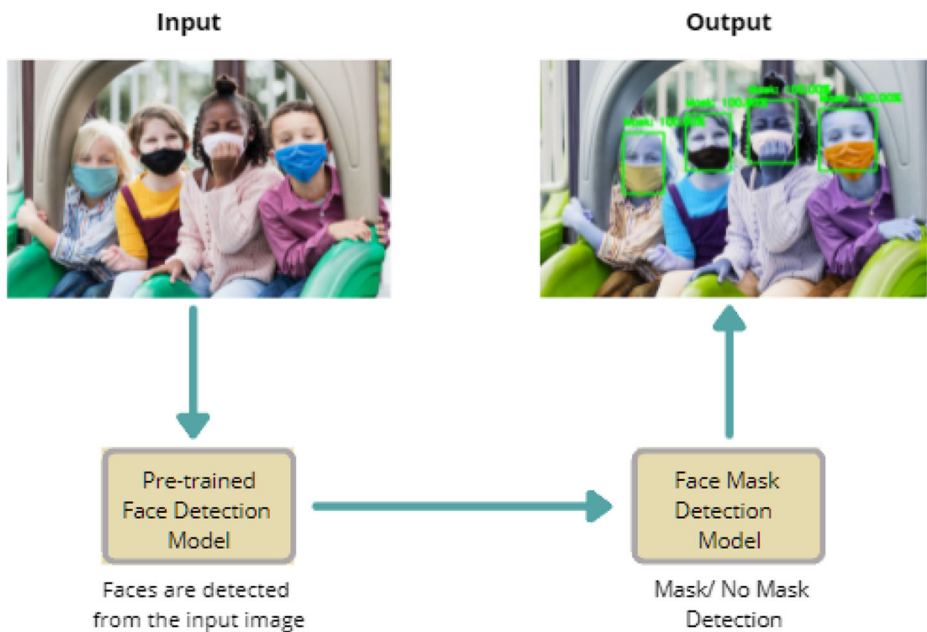
**Fig. 5** Diagram showing implemented Face Mask Model

Table 5 Classification Report

	Precision	Recall	f1-score	Support
dataset/with_mask	0.98	0.97	0.98	400
dataset/without_mask	0.97	0.98	0.98	400
accuracy			0.98	800
macro avg	0.98	0.98	0.98	800
weighted avg	0.98	0.98	0.98	800

The various metrics used to evaluate the model are accuracy, precision, recall, and f1 score.

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \quad (4)$$

$$Precision = \frac{Tp}{Tp + Fp} \quad (5)$$

$$Recall = \frac{Tp}{Tp + Fn} \quad (6)$$

$$F1 = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (7)$$

In the above, Tp represents: True positive, Tn : True negative, Fp : False positive, and Fn : False negative.

True positives are accurately predicted as being in a positive class, whereas false positives are images that were incorrectly predicted as being in a positive class. *True negatives* are accurately predicted to be in the negative class, whereas false negatives are incorrectly predicted to be in the negative class.

The accuracy of the masked individual identified by the developed model provides a good standard of prediction. In Fig. 6 above, which shows training accuracy, the model had struggled to acquire features until it reached 40 epochs, following which the curve remained steady. The accuracy was around 98% after 100 epochs. The green curve displays the training accuracy, while the blue line gives the validation dataset. Moreover, in Fig. 7, which shows the training and validation loss curve, the green line represents loss in the training dataset, which is smaller than 0.1, and the loss in the validation dataset is represented by the blue line, which is also less than 0.2 after 100 epochs.

– Model Testing

The model was tested on various diverse images, and some of them are exhibited below in the Fig. 10. The green rectangular box demonstrates a person correctly wearing the mask along with the accuracy score at the top, whereas the rectangular red box displays that the individual is without any mask. In summary, the model learns from the training dataset in order to label and then predict.

According to the Receiver Operating Characteristic Curve (ROC), a classification model can be evaluated by comparing its true positive rate with its false positive rate at various threshold values (see Fig. 8). The curve is at the top-left corner, which is considered a sign of excellent performance as smaller values on the x-axis mean lower false positives and higher true negatives. In contrast, more significant values on the y-axis mean lower false negatives and higher true positives.

– Comparison with other models

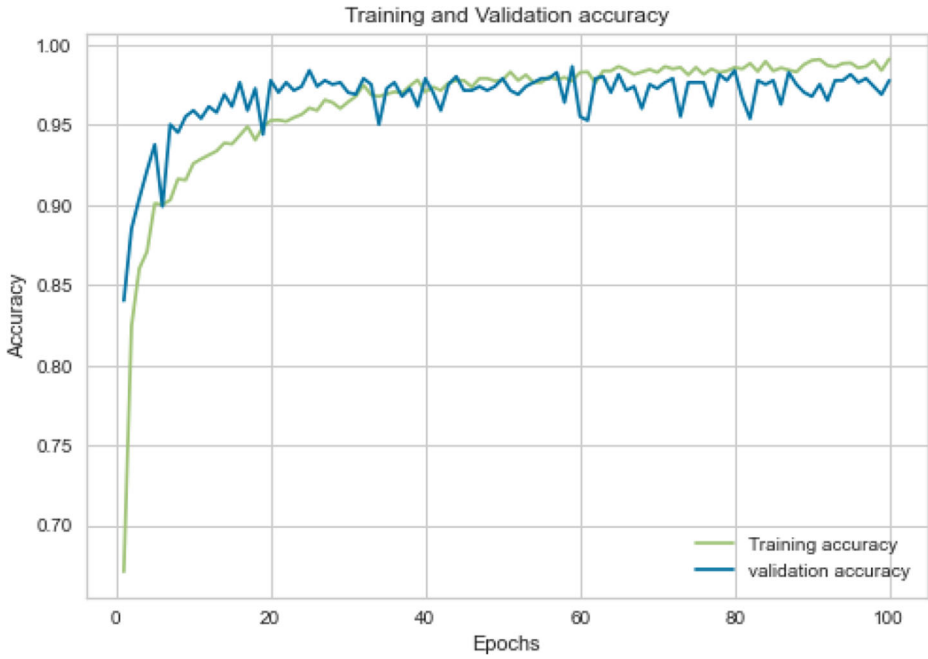


Fig. 6 Accuracy test results during Model Training

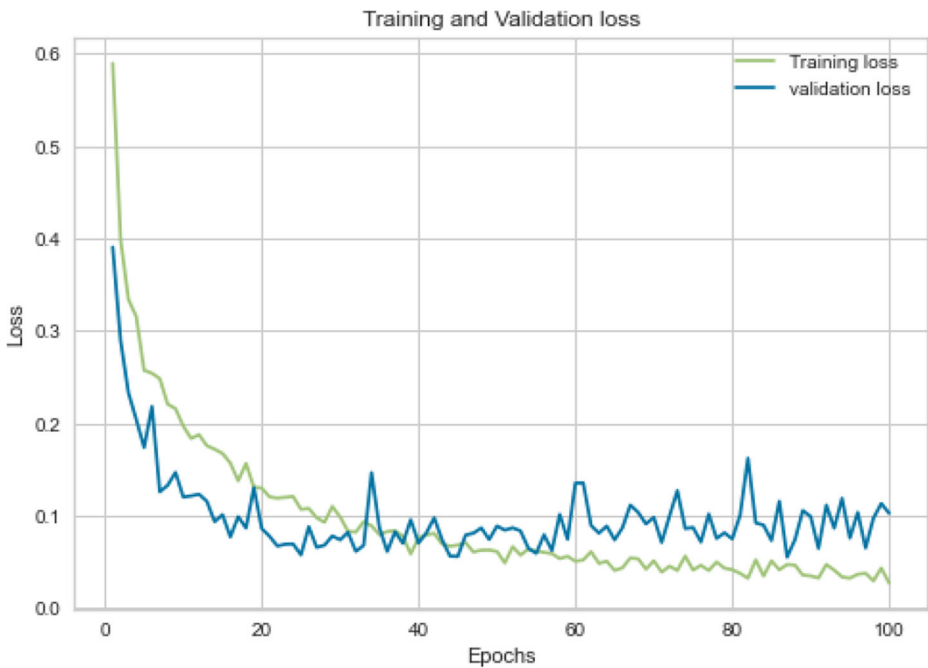


Fig. 7 Loss test results during Model Training

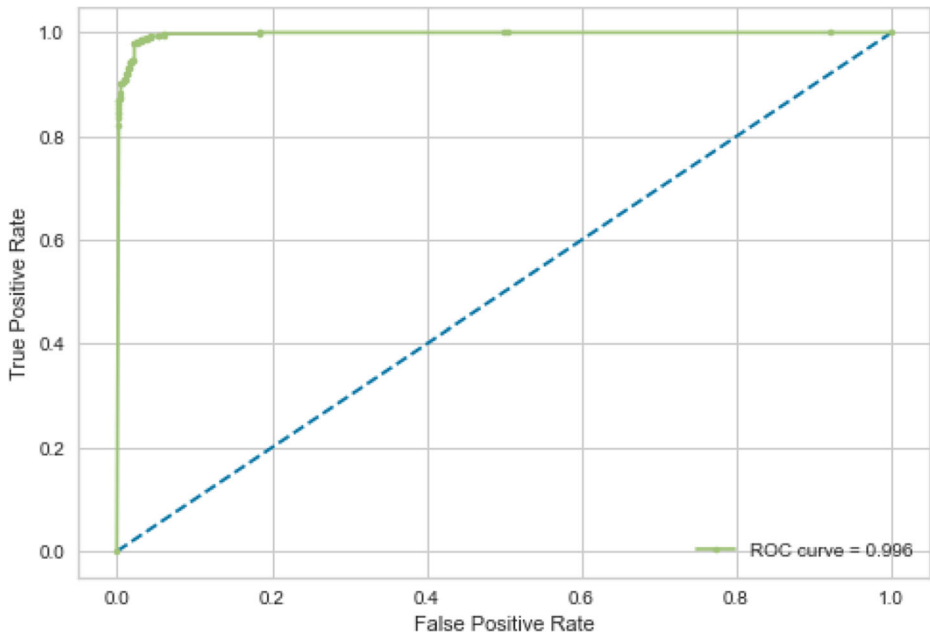


Fig. 8 ROC Curve Plot

Apart from the custom CNN architecture implemented in this research, there exists some other architectures, such as VGG -19 [8], MobilenetV2 [6], ResNet-50 [5], etc. This model was compared to several models by training them on the same dataset.

MobileNetV2 is a CNN model of 53 layers and 19 blocks [6].

DenseNet-121 is a modification of CNN. Each layer is connected to every other layer, hence the term DenseNet [3]. The combination of feature maps from preceding layers is the input of a layer [25].

Inception-v3 is a 48-layer pre-trained deep convolutional neural network trained on database with numerous pictures [13].

VGG-19 is a 19 layered pre-trained CNN model loaded with rich feature representations for a variety of pictures. The network accepts images with a resolution of 224 by 224 on its input [8].

Table 6 Comparison of different models

Model	Accuracy [%]	Time Per Epoch [s]	Model Size [MB]
MobilenetV2	97	10.16	11
DenseNet-121	98	10.94	96
Inception-V3	96	9.82	89
VGG-19	95	10.34	79
Proposed Method	98	8.95	33

The results shown in Table 6 are computed on machine equipped with NVIDIA Tesla T4 GPU (16 GB VRAM), 12 cores of AMD EPYC 7542 CPU (2.90 GHz) and 50 GB of RAM with a batch size of 32.

As can be seen in the table, different models were tested on the same data-set, giving an accuracy above 95%. The VGG-19 model had the lowest overall performance, down by 3% in accuracy from the proposed model. Nevertheless, these results show that the models show satisfactory performance to differentiate between a masked and a non-masked person. The top performer among these models was our proposed model, which shows a significant difference in performance with high accuracy, low time per epoch, and less model size compared to the other models.

On comparing several models with the proposed model based on accuracy, size, and training speed, we can find that DenseNet-121, for example, performs best but is among the slowest in terms of training time and also has the most significant memory footprint. On the other hand, MobileNet-V2 performs marginally more inferior and is substantially slower to train but has a smaller memory footprint.

On the contrary our proposed model, exhibits a slightly weaker recognition performance than Dense-Net-121, but is among the fastest regarding training speed and is considerably smaller than Dense-Net-121.

The graph in Fig. 9 compares these different models based on accuracy, time per epoch, and model size. As can be clearly seen, the proposed model had the least time per epoch, since it is located close to the x-axis with a reasonably good model size represented by the circle's diameter. The Dense-Net model had the highest time per epoch and model size, whereas MobileNetV2 was the smallest but took more time than inception-V3 and the proposed model. Hence, the proposed model is relatively faster than the other models shown in the graph.



Fig. 9 Different model comparison w.r.t accuracy, size and training speed

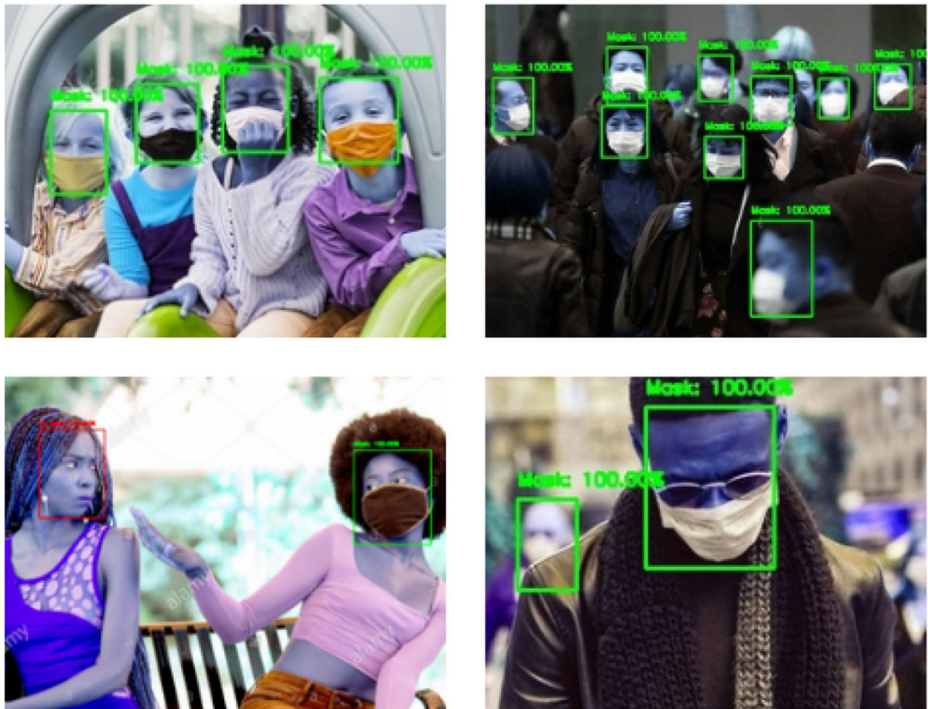


Fig. 10 Predictions on Test Images

6 Conclusion

This manuscript proposes a face mask recognition system for static images and real-time video that automatically identifies if a person is wearing a mask (see Fig. 10), which is an excellent solution for deterring the spread of the COVID-19 pandemic. By using Keras, OpenCV, and CNN, the proposed system is able to detect the presence or absence of a face mask and the model gives precise and quick results. The trained model yields an accuracy of around 98%. Trials were conducted to compare it with other pre-existing popular models which demonstrates that the proposed model performs better than DenseNet-121, MobileNet-V2, VGG-19, and Inception-V3 in terms of processing time and accuracy. This methodology is an excellent contender for a real-time monitoring system because of its precision and computing efficiency.

7 Future work

In the future, physical distance integration could be introduced as a feature, or coughing and sneezing detection could be added. Apart from detecting the face mask, it will also compute the distances among each individual and see any possibility of coughing or sneezing. If the mask is not worn properly, a third class can be introduced that labels the image as ‘improper mask’. In addition, researchers could propose a better optimiser, improved parameter configuration, and the use of adaptive models.

Author Contributions Hiten Goyal and Charanjeet Singh built the model and the computational framework and examined the data. Karanveer Sidana and Hiten Goyal carried out the implementation. Charanjeet Singh conducted the computations. Karanveer Sidana and Hiten Goyal wrote the paper with collaboration from all contributors. Abhilasha Jain and Swati conceived the study and was responsible for its direction and planning.

Funding This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Availability of data and materials Yes. <https://bit.ly/3sLfBX5>

Code Availability Yes <https://bit.ly/3yfdkFO>

Declarations

Conflict of Interests Hiten Goyal, Karanveer Sidana, Charanjeet Singh, Abhilasha Jain and Swati Jindal declare that they have no conflict of interest.

References

1. Chen D, Hua G, Wen F, Sun J (2016) Supervised transformer network for efficient face detection. In: European conference on computer vision. Springer, pp 122–138. https://doi.org/10.1007/978-3-319-46454-1_8
2. Chen Y, Menghan Hu, Hua C, Zhai G, Zhang J, Li Q, Yang SX (2021) Face mask assistant: detection of face mask service stage based on mobile phone. *IEEE Sensors J* 21(9):11084–11093. <https://doi.org/10.1109/JSEN.2021.3061178>
3. Cruz AP, Jaiswal J (2021) Text-to-image classification using atngan with densenet architecture. In: Proceedings of international conference on innovations in software architecture and computational systems. Springer, pp 1–17
4. Ejaz MS, Islam MR, Sifatullah M, Sarker A (2019) Implementation of principal component analysis on masked and non-masked face recognition. In: 2019 1St international conference on advances in science, engineering and robotics technology (ICASERT). IEEE, pp 1–5. <https://doi.org/10.1109/ICASERT.2019.8934543>
5. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778. <https://doi.org/10.1109/v>
6. Huang Y, Qiu C, Wang X, Wang S, Yuan K (2020) A compact convolutional neural network for surface defect inspection. *Sensors* 20(7):1974. <https://doi.org/10.3390/s20071974>
7. Hussain SA, Al Balushi ASA (2020) A real time face emotion classification and recognition using deep learning model. In: *Journal of physics: Conference series*, vol 1432. IOP Publishing, Bristol, p 012087
8. Karen S, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*
9. Khan N, Yaqoob I, Hashem IAT, Inayat Z, Ali WKM, Alam M, Shiraz M, Gani A (2014) Big data: survey, technologies, opportunities, and challenges. *Scientif World J*, 2014. <https://doi.org/10.1155/2014/712826>
10. Lawrence S, Lee Giles C, Tsoi AC, Back AD (1997) Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks* 8(1):98–113. <https://doi.org/10.1109/72.554195>
11. Li S, Ning X, Yu L, Zhang L, Dong X, Shi Y, He W (2020) Multi-angle head pose classification when wearing the mask for face recognition under the covid-19 coronavirus epidemic. In: 2020 International conference on high performance big data and intelligent systems (HPBD&IS). IEEE, pp 1–5. <https://doi.org/10.1109/HPBDIS49115.2020.9130585>
12. Li C, Wang R, Li J, Fei L (2020) Face detection based on yolov3. In: Recent trends in intelligent computing, communication and devices. Springer, pp 277–284. https://doi.org/10.1007/978-981-13-9406-5_34

13. Mednikov Y, Nehemia S, Zheng B, Benzaquen O, Lederman D (2018) Transfer representation learning using inception-v3 for the detection of masses in mammography. In: 2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE, pp 2587–2590. <https://doi.org/10.1109/EMBC.2018.8512750>
14. Militante SV, Dionisio NV (2020) Real-time facemask recognition with alarm system using deep learning. In: 2020 11th IEEE control and system graduate research colloquium (ICSGRC). IEEE, pp 106–110. <https://doi.org/10.1109/ICSGRC49013.2020.9232610>
15. Militante SV, Gerardo BD, Dionisio NV (2019) Plant leaf detection and disease recognition using deep learning. In: 2019 IEEE Eurasia conference on iot, communication and engineering (ECICE). IEEE, pp 579–582. <https://doi.org/10.1109/ECICE47484.2019.8942686>
16. Modes of transmission of virus causing covid-19. <https://bit.ly/3DAyqRu>. Accessed: 22-Nov-2021
17. Ning X, Duan P, Li W, Zhang S (2020) Real-time 3d face alignment using an encoder-decoder network with an efficient deconvolution layer. IEEE Signal Process Lett 27:1944–1948. <https://doi.org/10.1109/LSP.2020.3032277>
18. Nwankpa C, Ijomah W, Marshall S (2018) Stephen marshall. Activation functions: Comparison of trends in practice and research for deep learning. arXiv:1811.03378
19. Ochin S (2019) Deep challenges associated with deep learning. In: 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon). IEEE, pp 72–75. <https://doi.org/10.1109/COMITCon.2019.8862453>
20. Qin B, Li D (2020) Identifying facemask-wearing condition using image super-resolution with classification network to prevent covid-19. Sensors 20(18):5236. <https://doi.org/10.3390/s20185236>
21. Shashi Y (2020) Deep learning based safe social distancing and face mask detection in public areas for covid-19 safety guidelines adherence. Int J Res Appl Sci Eng Technol 8(7):1368–1375. <https://doi.org/10.22214/ijrasnet.2020.30560>
22. Ud Din N, Javed K, Bae S, Yi J (2020) A novel gan-based network for unmasking of masked face. IEEE Access 8:44276–44287. <https://doi.org/10.1109/ACCESS.2020.2977386>
23. Venkateswarlu IB, Kakarla J, Prakash S (2020) Face mask detection using mobilenet and global pooling block. In: 2020 IEEE 4th conference on information & communication technology (CICT). IEEE, pp 1–5. <https://doi.org/10.1109/CICT51604.2020.9312083>
24. Who coronavirus disease (covid-19) dashboard. <https://www.worldometers.info/coronavirus/>. Accessed: 22-Nov-2021
25. Yang G, Gewali UB, Ientilucci E, Gartley M, Monteiro ST (2018) Dual-channel densenet for hyperspectral image classification. In: IGARSS 2018–2018 IEEE International geoscience and remote sensing symposium. IEEE, pp 2595–2598

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.