Multimedia Tools and Applications (2023) 82:25131–25153
https://doi.org/10.1007/s11042-023-14408-y

Ensemble of deep transfer learning models for real-time automatic detection of face mask

Rubul Kumar Bania [1]

Received: 9 May 2022 / Revised: 12 November 2022 / Accepted: 21 January 2023 /
Published online: 1 February 2023

Abstract
The COVID-19 pandemic is causing a global health crisis. Public spaces need to be safeguarded from the adverse effects of this pandemic. Wearing a facemask has become an adequate protection solution many governments adopt. Manual real-time monitoring of face mask wearing for many people is becoming a difficult task. This paper applies three heterogeneous deep transfer learning models, viz., ResNet50, Inception-v3, and VGG-16, to prepare an ensemble classification model for detecting whether a person is wearing a mask. The ensemble classification model is underlined by the concept of the weighted average technique. The proposed framework is based on two phases. An off-line phase that aims to prepare a classification model by following training-testing steps to detect and locate facemasks. Then in the second online phase, it is deployed to detect real-time faces from live videos, which are captured by a web-camera. The prepared model is compared with several state-of-the-art models. The proposed model has achieved the highest classification accuracy of 99.97%, precision of 0.997, recall of 0.997, F1-score of 0.997 and kappa coefficient 0.994. The superiority of the model over state-of-the-art compared methods is well evident from the experimental results.

Keywords Face mask · Transfer learning · Ensemble · Covid-19

1 Introduction

Thousands of individuals worldwide die from the coronavirus 2019 (COVID-19) virus. COVID-19 is a respiratory disorder due to severe acute respiratory syndrome coronavirus-2 (SARS-CoV-2) virus [5]. The disease spreads when the COVID-19 infected person coughs, sneezes, or breathes the virus into the airspace. Also, an ordinary person can acquire it through salivation beads, respiratory droplets, or nasal droplets from infected individuals [5, 10].

✉ Rubul Kumar Bania
rubul.bania@gmail.com; rubul.bania@nehu.ac.in

[1] Department of Computer Application, North-Eastern Hill University, Tura Campus, Tura, Meghalaya 794002, India

Therefore, in the shadow of the COVID-19 pandemic, face mask-wearing has become mandatory in many public places. It has become a helpful solution that has proven to protect the places and reduce the spread of this pandemic [12, 24]. Though individual nations' governments enforce several rules to wear a face mask in public and workplaces, it is not easy to monitor such criteria in highly populated countries like India. Especially in hotspot areas, it is mandatory to wear masks by individuals. However, not every individual is aware or does not bother, thus risking their life and the lives of others by not wearing a mask [12].

Real-time monitoring of face mask wearing for government agencies becomes a difficult job. Manual tracking is generally hard to enforce because of the human resources needed to efficiently protect public spaces and ensure that individuals wear masks correctly. An effective and efficient computer vision strategy intends to develop a real-time application that publicly monitors individuals, whether they are wearing face masks. The first stage of identifying the existence of a face mask on the face is to detect the face, and then the face mask is caught. Thus, the whole process is divided into face detection and mask detection on the face.

In some previous studies, a facemask detection model is applied in the medical operating room. The model was based on the Viola-Jones face detection techniques [8]. This method used the color filter to differentiate between masked face and bare face by portioning the faces into two halves, i.e., upper, and lower half. But the feature extraction capability is not so effective as it has hand-coded features. Thus, the obtained accuracy is not so significant. Very recently, deep learning (DL) and computer vision have been used in numerous domains of engineering applications and solved several complex problems [2, 8, 12]. In the literature, DL architectures have shown an outstanding role in object detection. These architectures can be combined to detect the mask on a face. Researchers have designed several models/ architectures for facemask detection using lightweight classifiers of the Convolutional Neural Network (CNN) family, such as MobileNetV1 and MobileNetV2 [11, 18, 19, 22]. All these models are transferred learning-based techniques. The transfer learning technique encompasses using knowledge learned in some tasks and applying it to resolve the problem in the related target task [12].

In machine learning and deep learning research, ensemble learning is an influential and powerful tool [3, 10]. This technique combines the conclusion of multiple experts or classifiers based on the proverb "two brains are better than one." Usually, ensemble models for supervised learning have two essential steps; the first step involves selecting a set of different learning models known as the base models. The second step is a judicious combination of the results. Moreover, in the literature, two basic approaches are available, viz., homogeneous, and heterogeneous ensemble learning models [3].

1.1 Related work

In the literature, many of the developments in the direction of face-mask detection methods have been reported recently. This section presents the most relevant academic and industry related works for detecting the face mask. The implication of deep learning in computer vision has inspired the investigators to use it for face mask detection. Loey et al., [12] have prepared a facemask classification model by applying ResNet50, decision tree, support vector machine (SVM) and ensemble algorithm. ResNet50 is used extracting features from the image datasets. Finally, the features are passed through the machine learning models. SVM has achieved highest classification accuracy. The model was not applied on real time videos. In another recent study [13], authors have proposed a hybrid model based on ResNet50 and You Only

Look Once (YOLO) for detecting face mask in crowd. To train and validate the proposed detection model, a new dataset based on two public masked face datasets is considered. The experiments that are conducted on this new dataset have shown an average precision of 81%. In [24], the authors have proposed to use the Faster Region-Based Convolutional Neural Network (R-CNN) algorithm to detect masks and monitor social distancing. The proposed training model for mask detection is based on Single-Shot Multibox Detector (SSD), and You Only Look Once (YOLO) version 2. The testing of this model is performed on complex images, including face turning, wearing classes, beard faces, and scarf images. The testing accuracy for this model attains 93.4%. Militante et al. [14] have applied the Principal Component Analysis (PCA) model to recognize the masked and unmasked faces. Face without a mask gives a better recognition rate in PCA. Also, the authors have mentioned that extracting features from a masked face is less than an unmasked face. They found that the accuracy of mask face classification using the PCA relates to wearing masks. When wearing a mask, the accuracy decreases to 70%. Some other recent studies are summarized in Table 1.

1.2 Motivation and objectives

In existing research, the most frequently used methods for face mask detection are single CNN based models or single transfer learning of pre-trained deep learning model. All the models are unable to attain better classification and recognition performance. Moreover, in single transfer learning, overfitting and negative transfer are the most alarming limitations. So, to address these limitations for face mask detection, one novel system is urgently required.

Therefore, in this paper, by leveraging the benefits of the pre-trained deep transfer learning models [23] and heterogeneous ensemble learning techniques, one computational model is

Table 1 Summarized information of the state-of-the-art work

Contributors	Purpose	Techniques	Results
Asghar et al. [2]	Face mask detection.	Depth-wise separable convolution layers based on MobileNet is applied instead of 2D convolution layers. It is applied on AIZOO face mask dataset.	Acc. = 93.14, Pre. = 0.92, Recall=0.92, F-score= 0.92
Teboulbi et al. [21]	Face mask detection.	Pretrained models such as the MobileNet, ResNet Classifier, and VGG are used on Kaggle face mask dataset.	Acc. = 97.14, Pre. = 0.982, Recall=0.973, F-score= 0.96
Goyal et al. [7]	Face mask detection.	Convolution Neural Network layers (CNN) are used as its backbone architecture to create different layers. Kaggle face mask dataset is used.	Acc. = 98.0, Pre. = 0.98., Recall=0.97, F-score= 0.98
Kumar et al. [11]	Face mask detection.	Raspberry Pi circuit with CNN model are used on Kaggle face mask dataset is used.	Acc. = 97.01, Pre. = 0.98, Recall=0.98
Sethi et al. [15]	Face mask detection.	Three popular baseline models viz., ResNet50, AlexNet and MobileNet are on Kaggle face mask dataset	Acc. = 98.20, Pre. = 0.98, Recall=0.98
Das et al. [6]	Face mask detection.	Prepared a CNN model with 200 filters and reshaping of the input images are done.	Acc. = 95.77
Taneja et al. [20]	Face mask detection.	Lightweight MobileNetV2 architecture is used on Kaggle face mask dataset.	Acc. = 99.02
Hussain et al. [9]	Face mask detection.	Apply deep convolution neural network (DCNN) and MobileNetV2-based transfer learning models.	Acc. =98.00, Pre. = 0.98, Recall=0.98

proposed with an objective to detect whether an individual is wearing face mask or not. After preparing the model the performance and evaluations of the model shall be tested on real time videos or scenes. By considering these objectives this research work has the following importance.

i) The proposed model may assist the authorities or administrators to monitor individuals in office premises, commercial spaces, public spaces, or even in hotspot areas.
ii) The novelty and importance of this proposed model over other existing works is that it can efficiently and accurately detect face masks in real-time videos.
iii) To the best of the author's knowledge, this is one of the first work that ensembled deep transfer learning models for detecting face masks in real-time videos. So, this study may lead to a new direction of research in deep learning fields.

1.3 Contributions

The major contributions of the proposed work are given below:

i) Proposing a deep transfer learning-based weighted average ensemble model for face mask detection by combining the outputs of ResNet50, Inception-v3, and VGG-16 models.
ii) Proposed model is tested on real-time videos which are captured through web-camera.
iii) Proposed model is compared with several existing models.

The organization for the rest of the paper is as follows. Section 2 reviews previous background study related to face mask detection; Section 3 illustrates the proposed approach for real-time face-mask detection. Section 4 reports the experimental results and analyses, and finally Section 5 presents the conclusions and future work of this research work.

2 Background study

This section explains some fundamental concepts that are required to understand and implement the work thoroughly.

2.1 Convolution neural networks

Convolution Neural Networks (CNN) have made a significant and pre-dominant revolution in the study of image processing and computer vision. CNN is one type of deep learning architecture. The word "deep" in "deep learning" refers to the number of layers via which the data is transformed [23]. Due to the availability of high-performing computing machines, image recognition, classification, and object detection can undoubtedly be solved by using CNN. One typical CNN architecture consists of an input, hidden, and output layer. The hidden layers include layers that perform convolutions [1]. So, CNN is a specialized artificial neural network that uses a mathematical operation called convolution in place of general matrix multiplication. Extraction of informative features from an image is collected through a series of convolution layers, non-linear activation functions, pooling (down-sampling), and fully connected dense layers [8. 18]. The feature extraction procedure takes place in both convolutional

and pooling layers. After that, the classification process happens in the fully connected layer. The details of each layer are inspected successively in the following.

2.1.1 Convolutional layer

This is the base layer of CNN. It is responsible for determining the features of the image or pattern. In this layer, the input image is passed through a filter [1]. The values resulting from filtering consist of the feature map. This layer applies some kernels that slide through the image matrix to extract low- and high-level features. Thus, after passing through a convolutional layer, the image becomes abstracted to a feature map, also called an activation map. Let us consider an image matrix of size 5 × 5, image pixel values are 0, 1, and filter (kernel) matrix of size 3 × 3 with stride 1 as shown in Fig. 1. Stride here means the number of pixels shift over the image matrix. If the stride is 1 then we move the filter to 1 pixel at a time, and if the stride is 2 then we move the filter to 2 pixels at a time.

If a filter does not fit the input image we need to use padding by adding zeros in the image matrix or by removing the part of the image which is not fitting. The output of the convolution layer then passes through a non-linear activation function called as rectified linear unit (ReLU) which computes the function, $f(x) = \max(0,x)$.

2.1.2 Pooling layer

The pooling layer is the second layer after the convolutional layer [1]. A pooling layer is usually applied to the created feature maps to reduce the number of feature maps and network parameters by applying corresponding mathematical computation. In this study, we used max-pooling and global average pooling. The max-pooling process selects only the maximum value using the matrix size specified in each feature map, resulting in reduced output neurons. A global average pooling layer is only used before the fully connected layer, reducing data to a single dimension. After the global average pooling layer, it is connected to the fully connected layer. The other intermediate layer used is the dropout layer. The primary purpose of this layer is to prevent network overfitting and divergence.

2.1.3 Fully connected layer

The fully connected layer is CNN's last and most crucial layer [1]. This layer functions like a multilayer perceptron. The output of the previous convolution and pooling layer is flattened

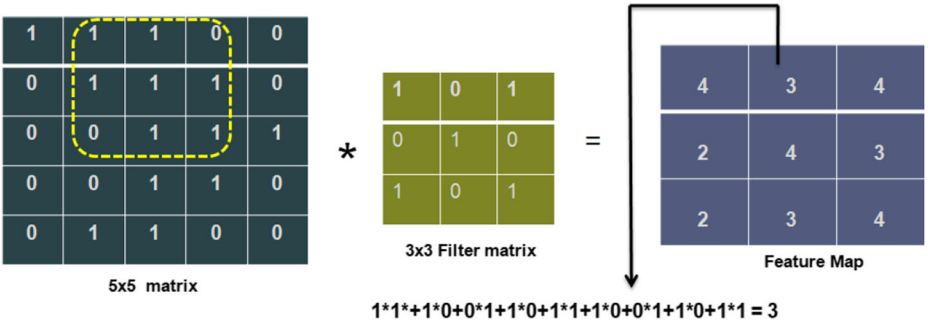

Fig. 1 Convolution operation

into a vector (1D array) and passes through the fully connected layer. Rectified linear unit (ReLU) activation function is also commonly used on the fully connected layer. In contrast, Softmax activation function is used to predict output images in the last layer of the fully connected layer [1, 18]. Activation functions are the nodes placed at the end or among neuronal networks (layers). They decide whether the neuron fires. Choice of activation function at hidden and output layers is essential 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.

2.2 Deep transfer learning models

In the deep learning domain, transfer learning has become more prevalent [12, 23]. The base intuition behind transfer learning is to take a model previously trained on a large dataset and transfer its knowledge. Training a deep CNN architecture with millions of parameters from scratch is very time consuming and requires equipment with high performance. Parameters and weights of models trained on different datasets are transferred to the new model. Usually, this approach is very significant and helpful when we have a relatively small dataset for training or have minimal computational power. This technique is highly used in computer vision and Natural Language Processing tasks. In addition, this method allows for obtaining results faster with lower calculation costs.

This research study uses three pre-trained transfer learning models under deep CNN architecture, ResNet50, Inception-v3, and VGG16 [17, 23, 26]. All these models were trained on the ImageNet dataset, which are imported and used for extracting essential image features from the face and non-face mask images. All the transfer learning models consist of two parts: the feature extraction part with CNN, and another is the classification part with fully-connected and Softmax layers.

2.2.1 ResNet model

ResNet (residual neural) was proposed in 2015 by researchers at Microsoft Research [13, 25]. The architecture of the model is shown in Fig. 2. ResNet has many variants that run on the concept of CNN but have different numbers of layers. Resnet50 is used to denote the variant that can work with 50 neural network layers. There is a common problem in deep learning associated with that called vanishing/exploding gradient. This causes the gradient to become 0 or too large. Thus, when we increase the number of layers, the training and test error rate also increases. After analyzing more on error rate, the authors were able to reach conclusion that it is caused by vanishing/exploding gradient. In order to solve the problem of the vanishing/exploding gradient Residual network is used. In this network skip connections are used. This skip connection basically skips training from a few layers and connects directly to the output. The approach behind this network is instead of layers learn the underlying mapping; it allows the network to fit the residual mapping. So, instead of say $H(x)$, initial mapping, the network will fit, $F(x) := H(x) - x$ which gives $H(x) := F(x) + x$.

2.2.2 Inception-v3 model

Inception-v3 is a pre-trained CNN model which has 48 layers. The network has an image input size of 299-by-299. The pictorial architecture is shown in Fig. 3. The model extracts general

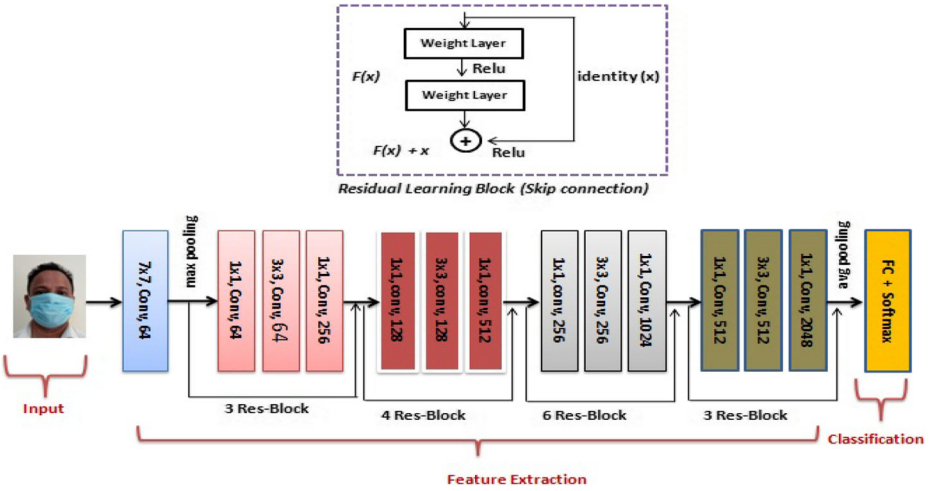

Fig. 2 Architecture of ResNet-50 model

features from input images in the first part and classifies them based on those features in the second part. This model comprises over 20 million parameters, and has been trained by one of the industry's top hardware experts. The model itself comprises symmetrical and asymmetrical building blocks, where each block consists of various convolutional, average, and max pooling, concatenate, dropouts, and fully connected layers. In addition, batch normalization is commonly used and applied to the activation layer input into this model. Then with fully connected layers softmax activation function is used to classify the images.

2.2.3 VGG-16 model

VGG-16 is a variant of the VGG model, consisting of 16 layers (13 convolution layers, three fully connected layers, five MaxPool layers, and one SoftMax layer) [4, 18, 26]. It is a CNN model that is trained on more than a million images from the ImageNet database. It was proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab of Oxford University in 2014. The model was prepared to classify images into 1000 numbers of

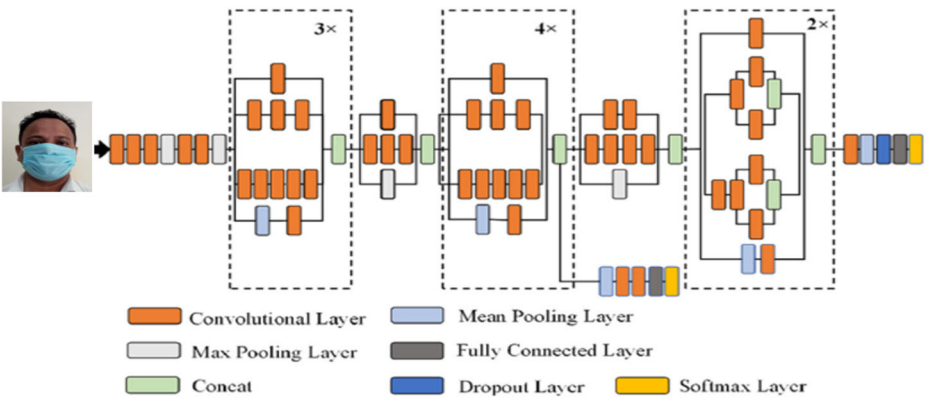

Fig. 3 Architecture of Inception-v3 model

object categories. Like the earlier pre-trained model, the input to the network of VGG-16 is also an RGB image of dimensions *(224, 224, 3)*. The image is passed through a stack of convolution layers, which uses the kernels of (3 × 3) size with a stride size of 1 pixel. It enabled them to cover the whole notion of the image. Spatial padding of 1-pixel for 3 × 3 conv. Layers were used to preserve the spatial resolution of the image. One of the configurations also utilizes 1 × 1 convolution filters, which can be considered a linear transformation of the input channels (followed by non-linearity). Spatial pooling is carried out by five max-pooling layers, which follow some of the convolution 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 convolution layers (which has a different depth in different architectures): the first two have 4096 channels each, and the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the softmax layer. The overall pictorial architecture is shown in Fig. 4.

3 Proposed methodology

The pipeline of the devised framework followed in this research is shown as a block diagram in Fig. 5. It is mainly based on two phases: offline and online processing. The offline processing aims to create a deep transfer learning-based ensemble model that can detect whether an individual is wearing a facemask. The model must learn from a well-defined dataset to train and predict whether a person has put on a mask. Thus, in this phase, data preprocessing of the images is required. Then in the online phase, it is aimed to deploy or apply the deep learning model in real-time video captured by the web camera to detect masks in live stream video.

3.1 Data preprocessing

The accuracy of a model is dependent on the quality of the dataset. The initial data cleaning is performed to eliminate the faulty pictures discovered in the dataset. The images are resized into a fixed size of (224 × 224 × 3), which helps to reduce the load on the machine while training and provides better results. The images are properly labeled with a one-hot encoding system. The array of images is then transformed into a NumPy array for quicker computation. The data

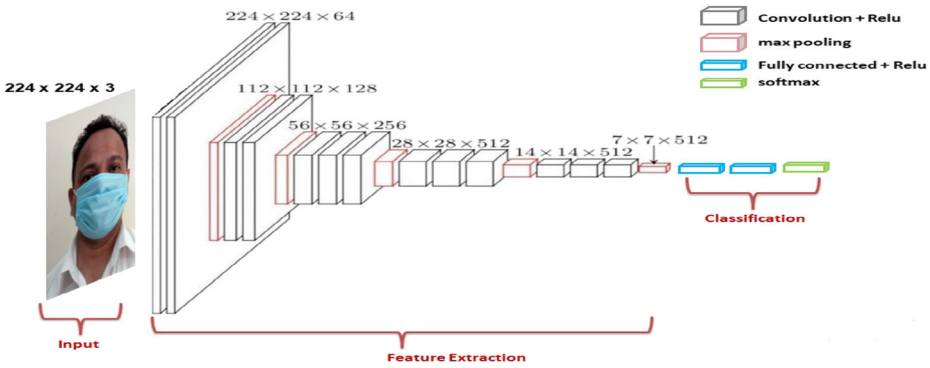

Fig. 4 Architecture of VGG-16 model

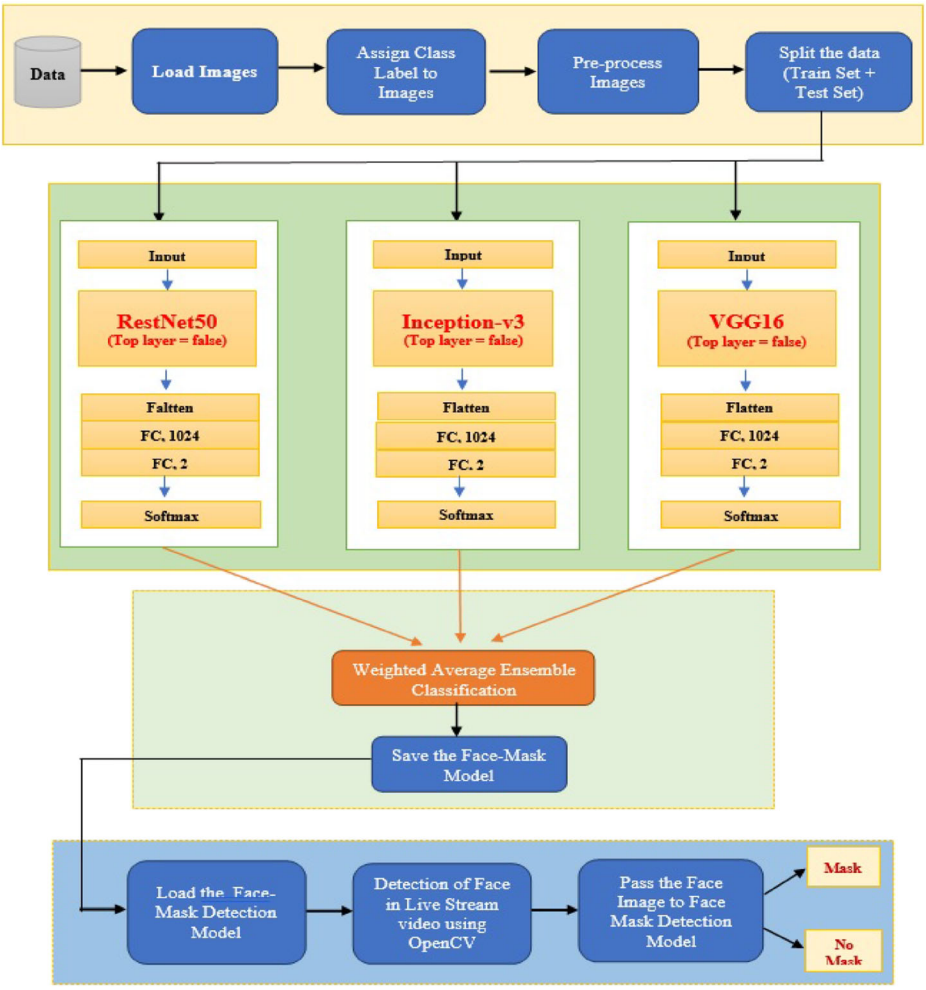

Fig. 5 Framework of the proposed methodology

augmentation technique is utilized to increase the sample size of training datasets and improve their quality. *ImageDataGenerator* is used with appropriate rotation, zoom, and horizontal or vertical flip values to generate numerous versions of the same picture [6, 16]. It is worth mentioning here that the elimination of faulty pictures and applying the data augmentation technique is applied on Dataset-I only. In the training set data, before applying the data preprocessing steps, there are 3064 images. But after the removal of some faulty pictures, it was 2883, and after selectively using the data augmentation on several images, the number of images is 3556. As the Dataset-II size is large so data augmentation is not applied. Details of the datasets are given in Section 5.2.

3.2 Proposed deep transfer learning-based ensemble model

After performing the data pre-processing part on the image data, training set images are applied to the three heterogeneous pre-trained transfer learning models viz., RestNet50,

Inception-v3 and VGG16. All the models were imported with pre-trained weights matrices, which were trained on the ImageNet Dataset. The main reason of using these models was to extract the essential and valuable features from the image data. It is also worth mentioning that as face-mask image data availability is less, it is challenging to train each of these models from scratch. Hence, to subjugate this issue, this study decided to use the transfer learning approach and then fine-tune the pre-trained deep CNN models on the collected face-mask dataset. By observing the exceptionally sound performance for the classification task in this work, the author is motivated to use these three pre-trained models. Also, all the models complex architecture deals wisely with the vanishing gradient descent problem and minimizes noise and variance problems. The other motivation to use the Ensemble technique for the output produced by the three models to allow better predictive performance and robustness in comparison to a single CNN classification model. All the deep transfer learning CNNs were separately trained on the collected dataset to perform the classification task. The input given to them is a (224,224,3) dimensional image. Figure 6 illustrates the detailed pictorial representation of the proposed ensemble model.

Applying the pre-trained weights of the transfer learning CNN models, features from the images are extracted. Features vectors are flattened into a single vector in the Flatten layer to pass it to the fully connected network for classification. The fully connected layer of each model has two dense layers with Relu as an activation function with a dropout threshold of 0.5, and a Softmax activation function follows it. The classification output produced by each classifier is ensembled with the help of the weighted averaging technique. In the literature, other approaches are available for designing ensemble models. They are namely, max and majority voting. By observing several well-known previous studies, the weighted averaging technique is being adapted in this research.

A weighted ensemble is an extension of a model averaging ensemble where the model's performance weights each member's contribution to the final prediction. The weighted average method assigns weights to each model, defining the importance of each model in the projection. It allows the weights to indicate each model's percentage of trust or expected

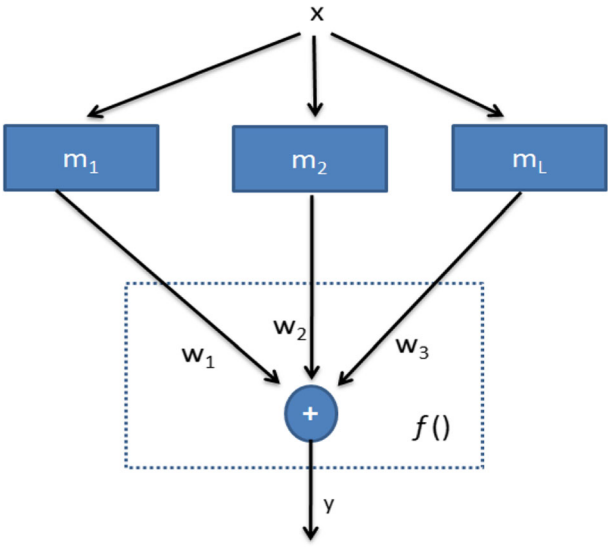

Fig. 6 Sample of weighted average model

performance. Figure 6 shows a block diagram of the weighted average ensemble process. The prediction outcome of the weighted average ensemble model for the sample image x_i with respect to L number of base models is shown in Eq. (1). The model weights are small positive values, and the sum of all weights equals one.

$$y_i = \frac{1}{L} \sum_{j=1}^{L} w_i m_{ji} \; ; \; \sum_{j=1}^{L} w_i = 1 \tag{1}$$

Thus, in the novel ensemble model approach, 0.4, 0.3, and 0.3 weight values (for Dataset-I) are assigned for the ResNet50, Inception-v3, and VGG-16 models. These values are achieved by applying the Grid search approach [4]. These weights gave a minor error for the ensemble model in training data, and that is why they are considered as the final weights in the model for testing. The grid search method performs an exhaustive searching operation through a subset of parameter space of the algorithm, followed by a performance metric. In this work, three loops are executed in the range of 0 to 5 to configure the weights for the three models. Then assigned weights are divided by 10, which is followed by a *tensordot* operation. Finally, out of several weights received in each run, those weights are considered, producing maximum classification accuracy.

In the second phase i.e., the online phase of the methodology, first task is to detect the face in the live-stream videos. For which OpenCV framework is used. Deep learning framework named as Caffe, which is created and maintained by Berkeley AI Research (BAIR) is applied in this research for faster and effective object detection techniques [7]. 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 300 × 300 ssditer 140,000.caffemodel* is the model which has weights of the layers. The *cv2.dnn.readNet* function takes two parameters ("path/to/prototxtfile", "path/to/caffemodelweights"). After identifying the faces, it is provided as an input to the face mask detection model. It will detect whether an individual is wearing a mask or not. So, the faces on individuals can be detected using this face detection model in both static pictures and real-time video streams.

4 Experimental results and discussions

After discussing the methodology, this section will describe the experimental details and findings of this study.

4.1 Experimental setup and evaluation measures

To get uniform experimental results, all the methods are implemented in Python. Programs are simulated in a machine with Processor: Xenon(R) CPU- E5–1630, 3.70 GHz clock speed and random-access memory of 32 GB having Windows 10 environment. The detailed experimental setup are as follows:

(i) All the methods and functions are implemented in Jupyter with Python 3.9 environment. Data structures like data frames, dynamic lists, collections, and dynamic arrays are used.
(ii) The deep learning models are implemented using Tensorflow, Keras functional API, which provides a flexible way to design neural networks with non-linear topology and shared layers.

(iii) The partitioned of the individual datasets are performed according to a train-test (80% and 20%) spilt scheme.

Four different classification validity measures such as (i) average accuracy, (ii) average precision, (iii) average recall and (iv) F-score measure, and (v) kappa-coefficient are used.

(i) **Accuracy (Acc)**: Overall effectiveness of the classifier.

$$\frac{TP + TN}{TP + FN + FP + TN}$$

(ii) **Precision (Pre)**: It calculates the number of positive class predictions which actually belongs to the positive class.

$$\frac{TP}{TP + FP}$$

(iii) **Recall (Rec):** The ratio of correctly predicted positive observations to the all observations in actual class as positive.

$$\frac{TP}{TP + FN}$$

(iv) **F-score:** Precision and Recall calculations can be combined to calculate the F-score or F-measure. Harmonic mean of the two fractions is computed.

$$\frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

(v) **Kappa-coefficient:** Kappa coefficient is the measure of agreement between predicted and true values in testing datasets. The value of kappa can be 0 to 1.

$$\frac{Tota\ Accuracy - Random\ accuracy}{(1 - random\ accuracy)}$$

Random accuracy for binary classification can be calculated as:

$$Random\ accuracy = \frac{1}{Total\ no.of\ \ classes}$$

4.2 Dataset in used

The datasets used in this research (with/without mask dataset) are collected from www. kaggle.com/dataset. Both datasets have different images, and the number of pictures differs. The first

dataset (Dataset-I) has a total of 3832 RGB images. Further, this is divided into two classes: images with a mask are 1914 and images without a mask is 1918. Dataset-I consists of both PNG and JPG formats images. Similarly, in the second dataset (Dataset-II) (https://www.kaggle.com/datasets/omkargurav/face-mask-dataset) there is a total of 7553 RGB images. Further, this is divided into two classes: images with a mask are 3725, and images without a mask are 3828. All the images in Dataset-II are of type JPG.

4.3 Results discussions

Extensive experiments are carried out to detect and classify the faces with the mask or no mask classes. To visualize the internal views of the deep CNN models' sample of the Heatmap representation for two different images on the Inception-v3 model are shown in Fig. 7a and b.

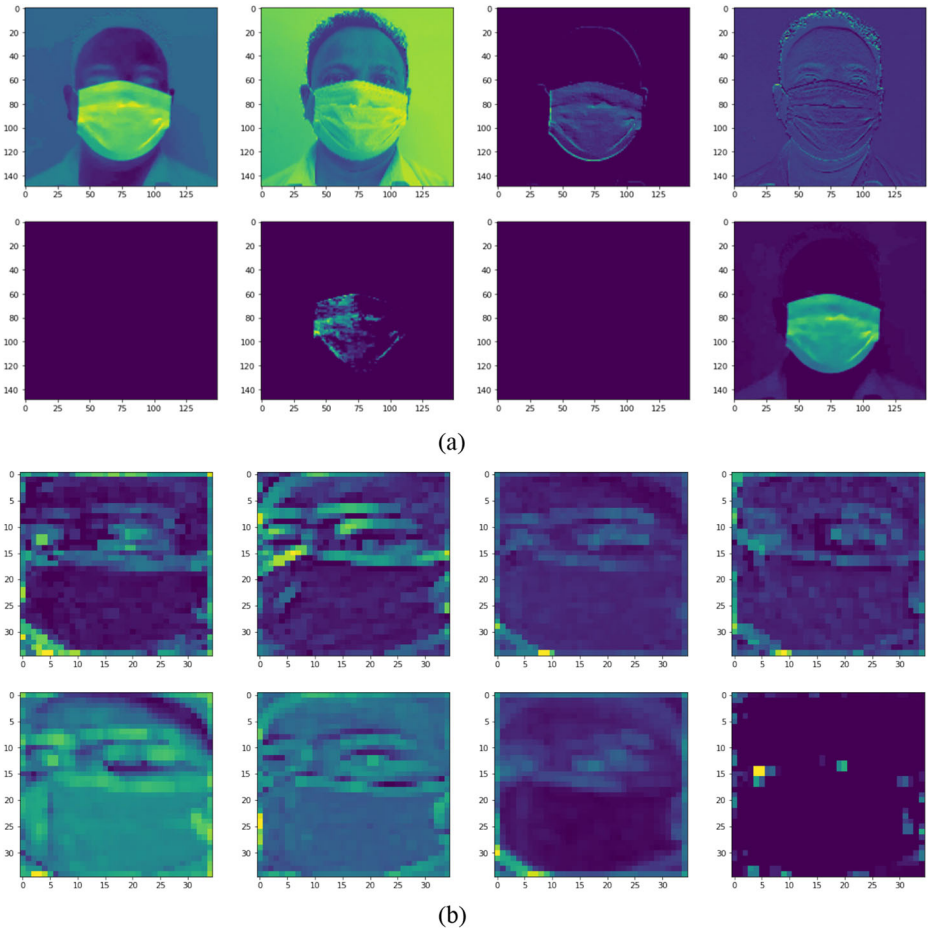

Fig. 7 Sample of Heatmap visualization of Inception-v3 model after 2nd layer and15th layer on two different images

Table 2 Summary of experimental results achieved by various models on Dataset-I

Methods	Precision	Recall	F-score	Kappa	Accuracy
Asghar et al. [2]	0.920	0.920	0.920	0.862	93.14%
Teboulbi et al. [21]	0.982	0.973	0.960	0.942	97.14%
Goyal et al. [7]	0.980	0.970	0.980	0.960	98%
Kumar et al. [11]	0.980	0.970	–	0.940	97%
Das et al. [6]	0.923	0.917	0.917	0.901	95.05%
Hussain et al. [9]	0.977	0.975	0.975	0.972	98.67%
VGG-16	0.961	0.960	0.960	0.954	94.22%
ResNet50	0.988	0.988	0.988	0.976	98.88%
Inception-v3	0.988	0.988	0.988	0.976	98.88%
Proposed Model	**0.995**	**0.992**	**0.993**	**0.986**	**99.30%**

The summary of the experimental results (on Dataset-I and Dataset-II) achieved by the different models in terms of four validity measures: percentage accuracy, precision, recall, F1-score, and kappa coefficient are reported in Tables 2 and 3. The bold fonts in the tables represent the best results. It can be observed in the tables that all the models have given an accuracy above 93%. The model developed by Asghar et al. [2] has the lowest performance in accuracy in both datasets. Also, it is noticeable that base model VGG-16 has the most deficient performance compared to ResNet50 and Inception-v3. Nevertheless, these results show that all the models perform satisfactorily in differentiating between a masked and a non-masked person. However, the top performer among these models is the proposed ensemble model, which shows a significant difference in performance with high accuracy. The sample of prediction results (with class label 1 or 0) for with mask and without mask images can be seen in Fig. 8.

During the training, one of the most used plots to debug a network is a cross-entropy (loss) curve. It provides a snapshot of the training process and the direction in which the network learns. Another most used curve to understand the progress of networks is an accuracy curve. It plots both training and validation accuracy. Thus, the learning graphs/curves based on the

Table 3 Summary of experimental results achieved by various models on Dataset-II

Methods	Precision	Recall	F-score	Kappa	Accuracy
Asghar et al. [2]	0.934	0.936	0.936	0.876	93.82%
Teboulbi et al. [21]	0.977	0.977	0.976	0.950	97.57%
Goyal et al. [7]	0.964	0.959	0.964	0.936	96.85%
Kumar et al. [11]	0.976	0.978	0.977	0.972	98.06%
Das et al. [6]	0.903	0.902	0.903	0.862	93.18%
Hussain et al. [9]	0.955	0.954	0.955	0.937	96.89%
VGG-16	0.912	0.933	0.933	0.906	95.37%
ResNet50	0.969	0.971	0.971	0.951	97.53%
Inception-v3	0.977	0.975	0.988	0.971	98.56%
Proposed Model	**0.997**	**0.997**	**0.997**	**0.994**	**99.97%**

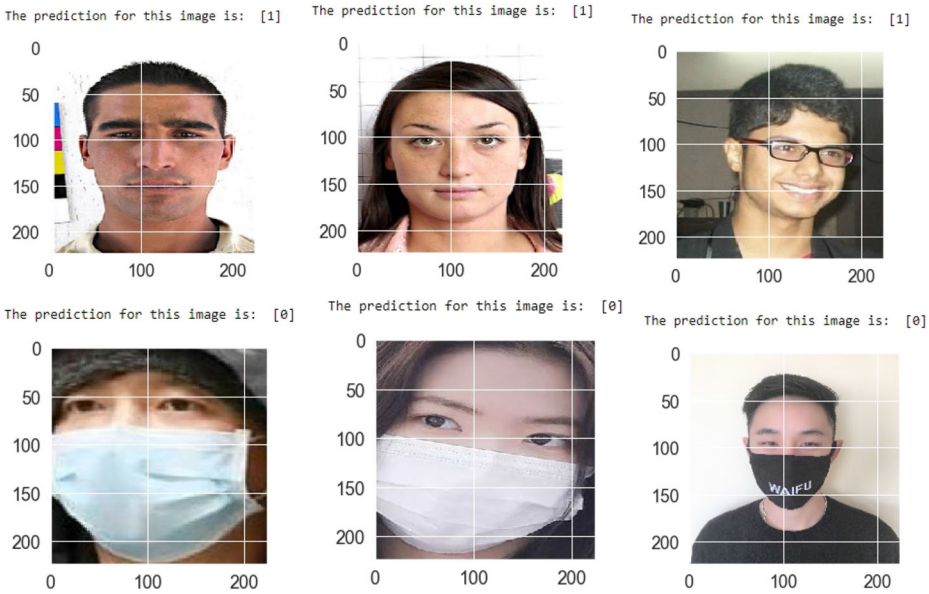

Fig. 8 Sample of Test images with their predicted numeric class labels without mask (1) and with mask (0)

history of the base models viz., Inception-v3, ResNet50, and VGG16 are plotted in Figs. 9 and 10. In Figs. 9a–e and 10a–e, the x-axis represents the number of epochs till which the model has stopped training after observing no further improvement while training the model. The y-axis represents the model's loss after the model's training.

Similarly, in Figs. 9b–f and 10b–f, the x-axis represents the number of epochs until the model has stopped training after observing no further improvement while training model. The y-axis represents the model's accuracy after the model's training. So, from those figures, it is observed that all the models provide better scores for the training and validation accuracy and training and validation loss. Thus, it confirms that the ensemble of these three models may provide good classification results.

Figure 11 depicts the confusion matrix drawn for Dataset-I by the proposed model. Among the 768 testing image samples (20% of dataset), 5 images were misclassified by the proposed ensemble architecture. In Fig. 12 ROC-AUC curve is also shown. Thus, ROC curve is plotted with TPR (True-Positive rate) against the FPR (False-Positive rate). TPR is on the y-axis, and FPR is on the x-axis. The AUC value is 0.99, which is more significant. Similarly, Figs. 13 and 14 represent the confusion matrix and ROC-AUC curve for the Dataset-II generated by the proposed model.

To validate the perfomance of the model, it is deployed to detect real-time faces from live streaming videos using the web camera. Some of the snapshots which were taken during the testing phase are shown in Figs. 15 and 16. It is noticeable from the figures that model has correctly detected the faces with mask and without mask. Red bounding boxes reflects the faces without face mask and by green bounding box with mask faces are shown. In crowded

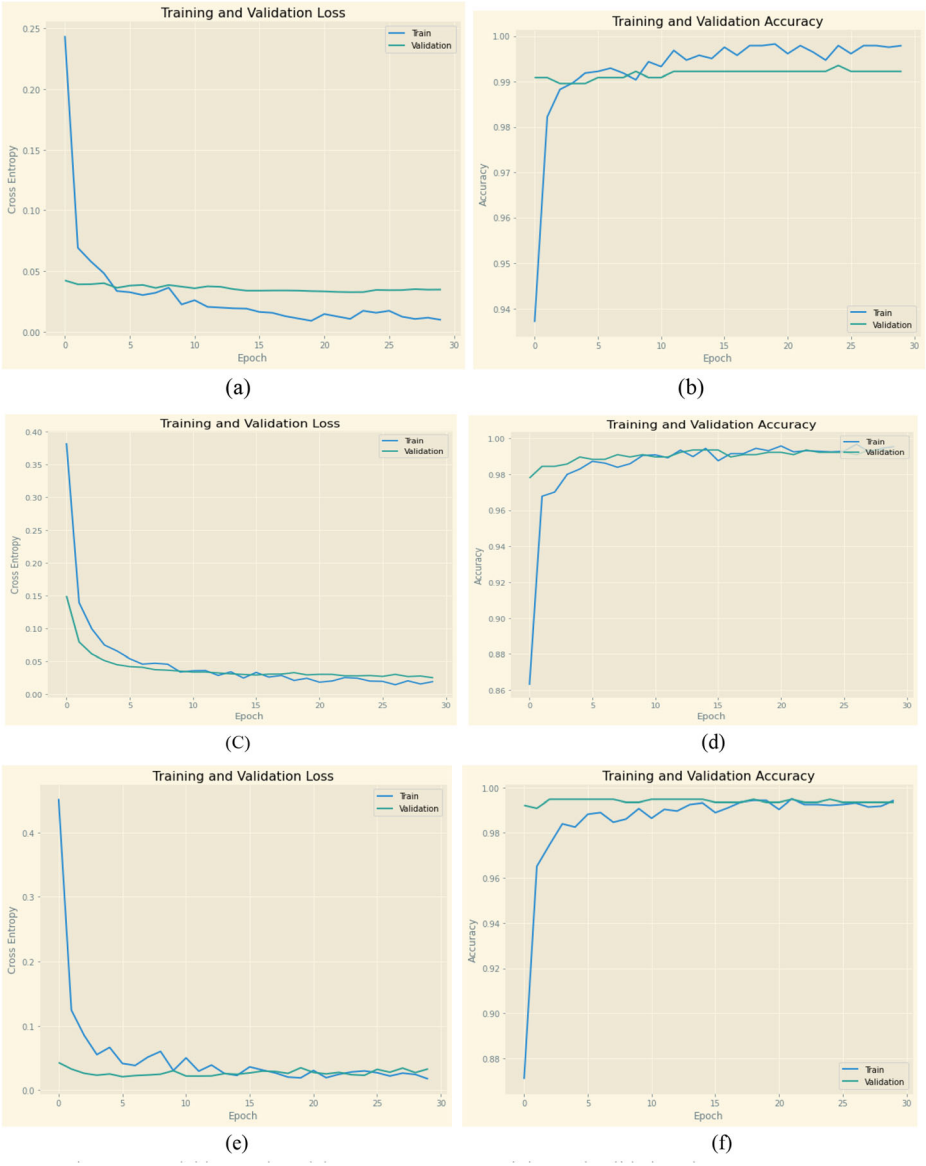

Fig. 9 Model loss and model accuracy curve on training and validation phases on Dataset-I

situation and when a person is in motion, the model can correctly detect the faces with mask and without mask and it is shown in Fig. 16. It is worth to mention here that the hardware configuration of the web camera in used for this study is low therefore the quality of the images received are bit low quality.

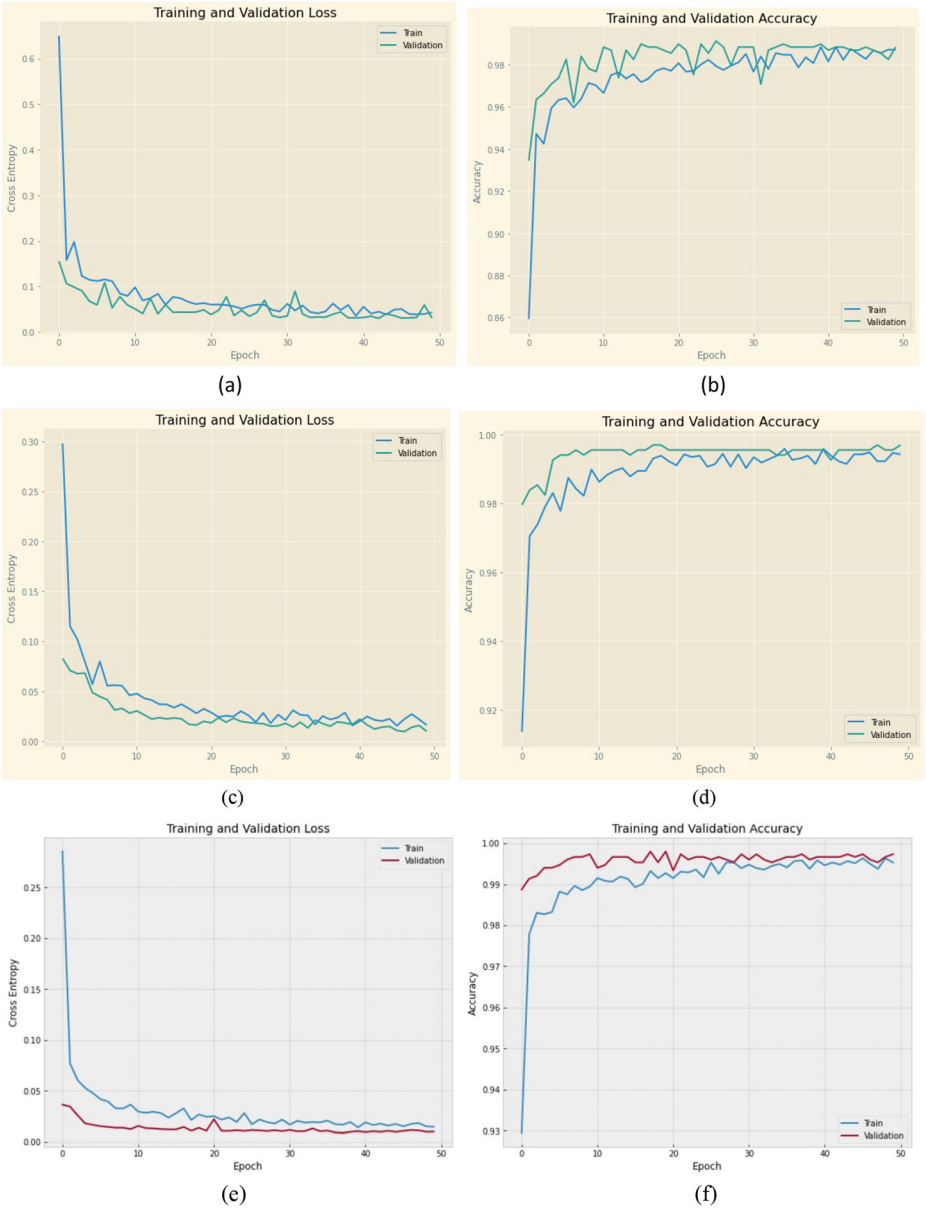

Fig. 10 Model loss and model accuracy curve on training and validation phases on Dataset-II

A comparison of computational time among the base models and ensemble model on Dataset-I and Dataset-II is shown in Table 4. The bold fonts in the table represent the best results.The results shown in Table 4 are computed on a machine equipped with Xenon(R) CPU- E5–1630, 3.70 GHz clock speed, and random-access memory of 32 GB having a

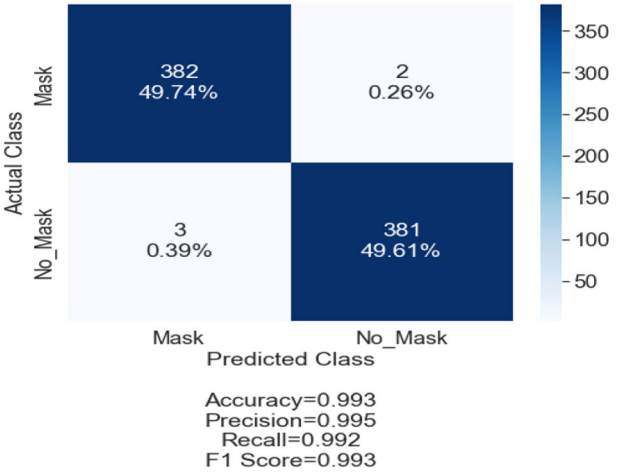

Fig. 11 Confusion matrix generated by the proposed model on Dataset-I

Windows 10 environment. The table compares the models in terms of execution time (in minutes) by considering time per epoch, overall training time on 80% split data, and overall testing time on 20% split data. Training time is not applicable for the proposed ensemble model, as it depends on the base model's performance. It can be observed that the ensemble model has taken less amount of time on test set data for both datasets.

From the experimental results it can be observed that proposed model has achieved good reasonable amount of accuracy along with other metrices. This robustness is achieved because all the three models viz., ResNet50, Inception-v3, and VGG-16 complex architecture deals

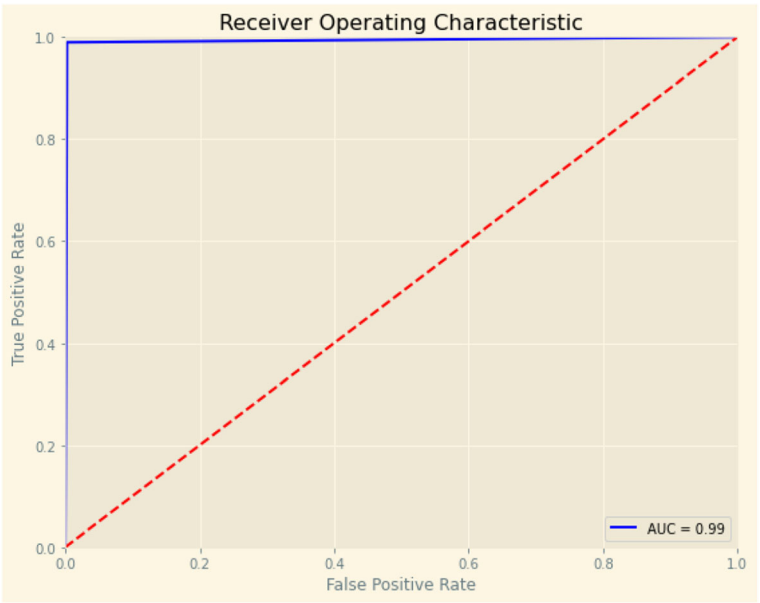

Fig. 12 ROC-AUC curve of proposed model on Dataset-I

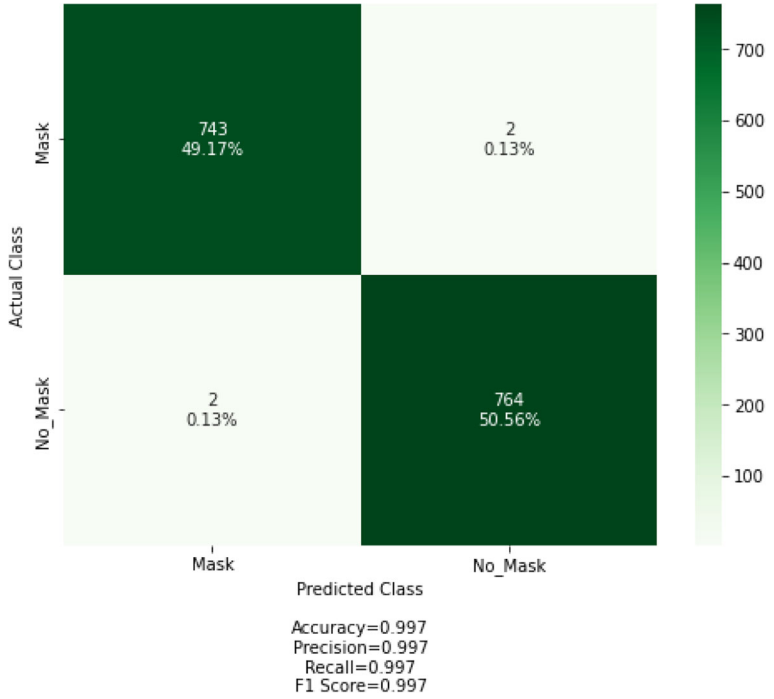

Fig. 13 Confusion matrix generated by the proposed model on Dataset-II

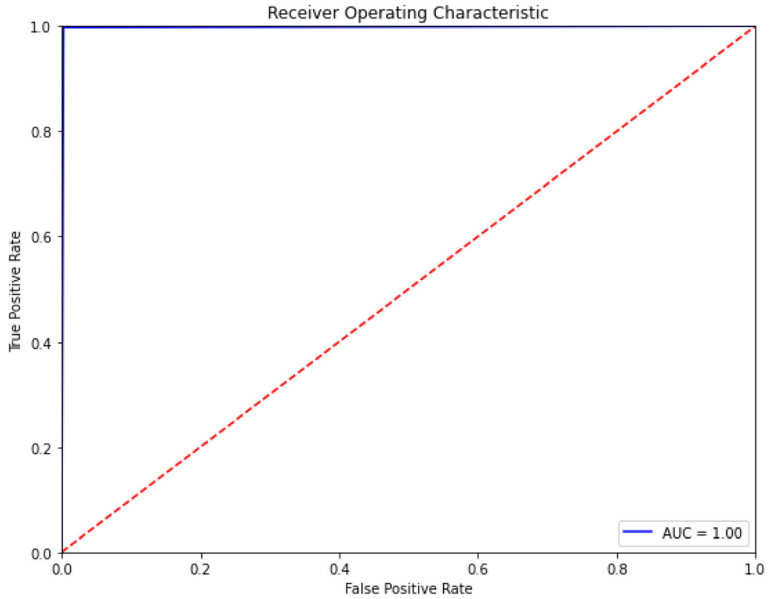

Fig. 14 ROC-AUC curve of proposed model on Dataset-II

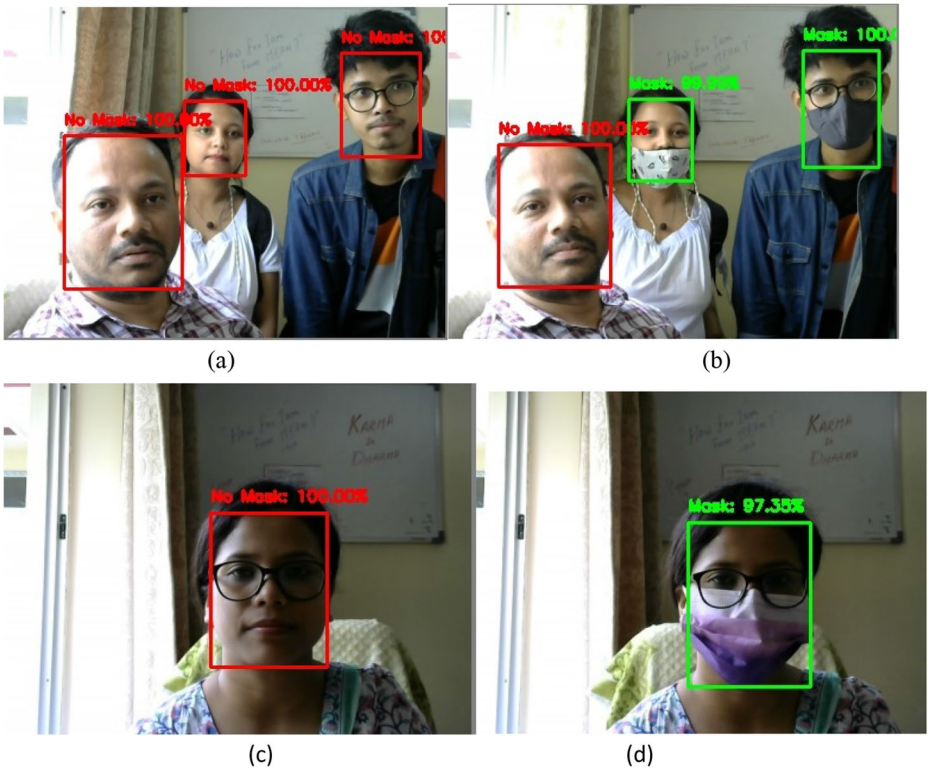

Fig. 15 **a** Detection of 3 faces without mask **b** Detection two faces with mask and one face without mask **c** Detection of face without mask **d** Detection of face with mask

wisely with the vanishing gradient descent problem and they can minimize the noise and variance problems. Moreover, the use of weighted average ensemble approach with grid search technique has judiciously produced the correct output for which it allows better predictive performance in comparison to the other state-of-the-art CNN classification models.

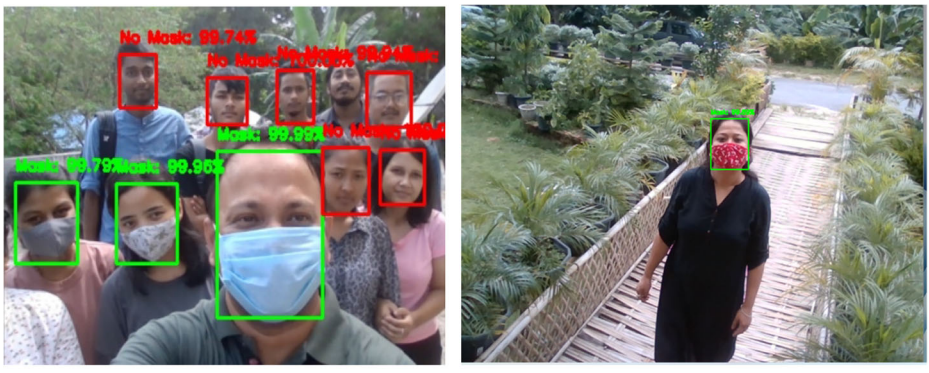

Fig. 16 **a** Mask detection in a group of people **b** Mask detection when a person is in motion in live stream

Table 4 Comparison of computational time various models on Dataset-I and Dataset-II

Dataset	Model	Time Per Epoch in minutes	Overall training time in minutes (On 80% spilt data)	Overall testing time in minutes (On 20% spilt data)
Dataset-I	VGG16	2.451	76.066	1.022
	Inception-v3	2.332	63.892	0.542
	ResNet50	2.066	61.092	0.511
	Proposed Model	N/A	N/A	**0.492**
Dataset-II	VGG16	3.669	132.077	1.113
	Inception-v3	3.122	129.343	1.044
	ResNet50	4.022	152.022	0.582
	Proposed Model	N/A	N/A	**0.533**

5 Conclusion and future directions

The urgency of controlling COVID-19 has increased the application value and importance of real-time mask detection. Manual real-time monitoring of face mask-wearing is a complicated and tedious task. This paper combines three heterogeneous deep transfer learning-based models, viz., ResNet50, Inception-v3, and VGG-16, to prepare an ensemble classification model for detecting whether a person is wearing a mask or not. By using several images, (with mask and without mask) model was trained and tested. Then trials were conducted on live stream videos (captured by web camera) to test the performance of the prepared model. The experimental results show that the proposed model has achieved a highest classification accuracy of 99.97%, precision of 0.997, recall 0.997, F1-score 0.997 and kappa coefficient 0.994. Compared to some of the state-of-the-art models, it has shown better performance. The proposed model can be used with surveillance cameras to detect persons who do not wear face masks and hence may restrict the COVID-19 transmission. In future, by using a high-end web-camera model will be deployed in more real time crowded environment to check its performance.

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

Data availability Image data used in this work is available at www.kaggle.com/datasets.

Declarations

Conflict of interests None declared.

References

1. Alzubaidi L, Zhang J et al (2021) Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8(53):1–74. https://doi.org/10.1186/s40537-021-00444-8
2. Asghar MS, Albogmay FR (2022) Facial mask detection using depth-wise separable convolutional neural network model during COVID-19 pandemic. Public Health 10:855254. https://doi.org/10.3389/fpubh.2022.855254

3. Bania RK (2021) Heterogeneous ensemble learning framework for sentiment analysis on COVID-19 tweets. INFOCOMP J Comput Sci 20(2):1–14
4. Belete DM, Huchaiah MD (2022) Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. Int J Comput Appl 44(9):875–886. https://doi.org/10.1080/1206212X.2021.1974663
5. Coronavirus World Heal Organ (2022) https://www.who.int/healthtopics/coronavirus. Accessed 25 July 2022
6. Das A, Ansari MW, Basak R (2020) Covid-19 face mask detection using TensorFlow, Keras and OpenCV. IEEE 17th India council international conference (INDICON), New Delhi, India. https://doi.org/10.1109/INDICON49873.2020.9342585
7. Goyal H, Sidana K, Singh C, Jain A, Jinda S (2022) A real time face mask detection system using convolutional neural network. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-12166-x
8. Gupta P, Saxena N, Sharma M, Tripathi J (2018) Deep neural network for human face recognition. Int J Eng Manufact 8(1):63–71. https://doi.org/10.5815/ijem.2018.01.06
9. Hussain D, Ismail M, Hussain I et al (2022) Face mask detection using deep convolutional neural network and MobileNetV2-based transfer learning. Wirel Commun Mob Comput 2022:1–10. https://doi.org/10.1155/2022/1536318
10. Kedia P, Katarya R (2021) CoVNet-19: A Deep Learning model for the detection and analysis of COVID-19 patients. Appl Soft Comput J 104:107184
11. Kumar TA, Rajmohan R, Pavithra M, Gaber T (2022) Automatic face mask detection system in public transportation in smart cities using IoT and deep learning. Electronics 11:904. https://doi.org/10.3390/electronics11060904
12. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement 167:Article ID 108288
13. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) Fighting against COVID-19: a novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. Sustain Cities Soc 65:102600
14. Militante SV, Dionisio NV (2020) Real-time facemask recognition with alarm system using deep learning. In: Proceedings of 11th IEEE control and system graduate research colloquium (ICSGRC). IEEE, pp 106–110. https://doi.org/10.1109/ICSGRC49013.2020.9232610
15. Sethi S, Kathuria M, Kaushik T (2021) Face mask detection using deep learning: an approach to reduce risk of coronavirus spread. J Biomed Inform 120:103848
16. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):1–48. https://doi.org/10.1186/s40537-019-0197-0
17. Sitaula C, Hossain MB (2021) Attention-based VGG-16 model for COVID-19 chest X-ray image classification. Appl Intell 51:2850–2863. https://doi.org/10.1007/s10489-020-02055-x
18. Suresh K, Palangappa MB, Bhuvan S (2021) Face mask detection by using optimistic convolutional neural network. 6th International Conference on Inventive Computation Technologies (ICICT), pp. 1084–1089
19. Talahua JS, Buele J, Calvopina P, Varela-Aldas J (2021) Facial recognition system for people with and without face mask in times of the COVID-19 pandemic. Sustainability (Switzerland) 13(12):6900. https://doi.org/10.3390/su13126900
20. Taneja S, Nayyar A, Vividha, Nagrath P (2021) Face mask detection using deep learning during COVID-19, Proceedings of Second International Conference on Computing, Communications, and Cyber-Security, Lecture Notes in Networks and Systems, vol. 203. Springer, Singapore. https://doi.org/10.1007/978-981-16-0733-2_3
21. Teboulbi S, Messaoud S, Hajjaji MA, Mtibaa A (2021) Real-time implementation of AI-based face mask detection and social distancing measuring system for COVID-19 prevention. Sci Program 2021:Article ID 8340779. https://doi.org/10.1155/2021/8340779
22. Ullah N, Javed A et al (2022) A novel deep mask net model for face mask detection and masked facial recognition. J King Saud Univ Comput Inf Sci 34(10):9905–9914. https://doi.org/10.1016/j.jksuci.2021.12.017
23. Weiss K (2016) A survey of transfer learning. J Big Data 3(9):1–40. https://doi.org/10.1186/s40537-016-0043-6
24. Weitz JS, Beckett SJ, Coenen AR, Demory D, Dominguez-Mirazo M, Dushoff J, Leung CY, Li G, Măgălie A, Park SW, Rodriguez-Gonzalez R, Shivam S, Zhao CY (2020) Modeling shield immunity to reduce COVID-19 epidemic spread. Nat Med 26(6):849–854

25. Yang D, Martinez C et al (2021) Detection and analysis of COVID-19 in medical images using deep learning techniques. Sci Rep 11:19638. https://doi.org/10.1038/s41598-021-99015-3
26. Zhu J, Shen B et al Deep transfer learning artificial intelligence accurately stages COVID-19 lung disease severity on portable chest radiographs. PLoS One 15(7):e0236621. https://doi.org/10.1371/journal.pone.0236621