SN Computer Science (2023) 4:288
https://doi.org/10.1007/s42979-023-01738-9

ORIGINAL RESEARCH

RRFMDS: Rapid Real-Time Face Mask Detection System for Effective COVID-19 Monitoring

Burhan ul haque Sheikh[1] · **Aasim Zafar**[1]

Received: 8 July 2022 / Accepted: 15 February 2023 / Published online: 27 March 2023

Abstract

The primary mode of COVID-19 transmission is through respiratory droplets that are produced when an infected person talks, coughs, or sneezes. To avoid the fast spread of the virus, the WHO has instructed people to use face masks in crowded and public areas. This paper proposes the rapid real-time face mask detection system or RRFMDS, an automated computer-aided system to detect a violation of a face mask in real-time video. In the proposed system, single-shot multi-box detector is utilized for face detection, while fine-tuned MobileNetV2 is used for face mask classification. The system is lightweight (low resource requirement) and can be merged with pre-installed CCTV cameras to detect face mask violation. The system is trained on a custom dataset which consists of 14,535 images, of which 5000 belong to incorrect masks, 4789 to with masks, and 4746 to without masks. The primary purpose of creating such a dataset was to develop a face mask detection system that can detect almost all types of face masks with different orientations. The system can detect all three classes (incorrect masks, with mask and without mask faces) with an average accuracy of 99.15% and 97.81%, respectively, on training and testing data. The system, on average, takes 0.14201142 s to process a single frame, including detecting the faces from the video, processing a frame and classification.

Keywords Face mask detection · COVID-19 · Transfer learning · Deep learning · Face detection

Introduction

COVID-19, also known as SARS-CoV-2, is highly contagious. The COVID-19 pandemic has impacted nearly every country, wreaking havoc on available healthcare facilities and treatment systems. Direct contact with infected respiratory droplets spreads the virus (produced via sneezing and coughing). Anyone who comes into contact with virus-infected surfaces and subsequently touches their face may also become sick and experience symptoms such as shortness of breath, cough and fever. Although COVID-19 vaccinations have been developed and their widespread distribution began in early December 2020, they do not eradicate the virus; rather, they minimize complications and morbidity associated with COVID-19.

The WHO emphasizes wearing a face mask in crowded and public places, as it prevents virus transmission via the nose or oral passages [1–3]. In COVID-19-affected countries, the government has instituted laws mandating the wearing a face mask. Face masks (e.g., cotton, surgical, N-95) offer 50–95% protection against the COVID-19 virus [4]. In the circumstances described, it is an excellent practice to always use a face mask to prevent exposure to COVID-19. In this context, determining whether or not a person in a public gathering or an organization is wearing a mask has been the subject of a significant amount of research. Conventional procedures for checking a face mask violation are not always feasible and are error prone. Conventional procedures include human force monitoring people for not wearing face masks manually. Therefore, there is a need for a system that can automatically do this task. Moreover, the human force could be saved and deployed to other essential tasks.

This article is part of the topical collection "Computer Aided Methods to Combat COVID-19 Pandemic" guest edited by David Clifton, Matthew Brown, Yuan-Ting Zhang and Tapabrata Chakraborty.

✉ Burhan ul haque Sheikh
sbuhaque@myamu.ac.in

Aasim Zafar
azafar.cs@amu.ac.in

[1] Department of Computer Science, Aligarh Muslim University, Aligarh 202002, Uttar Pradesh, India

During the last 10 years, significant strides have been made in the field of computer-aided deep neural network (DNN) approaches, which have demonstrated promising results in classification tasks [5, 6], pattern recognition [7], and other areas. DNN models can detect minute variations between images, ultimately allowing them to precisely recognize an object in an image. In face mask detection systems, strategies based on deep learning (DL) and machine learning (ML) have been used to develop reliable and accurate systems that are both quick and efficient. A large number of researchers have developed a variety of DNN architectures for the purpose of detecting face mask violations. Most of these researchers have chosen to base their methods on transfer learning and hybrid approaches (combination of deep learning and machine learning). DL models have shown high sensitivity and specificity when detecting face mask violations, as shown in Table 6.

This paper proposes a system named RRFMDS (Rapid Real-Time Face Mask Detection System) for effective and accurate face mask detection. During the entirety of the development process, deep learning strategies, such as convolution neural networks (CNN), were applied throughout the process. The system consists of a face detector, intermediate block, and face mask detection modules. To construct a face identification module, we employed a single-shot multi-detector, based on Resnet-10 [8, 9]. Furthermore, to identify face masks, we used the transfer learning of the state-of-the-art model, MobileNetV2 [10].

In the first place, faces from video data frames are detected using a face detection module and an intermediate block performs necessary operations on the detected faces and, finally, detected faces are classified as unmasked faces, masked faces, or incorrectly masked faces using fine-tuned MobileNetV2 model. The reason behind selecting these models was to develop a lightweight system that could be incorporated with CCTV cameras. The system is suitable for use in a diverse assortment of scenarios, such as it could be installed in hospitals, supermarkets, and educational institutes.

The following is a list of the primary contributions of this paper:

- A face mask detection system is developed to simultaneously identify and classify multiple faces from the video data.
- The model is able to detect different types of masks, occluded faces, and faces in various orientations accurately.
- The custom dataset developed for this problem is made publicly available on a KAGGLE repository (https://www.kaggle.com/datasets/shiekhburhan/face-mask-dataset).
- The dataset can be used for other problems by the researchers such as face detection, facial landmarks, occluded face detection and recognition of facial expression.
- We discuss the potential for the expansion of a system's functionality in this paper. It may be useful for researchers to build a more accurate face mask detection system.

The structure of this paper is as follows. Following the introduction, "Related Work" deals with the review of extant literature. The next section describes the dataset description, followed by "Proposed Methodology" which discusses the proposed methodology. "Experimental Results and Discussion" provides the experimental results and demonstrates the output of the proposed system. The scope of the present study is captured in "Future Work"; finally, in the last section, we have discussed the conclusion.

Related Work

Due to the ongoing epidemic, there has been a lot of interest in projects with similar purposes. Most researchers utilized CNN from all the approaches mentioned in the literature because of its outstanding performance and capacity to extract valuable characteristics from the image data. Other methods have employed hybrid strategies that use ML methodologies with or without deep learning.

CNN-Based Approaches

Contrary to ML approaches, we do not need to extract the features in CNN-based methods manually. CNN uses convolution and pooling techniques to extract valuable features from the input. We have discussed some popular CNN-based face mask detection models in the following.

In Ref. [11], similar to our model, MobileNetV2 was used to classify the face mask and Caffe-based face detector. A small dataset of 4095 images was used. Additionally, the dataset has only two classes, masked and unmasked. Hence, the model trained is not able to detect the incorrectly masked faces (i.e., having his or her mask below the nose). It achieved a decent F1-score of 0.93.

In Ref. [12], the MAFA or Masked Faces face mask dataset was initially produced. They built a CNN model capable of detecting facial occlusion, including masks. They divided their concept into three key components: the proposal module, the embedding module, and the verification module. The initial module combines two CNNs and retrieves facial image characteristics. The second module focuses on detecting facial landmarks that are not obscured by occlusion. The LLE algorithm is implemented at this stage. In the final module, classification and regression tasks are carried out

using a CNN to determine if an item is a face and to scale the position of missing facial signals. Identifying side-facing faces degraded the model's performance, and the dataset contains more occluded than masked faces. Therefore, training with this dataset is not always viable for face mask identification alone. The performance was determined by calculating the precision of each parameter and averaging the precision of various parameters. Recorded precision averaged 74.6%.

In Ref. [13], a dataset known as "MASKED FACE DATASET" was proposed and three CNN architectures were cascaded for face mask detection. The dataset only consists of 200 images. To overcome the problem of overfitting, they used the concept of transfer learning and fine-tuned the model with the WiderFace dataset [14]. The first CNN consists of five layers and is used to scale the input image. The second and third layers consist of seven layers each. The advantage of using three cascaded CNN is that each false detection is eliminated, thus making a prediction stronger. However, using three CNN makes it computationally expensive. The model achieved an accuracy of 86.6% and recall of an 87.8% on their proposed dataset.

In Ref. [15], the authors presented the SRCNet model for face mask detection. The model comprises two networks: a classification network and an image super-resolution (SR) network. The model is capable of adequately classifying incorrect face mask wearing (IFW), correct face mask wearing (CFW), and no face mask wearing (NFW). The model was trained using the MMD or Medical Masked dataset [16] and the MobileNetV2 CNN algorithm was adopted [10]. The design was well organized and effective. However, the dataset used for training was very small and the inference speed was slower than other algorithms. The model achieved an accuracy of 98.07%.

The approach in Ref. [17] identifies three classification categories: no mask, improper face mask, and with mask. The model was trained on a dataset consisting of 35 masked and unmasked face images. Before training, the dataset was first preprocessed and scaled to the necessary dimensions. The model first identifies the face, extracts the face from the input, and then applies the face mask net model for classification. It includes extremely limited and regionally specific data. The accuracy of the model was reported to be 98.06%.

In Ref. [18], a novel face mask detection technique was proposed using YOLOv2 and ResNet50 together. They used the FMD [31] and MMD [16] datasets to train and test a model. SGDM and adaptive moment estimation (ADAM) optimizers were used to compare the performances. The model achieved an average precision of 81%. In [19], the VGG16 architecture was utilized to identify and categorize face expressions. The accuracy of their VGG16 model trained on the KDEF database is 88%.

The transfer learning of the InceptionV3 model was used in Ref. [20]. The last layer of the model was removed and five new trainable layers were added. The last layer consists of two neurons, followed by a softmax activation function where each neuron corresponds to a masked face and an unmasked face, respectively. The model obtained an accuracy of 99.91% training and 100% testing accuracy in 80 epochs.

In Ref. [21], VGG-16 CNN was used for face mask detection. The dataset they developed consists of 25,000 images, and the model was trained on it. The mask-covered area in an image was first segmented and extracted. The proposed model used the Adam optimizer as an optimization function. Their algorithm was 96% accurate at spotting face masks.

The SSDMNV2 model is proposed in Ref. [22]. They used a similar approach to ours—for face detection, they utilized a single-shot multi-box detector and MobileNetV2 for classification. The classification accuracy was around 92% and the F1-score was 0.93. Our proposed system outperforms it with 98.6% training and 97% testing accuracy and a 0.95 F1-score.

In Ref. [23], for face detection, YOLOv3 was used. It was trained on celebi and wider face [14] databases. The model was later evaluated on the FDDB database [24] and achieved an accuracy of 93.9%.

Hybrid-Based Approaches

The algorithms for deep learning and machine learning were combined in Ref. [25]. The deep learning model ResNet50 was employed for feature extraction, while machine learning methods such as support vector machines and decision tree algorithms were used for classification. One of the four types of datasets contains both actual and fake face masks. On the training dataset containing actual face masks, the decision trees classifier did not obtain a decent classification accuracy (68%) on false face masks.

In Ref. [26], they proposed a model that triggers an alarm for surgical face mask violation in the operating room for face detection. They used Viola–Jones face detection and LogitBoost for face mask detection. One of the problems with the model was that it would make a mistake if clothing was found near the face. Synthetic rotation was used to find a solution to this problem. In addition, the model was trained only on surgical face masks. The recall was said to be above 95%, and the rate of false positives was less than 5%.

In Ref. [27], a haar-cascade-based feature detector was utilized to recognize a nose and mouth from the detected face. The model identifies the nose and mouth and predicts an unmasked face. If it detects only the nose, it predicts an incorrectly masked face and a correctly masked face if neither is detected. This method is quick and straightforward, but it can only interpret full-frontal faces and can be tricked

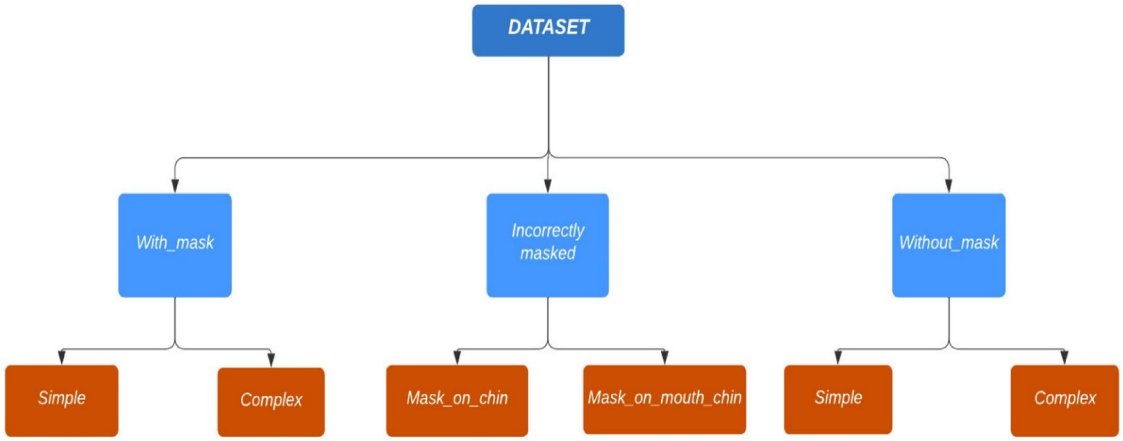

Fig. 1 Dataset organization structure in a graphical format

Fig. 2 Example of images in the dataset

by covering the mouth and nose. Our proposed model is able to predict correctly from different orientations of a face and occlusion, such as a hand on a face or hair on a face.

Principal component analysis (PCA) algorithm was implemented in Ref. [28] for face mask violations. It performed well with an accuracy of 96.25 for detecting the faces without the mask, but while detecting the faces with a mask, the performance was reduced to 68.75%.

Dataset Description

For our proposed model, a customized dataset was created named ***Efficient Face mask Dataset***.

The dataset contained almost all varieties of face masks and was collected manually from different sources. Some of the data were collected from related research [29]. Some of the data were compiled from different standard datasets, including MAsked FAces (MAFA) [12] and Masked Face Detection Dataset (MFDD) [30]. Additionally, some simulated images generated through the data augmentation techniques were also added to the dataset to enrich it. We gathered images from different sources, so our dataset would be diverse and unbiased. The primary reason for creating the dataset was to develop a highly accurate model that can detect a wide range of mask types and effectively identify the occlusion instances in front of the face. The organization of the dataset can be seen in Fig. 1.

The dataset is primarily divided into three categories: with mask, incorrectly masked, and without mask. In addition, we organized the data under these primary categories into distinct subcategories based on their properties. The reason for such an organization is that it can be utilized for other computer vision problems. Besides the face mask detection model, the dataset can be used for face recognition and occlusion face detection. For example, face detection problems may utilize the simple without mask subcategory, whereas face occlusion detection can utilize the complex with mask category. Some of the images of the dataset can be seen in Fig. 2.

The various types of images used in the dataset are as follows:

- *Mask_on_chin* images: These are the images in which masks are put on the chin only. The mouth and nose of a person are visible.

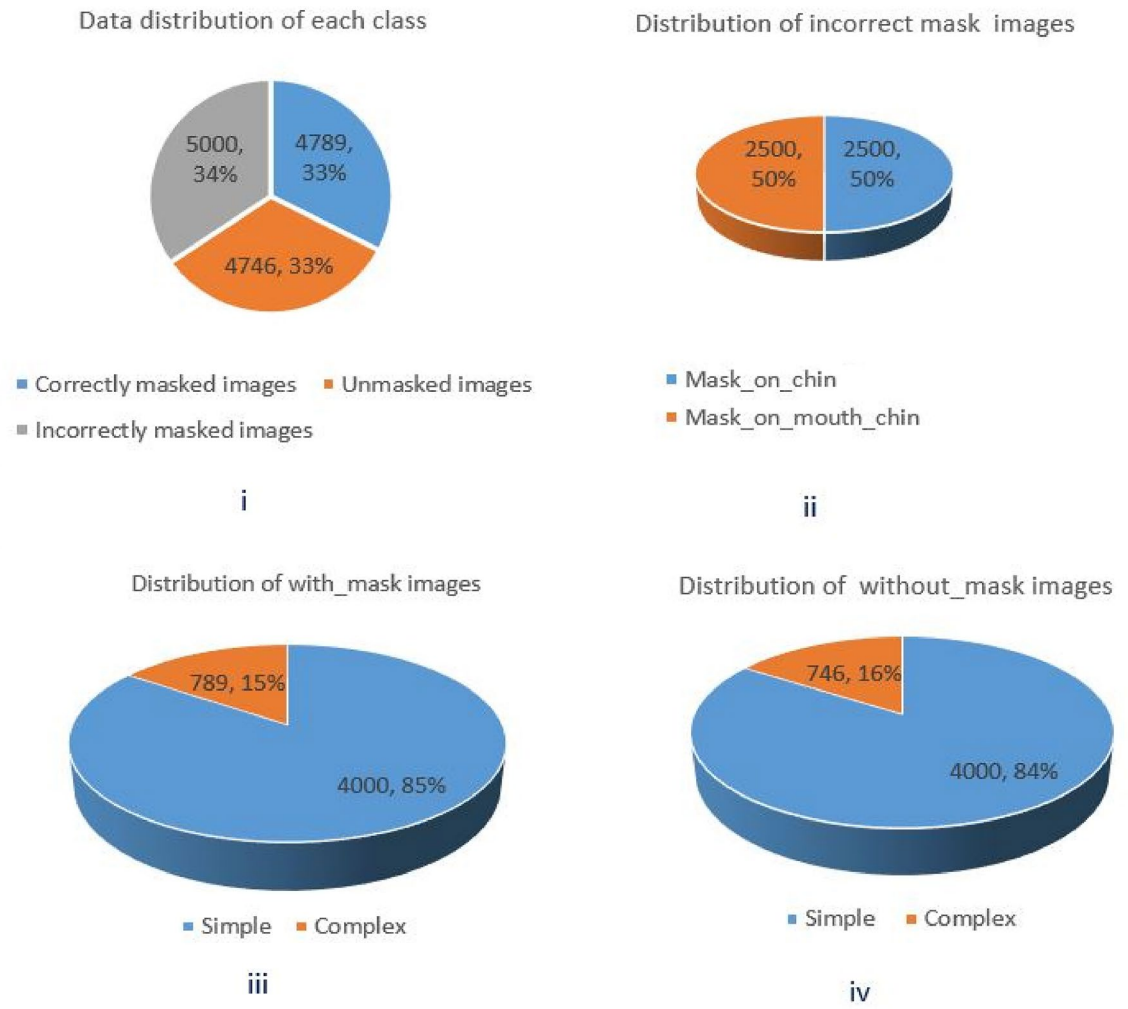

Fig. 3 Data distribution, (i) each class distribution. (ii) Mask_on chin and Mask_on_mouth_chin images in the incorrectly masked images. (iii) Distribution of simple and complex images in correctly masked images. (iv) Distribution of the unmasked images

- *Mask_on_chin_mouth* images: In this, the mask covers the chin and the mouth area. The nose of a person is not covered.
- *Simple with_mask* images: These consist of data samples of face masks without any texture or logos.
- *Complex with_mask* images: It includes images of complex face masks with textures, logos, or printed designs.
- *Simple without_mask* images: These are images without any occlusion.
- *Complex without_mask* images: It consists of faces with occlusion, such as beard, hair, and hands covering the face.

The dataset has a total of 14,535 images. The *incorrect_masked* class consists of 5000 images, of which 2500 are *Mask_on_Chin* and 2500 are *Mask_on _Mouth_Chin.* The *With_mask* class has 4789 images, of which 4000 are simple with_mask and 789 are complex with_mask images. Similarly, *without_mask* has 4746 images, of which 4000 are simple and 746 are complex images. In Fig. 3, we have shown the distribution of the dataset. It consists of four subfigures: data distribution of each class, distribution of incorrect_mask images, distribution of with mask_images, and distribution of without_mask images.

In addition, the proposed dataset has been compared to the standard datasets typically used for face mask identification algorithms in Table 1.

Proposed Methodology

We trained multiple state-of-the-art CNN models, namely VGG-16, Resnet-50 and MobileNetV2, on the proposed dataset to choose the most accurate and fast. The result shows that the model based on the MobileNetv2 performs better in accuracy and inference speed. We have discussed

Table 1 Comparison of various standard face mask datasets with the proposed dataset

Paper	Dataset	Composition of dataset	No. of images	Characteristics	Shortcomings
[31]	FMDD	Contains only masked face images	853	Available publicly	Limited images and only masked images
[30]	MFDD	Solely masked face images	24,471	public	Biased to Chinese faces
[12]	MAFA	Contains a masked face and any sort of occlusion on the face	30,811	Categorical classification is easily deployable since mask type is specified	Mostly preferable for occlusion detection rather than physical mask detection
[32]	MaskedFaceNet	Contains improperly worn masked face data along with masked faces	1,37,016	Benchmark dataset, categorical classification is easier than with other datasets	Biased toward surgical masks
[33]	RMFRD	Masked and unmasked face images of the same subject	95,000	Effective in accuracy, since the dataset is very large	Biased toward Asian facial images
[34]	LFW	Composed of celebrity images of different orientations	13,233	Benchmark dataset for face recognition	Does not contain any masked face image
Proposed dataset	Efficient Face Mask Dataset	Composed of the following: (1) simple and complex masked images (2) simple and complex unmasked images (3) mask-on-chin incorrectly masked and mask-on mouth–chin incorrectly images	14,535	The proposed dataset is broad and adaptable, not skewed toward a single face. It has been constructed by considering the prior dataset's shortcomings. The dataset contains numerous types and colors of masks that were also included to not make it biased toward any single form of a mask. The model constructed on this dataset will recognize practically any form of the mask with varied orientations and can detect occlusion in front of the face easily and reliably	_____

the proposed RRFMDS based on MobileNetV2 in this section. The proposed system consists of the following modules:

1. *Data augmentation*: this module increases the dataset's size and avoids overfitting.

2. *Face Detection module*: this module is responsible for detecting faces from real-time video data.

3. *Intermediate block*: it performs extraction of the detected faces and preprocessing.

4. *Face Mask Detection module*: this module is first trained on the custom dataset and then used for detecting faces as unmasked, masked, or incorrectly masked faces.

RRFMDS Overview

At first, the face mask classifier is trained on an augmented dataset. The face detection module then efficiently locates faces from the frames of a real-time video, even in overlapping scenarios. The faces, or regions of interest, that have been detected are extracted and passed to the intermediate block.

These detected faces are then processed, batched together, and sent to the trained classifier. The classifier predicts the output of the detected faces as probabilities. The final result is the localized faces in the video frame with class prediction and probability of the class.

Figure 4 depicts the general architecture of the system, and Table 2 provides information on the models that were utilized for each individual component.

Data augmentation: Deep learning models are data hungry, i.e., the model significantly improves when a large quantity of training data is provided. The model may be susceptible to overfitting if insufficient training data exists. The model might not be able to generalize learned features well on unseen data. “Data augmentation” refers to the practice of increasing the number of training examples in a dataset. Data is generated from existing images by changing the brightness, orientation, and size.

In our work, we utilized a number of different data augmentation strategies, including flipping, rotation, contrast, shearing, and zooming, with the goal of reducing class imbalance and expanding the size of the training dataset. In addition to helping prevent overfitting, this would also help the model become more robust in identifying previously undiscovered data. We implemented the ImageDataGenerator class of the TensorFlow package to perform

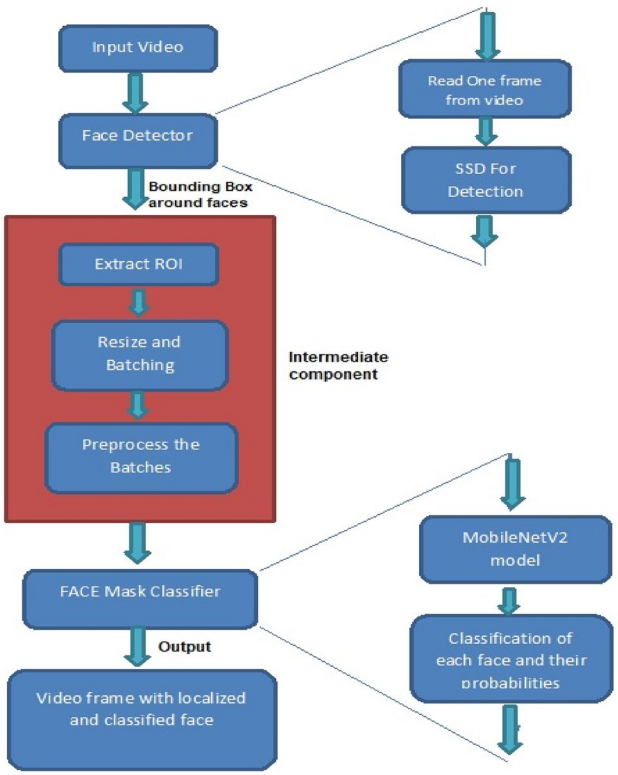

Fig. 4 Illustration of working of face detection, intermediate, and face mask detector

Table 2 Models used for each component

Component	Models considered
Data augmentation	ImageDataGenerator class of tensorflow
Face detection	SSD (single-shot multi-box detector) based on ResNet 50
Face mask classifier/detection	Transfer learning of MobileNetV2

data augmentation. In each epoch, the ImageDataGenerator applied a transformation on all the training images we have and used the transformed images for training. By doing this, we are somehow creating new data (i.e., also called data augmentation), but obviously, the generated images are not totally different from the original ones. This way, the learned model may be more robust and accurate as it is trained on different variations of the same image. Augmentation here does not increase the number of training images per epoch. Instead, it uses a different transformation of each image in each epoch. Since we train our model for 30 epochs, we have used 30 different versions of each original image in training (or 14,535 * 30 = 436,050 different images in the whole training) instead of just the 14,535 original images in the whole training). Put differently, the total number of unique images increases in the whole training from start to finish, not per epoch. Augmented images need to have their dimensions adjusted according to the default input size of MobileNetV2 (224*224*3), and also images are required to be normalized for faster training. Once more, we accomplished this with the assistance of ImageDataGenerator. The main advantage of using ImageDataGenerator is that it generates images on the go during training time. It returns the training batches of the original and augmented images. Some of the augmented images generated can be seen in Fig. 5.

Fine tuning and training: The proposed system employs transfer learning of MobileNetV2 for the classifier. It allows us to use the weights of the state-of-art model, which has been trained on the Imagenet dataset, as the starting point in the training. The base layers are frozen to preserve the features that have already been learned. After that, an additional four trainable layers are added, and those layers are then trained with the help of the custom dataset. The last layer of the model consists of three neurons where each neuron corresponds to the required class. The first neuron corresponds to the incorrect mask, the second corresponds to masked faces, and the third neuron corresponds to the unmasked faces.

During the training of the model, we used 93% of the complete data and remaining 7% were used for a testing purpose. Following this, the model was fine-tuned, and the weights were stored. By using pre-trained weights, it is possible to save a significant amount of computational costs while also improving the end result. Figure 6 illustrates the architecture of the face mask detection model. Figure 7 provides a visual representation of the training process. During the model's training process, the hyperparameters used are shown in Table 3.

Face Detector: The faces of the persons need to be detected first before they are classified. This model is responsible for face detection and acts as the first stage in face mask detection system. We used a pre-trained DNN module named SSD (single-shot multi-box detector) based on ResNet 10 or the face detector in our proposed system. It is an open-source deep neural network model available on the GitHub repository of Open-CV. It is comparable to the YOLO object identification method, which likewise only requires one shot to detect multiple objects using multi-box. The reason for using such a face detector module is that it is significantly lightweight, faster, and accurate and can be embedded with any device. There are two versions available and we have used the Caffe implementation [35] (floating point 16 version). An OpenCV method cv2.dnn.readnet ('path to Caffe model', 'path to prototxt file') was used to load the Caffe model and prototxt file.

A frame is extracted from the real-time video data and then sent to the face detector. The face detector will determine each and every face contained inside a frame, after

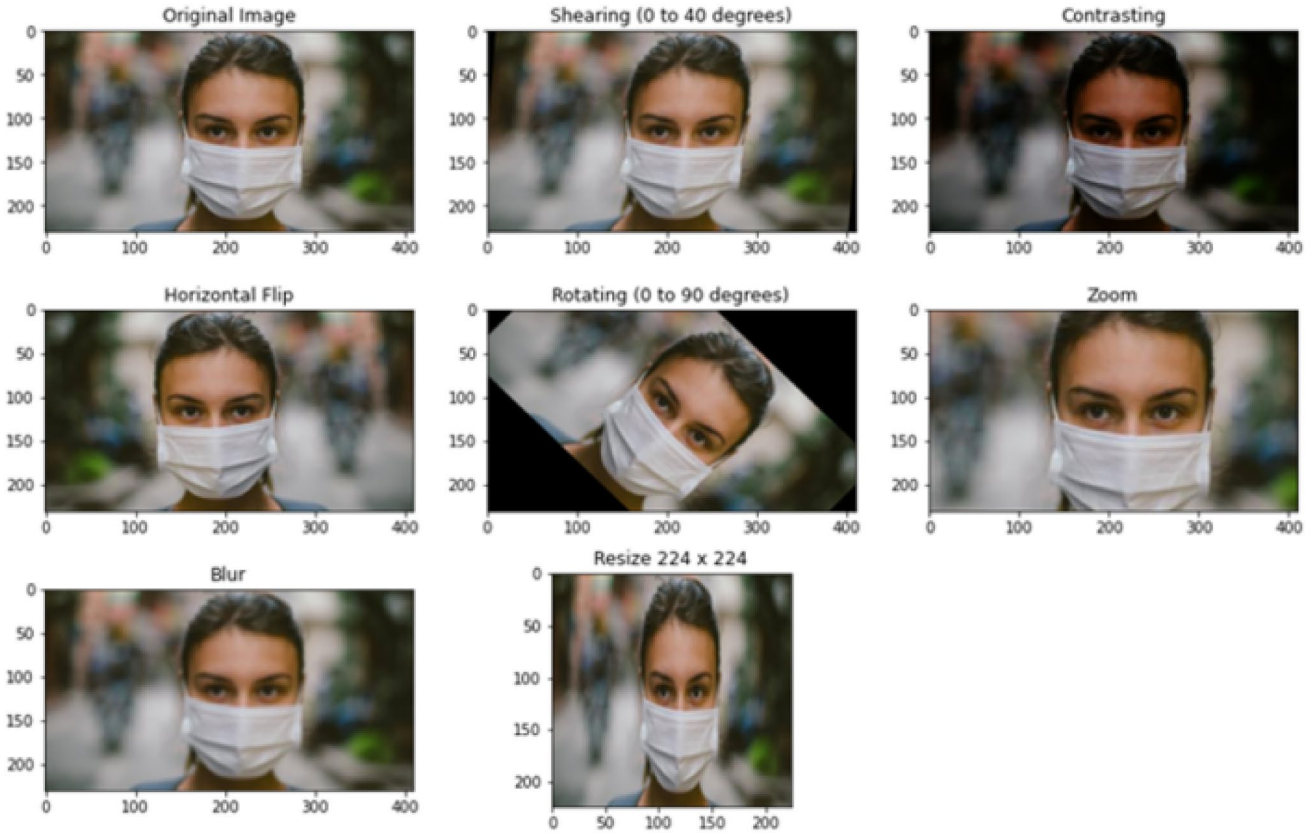

Fig. 5 Augmented images of one of the data samples using the ImageDataGenerator method

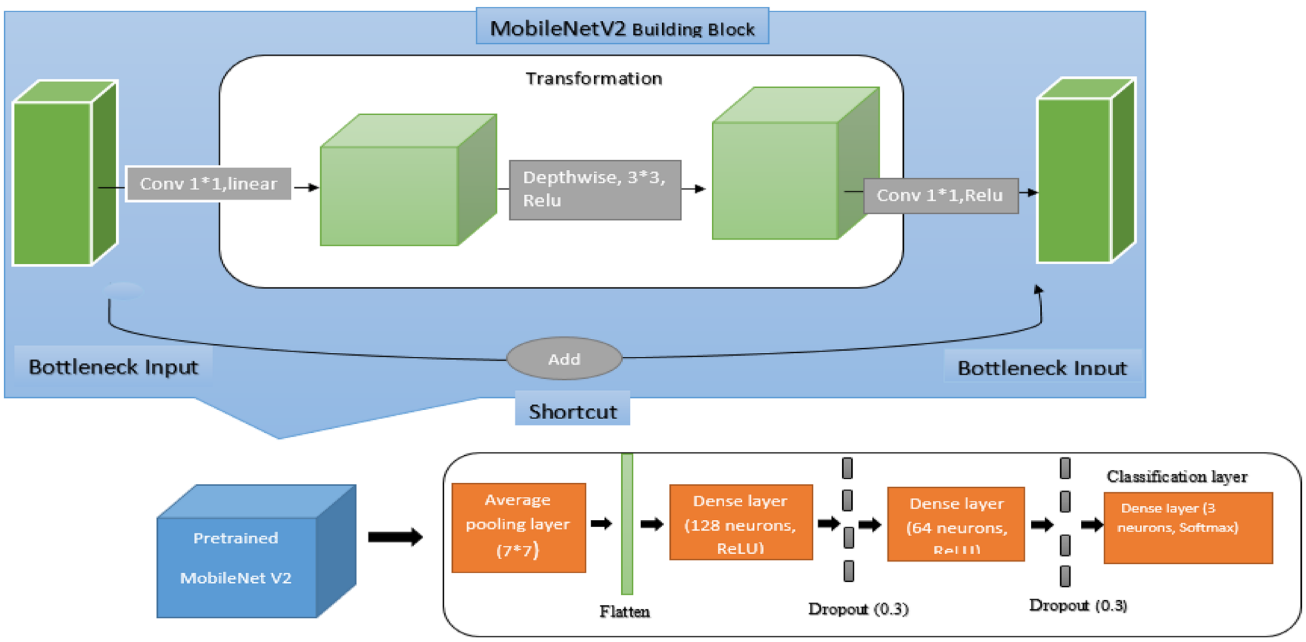

Fig. 6 Architecture of the face mask detector based on MobileNetV2

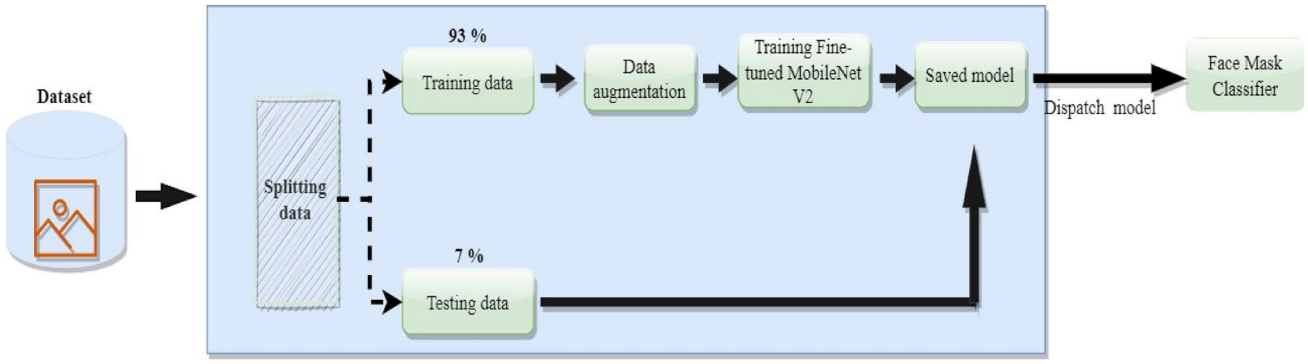

Fig. 7 Training process of the face mask classifier

Table 3 Hyperparameter setting of a face mask detection model

Hyperparameter	Value
Learning rate	1e-3 with decay rate = learning rate/ epoch number
Batch size	32
Epochs	30
Dropout rate	0.3, 0.3
Input layer size	224 *224*3
Output layer size	224*224*3
Optimization	ADAM
Loss function	CategoricalCrossentropy

which it will output the detected faces together with the bounding box coordinates. Following that, the outputs are sent to an intermediate block, where they are subjected to further processing before being transferred to the face mask detector.

The following is the algorithm of the face detector and intermediate processing block.

In STEP 1, the frames from the real-time video are captured and then the faces from the captured frame are identified in STEP 1.1. If the number of faces is greater than 1, we crop the region of interest (ROI) from the frame, i.e., the face region and save them separately, as shown by STEP 1.2. In STEP 2, cropped image(s) is preprocessed to make it MobileNetV2 compatible. In STEP 3, the processed image(s) are sent to the Classifier for classification.

Intermediate Processing Block: After the faces have been identified, they need further processing. This module is responsible for processing the identified faces and arranging them together in batches for classification. Input to this block is the frame with bounding boxes received from the face detection module. It is responsible for extracting the region of interest (ROI) by cropping the bounding box region of a frame. This is done because the face mask detection module requires only the person's entire face to make a classification. After cropping the ROI, the extracted faces are processed by the *preprocess_input* method of the *Mobilenet_v2* class in Tensorflow. Preprocessing involves resizing the extracted faces into 224*224*3 dimensions and normalizing an image from 0 to 1 so that a classifier can process it. Furthermore, all the faces are batched together for batch inference.

Face mask detection: This module determines whether the input it receives from the intermediate processing block should be categorized as masked, unmasked, or incorrectly masked. To train the face mask classifier, a model called MobileNetV2 is used based on transfer learning. Because of its lightweight architecture, low latency, and high performance, this model is well suited for video analysis. As a result of this step, a video frame will be produced with localized faces and will be categorized as either incorrectly masked, masked, or unmasked.

After the images have been preprocessed as required by MobileNetV2, the features of the preprocessed image are computed in STEP 1. After computing the features, the newly added layer does the further computation, and the last layer, which is a softmax layer, computes the probability of each face belonging to the particular class.

Algorithm 1 Face detector and intermediate block

Input: Frame from a real-time video
Output: Preprocessed Image for Classifier
STEP 1: **for** *each frame in a video* **do**
STEP 1.1: Identify faces from the frame with a bounding box.
STEP 1.2: If number of bounding box > 1
for *each bounding-box* **do**
Crop the bounding box regions to extract ROI and save the images separately
end
else
Crop the bounding box region to extract ROI and save it
end

STEP 2: Preprocess the cropped image (s) as required by the Face mask classifier MobileNetV2
STEP 3: Preprocessed image(s) are sent to the MobileNetV2 Classifier for classification
STEP 4: Repeat step 1 until the video is closed.

Algorithm 2 Face mask detection

Input: Preprocessed image
Output: Classification of the localized faces from the real-time video data

STEP 1: Compute the feature map from the input image(s).

STEP 2: Computed features are processed by the newly added layers.

STEP 3: Output from the last layer (consist of 3 neurons) is given to the softmax activation function.

STEP 4: Softmax function gives the probabilities of each class.

STEP 5: The localized face(s) are classified as the class with maximum probability

Experimental Results and Discussion

Google colaboratory was used to train a model. It is a platform that enables us to write Python scripts for machine learning, deep learning, and data analysis and then execute those programs while having unrestricted access to the cloud's resources. Whenever a new Colab session is started, the computer will automatically assign a random GPU, CPU, and amount of disc storage. There is a wide variety of graphic processing units (GPU) that are readily accessible, including T4s, P4s, Nvidia K80s, and P100s.

In this paper, Keras, Tensorflow, Sklearn, Matplotlib, and Numpy APIs were used. Keras and Tensorflow are advanced neural network packages used to design a classifier.

- *Keras and Tensorflow:* Used to design a classifier, MobileNetV2.
- *SKlearn:* For data analysis, such as computing the metrics of the model.
- *OpenCV::* To perform image processing operations and also used to load the face detector model.
- *Imutils:* For video streaming.
- *Matplotlib:* To plot the learning curves of accuracy and loss.
- *Time:* To compute the average processing of the frame.

There are a number of metrics that are used to assess a model's performance, including recall, precision, and F1-score, accuracy, as well as the macro average and weighted average and the average frame rate (FPS). All these metrics are defined below:

- *Accuracy*: Represents the number of correctly classified data instances over the total number of data instances.
- *Precision:* Is the proportion of correctly predicted positive observations to the total predicted positive observations.
- *Recall or sensitivity*: Is given by the proportion of true positive to all positives.
- *F1-score:* Is the harmonic mean of the recall and precision, i.e., mathematically, it is computed as a weighted average of both.

$$\text{F1 - score} = 2 * (\text{recall} * \text{precision}) / (\text{recall} + \text{precision}).$$

Table 4 Performance of RRFMDS

Performance metrics	Class	Result (1000 test samples)	Macro average	Weighted average
Precision	Incorrect mask	**0.98**	**0.98**	**0.98**
	Unmasked	**0.98**		
	With mask	**0.98**		
Recall	Incorrect mask	**1.00**	**0.98**	**0.98**
	Unmasked	**0.97**		
	With mask	**0.98**		
F1-score	Incorrect mask	**0.99**	**0.98**	**0.98**
	Unmasked	**0.97**		
	With mask	**0.98**		
Average accuracy (testing data)	–	**97.81%**	–	–

The results of the proposed system are in bold.

Table 5 Frame processing time of the proposed model

Average FPS (without GPU)	Average time to process a single frame
7	**0.13201142** s

The result of the proposed system is in bold.

We have implemented the *classification_report* method of the SKlearn package to compute all these metrics. The performance of a model on test data is depicted in Tables 4 and 5.

Although there are Tensorflow packages available to compute the frames per second (FPS), we preferred to write a code to get the FPS. The FPS was computed by taking the mean of ten runs, where each run was for 60 s and the average processing time of a frame was computed by using the *Time* package of Python. It was calculated by taking the mean of the ten frames. A system's frame rate and frame processing time were computed on a device without a GPU .

The model was trained for 30 epochs and achieved an accuracy of 99.15% and 97.81% on training and testing data. The learning curves for the model are shown in Fig. 8. The plot makes it clear that as the number of epochs in training and validation increases, both the accuracy of training and validation increases, while the loss of training and validation decreases. It is evident from the plot that the testing and training accuracy are not far from each other, which concludes the model is not overfit.

Additionally, the RRFMD model was compared to other pre-existing models that had been trained on a variety of datasets. Table 6 shows the result of the comparison. The majority of the models in the table are based on transfer learning of advanced CNN models such as ResNet50, VGG, and MobileNet. However, the proposed model performed significantly better than all of these models in terms of accuracy and F1-score. In Table 7, we have compared the various parameters of existing state-of-the-art models to the proposed model. Moreover, Table 8 shows the performance of the various existing face mask models in terms of the time complexity required to process the frame or average frames processed per second and the memory complexity. The results of some of these models may appear better than the suggested models because nearly all of these models were trained on GPU-based computers. On a machine with a GPU, the proposed model will likely perform better.

Comparison of Models

We also trained some pre-existing state-of-the-art models on the same dataset and compared the results with the proposed RRFMDS. We chose VGG16 and Resnet50 for this purpose and the comparison of the results with the proposed model in terms of accuracy and F1 score is shown in Table 9. We also compared the average frames per second (FPS) processed of the selected models in Table 9.

VGG16 is a 16-layer pre-trained CNN model that won first prize in image classification in the 2014 ILSVRC challenge [49]. However, the model is computationally heavy and has 128 million parameters. The model performed at the rate of 2.13 FPS, making it unsuitable for low-ended devices such as CCTV cameras.

ResNet-50 is a 50-layer deep network [49]. It has 48 convolution layers along with 1 MaxPool and 1 Average Pool layer. ResNet is based on the deep residual learning framework. It solves the problem of the vanishing gradient problem even with extremely deep neural networks. Despite

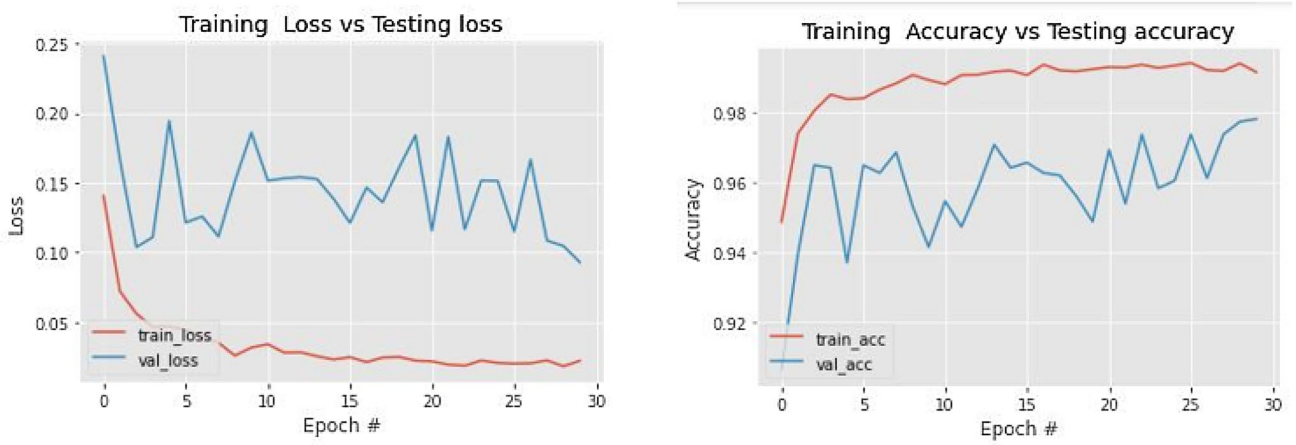

Fig. 8 Learning curves of models accuracy and loss

having 50 layers, it has over 23 million trainable parameters, which is much smaller than existing architectures. The model provides good accuracy, but cannot be deployed to the resource-constrained device due to high computational demand. During the testing, it only provided an average of 2.78 FPS.

The proposed model outperforms all these selected models in terms of accuracy and FPS processing time. The proposed model is lightweight in terms of total parameters and the model size is also small compared to the others. This makes the proposed model suitable for small devices such as CCTV cameras.

Output

The video was captured from an HP HD web camera (05c8:03b1) which is approximately 2.0 megapixels. As depicted in Fig. 9, the output of the model for the current frame of the video is a bounding box around the face with red, green, or blue colors where the red bounding box implies an unmasked face, green indicates wearing a mask, and blue means a person is incorrectly wearing a mask. The class prediction and class probability are also shown on top of the bounding box.

Future Work

In the future, we would like to work on the following areas:

- The real-world application of a model is a little challenging. Although the model performs well and has reasonable accuracy, the performance in the real world can degrade. Due to the weather and other factors, the model might not receive a good-quality video. There is a possibility that the received frame is blurry and has poor luminance. In that case, the model needs some improvement. The same model can further be improved by adding the dataset with blur and low-contrast images.
- In the future, we would like to compare the performance of multiple state-of-the-art models on benchmark and proposed datasets.
- At the moment, the model provides an inference speed on a CPU of 7 frames per second. Our goal in the future is to improve it and make it viable for CCTV cameras even without GPU.
- In the future, the models used for the system's components can be changed to ones that are more accurate and have lower latency.
- The alarm functionality can be added to the system, i.e., whenever there is a face mask violation, an alarm raises.
- The prediction quality can be improved by using a high-definition video camera for video capture.

Table 6 Comparison of various state-of-the-art models with the proposed model

Paper	Algorithm	Classification type	Model description	Dataset	Performance metric	Performance	Shortcomings
[12]	LLE-NET	Binary	A novel dataset and face occlusion detection technique using LLE-CNN and its three separate modules	MAFA	Average precision	74.6%average precision	1. Side face orientation affects the model's performance 2. Detects any occlusion regardless of face masks
[13]	Cascaded CNN	Binary	Three CNN classifiers were combined to detect a masked face and named as cascade framework of CNN	1. WiderFace (pre-trained) [14] 2.MASKED FACES [13]	Accuracy, recall, IoU	Accuracy is 86.6% and recall is 87.8%	Use of three cascaded CNN might bring complexity and easy replaceability by a robust classifier
[15]	SRCNet	Categorical	Image super-resolution network (SR) and a classification network were combined to identify face mask-wearing conditions Images with low resolution improved with an SR network, and MobileNetV2 was used for the classification network	Medical Mask Dataset (MMD) MMD[16]	Accuracy	Accuracy is 98.70%	1. Relatively small dataset 2. Lack of identification on video streams 3. Detection speed is comparatively high
[36]	SSD, MobileNetV2	Binary	A real-time social distance maintaining and a face mask detector was deployed in raspberry pi4 using a combination of SSD and MobileNetV2 Violating instructions resulted in alarm	Customized	Confidence, precision, recall	Precision of 91.7% with a confidence score of 0.7	Performance is evaluated with the detection of social distancing
[18]	ResNet-50, YOLOv2	Binary	A face mask classification algorithm with the help of YOLOv2 and ResNet-50, along with a novel dataset	1. FMDD [31] 2. MMD [16]	Average precision, log average miss rate	Average precision is 81%, LAMR was 0.4	1. Relatively small dataset 2. Lack of video streams

Table 6 (continued)

Paper	Algorithm	Classification type	Model description	Dataset	Performance metric	Performance	Shortcomings
[36]	ResNet, SVM	Binary	Detects face mask from speech using the data augmentation technique with multiple ResNet that GANs have trained	Mask Augsburg Speech Corpus (MASC)	Accuracy	74.6% accuracy	Processing time is high
[26]	AdaBoost	Binary	This method used color filters for the classification of face and face mask in an operating room with the help of skin texture in HSV color space	1. BAO [37] 2. LFW [34]	Recall, false positive	Recall above 95%, false positive rate below 5%	1. Only trained to detect surgical masks in the operation room 2. Use of color filter does not provide high performance always
[11]	MobileNetV2, Caffe-based detector	Binary	Similar to our model, it used mobilenetV2 for classification of masked and unmasked for face detection It used Caffe-based detector	Custom dataset of 4095 images	F1-score	0.93 F1-score	It has a limited dataset. The model can classify only masked and unmasked faces
[22]	SSDMNV2	Binary	Similar to our proposed model, it uses MobileNetV2 for classification and SSD detector for face detection	Custom dataset of 5521 images	Accuracy, precision, recall, F1-score	Test accuracy of 92%, F1-score 0.93, precision 0.94, recall 0.93	Small dataset and cannot detect in correctly masked faces
[21]	VGG-16	Binary	The mask-covered area in an image was first segmented and extracted and then trained on VGG-16	Custom	Accuracy, precision, recall, F1-score	96% accuracy, precision 0.96, recall 0.96, F1-score 0.96	Processing time is high

Table 6 (continued)

Paper	Algorithm	Classification type	Model description	Dataset	Performance metric	Performance	Shortcomings
Proposed model	Mobile Net V2 and Caffe-based SSD based on Resnet1-10	Categorical (masked, unmasked, incorrectly masked)	The proposed model was trained on transfer learning of MobileNetv2 and for face detection module ResNet10-based single-shot multi-box detector was used	Proposed dataset	Accuracy, precision, recall F1-score, Avg frame, frame per second (FPS), average time to process a single frame (FPT)	Training accuracy = 99.15%, test accuracy = 97.81%, precision = 0.98, recall = 0.98, F1-score = 0.98, FPS = 8 (ON CPU), FPT = 0.12201142 s (without GPU)	–

Conclusion

This paper proposes a system for rapid face mask detection that uses a lightweight and efficient approach for face detection and mask classification from real-time video data. The system consists of a face detection module and a face mask classifier, employing a single-shot multi-box detector based on ResNet50 and a MobileNetv2 convolutional neural network classifier, respectively. The proposed system is able to process a single frame from the live video data in 0.13201142 s and processes 7 frames per second, as shown in Table 3. The system was trained on the diverse and challenging custom dataset and achieved an accuracy of 97.81% on the test data, as shown in Fig. 8 and Table 4. In addition, the system was compared with some standard face mask detection models in Table 6 and the results show that the proposed system outperformed all of them. By integrating this system with pre-installed CCTV cameras, it can be applied to monitor face mask violations in various public places, including educational institutes, hospitals, and traffic signals. The dataset used in this work is publicly available, providing a valuable resource for future research on face mask detection. In Table 1, we also compared the proposed dataset with some benchmark datasets. To enhance the size of the dataset, the data augmentation technique was employed. Overall, this system has the potential to contribute to the mitigation of COVID-19 transmission and improve public health and safety.

Table 7 Parameter comparison of various models with the proposed system

Article	Parameters					
	# classifier	# Layers	Learning rate	Optimization	No. of images	Epoch
[12]	2	–	–	–	35,806	–
[13]	2	5,7,7	–	–	–	200
[15]	3	53	10^-4	ADAM	3843	50
[36]	2	18–101	10^-4	ADAM	1415	100
[21]	2	16	10^-3	ADAM	25,000	100
Proposed system	3	MobileNet + 4 new trainable layers	10^-3	ADAM	14,535	30

Table 8 Comparison of time and memory complexity of various existing models with the proposed model

Article	Model	(Frames per second) FPS	Inference time of an image	Size or number of parameters
[38]	(Squeeze and excitation) SE-YOLOv3	Not available	The average execution time of the system for processing one image frame is 0.13 s	Not available
[39]	YOLO v4	The average FPS generated by the face detection Is about 11,1 FPS	Not available	Not available
[40]	Multi-task cascaded convolutional network, due to its superior speed and efficiency (MTNN) [41] + Resnet18 [42]	6 FPS	(Inferences/s (on Tesla V100)) ResNet18 (680.83) MobileNetv2 (577.15) DenseNet161 (301.49) Inceptionv3 (425.99)	Not available
[43]	MobileNetV2, DenseNet121 NASNet	Currently, the model gives 5 FPS inference Speed on a CPU	Average inference Time for a 720p resolution image (s) 0.295 (MobileNetV2) 0.353 (DenseNet121) 0.118 (NASNet)	4.88 (million) 8.52 (million) 4.07 (million)
[22]	SSDMNV2 LeNet AlexNet VGG-16 ResNet-50	15.71 14.55 6.31 2.76 2.89	Not available	Not available
[26]	Viola–Jones face detection and LogitBoost for face mask detection	10 fps	Not available	Not available
[44]	Custom CNN(Proposed Method) MobilenetV2 DenseNet-121 Inception-V3 VGG-19	Not available	Not available	11 Mb (MobileNetV2) 96 MB (DenseNet-121) 89 MB (Inception-V3) 79 MB (VGG-19) 33 MB (proposed model, around 3 million parameters)
[45]	Efficient-YOLOv3	14.62	Not available	15.91 million parameters
[46]	YoloV3	12.91	Not available	61.58 million parameters
[47]	YoloV4	11.32	Not available	64.01 million parameters
Proposed model	**MobileNetV2**	**7 fps (without GPU)**	**0.13201142**	**11 MB** (~3 million parameters)

The results of the proposed system are in bold.

Table 9 Comparison of the chosen state-of-the-art models with the proposed model using accuracy, F1-score, and FPS parameters

Architecture	Average accuracy (testing data)	F1-score (%)	Average frame per second (FPS)	Size	No. of trainable parameters
VGG16	90.07	90.8	2.13	69 MB	Around 129 million
ResNet-50	93.05	92.89	2.78	28 MB	Around 24 million
Proposed model	**97.81**	**0.98**	**7**	**11 MB**	Around 3 million

The results of the proposed system are in bold.

Fig. 9 The output of the model. Case 1: system correctly classifies faces as with_masks with an accuracy of about 100%. Case 2: it correctly detects the occlusion (hand) in front of the face and classifies it as no mask with 99.99% accuracy. Case 3: it classifies faces as incorrect mask and correct mask accurately. Case 4: it correctly classifies faces with no masks. Case 5: it classifies persons with an incorrect mask with good accuracy

Data Availability The datasets generated during and/or analyzed during the current study are available in the KAGGLE repository, https://www.kaggle.com/datasets/shiekhburhan/face-mask-dataset.

Code Availability The code generated during and/or analyzed during the current study is available from the corresponding author upon reasonable request.

Declarations

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ellis R. WHO changes stance, says public should wear masks. World Health Organization; 2020.
2. Feng S, Shen C, Xia N, Song W, Fan M, Cowling BJ. Rational use of face masks in the COVID-19 pandemic. Lancet Respir Med. 2020;8(5):434–6.
3. World Health Organization. Advice on the use of masks in the context of COVID-19: interim guidance, 6 April 2020 (No. WHO/2019-nCov/IPC_Masks/2020.3). World Health Organization; 2020.
4. "How mask antiviral coatings may limit covid- 19 transmission". 2020 [Online]. Available: https://www.optometrytimes.com/view/

how-mask-antiviral-coatings-may-limit-COVID-19-transmission. Accessed 17 Jun 2021.
5. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems. ACM; 2012. p. 1097–105.
6. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations. 2015.
7. Jain AK, Duin RPW, Mao J. Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell. 2000;22(1):4–37.
8. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: single shot multibox detector. In: European conference on computer vision. Cham: Springer; 2016. p. 21–37.
9. Anisimov D, Khanova T. Towards lightweight convolutional neural networks for object detection. In: 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS). IEEE; 2017. p. 1–8.
10. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Adam, H. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2017.
11. https://github.com/chandrikadeb7/Face-Mask-Detection
12. Ge S, Li J, Ye Q, Luo Z. Detecting masked faces in the wild with lle-cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE; 2017. p. 2682–90.
13. Bu W, Xiao J, Zhou C, Yang M, Peng C. A cascade framework for masked face detection. In: 2017 IEEE international conference on cybernetics and intelligent systems (CIS) and IEEE conference on robotics, automation and mechatronics (RAM). IEEE; 2017. p. 458–62.
14. Yang S, Luo P, Loy CC, Tang X. Wider face: a face detection benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE; 2016. p. 5525–33.
15. Qin B, Li D. Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. Sensors. 2020;20(18):5236.
16. "Medical mask dataset". [Online]. Available: https://www.kaggle.com/shreyashwaghe/medical-mask-dataset. Accessed 17 Jun 2021.
17. Inamdar M, Mehendale N. "Real-time face mask identification using facemasknet deep learning network". Available at SSRN 3663305, 2020. [Online]. Available: https://dx.doi.org/https://doi.org/10.2139/ssrn.3663305.
18. Loey M, Manogaran G, Taha MHN, Khalifa NEM. Fighting against COVID-19: a novel deep learning model based on yolo-v2 with resnet-50 for medical face mask detection. Sustain Cities Soc. 2021; 65: 102600. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210670720308179.
19. Hussain SA, Al Balushi ASA. A real time face emotion classification and recognition using deep learning model. J Phys: Conf Ser. 2020;1432(1):012087.
20. Jignesh Chowdary G, Punn NS, Sonbhadra SK, Agarwal S. Face mask detection using transfer learning of inceptionv3. In: International conference on big data analytics. Cham: Springer; 2020. p. 81–90.
21. Militante SV, Dionisio NV. Real-time facemask recognition with alarm system using deep learning. In: 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC). IEEE; 2020. p. 106–10.
22. Nagrath P, Jain R, Madan A, Arora R, Kataria P, Hemanth J. SSDMNV2: a real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. Sustain Cities Soc. 2021;66: 102692.
23. Li C, Wang R, Li J, Fei L. Face detection based on YOLOv3. In: Recent trends in intelligent computing, communication and devices. Singapore: Springer; 2020. p. 277–84.
24. Jain V, Learned-Miller E. Fddb: a benchmark for face detection in unconstrained settings (Vol. 2, No. 6). UMass Amherst technical report. 2010.
25. Loey M, Manogaran G, Taha MHN, Khalifa NEM. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement. 2021;167: 108288.
26. Nieto-Rodríguez A, Mucientes M, Brea VM. System for medical mask detection in the operating room through facial attributes. Springer; 2015. p. 138–45.
27. Cabani A, Hammoudi K, Benhabiles H, Melkemi M. Masked-Face-Net–a dataset of correctly/incorrectly masked face images in the context of COVID-19. Smart Health. 2021;19: 100144.
28. Ejaz MS, Islam MR, Sifatullah M, Sarker A. 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; 2019. p. 1–5.
29. https://www.kaggle.com/datasets/andrewmvd/face-mask-detection
30. Wang Z, Wang G, Huang B, Xiong Z, Hong Q, Wu H, Yi P, Jiang K, Wang N, Pei Y et al. Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093, 2020. [Online]. Available: arXiv:2003.09093
31. "Larxel's face mask detection dataset: Fmdd". [Online]. Available: https://www.kaggle.com/andrewmvd/face-mask-detection. Accessed 17 Jun 2021.
32. Cabani A, Hammoudi K, Benhabiles H, Melkemi M. Masked-facenet – a dataset of correctly/incorrectly masked face images in the context of COVID-19. Smart Health. 2021; 19: 100144. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352648320300362.
33. "Real-world masked face recognition dataset". [Online]. Available: https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset. Accessed 17 Jun 2021.
34. Huang GB, Mattar M, Berg T, Learned-Miller E. Labeled faces in the wild: a database for studying face recognition in unconstrained environments. In: Workshop on faces in' real-life' images: detection, alignment, and recognition. Marseille, France: Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, Oct. 2008. [Online]. Available: https://hal.inria.fr/inria-00321923.
35. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Darrell T. Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on multimedia. ACM; 2014. p. 675–8.
36. Yadav S. 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. 2020;8(7):1368–75.
37. Frischholz R. Bao face database at the face detection homepage. 2012. Accessed 17 Jun 2021.
38. Jiang X, Gao T, Zhu Z, Zhao Y. Real-time face mask detection method based on YOLOv3. Electronics. 2021;10(7):837.
39. Susanto S, Putra FA, Analia R, Suciningtyas IKLN. The face mask detection for preventing the spread of COVID-19 at Politeknik Negeri Batam. In: 2020 3rd International Conference on Applied Engineering (ICAE). IEEE; 2020. p. 1–5.
40. Li Z. Face mask detection: MTCNN + ResNet18 demo.
41. Zhang K, Zhang Z, Li Z, Qiao Y. Joint face detection and alignment using multi-task cascaded convolutional networks. IEEE Signal Process Lett. 2016;23(10):1499–503.

42. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. pp. 770–778.
43. Chavda A, Dsouza J, Badgujar S, Damani A. Multi-stage CNN architecture for face mask detection. In 2021 6th International Conference for Convergence in Technology (i2ct), IEEE; 2021. pp. 1–8
44. Goyal H, Sidana K, Singh C, Jain A, Jindal S. A real time face mask detection system using convolutional neural network. Multimed Tools Appl. 2022;81(11):14999–5015.
45. Su X, Gao M, Ren J, Li Y, Dong M, Liu X. Face mask detection and classification via deep transfer learning. Multimed Tools Appl. 2022;81(3):4475–94.
46. Redmon J, Farhadi A. Yolov3: an incremental improvement. arXiv e-prints. 2018.
47. Bochkovskiy A, Wang CY, Liao HYM. Yolov4: Optimal speed and accuracy of object detection. 2020.
48. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. *arXiv preprint* arXiv:1409.1556.
49. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE; 2016. p. 770–8.