electronics

MDPI

Article

Real-Time Facemask Detection for Preventing COVID-19 Spread Using Transfer Learning Based Deep Neural Network

Mona A. S. Ai [1,2,*], Anitha Shanmugam [3], Suresh Muthusamy [4], Chandrasekaran Viswanathan [5], Hitesh Panchal [6], Mahendran Krishnamoorthy [7], Diaa Salama Abd Elminaam [8,9,*] and Rasha Orban [2]

1 Computer Science Department, College of Computer Science and Information Technology, King Faisal Unversity, Hofuf 31982, Saudi Arabia
2 Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha 12311, Egypt; rasha.abdelkreem@fci.bu.edu.eg (R.O.)
3 Department of Information Technology, Kongu Engineering College (Autonomous), Perundurai 638060, Tamil Nadu, India; anithame@kongu.ac.in
4 Department of Electronics and Communication Engineering, Kongu Engineering College (Autonomous), Perundurai 638060, Tamil Nadu, India; infostosuresh@gmail.com
5 Department of Medical Electronics, Vellalar College of Engineering and Technology, Thindal 638012, Tamil Nadu, India; vc4sachin@gmail.com
6 Department of Mechanical Engineering, Government Engineering College, Gandhinagar 382028, Gujarat, India; engineerhitesh2000@gmail.com
7 Department of Electrical and Electronics Engineering, Jansons Institute of Technology, Karumathampatti, Coimbatore 641659, Tamil Nadu, India; mahae1987@gmail.com
8 Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha 12311, Egypt
9 Computer Science Department, Faculty of Computer Science, Misr International University, Cairo 12585, Egypt
* Correspondence: mona.abdelbaset@fci.bu.edu.eg or m.ali@kfu.edu.sa (M.A.S.A.); diaa.salama@fci.bu.edu.eg (D.S.A.E.)

Citation: Ali, M.A.S.; Shanmugam, A.; Muthusamy, S.; Viswanathan, C.; Panchal, H.; Krishnamoorthy, M.; Elminaam, D.S.A.; Orban, R. Real-Time Facemask Detection for Preventing COVID-19 Spread Using Transfer Learning Based Deep Neural Network. *Electronics* **2022**, *11*, 2250. https://doi.org/10.3390/electronics11142250

Academic Editors: Amir H. Gandomi, Fang Chen, Laith Abualigah, Amir Mosavi and Daniel Morris

Received: 14 April 2022
Accepted: 12 July 2022
Published: 18 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Abstract: The COVID-19 pandemic disrupted people's livelihoods and hindered global trade and transportation. During the COVID-19 pandemic, the World Health Organization mandated that masks be worn to protect against this deadly virus. Protecting one's face with a mask has become the standard. Many public service providers will encourage clients to wear masks properly in the foreseeable future. On the other hand, monitoring the individuals while standing alone in one location is exhausting. This paper offers a solution based on deep learning for identifying masks worn over faces in public places to minimize the coronavirus community transmission. The main contribution of the proposed work is the development of a real-time system for determining whether the person on a webcam is wearing a mask or not. The ensemble method makes it easier to achieve high accuracy and makes considerable strides toward enhancing detection speed. In addition, the implementation of transfer learning on pretrained models and stringent testing on an objective dataset led to the development of a highly dependable and inexpensive solution. The findings provide validity to the application's potential for use in real-world settings, contributing to the reduction in pandemic transmission. Compared to the existing methodologies, the proposed method delivers improved accuracy, specificity, precision, recall, and F-measure performance in three-class outputs. These metrics include accuracy, specificity, precision, and recall. An appropriate balance is kept between the number of necessary parameters and the time needed to conclude the various models.

Keywords: deep learning; facemask; computer vision; CNN; COVID-19

1. Introduction

The research supporting people wearing masks in public locations to prevent COVID-19 transmission is advancing rapidly. Disease spread can be delayed by physically separating sick persons from others, taking additional precautions, and minimizing the probability of transmission per interaction. A mask minimizes transmissibility per encounter in laboratory and clinical settings by limiting the transmission of contaminated respiratory particles. When public mask-wearing is widespread, it effectively reduces virus spread [1]. Recently, researchers at the University of Edinburgh published a study to better understand the COVID-19 pandemic response measures. Wearing a face mask or other covering over the nose and mouth was proven to significantly minimize the risk of coronavirus spread by avoiding the forward distance traveled by an individual's exhaled air [2]. Face mask detection is determining whether or not someone is wearing a mask. In computer vision and pattern recognition, face detection is a crucial component. The face is recognized using various machine learning techniques [3]. The existing systems have several flaws, including high feature complexity and low detection accuracy. Face identification approaches based on deep convolutional neural networks (CNNs) have been popular in increasing detection performance [4]. Even though many academicians have worked hard to develop fast face detection and recognition algorithms, there is a significant difference between 'detection of the face under mask' and 'detection of mask over face'. In practice, it is difficult to spot mask abuse. The key challenge is the dataset limitation. Mask-wearing status datasets are often minimal and merely identify the presence of masks. There is very little study on detecting masks over the face in the literature. The proposed research intends to create a technique that can accurately detect masks over the face in public places (such as airports, train stations, crowded markets, and bus stops) to prevent coronavirus transmission, thus contributing to public health. Furthermore, detecting faces with or without a mask in public is difficult due to the little data available for detecting masks on human faces, making the model difficult to train. As a result, the concept of transfer learning is utilized to transfer learned kernels from networks trained on a large dataset for similar face detection.

In this pandemic situation, there is a need to monitor the people wearing masks to control the spread of COVID-19. It is necessary to alert the people to wear masks properly in public places by comparing the captured image with the datasets. If CCTV cameras record videos, the faces appear small, hazy, and low-resolution. Because people do not always stare straight at the camera, the facial angles change. These real-world videos differ significantly from those obtained by webcams or selfie cameras, making face mask recognition in practice much more difficult. Residual blocks were integrated into the depth-wise separable convolution layer developed by MobileNetV2. Residual networks enable deep network training by creating the network using residual models. Residual network ResNet50V2 is one of the fastest object detection techniques for face detection using CNN. Inception-ResNetV2 is a convolutional neural architecture based on the Inception family of architectures that includes residual connections (which replace the filter concatenation stage in the Inception). Efficient Net examines neural network scaling, thereby simultaneously scaling the network's width, resolution, and depth. The objective was to create a face detector using ResNet50V2 to improve the efficiency of identifying faces with better accuracy and speed. The system's goal was to reduce manual labor by identifying people through video analysis and determining whether or not they are wearing masks. A heuristic evaluation with numerous users was conducted to examine usability, concluding that the implemented system is more user-friendly, faster, and more efficient than existing solutions. A unified approach of ResNet50V2 combines many features and classifiers, where all features can be used to identify the mask through video detection. ResNet50V2 can identify multiple objects in a single frame or image, providing better accuracy since many persons can be in a single frame. ResNet50V2 correctly identifies and warns the user with better accuracy compared toResNet101V2. The object recognition algorithm performs better only when the images are captured, and it takes more time to

process the image and produce the output. Comparatively, ResNet50V2 gives the result by detecting the object in the video stream. The proposed method is faster, more efficient, and more accurate.

The main contributions of the proposed work (techniques and benefits) are as follows:

1. A real-time system was built for determining whether a person on a webcam is wearing a mask or not.
2. A balanced dataset for a facemask with a nearly one-to-one imbalance ratio was generated using random oversampling (ROS)
3. An object detection approach (ensemble) combined a one-stage and two-stage detector to recognize objects from real-time video streams with a short inference time (high speed) and high accuracy.
4. Transfer learning was utilized in ResNet50V2 for fusing high-level semantic information in diverse feature maps by extracting new features from learned characteristics.
5. An improved affine (bounding box) transformation was applied in the cropped region of interest (ROI) as there are many changes in the size of the face and location.

The remainder of the paper is arranged as follows: Section 2 presents the literature review. The proposed methodology is given in Section 3. Section 4 provides the implementation results and discussion. Lastly, the conclusions and future work are specified in Section 5.

2. Literature Review

Transfer learning is an approach in which knowledge acquired by a CNN from provided and related data is used to solve the problem. Deep learning networks pretrained on previous datasets can be fine-tuned to achieve high accuracy with a smaller dataset. The methods which are used for deep learning are discussed below. Sethi et al. [5] proposed a multigranularity masked face recognition model developed using MobileNetV2 and achieved 94% accuracy. Sen et al. [6] built a system that differentiates those who use face masks and those who do not utilizing a series of photographs and videos. The suggested method employed the MobileNetV2 model and Python's PyTorch and OpenCV for mask detection, with 79.24% accuracy. Balaji et al. [7] included an entrance system to public locations that distinguish persons who wear masks from those who do not. Furthermore, if a person violates the rule of wearing a facemask, this device produces a beep as an alert. The video was captured with a Raspberry-PI camera and then converted into pictures for further processing. The usage of masks significantly slow the virus's spread, according to Cheng et al. [8]. It was determined that YOLO v3-tiny (You Only Look Once) can detect mask use in real time. It is also small, fast, and excellent for real-time detection and mobile hardware deployment. Sakshi et al. [9] created a face mask detector based on MobileNetV2 architecture utilizing Keras/TensorFlow. The model was changed to guarantee face mask recognition in real-time video or still pictures. The ultimate goal is to employ computer vision to execute the concept in high-density areas, such as hospitals, healthcare facilities, and educational institutions. Using a featured image pyramid and focus loss, a single-stage object detector can detect dense objects in images over several layers. Jiang et al. [10] proposed a two-stage detector that achieves amazing accuracy and speeds comparable to the single-stage detector. It divides a picture into GxG grids, each providing N-bound box predictions. Each bounding box can only have one class during the prediction, preventing the network from finding smaller items. Redmon et al. [11] introduced YOLO, which uses a one-phase prediction strategy with impressive inference time, but the localization accuracy was low for small images. YOLOv2 with batch normalization, a high-resolution classifier, and anchor boxes were added to the YOLO network.

YOLOv3 is an improved version of YOLOv2, featuring a new feature extraction network, a better backbone classifier, and multiscale prediction. Although Kumar et al. [12]

suggested a two-stage detector with high object detection accuracy, it is limited for video surveillance due to sluggish real-time inference speed. Although Morera et al. [13] suggested YOLOv3, it achieved the same classification accuracy as a single-shot detector (SSD). Furthermore, YOLOv3's inference demands significant CPU resources, making it unsuitable for embedded systems. SSD networks outperform YOLO networks due to their compact filters of convolution type, extensive feature maps, and estimation across manifolds. The YOLO network has two fully linked layers, while the SSD network utilizes varied-sized convolutional layers. The region-based convolutional neural network (R-CNN) presented by Girshick et al. [14] was the first CNN implementation for object detection and localization on a large scale. The model generated state-of-the-art results when tested on standard datasets. R-CNN first extracts a set of item proposals using a selective search strategy and then forecasts items and related classes using an SVM (support vector machine) classifier. He et al. [15] introduced SPPNet, which is a categorization system for gathering features and feeding them into a fully connected layer. SPNN can create feature maps in a single-shot detection for the whole image, resulting in a nearly 20-fold boost in object detection time over R-CNN. Both the detector and the regressor are trained simultaneously without changing the network configurations. Girshick et al. [16] introduced fast R-CNN in which the region of interest (RoI) pooling layer is used to fine-tune the model. Nguyen et al. [17] proposed fast R-CNN, which is an extension of R-CNN and SPPNet. Although fast R-CNN efficiently integrates the properties of R-CNN and SPPNet, its detection speed is still inferior to single-stage detectors. Fu et al. [18] proposed faster R-CNN, which combines fast R-CNN and RPN. It achieves nearly cost-free region proposals by gradually integrating individual components of the object detection system (e.g., proposal detection, feature extraction, and bounding box regression) in a single step. Even though this integration breaks beyond the fast R-CNN speed bottleneck, the subsequent detection stage has computation redundancy. The region-based fully convolutional network (R-FCN) method supports backpropagation for both training and inference, according to Dvornik et al. [19]. Liang et al. [20] introduced the feature pyramid network (FPN) to recognize nonuniform objects; however, academics rarely employ this network due to its high processing costs and memory requirements. He et al. [21] proposed mask R-CNN to improve faster R-CNN by utilizing segmented mask estimates on each RoI. Most existing systems use images to identify the presence or absence of a mask. Fewer algorithms give output with more than a 90% accuracy rate in video streaming. The users are also detected through images, but this works efficiently only when they remain stationary, posing a problem for real-time implementation. Capturing the user's image and then determining the presence/absence of a mask takes more time and is a little more complicated than in video streaming. ResNet50V2 correctly identifies the presence/absence of a mask with better accuracy compared to MobileNetV2. The video analysis method can be used for face mask detection. Of all the approaches proposed in the literature, ResNet50V2 appears to be the most promising face mask detection as it uses a fast and accurate object detection algorithm. The ResNet50V2 approach allows the accuracy of determining mask wearing in a video and identification/extraction of the pixels associated with each individual.

The existing literature study has some limitations, which are summarized as follows:

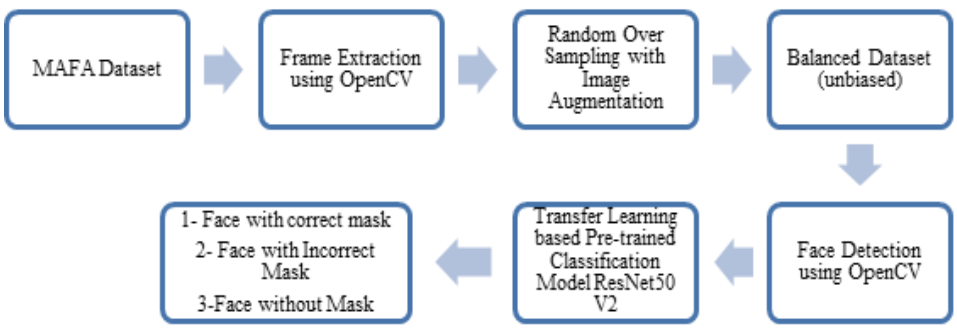
a. Various models have been pretrained on standard datasets, but only a limited number of datasets handle facemask detection to overcome the COVID-19 spread.
b. Due to the limitedness of facemask datasets, varying degrees of occlusion and semantics are essential for numerous mask types.
c. However, none of them are ideal for real-time video surveillance systems.

According to Roy et al., surveillance devices are constrained by a lack of processing power and memory [22]. As a result, these devices necessitate efficient object detection models capable of performing real-time surveillance while using minimal memory and maintaining

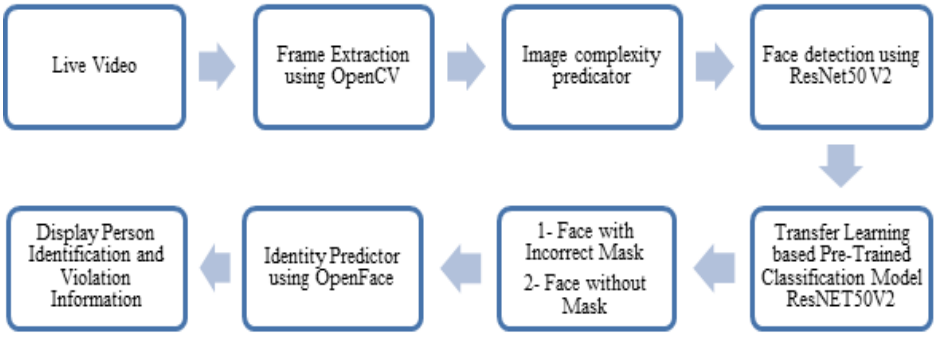
high accuracy. Although one-stage detectors are suitable for video surveillance in many applications, hey have limited accuracy [23]. Two-stage detectors offer accurate detection in case of multifaceted input at the expense of high computing time [24]. The aforementioned factors require creating a combined surveillance device, thereby saving computing time with improved accuracy. To obtain the required image complexity score, each image is represented in a pyramid, extracted features are combined, and L2 normalization is performed. The input samples chosen for analysis should always be balanced, containing the appropriate number of samples belonging to each class label. Handling an imbalanced dataset is important for better classification [25,26]. According to the papers described above, deep learning architectures are rapidly being applied to facemask detection to prevent COVID-19 spread using a transfer learning-based deep neural network [27–34]. Other deep learning [35–42] and optimization algorithms are also used to solve various optimization problems [43–51]. However, several gaps in using deep learning systems for real-time implementation and prevention strategies must be addressed, as indicated [52–57].

3. Proposed Methodology

Figure 1 shows the proposed real-time face mask detection system, which is implemented in two phases (training and deployment). The training and deployment phase algorithms are given in Algorithms 1 and 2, respectively. The training phase consists of 11 steps ranging from dataset collection to image classification (the dataset is classified into three classes: face with the correct mask, face with an incorrect mask, and face without a mask). In the first step, the data frame is extracted using OpenCV, followed by random oversampling to balance the unequal number of classes by performing imbalance computation and random oversampling (ROS) using ϱ. In the third step, image augmentation and face detection are applied by passing through many convolutional layers, which extract feature maps. In the next step, transfer learning is implemented by replacing the last predicting layer of the pretrained model with its predicting layers to implement fine-tuned transfer learning. Finally, in the last step, a pretrained classification model, ResNet50V2, is applied to classify images. The training phase is applied to the MAFA dataset [56] described in the next section. The deployment phase consists of 12 steps ranging from data collection (live video) to displaying personal information (such as identity and name) and violation information (such as location, timestamp, camera type, ID, and violation category, e.g., face without a mask and face with an incorrect mask). In the first step, the data frame is extracted using OpenCV, followed by image complexity prediction to identify whether the image is soft or hard. Object classification using a semi-supervised approach is applied. In the next step, transfer learning is implemented by replacing the last predicting layer of the pretrained model with its predicting layers to implement fine-tuned transfer learning. Finally, identity prediction by applying a pretrained classification model, ResNet50V2, is the last step in classifying images.



(**a**) Training phase



(**b**) Deployment phase

Figure 1. The proposed real-time model for face mask detection.

Algorithm 1. Training phase

Input: MAFA dataset containing videos

Processes: Frame extraction, random oversampling, image augmentation, face detection and transfer learning with pretrained classification

Frame extraction using OpenCV:

Step 1: Split the video captured using cv2.VideoCapture(<path_of_video>) through inbuilt camera into frames and save using cv2.imwrite()

Random over sampling:

Step 2: Unequal number of classes is balanced by performing imbalance computation and ROS using ϱ

$$\rho = \frac{\text{count (majority } (Di))}{\text{count (minority } (Di))},$$

where D_{mi} and D_{ma} are majority and minority classes of D,

Image augmentation and Face detection:

Step 3: At various locations, the image is passed through a large number of convolutional layers, which extract feature maps.

Step 4: In each of those feature maps, a 4 × 4 filter is used to determine a tiny low default box and predict each box's bounding box offset.

Step 5: Five predictions are included in each bounding box output: x, y, w, h, and confidence. The centroid of the box is represented by x and y in relation to the grid cell limits

Step 6: Conditional class probabilities are also predicted in each grid cell,Pr (class | object)

Step 7: The truth boxes are matched with the expected boxes using intersection over union (IOU),

$$\text{Intersection over union } (IOU) = \frac{\text{area of overlap}}{\text{area of union}}.$$

Transfer Learning:

Step 8: Replace last predicting layer of the pretrained model with its own predicting layers to implement fine-tuned transfer learning.

Step 9: Generic features are learnt by the network's initial lower layers from the pretrained Model, and its weights are frozen and not modified during the training.

Step 10: Task-specific traits are learned at higher layers which can be pretrained and fine-tuned.

Pre-trained classification:

Step 11: Apply pretrained classification model, ResNET50V2 to classify images.

Output: Images in the dataset are classified into three classes: face with correct mask, face with incorrect mask, and face without mask.

Algorithm 2. Deployment phase

Input:	Live video
Processes:	Frame extraction, image complexity predictor, transfer learning with pretrained classification and identity prediction
Frame extraction:	
Step 1:	Split the video captured using cv2.VideoCapture(<path_of_video>) through inbuilt camera into frames and save using cv2.imwrite()
Step 2:	Face detection using MobileNetV2 and ResNET50V2 is performed by comparing with trained images
Step 3:	During testing, class-specific confidence scores are obtained using $\Pr(\text{Class} \mid \text{Object}) \times \Pr(\text{Object}) \times [0U = \Pr(\text{Class}) \times 10U.$
Step 4:	The truth boxes are matched with the expected boxes using IOU, Intersection over union $= \frac{\text{area of overlap}}{\text{area of union}}$.
Image Complexity Prediction:	
Step 5:	To identify whether the image is soft or hard, object classification using semi-supervised approach is applied.
Step 6:	For predicting the class of soft pictures, the MobileNet-SSD model is used, $L = 1/N\ (\text{Lclass} + \text{Lbox}),$ where N is the total number of matched boxes with the final set of matched boxes, Lbox is the L1 smooth loss indicating the error of matched boxes, and Lclass is the softmax loss for classification.
Step 7:	For predicting challenging pictures, a faster RCNN based on ResNet50V2 is used.
Transfer Learning:	
Step 8:	Replace last predicting layer of the pretrained model with its own predicting layers to implement fine-tuned transfer learning.
Step 9:	Generic features are learnt by the network's initial lower layers from the pretrained Model, and its weights are frozen and not modified during the training.
Step 10:	Task-specific traits are learned at higher layers which can be pre-trained and fine-tuned.
Pre-trained classification:	
Step 11:	By applying pretrained classification model, ResNet50V2, images are classified into face with mask, face with no mask, and face with incorrect mask.
Identity prediction:	
Step 12:	OpenFace is applied to detect the face is with or without mask. Affine transformation is applied to detect the non-mask faces.
Output:	Display the personal information such as identity and name and violation information (such as location, timestamp, camera type, ID, and violation category, e.g., face without mask and face with incorrect mask).

In the proposed model for face mask detection, a simple and user-friendly system brings comfort to users. It uses a web camera as its hardware requirement and processes the video captured. The web camera can be placed where the shop's entrance, hotels, offices, etc., are visible so that a face mask can be easily detected. In the proposed methodology, the video is processed using transfer learning and an efficient deep learning method for detecting the face mask. The proposed solution is applied to the face mask dataset, i.e., MAFA (benchmark dataset), to check the efficiency of the proposed solution. The dataset name is face mask data consisting of 15,000 images with 7500 people wearing masks and 7500 images with people not wearing masks (Samples of the pictures from the dataset can be found in Figure 2, obtained from MAFA https://www.kaggle.com/datasets/revanthrex/mafadataset (accessed on 1 November 2021)).

(**a**) Samples of people wearing masks from the MAFA dataset

(**b**) Samples of people without masks from the MAFA dataset

Figure 2. Face mask dataset, samples for each class from the MAFA dataset.

Learning more features using learning algorithms is difficult due to the face mask dataset's small size and various image complexities. Transfer learning based on deep learning is used to pass knowledge learned from a source task to a related target task. It is also used to train the network more quickly, reliably, and cost-effectively. The proposed work generally consists of preprocessing with an image complexity predictor, pretrained classifier, and identity predictor.

The image is scaled, and the dataset is unbiased in the first step. Then, the image is divided into soft and hard types according to complexity. For predicting the class of soft pictures, the MobileNet-SSD model is used. For predicting challenging images, a faster RCNN based on ResNet50V2 is used. In the second step, transfer learning is applied. The classifier model classifies into three classes: face detection with a correct mask, face detection with an incorrect mask, and face detection with no mask. Personal identification and violation information are then displayed for further action. A full description of the three phases of the planned architecture is provided below.

3.1. Dataset Characteristics

This paper conducted experiments using the medical face mask dataset, i.e., MAFA [56], published by Shiming Ge. The MAFA dataset consists of 35,803 masked faces with a minimum size of 32 × 32. The faces in this dataset have a different orientation and occlusion degree. We selected 15,000 images that contained frontal faces from MAFA. The dataset was divided into three parts for training, validation, and testing with 11,000, 2000, and 2000 images, respectively. Figure 2 shows sample images from the MAFA dataset. The data are presented in Figure 2, adapted from [58].

3.2. Image Prepocessing and Face Detection

Bias denotes a dataset with an unequal number of classes. This bias was balanced, and frames were extracted from the videos and resized to 128 × 128 pixels. Data augmentation is a widely used approach for getting the most out of a data source. In CNN, the initial layers are in charge of extracting generic visual elements such as edges and textures. The subsequent layers look for more specific qualities on the basis of the preceding attributes. This procedure is applied for numerous layers until high-value semantic traits can be detected, such as detecting eyes or noses. Finally, the categorization is carried out using a traditional neural network. Variety of the training set can be obtained from changes made to the photos such as rotations, translations, or zooming.

The transfer learning approach is preferable in the case of limited samples available in the training set. Then, the dataset can be increased to a large size by performing a different arrangement of faces on a template. However, this cannot be meticulously followed in real time. Hence, face detection is achieved by removing the image boundaries with no useful information. For this purpose, an effective approach called rapid object detection with a boosted cascade of simple features is utilized.

OpenCV considers a snapshot of a live video in a given location and converts it into frames. The facial photos are extracted and used to distinguish the person not wearing a mask on their face. Face features are extracted from photos using the ResNet50V2 model, and these features are subsequently learned using many hidden layers. An alarm sound will be played to a person without a mask whenever the model recognizes someone without a mask. The various steps in image augmentation for classification are illustrated in Figure 3. The data are presented in Figure 3, adapted from [58].

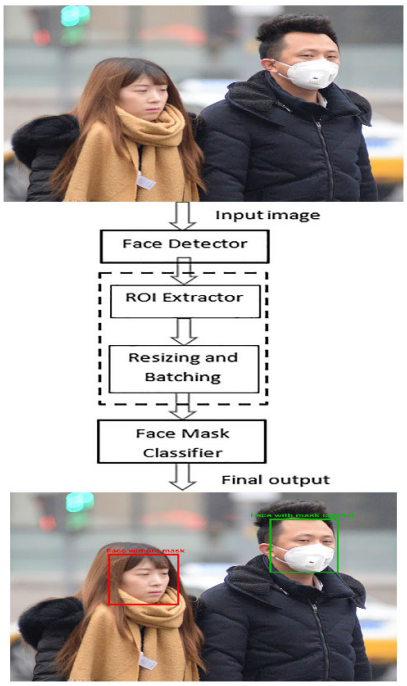


Figure 3. Steps in image augmentation for classification.

3.3. Image Complexity Prediction

The image complexity predictor uses a semi-supervised object classification strategy to split the data into soft or hard images. This strategy is preferred when limited labeled and unlabeled data are present in the dataset. Then, the soft images are processed using a

single-stage detector, and hard images are processed using the two-stage detector. Compared with trained images, the predictor optimizes face detection accuracy and computational time while detecting faces using MobileNetV2 and ResNet50V2. A fully connected layer follows two convolutional layers with 512 neurons and a readout layer with 10 neurons in the curriculum learning network. This helps in learning harder images step by step and is included in class labels, producing high accuracy even for small networks. The algorithm for the image complexity predictor is shown in Algorithm 3.

Algorithm 3. Image complexity predictor	
Input:	Images from the MAFA dataset containing videos
Processes:	Single- and two-stage detectors
Step 1:	Split soft images (very few people in an image) and hard images (group of people in different poses and locations with background).
Step 2:	Apply a single-stage detector to process soft images.
Step 3:	Apply a two-stage detector to process hard images.
Output:	Set of region proposals (R denotes image center position with height and width, and G denotes bounding box around the image)

3.4. Transfer Learning-Based Pretrained Classification Model

Transfer learning was founded on developing learning by transferring information from a previously learned task to a new task. There are two stages in transfer learning. In the initial feature extraction phase, only the classification layers are trained; however, in the second phase of fine-tuning, all layers in the system are retrained. Transfer learning using MobileNetV2, VGG19, ResNet50V2, AlexNet, and InceptionV3 is shown in Figure 4. The data are presented in Figure 4, adapted from [58].

AlexNet has five convolutional layers and three fully connected layers. It employs ReLu, which is much quicker than sigmoid, and adds a dropout layer after each FC layer to reduce overfitting. By substituting large kernel-sized filters (11 and five in the first and second convolutional layers, respectively) with multiple 3 × 3 kernel-sized filters one after the other, VGG19 outperforms AlexNet. The activations in Inception are sparsely connected, which means that not all 512 output channels are connected to all 512 input channels.

ResNet50V2 was fine-tuned with five additional layers: a 5 × 5 average pooling layer, a flattening layer, and a dense rectified linear unit (ReLU) layer with 128 neurons and 0.5 dropouts of an output layer consisting of softmax function as the activation function.

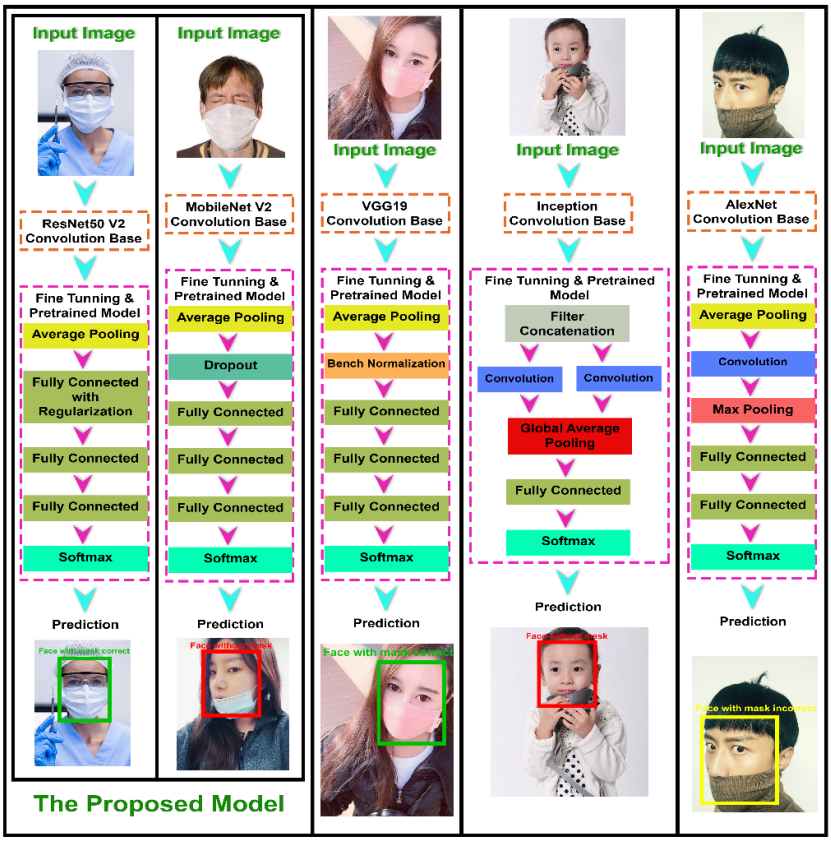


Figure 4. Transfer learning using MobileNetV2, VGG19, ResNet50V2, AlexNet, and InceptionV3.

3.5. Identity Prediction

The identity predictor is applied using OpenFace 0.20 for detecting faces with or without a mask. For processing non-masked faces, an affine transformation is applied, as shown in Figure 5. The data are presented in Figure 3, adapted from [56]. The pair (R, G) represents the region proposal where $R = (Rx, Ry, Rw, Rh)$ represents the pixel coordinates of the center of proposals along with width and height, and $G = (Gx, Gy, Gw, Gh)$ represents coordinates of each ground-truth bounding box. After applying the affine transformation, bounding box regression is applied for moving information from the region proposal (R) to bounding box (G), representing ground truth with no loss of information. Then, a scale-invariant transformation is performed on pixel coordinates of R, and then log space transformation is applied on the R's width and height. $Tx(R)$, $Ty(R)$, $Tw(R)$, and $Th(R)$ represent the corresponding four transformations, and coordinates of the ground-truth box are obtained using Equations (1)–(4).

$$G_x = T_x(R_x) + R_x, \tag{1}$$

$$G_y = T_y(R_y) + R_y, \tag{2}$$

$$G_w = T_w(R_w) + R_w, \tag{3}$$

$$G_h = T_h(R_h) + R_h, \tag{4}$$

where $f_6(R)$ represents the linear function of Pool_6 feature of R. The equation for $T_i(R)$ is shown in Equation (5).

$$T_i(R) = w_i f_6(R), \tag{5}$$

where w_i represents the weight learned by optimizing the regression, as given by Equation (6).

$$w_i = \sum_{n \in R} \left(t_i^n - \hat{w} f_6(R^n)\right)^2 + \lambda |\hat{w_i}|^2, \tag{6}$$

where t_i represents the regression target which is related to coordinates, width, and height of region proposal pair (R, G) as denoted in Equations (7)–(10), respectively.

$$t_x = \frac{G_x - R_x}{R_w}. \tag{7}$$

$$t_y = \frac{G_y - R_y}{R_h}. \tag{8}$$

$$t_w = \log\left(\frac{G_x}{R_w}\right). \tag{9}$$

$$t_h = \log\left(\frac{G_h}{R_h}\right). \tag{10}$$

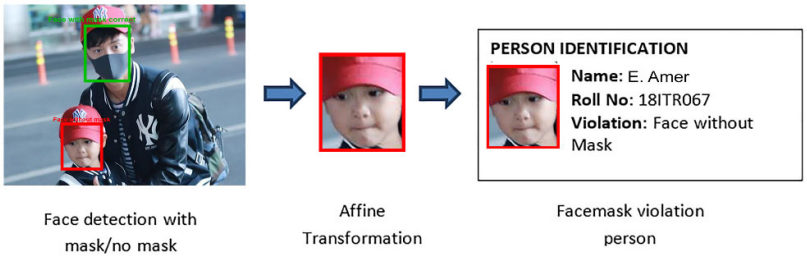


Figure 5. Localizing a face without a mask using affine transformation.

3.6. Loss Function and Optimization

In classification theory, the loss function and objective function transfer estimated distributions onto true distributions. This function's output should be minimized using an optimization method. Classification and regression losses are used in the single-shot detector. The confidence level in the predictions of each bounding box produced by the network is measured by the classification loss. Categorical cross-entropy is used to calculate this loss, as given by Equation (11).

$$Loss = \sum t(x) \log\left(e(x)\right), \tag{11}$$

where $t(x)$ and $e(x)$ represent true and estimated distributions over categorical variables, respectively. The regression loss is the difference between the network's predicted bounding boxes and the ground-truth bounding box.

3.7. Control of Overfitting

Even though data augmentation and an unbiased dataset are used, the model must be generalized to fit any pattern in the input and respond with appropriate results. When

the time to train the model increases, the model becomes overfitted. To reduce the overfitting problem to a negligible value, the feature selection process is performed with a better optimizer.

3.8. Evaluation Parameters

The standard evaluation parameters such as accuracy, recall (sensitivity), precision, F-measure, and specificity are calculated on the basis of the number of true positives (TP), the number of true negatives (TN), the number of false negatives (FN), and the number of false positives (FP) as given in Equations (12)–(16).

$$\text{Accuracy } = \frac{(\boldsymbol{TN} + \boldsymbol{TP})}{(\boldsymbol{TN} + \boldsymbol{TP} + \boldsymbol{FN} + \boldsymbol{FP})}. \quad (12)$$

$$\text{Precision } = \frac{\boldsymbol{TP}}{(\boldsymbol{TP} + \boldsymbol{FP})}. \quad (13)$$

$$\text{Recall } = \frac{\boldsymbol{TP}}{(\boldsymbol{TP} + \boldsymbol{FN})}. \quad (14)$$

$$\boldsymbol{F} - \text{ measure } = \frac{(\mathbf{2} \times \text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}. \quad (15)$$

$$\text{Specificity } = \frac{\boldsymbol{TN}}{(\boldsymbol{TN} + \boldsymbol{FP})}. \quad (16)$$

The TP, TN, FN, and FP values are calculated using the confusion matrix in Table 1, and the corresponding formulas are given below.

Table 1. Confusion matrix.

		Predicted Values		
		Face Wearing Mask Correctly	**Face Wearing Mask Incorrectly**	**Face Wearing No Mask**
Actual values	**Face wearing mask correctly**	a	b	c
	Face wearing mask incorrectly	d	e	f
	Face wearing no mask	g	h	i

- Face wearing mask correctly:

$$\boldsymbol{TP} = \boldsymbol{a}.$$

$$\boldsymbol{FN} = \boldsymbol{b} + \boldsymbol{c}.$$

$$\boldsymbol{FP} = \boldsymbol{d} + \boldsymbol{g}.$$

$$\boldsymbol{TN} = \boldsymbol{e} + \boldsymbol{f} + \boldsymbol{h} + \boldsymbol{i}.$$

- Face wearing mask incorrectly:

$$\boldsymbol{TP} = \boldsymbol{e}.$$

$$\boldsymbol{FN} = \boldsymbol{d} + \boldsymbol{f}.$$

$$\boldsymbol{FP} = \boldsymbol{b} + \boldsymbol{h}.$$

$$FP = b + h.$$

- Face wearing no mask:

$$TP = i.$$

$$FN = g + h.$$

$$FP = c + f.$$

$$TN = a + b + d + e.$$

Inference time represents the time spent reading from the input image to the final class prediction.

The number of trainable and nontrainable parameters present in different layers affects the capability of prediction, the complexity of the model, and the amount of memory required. The confusion matrix values combining all comparison models are depicted in Table 2.

Table 2. Confusion matrix results for all comparison models.

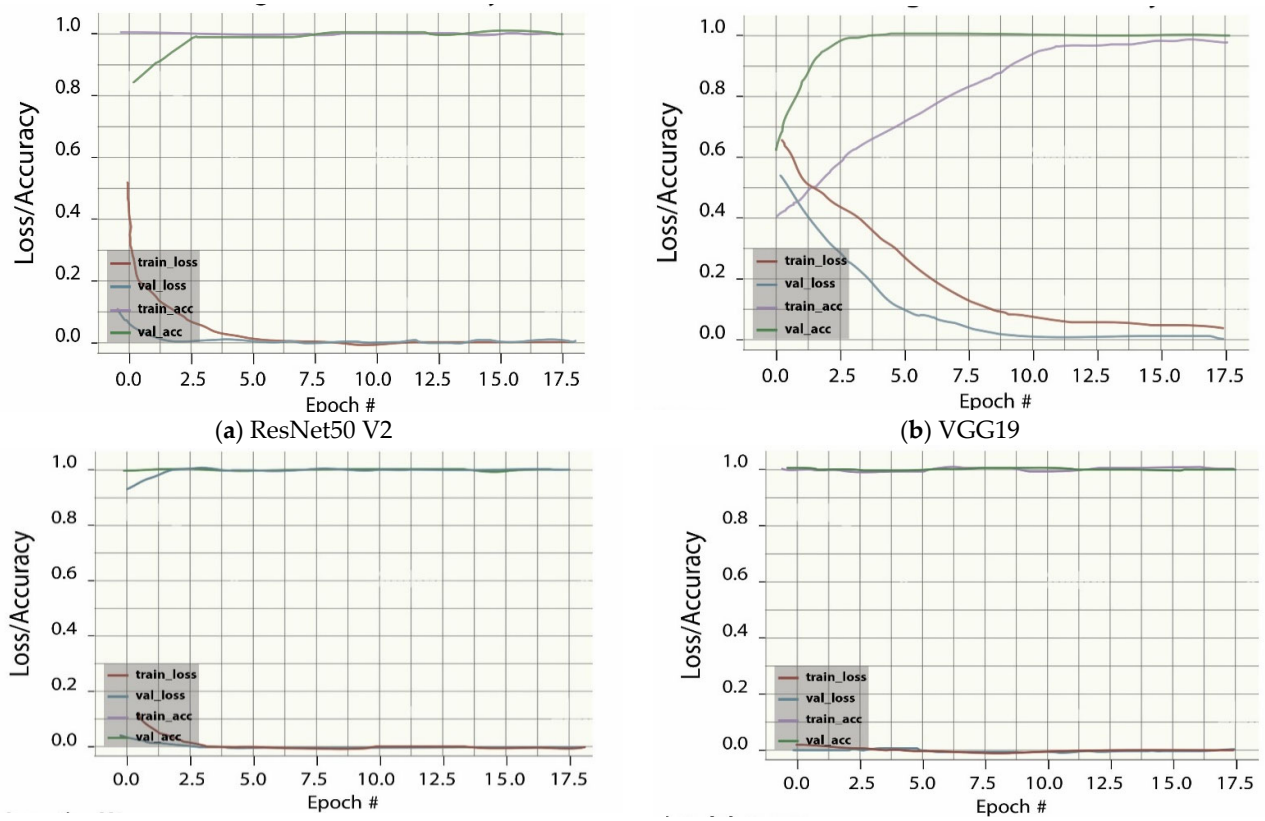
		Predicted Value		
		face wearing mask correctly	face wearing mask incorrectly	face wearing no mask
Actual Values	face wearing mask correctly	0.94	0.058	0.002
	face wearing mask incorrectly	0.079	0.9	0.021
	face wearing no mask	0.005	0.00497	0.99

4. Results and Discussion

The proposed algorithm was implemented in Python, and the integrated development environment was PyCharm, a popular Python IDE created by JetBrains to conduct Python language programming. The experiment was set up by loading different pretrained models using the Charm package (https://github.com/JetBrains/awesome-pycharm (accessed on 11 July 2022)) run on an Intel(R) Core i7 2.80 GHz CPU with 8 GB RAM and the Windows 10 operating system. The proposed system uses MaskedFace-Net (MFN) and Flickr-Faces-HQ Dataset (FFHQ) with 67,193 pictures of faces with a correct mask, 66,899 pictures of a face with an incorrect mask, and 66,535 pictures of a face with no mask. Figure 6a–d exhibit the accuracy of face mask detection for various models.

After numerous experiments, it was discovered that translation and zoom operations did not affect the results. Finally, the training dataset was rotated and flipped horizontally randomly in the range of [−5°, +5°].

The proposed technique with a multilayer convolutional network for recognizing objects from color images achieves larger mAP, more frames to increase speed, and acceptable accuracy. The training accuracy increases sharply and becomes stable at epoch 2 due to a balanced dataset in ResNet50V2, as depicted in Figure 6a. The training and validation accuracy differs at the maximum by 25% at epoch 2.5 in VGG19, as shown in Figure 6b. The training and validation accuracy differs at the initial epoch and becomes stable at epoch 1 in InceptionV3, as shown in Figure 6c. The training and validation accuracy is the same at every epoch of the execution in MobileNetV2, as shown in Figure 6d.



(**a**) ResNet50 V2 (**b**) VGG19

(**c**) Inception V3 (**d**) Mobile Net V2

Figure 6. Accuracy and loss of face mask detection in different models.

The accuracy of the ResNet50V2 model was higher when compared to other models, namely, VGG19, MobileNetV2, and InceptionV3, by 2.92, 0.44, and 3.38, and by 9.49, 1.31, and 4.9 when using the biased dataset in both training and validation phases, respectively, as depicted in Figure 7.

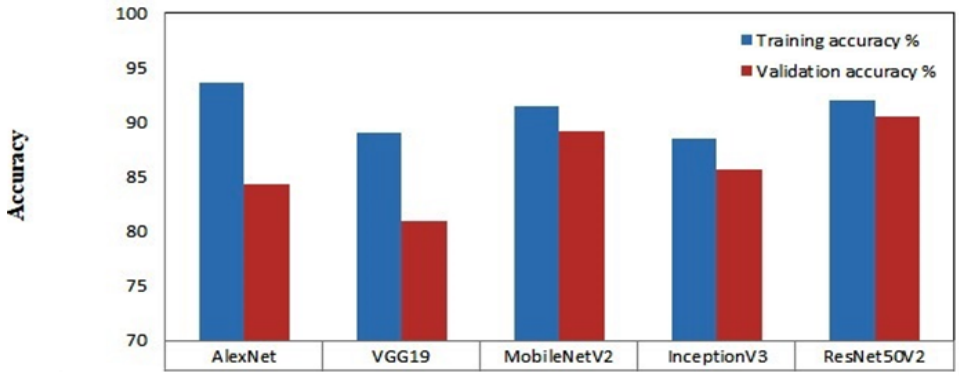


Figure 7. Training and validation accuracy in face mask detection.

The input picture size was 840 × 840 with a batch size of 2 in strong ResNet50V2 backbone, while the input image size is 640 × 640 with a batch size of 32 in light MobileNetV2backbone. Although multiple network components can improve detection performance, the ResNet50V2 backbone achieved the greatest improvement in several parameters with the balanced dataset. ResNet50V2 showed an improvement in parameters precision, recall, F-measure, accuracy, and specificity by 1.22, 3.21, 0.43, and 2.22 and by 0.17, 2.13, 0.16, and 1.08, respectively when compared to AlexNet and MobileNetV2 (see Figure 8).

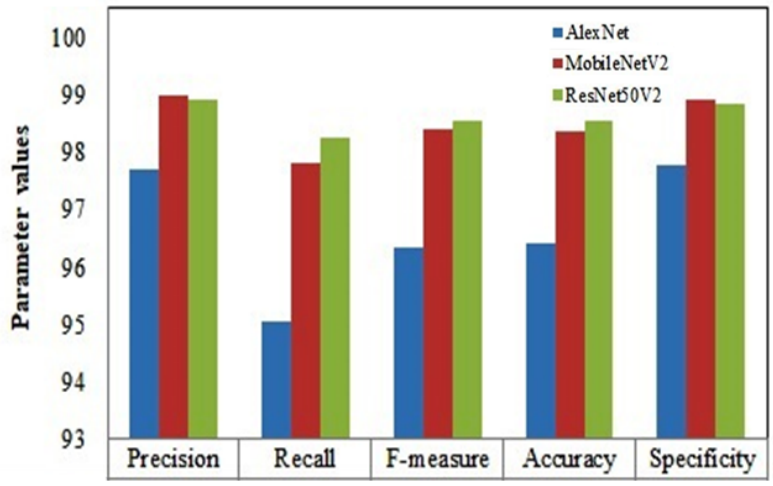


Figure 8. Performance of various parameters in the face mask detection.

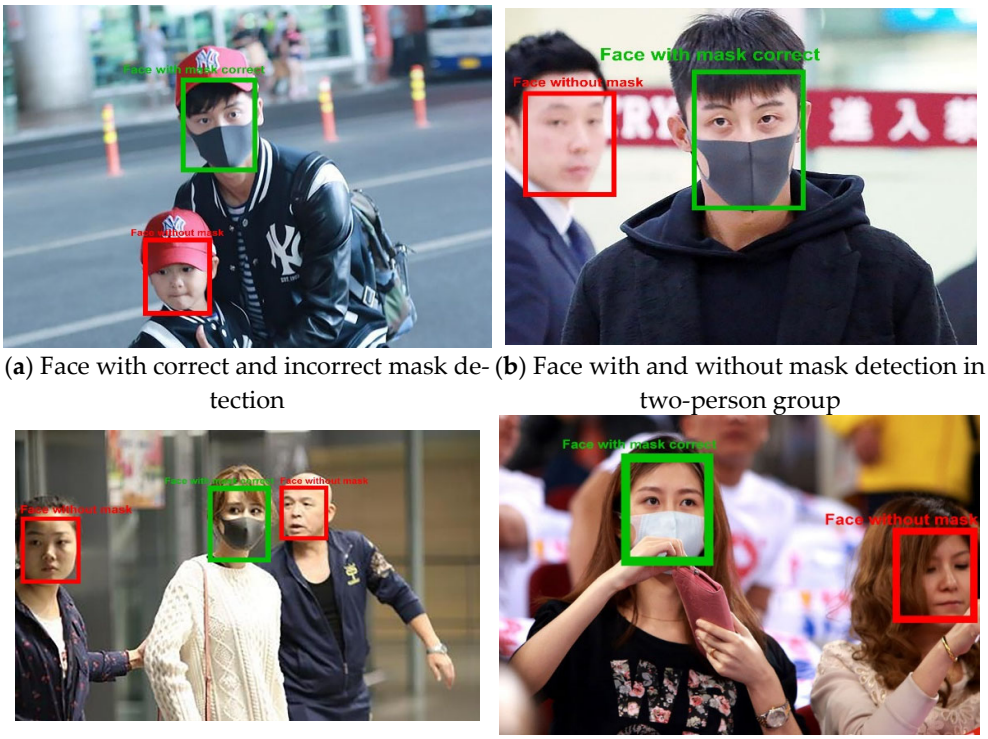
The total number of trainable and nontrainable parameters in ResNet50V2 was 24 million; hence, high computation cost was incurred, whereas, in MobileNetV2, VGG19, and InceptionV3, is the numbers were 2.4 million, 20 million, and 22 million, respectively. MobileNetV2 uses Dropout in the fully connected network, such that the number of parameters in the network is greatly reduced compared to ResNet50V2. InceptionV3 uses filter concatenation in the convolution layer rather than the connected layer to reduce the number of parameters. As VGG19 uses batch normalization in the connected layer, the number of parameters is reduced. MobileNetV2 extracts features using depth-wise convolutions and adjusts channel numbers using channel-wise convolutions; hence, the computational cost of MobileNetV2 is substantially lower than networks utilizing traditional convolutions. Thus, MobileNetV2 performs well in real-time object detection using surveillance devices with less memory consumption.

ResNet50V2 achieved 97% when averaging precision, recall, and F1-score values as the bias was removed among the three classes of facemask detection. Compared to other models, MobileNetV2, InceptionV3, and VGG19, ResNet50V2 led to improvements by 0.5%, 6.5%, and 3%, as shown in Table 3. ResNet50V2's inference time was reduced by 4 ms, while InceptionV3, VGG19, and MobileNetV2's inference times were reduced by 2 ms. The integrated system used MobileNet-SSD for face detection and ResNet50V2 as the backbone for facemask classification; hence, high accuracy with decreased inference time was obtained. A deep neural network is costly since it demands substantial computing power and requires time to train huge datasets. Deep-learning-based transfer learning is utilized to pretrain the ResNet50V2 network faster and cost-effectively by deleting the batch normalization layer in the network at test time. As faster R-CNN uses RPN, it is more efficient in generating ROI and runs at 10 ms per image. The parameter values for different models are denoted in Table 3.

Table 3. Different models and their parameter values.

Models	Total Number of Trainable and Nontrainable Parameters	Precision	Recall	F1-score	Inference Time (ms)	Training Accuracy	Validation Accuracy
MobileNetV2	2,422,339	0.97	0.96	0.97	15	91.49	89.18
VGG19	20,090,435	0.94	0.94	0.94	11	89.01	81
InceptionV3	22,065,443	0.91	0.9	0.91	9	88.55	85.59
ResNet50V2	**23,827,459**	**0.97**	**0.97**	**0.97**	**7**	**91.93**	**90.49**

Figure 9a–d show various screenshots illustrating facemask detection with three-class outputs.



(**a**) Face with correct and incorrect mask detection

(**b**) Face with and without mask detection in two-person group

(**c**) Face with and without mask detection in three-person group

(**d**) Face with and without mask detection in three-person group

Figure 9. Screenshots of three-class face mask detection.

The four pretrained models, ResNet50V2, MobileNetV2, InceptionV3, and VGG19, were compared for performance analysis. ResNet50V2 was the optimal model in terms of inference time, error rate, detection speed, and memory usage among the compared models. Figure 9 shows screenshots of three-class face mask detection. The data are presented in Figure 9, adapted from [58].

5. Comparison with Related Works and Discussion

Developing a system that can be easily embedded in a device that can be utilized in public spaces to aid in the prevention of COVID-19 transmission requires more accurate face detection, precise localization of the individual's identity, and avoidance of overfitting. To further demonstrate the quality of the proposed model, we compared AlexNet, MobileNet, and YOLO baseline models in terms of accuracy, precision, and recall for detecting human images with and without a mask. Table 4 summarizes the outcomes of this comparison, demonstrating that the proposed system outperformed other models in terms of accuracy, precision, and recall. Experiments demonstrated that the proposed system can accurately detect faces and masks while consuming less inference time and memory than previously developed methods. To address the data imbalance issue identified in the previously published dataset, efforts were made to create an entirely new, unbiased dataset that is well suited for mask detection tasks related to COVID-19, among other applications.

Table 4. Performance comparison of different methods in terms of accuracy (AC) and average precision (ap).

Reference	Methodology	Classification	Detection	Result
(Ejaz et al., 2019) [59]	PCA	Yes	No	AC = 70%
(Ud Din et al., 2020) [60]	GAN	Yes	Yes	-
Proposed	Improved ResNetV2	Yes	Yes	AC = 91.93% AP = 97% Recall = 97% F-Score = 97%

6. Improvement in Accuracy Using MAFA Dataset in the Proposed Method

- In the original imbalanced MAFA dataset, random oversampling was applied to obtain the balanced dataset, and image augmentation was also performed to improve the accuracy.
- In a large dataset ranging from simple to complex images, transfer learning was applied to pretrain the parameters, especially for small objects, which also improved the accuracy of the model.
- In the fine-tuning phase of transfer learning, all the layers in the system were pretrained. Accordingly, the optimum value for each parameter was obtained, which was used to improve the accuracy of the model.
- Using MobileNetV2 for face detection and ResNet50V2 for mask detection improved the accuracy of the system.
- Two-stage detection for classifying images according to groups, distant views, and occlusions also improved the accuracy but at the cost of computational time.
- The batch normalization layer in ResNet50V2 for pretraining the parameters and L2 normalization to accurately predict the image complexity were used to improve the accuracy of the system.

7. Conclusions

The proposed work presented a deep-learning-based solution for identifying masks over faces in public locations to reduce coronavirus community spread. The ensemble approach aids in reaching high accuracy, but it also significantly improves detection speed. Furthermore, transfer learning on pretrained models and rigorous testing on an unbiased dataset resulted in a reliable and low-cost solution. The findings support this application's viability in real-world scenarios, thus helping to prevent pandemic spread. Compared with existing approaches, the proposed method achieved better performance in terms of accuracy, specificity, precision, recall, and F-measure in three-class outputs. A proper tradeoff was maintained between several required parameters and inference time using different models. To improve the proposed method's performance, different datasets with different sizes can be used for medical face masking detection. Furthermore, data can be pretrained using new deep learning methods, which can result in a huge number of features in the datasets. Hence, to improve accuracy, we can apply a new metaheuristic algorithm for solving the image classification problem based on feature selection. Moreover, to improve the accuracy of the new metaheuristic algorithm, we can use hybrid algorithms or a different operator to enhance the exploitation stage, such as random opposition-based learning (ROBL) and opposition-based learning (OBL) to prevent local optima and accelerate the convergence. Future work can be expanded to include other mask-wearing issues to improve accuracy. The developed model can be implemented using surveillance devices for biometric applications, especially in polluted industries with facial landmark detection and face masks.

Author Contributions: Conceptualization, M.A.S.A., A.S., S.M., C.V., H.P., M.K., R.O. and D.S.A.E.; Data curation M.A.S.A., A.S., S.M., C.V., H.P., M.K., R.O. and D.S.A.E.; Formal analysis, M.A.S.A., A.S., S.M., C.V., H.P., M.K., R.O. and D.S.A.E.; Funding, M.A.S.A. acquisition; Methodology, M.A.S.A., A.S., S.M., C.V., Hitesh Panchal Investigation, M.K., R.O. and D.S.A.E.; Supervision; M.A.S.A., A.S., S.M., C.V., H.P., M.K., R.O. and D.S.A.E.; Software, M.A.S.A., A.S., S.M., C.V., H.P., M.K., R.O. and D.S.A.E.; Validation, M.A.S.A., A.S., S.M., C.V., H.P., M.K., R.O. and D.S.A.E.; Writing—original draft; Writing—review & editing M.A.S.A., A.S., S.M., C.V., H.P., M.K., R.O. and D.S.A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Scientific Research, King Faisal University, grant number GRANT207, and the APC was funded by the Deanship of Scientific Research, King Faisal University.

Acknowledgments: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Project No. GRANT207).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviation

CNN	Convolutional neural network
ResNet	Residual network
ROI	Region of interest
YOLO	You only look once
R-CNN	Region-based convolutional neural network
ROS	Random oversampling
IOU	Intersection over union
MFN	MaskedFace-Net
FFHQ	Flickr-Faces-HQ Dataset
ROBL	Random opposition-based learning
OBL	Opposition-based learning

References

1. Howard, J.; Huang, A.; Li, Z.; Tufekci, Z.; Zdimal, V.; van der Westhuizen, H.-M.; von Delft, A.; Price, A.; Fridman, L.; Tang, L.-H.; et al. An evidence review of face masks against COVID-19. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2014564118. https://doi.org/10.1073/pnas.2014564118.
2. Godoy, L.R.G.; Jones, A.E.; Anderson, T.N.; Fisher, C.L.; Seeley, K.M.; Beeson, E.A.; Zane, H.K.; Peterson, J.W.; Sullivan, P.D. Facial protection for healthcare workers during pandemics: A scoping review. *BMJ Glob. Health* **2020**, *5*, e002553.
3. Nanni, L.; Ghidoni, S.; Brahnam, S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognit.* **2017**, *71*, 158–172. https://doi.org/10.1016/j.patcog.2017.05.025.
4. Erhan, D.; Szegedy, C.; Toshev, A.; Anguelov, D. Scalable Object Detection using Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2147–2154.
5. Sethi, S.; Kathuria, M.; Kaushik, T. Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread. *J. Biomed. Inform.* **2021**, *120*, 103848. https://doi.org/10.1016/j.jbi.2021.103848.
6. Sen, S.; Sawant, K. Face mask detection for COVID-19 pandemic using pytorch in deep learning. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1070*, 012061. https://doi.org/10.1088/1757-899x/1070/1/012061.
7. Balaji, S.; Balamurugan, B.; Kumar, T.A.; Rajmohan, R.; Kumar, P.P. A Brief Survey on AI Based Face Mask Detection System for Public Places. *Ir. Interdiscip. J. Sci. Res.* **2021**, *5*, 108–117.
8. Cheng, G.; Li, S.; Zhang, Y.; Zhou, R. A Mask Detection System Based on Yolov3-Tiny. *Front. Soc. Sci. Technol.* **2020**, *2*, 33–41.
9. Sakshi, S.; Gupta, A.K.; Yadav, S.S.; Kumar, U. Face Mask Detection System using CNN. In Proceedings of the 2021 IEEE International Conference on Advanced Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 4–5 March 2021; pp. 212–216.
10. Jiang, M.; Fan, X.; Yan, H. RetinaMask: A Face Mask Detector. 2020. Available online: http://arxiv.org/abs/2005.03950 (accessed on 5 April 2021).
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 2016, pp. 779–788.
12. Kumar, Z.; Zhang, J.; Lyu, H. Object detection in real time based on improved single shot multi-box detector algorithm. *EURASIP J. Wirel. Commun. Netw.* **2020**, *1*, 1–18
13. Morera, Á.; Sánchez, Á.; Moreno, A.B.; Sappa, Á.D.; Vélez, J.F. SSD vs. YOLO for Detection of Outdoor Urban Advertising Panels under Multiple Variabilities. *Sensors* **2020**, *20*, 4587. https://doi.org/10.3390/s20164587.
14. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. https://doi.org/10.1109/tpami.2015.2437384.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916.
16. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
17. Nguyen, N.-D.; Do, T.; Ngo, T.D.; Le, D.-D. An Evaluation of Deep Learning Methods for Small Object Detection. *J. Electr. Comput. Eng.* **2020**, *2020*, 3189691. https://doi.org/10.1155/2020/3189691.
18. Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:1701.06659.
19. Dvornik, N.; Shmelkov, K.; Mairal, J.; Schmid, C. BlitzNet: A Real-Time Deep Network for Scene Understanding. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
20. Liang, Z.; Shao, J.; Zhang, D.; Gao, L. Small Object Detection Using Deep Feature Pyramid Networks. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2018; Volume 11166, pp. 554–564. https://doi.org/10.1007/978-3-030-00764-5_51.
21. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision Venice, Italy, 22–29 October 2017; pp. 2980–2988.
22. Roy, B.; Nandy, S.; Ghosh, D.; Dutta, D.; Biswas, P.; Das, T. MOXA: A Deep Learning Based Unmanned Approach For Real-Time Monitoring of People Wearing Medical Masks. *Trans. Indian Natl. Acad. Eng.* **2020**, *5*, 509–518. https://doi.org/10.1007/s41403-020-00157-z.
23. Ionescu, R.T.; Alexe, B.; Leordeanu, M.; Popescu, M.; Papadopoulos, D.P.; Ferrari, V. How hard can it be? Estimating the difficulty of visual search in an image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2157–2166.
24. Soviany, P.; Ionescu, R.T. Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction. In Proceedings of the 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 20–23 September 2018. https://doi.org/10.1109/synasc.2018.00041.
25. Devi Priya, R.; Sivaraj, R.; Anitha, N.; Devisurya, V. Forward feature extraction from imbalanced microarray datasets using wrapper based incremental genetic algorithm. *Int. J. Bio-Inspired Comput.* **2020**, *16*, 171–180.
26. Devi Priya, R.; Sivaraj, R.; Anitha, N.; Rajadevi, R.; Devisurya, V. Variable population sized PSO for highly imbalanced dataset classification. *Comput. Intell.* **2021**, *37*, 873–890.
27. Chen, K. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
28. Goyal, H.; Sidana, K.; Singh, C. A real time face mask detection system using convolutional neural network. *Multimed. Tools Appl.* **2022**, *81*, 14999–15015.

29. Farman, H.; Khan, T.; Khan, Z.; Habib, S.; Islam, M.; Ammar, A. Real-Time Face Mask Detection to Ensure COVID-19 Precautionary Measures in the Developing Countries. *Appl. Sci.* **2022**, *12*, 3879. https://doi.org/10.3390/app12083879.
30. Mbunge, E.; Simelane, S.; Fashoto, S.G.; Akinnuwesi, B.; Metfula, A.S. Application of deep learning and machine learning models to detect COVID-19 face masks—A review. *Sustain. Oper. Comput.* **2021**, *2*, 235–245.
31. Tomás, J.; Rego, A.; Viciano-Tudela, S.; Lloret, J. Incorrect Facemask-Wearing Detection Using Convolutional Neural Networks with Transfer Learning. *Healthcare* **2021**, *9*, 1050. https://doi.org/10.3390/healthcare9081050.
32. Jiang, X.; Gao, T.; Zhu, Z.; Zhao, Y. Real-Time Face Mask Detection Method Based on YOLOv3. *Electronics* **2021**, *10*, 837. https://doi.org/10.3390/electronics10070837.
33. Hussain, S.; Yu, Y.; Ayoub, M.; Khan, A.; Rehman, R.; Wahid, J.; Hou, W. IoT and Deep Learning Based Approach for Rapid Screening and Face Mask Detection for Infection Spread Control of COVID-19. *Appl. Sci.* **2021**, *11*, 3495. https://doi.org/10.3390/app11083495.
34. Awan, M.J.; Bilal, M.H.; Yasin, A.; Nobanee, H.; Khan, N.S.; Zain, A.M. Detection of COVID-19 in Chest X-ray Images: A Big Data Enabled Deep Learning Approach. *Int. J. Environ. Res. Public Health* **2021**, *18*, 10147. https://doi.org/10.3390/ijerph181910147.
35. Ardabili, S.; Mosavi, A.; Várkonyi-Kóczy, A.R. Systematic review of deep learning and machine learning models in biofuels research. In *Engineering for Sustainable Future*; Springer: Cham, Switzerland, 2020; pp. 19–32. https://doi.org/10.1007/978-3-030-36841-8_2.
36. Abdelminaam, D.S.; Ismail, F.H.; Taha, M.; Taha, A.; Houssein, E.H.; Nabil, A. Coaid-deep: An optimized intelligent framework for automated detecting COVID-19 misleading information on Twitter. *IEEE Access* **2021**, *9*, 27840–27867.
37. Emadi, M.; Taghizadeh-Mehrjardi, R.; Cherati, A.; Danesh, M.; Mosavi, A.; Scholten, T. Predicting and Mapping of Soil Organic Carbon Using Machine Learning Algorithms in Northern Iran. *Remote Sens.* **2020**, *12*, 2234. https://doi.org/10.3390/rs12142234.
38. Salama AbdELminaam, D.; Almansori, A.M.; Taha, M.; Badr, E. A deep facial recognition system using intelligent computational algorithms. *PLoS ONE* **2020**, *15*, e0242269.
39. Mahmoudi, M.R.; Heydari, M.H.; Qasem, S.N.; Mosavi, A.; Band, S.S. Principal component analysis to study the relations between the spread rates of COVID-19 in high risks countries. *Alex. Eng. J.* **2020**, *60*, 457–464. https://doi.org/10.1016/j.aej.2020.09.013.
40. Ardabili, S.; Mosavi, A.; Várkonyi-Kóczy, A.R. Advances in Machine Learning Modeling Reviewing Hybrid and Ensemble Methods. In *Engineering for Sustainable Future*; Springer: Cham, Switzerland, 2020; pp. 215–217. https://doi.org/10.1007/978-3-030-36841-8_21.
41. AbdElminaam, D.S.; ElMasry, N.; Talaat, Y.; Adel, M.; Hisham, A.; Atef, K.; Mohamed, A.; Akram, M. HR-Chat bot: Designing and Building Effective Interview Chat-bots for Fake CV Detection. In Proceedings of the 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 26–27 May 2021; pp. 403–408. https://doi.org/10.1109/miucc52538.2021.9447638.
42. Rezakazemi, M.; Mosavi, A.; Shirazian, S. ANFIS pattern for molecular membranes separation optimization. *J. Mol. Liq.* **2018**, *274*, 470–476. https://doi.org/10.1016/j.molliq.2018.11.017.
43. Torabi, M.; Hashemi, S.; Saybani, M.R.; Shamshirband, S.; Mosavi, A. A Hybrid clustering and classification technique for forecasting short-term energy consumption. *Environ. Prog. Sustain. Energy* **2018**, *38*, 66–76. https://doi.org/10.1002/ep.12934.
44. Ardabili, S.; Abdolalizadeh, L.; Mako, C.; Torok, B.; Mosavi, A. Systematic Review of Deep Learning and Machine Learning for Building Energy. *Front. Energy Res.* **2022**, *10*, 786027. https://doi.org/10.3389/fenrg.2022.786027.
45. Houssein, E.H.; Hassaballah, M.; Ibrahim, I.E.; AbdElminaam, D.S.; Wazery, Y.M. An automatic arrhythmia classification model based on improved Marine Predators Algorithm and Convolutions Neural Networks. *Expert Syst. Appl.* **2021**, *187*, 115936. https://doi.org/10.1016/j.eswa.2021.115936.
46. Deb, S.; Abdelminaam, D.S.; Said, M.; Houssein, E.H. Recent Methodology-Based Gradient-Based Optimizer for Economic Load Dispatch Problem. *IEEE Access* **2021**, *9*, 44322–44338. https://doi.org/10.1109/access.2021.3066329.
47. Elminaam, D.S.A.; Neggaz, N.; Ahmed, I.A.; Abouelyazed, A.E.S. Swarming Behavior of Harris Hawks Optimizer for Arabic Opinion Mining. *Comput. Mater. Contin.* **2021**, *69*, 4129–4149. https://doi.org/10.32604/cmc.2021.019047.
48. Band, S.S.; Ardabili, S.; Sookhak, M.; Chronopoulos, A.T.; Elnaffar, S.; Moslehpour, M.; Csaba, M.; Torok, B.; Pai, H.-T.; Mosavi, A. When Smart Cities Get Smarter via Machine Learning: An In-Depth Literature Review. *IEEE Access* **2022**, *10*, 60985–61015. https://doi.org/10.1109/access.2022.3181718.
49. Mohammadzadeh, S.D.; Kazemi, S.-F.; Mosavi, A.; Nasseralshariati, E.; Tah, J.H. Prediction of compression index of fine-grained soils using a gene expression programming model. *Infrastructures* **2019**, *4*, 26.
50. Deb, S.; Houssein, E.H.; Said, M.; Abdelminaam, D.S. Performance of Turbulent Flow of Water Optimization on Economic Load Dispatch Problem. *IEEE Access* **2021**, *9*, 77882–77893. https://doi.org/10.1109/access.2021.3083531.
51. Abdul-Minaam, D.S.; Al-Mutairi, W.M.E.S.; Awad, M.A.; El-Ashmawi, W.H. An Adaptive Fitness-Dependent Optimizer for the One-Dimensional Bin Packing Problem. *IEEE Access* **2020**, *8*, 97959–97974. https://doi.org/10.1109/access.2020.2985752.
52. Mosavi, A.; Golshan, M.; Janizadeh, S.; Choubin, B.; Melesse, A.M.; Dineva, A.A. Ensemble models of GLM, FDA, MARS, and RF for flood and erosion susceptibility mapping: A priority assessment of sub-basins. *Geocarto Int.* **2020**, 2541–2560. https://doi.org/10.1080/10106049.2020.1829101.
53. Mercaldo, F.; Santone, A. Transfer learning for mobile real-time face mask detection and localization. *J. Am. Med. Inform. Assoc.* **2021**, *28*, 1548–1554. https://doi.org/10.1093/jamia/ocab052.

54. Teboulbi, S.; Messaoud, S.; Hajjaji, M.A.; Mtibaa, A. Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention. *Sci. Program.* **2021**, *2021*, 8340779. https://doi.org/10.1155/2021/8340779.
55. Hussain, D.; Ismail, M.; Hussain, I.; Alroobaea, R.; Hussain, S.; Ullah, S.S. Face Mask Detection Using Deep Convolutional Neural Network and MobileNetV2-Based Transfer Learning. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1536318. https://doi.org/10.1155/2022/1536318.
56. Shaban, H.; Houssein, E.H.; Pérez-Cisneros, M.; Oliva, D.; Hassan, A.Y.; Ismaeel, A.A.; AbdElminaam, D.S.; Deb, S.; Said, M. Identification of parameters in photovoltaic models through a runge kutta optimizer. *Mathematics* **2021**, *9*, 2313.
57. Houssein, E.H.; Abdelminaam, D.S.; Hassan, H.N.; Al-Sayed, M.M.; Nabil, E. A hybrid barnacles mating optimizer algorithm with support vector machines for gene selection of microarray cancer classification. *IEEE Access* **2021**,*9*, 64895–64905.
58. Vibhuti; Jindal, N.; Singh, H.; Rana, P.S. Face mask detection in COVID-19: A strategic review. *Multimedia Tools Appl.* **2022**, 1–30. https://doi.org/10.1007/s11042-022-12999-6.
59. Ejaz, M.S.; Islam, M.R.; 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), Dhaka, Bangladesh, 3–5 May 2019; pp. 1–5.
60. Din, N.U.; Javed, K.; Bae, S.; Yi., J. A Novel GAN-Based Network for Unmasking of Masked Face. *IEEE Access* **2020**, *8*, 44276–44287. https://doi.org/10.1109/access.2020.2977386.