



Untargeted white-box adversarial attack to break into deep learning based COVID-19 monitoring face mask detection system

Burhan Ul haque Sheikh¹ · Aasim Zafar¹

Received: 1 July 2022 / Revised: 17 September 2022 / Accepted: 18 April 2023 /

Published online: 5 May 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023, corrected publication 2023

Abstract

The face mask detection system has been a valuable tool to combat COVID-19 by preventing its rapid transmission. This article demonstrated that the present deep learning-based face mask detection systems are vulnerable to adversarial attacks. We proposed a framework for a robust face mask detection system that is resistant to adversarial attacks. We first developed a face mask detection system by fine-tuning the MobileNetv2 model and training it on the custom-built dataset. The model performed exceptionally well, achieving 95.83% of accuracy on test data. Then, the model's performance is assessed using adversarial images calculated by the fast gradient sign method (FGSM). The FGSM attack reduced the model's classification accuracy from 95.83% to 14.53%, indicating that the adversarial attack on the proposed model severely damaged its performance. Finally, we illustrated that the proposed robust framework enhanced the model's resistance to adversarial attacks. Although there was a notable drop in the accuracy of the robust model on unseen clean data from 95.83% to 92.79%, the model performed exceptionally well, improving the accuracy from 14.53% to 92% on adversarial data. We expect our research to heighten awareness of adversarial attacks on COVID-19 monitoring systems and inspire others to protect healthcare systems from similar attacks.

Keywords COVID-19 · Adversarial example · Face mask recognition · Adversarial attacks · Deep learning · Robustness

1 Introduction

The Covid-19 epidemic has caused devastation to healthcare facilities and treatment systems in every country. The virus is disseminated by direct contact with contaminated respiratory droplets (produced via sneezing and coughing). Anyone who comes into contact

✉ Burhan Ul haque Sheikh
sbuhaque@myamu.ac.in; shiekhburhan2013@gmail.com

Aasim Zafar
azafar.cs@amu.ac.in

¹ Department of computer science, Aligarh Muslim University, Aligarh, Uttar Pradesh 202002, India

with virus-infected surfaces and touches their faces might get sick, with symptoms such as shortness of breath, cough, and fever. In crowded and public locations, the WHO recommends wearing a face mask to minimize viral transmission via the nose or oral passages [12, 14, 62]. The government of COVID-19-affected nations has passed laws mandating the wearing of face masks. As a result, large groups of individuals in public spaces and congested locations must be monitored for face mask violations. Conventional procedures for checking a face mask violation are not always feasible and are error-prone.

Computing-assisted deep neural network (DNN) techniques have made significant progress in the recent decade, showing promise in a variety of classification tasks [33, 55] and pattern recognition [26]. In the fight against the COVID-19 epidemic, artificial intelligence, and more specifically machine or deep learning, has proved to be a successful and crucial tool. It has proven to be beneficial in early predicting infection by the analysis of the previous data of the patients [5]. In addition, deep learning models have been proposed to detect COVID-19 from the readily available chest X-Ray or CT images [27]. As discussed earlier, wearing a mask is the most effective way to stop the rapid spread of COVID-19. However, ensuring that masks are worn properly in public or crowded places is difficult with conventional approaches, which rely on human monitoring. Fortunately, artificial intelligence has provided a solution with computer-aided face mask detection systems. In computer-aided face mask detection systems, methodologies primarily based on deep learning have been applied to produce rapid, accurate, and trustworthy systems. A large number of researchers have developed a variety of DNN architectures to detect face mask violations. Most of these researchers have based their methods on transfer learning [61], while some have developed novel architectures. DL models have shown high sensitivity and specificity when detecting face mask violations.

Deep learning models need a significant amount of data to train a model effectively in the training phase. When given a new unseen input sample, the deep learning model infers the corresponding output during the inferencing phase. For example, in the case of the face mask detection model, when the model is trained on the dataset, the model is then given unseen data and returns the classification result, i.e., *with mask*, *no mask*, or *incorrect mask*. Deep learning models have various confidential and substantial assets in deep learning models, including training data, trained models and model parameters. However, recent research has proven that these assets are vulnerable to various attacks during the training and inferencing phase of the model, allowing attackers to undermine the model's security and privacy. These attacks on deep learning algorithms are broadly categorized into four types: Adversarial attack, Data poisoning attack, Model Inversion attack and model extraction attack. Data poisoning is performed before the model's training process [6]. The training data is contaminated using various techniques, including adding mislabeled data or triggering patches to the input data. It aims to force the model to misclassify the input. Model inversion attack concerns the confidentiality of the data used for the model's training [31]. In a model extraction attack [28], the attacker tries to get the internal confidential details of the model, such as parameters, hyper-parameters, and the model's architecture. During an adversarial attack, input data samples are added with the modest, skillfully crafted perturbation that causes the model to misclassify during inferencing [53]. In computer vision, the adversarial attack is primarily applied to the classification models during the inferencing phase. In Fig. 1, we have demonstrated various attacks on deep learning models in their respective phases.

The presence of the adversarial attack calls into doubt the widespread use of deep learning models for face mask recognition. Especially in insecure situations, the prevalence of this adversarial approach raises concerns regarding the widespread deployment of deep

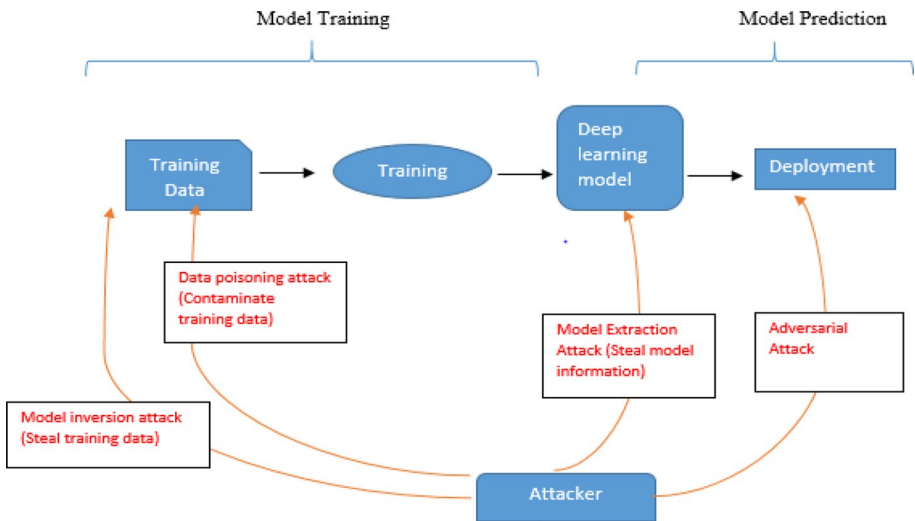


Fig. 1 It demonstrates the various attacks on the deep learning model's assets

learning models for face mask identification. Since various face mask detection systems based on deep learning have been proposed so far, none of them have discussed the possible vulnerabilities in the system and how to mitigate these weaknesses. This study reveals the vulnerabilities of face mask detection systems based on conventional deep learning and proposes a comparatively more robust system against adversarial attacks. As a part of this research, first, we developed a custom dataset for training the proposed face mask detection model. The model is based on transfer learning of the state-of-the-art classification model MobileNetV2 [54] and can recognize and classify the images into three classes, including an *incorrect face mask*, *with mask* and *no mask*, with excellent accuracy of 96%. After that, we assessed the commensurate vulnerabilities of the developed model against the adversarial examples. In particular, we attacked the model by the Un-targeted Fast Gradient Sign Method (FGSM) [17]. The adversarial samples were computed on different, very small epsilon parameter settings of FGSM and the model's performance was evaluated. The resultant adversarial image looks imperceptible and similar to the original image. These adversarial input images forced the face mask model to misclassify with high and alarming confidence. The result shows that the attack significantly degrades the performance of a model. In order to enhance the model's resilience against adversarial attacks, we implemented an adversarial training strategy with slight alterations [15]. Specifically, instead of retraining the model on the entire dataset, we trained it on a randomly chosen subset of the training data, along with their respective adversarial images. The outcome of this modified training approach demonstrated an improvement in the model's performance on adversarial examples.

1.1 Basic concepts

we have discussed some essential terminologies related to attacks on deep learning-based face mask detection models. The attacks on the system can be classified according

to whether the attacker has access to the model; in that case, we have two types of attacks: white box and black-box attacks. Additionally, the attacks can be classified according to whether the model is compelled to produce a particular label or not. As a result, we have two types of assaults: targeted and untargeted.

- **White box Attack Approach:** In this case, the attacker has full access to the system, say 'F' or the entire information of a model, including the gradient of the model, model parameters, hyperparameters, and training dataset. The adversary is provided with the classifier 'F' in the white-box threat model. Gradient descent on the adversarial loss $\ell_y(\cdot)$, or an approximation thereof is a powerful attack approach in this scenario. To ensure that the changes remain undetectable, one can regulate the perturbation norm $\|\delta\|_2$ either by stopping the loss optimization early or by incorporating the norm directly as a regularize or constraint into the loss optimization.
- **Black box Attack Approach:** The attacker only partially knows a model. Gradient parameters and training sets might not be accessible or known to the attacker. Arguably, many white-box assumptions are impractical in many real-world situations. For example, the model 'h' may be accessible to the public via an Application Programming Interface(API) that allows only input queries, obviating the attacker of access to the model's details.
- **Targeted Attack Approach:** The face mask detection system is compelled to output a certain/target label. For example, given an input image of a person wearing a mask, the model may be compelled to output an unmasked face.
- **Untargeted Attack Approach:** It is entirely focused on the misclassification of a model, regardless of the output.

1.2 Contribution

- The custom face mask dataset was developed and made publicly available on a KAGGLE repository (<https://www.kaggle.com/datasets/shiekhburhan/face-mask-dataset>).
- We developed the deep learning-based face mask detection model using the conventional approach. The model is based on transfer learning of MobileNetv2 and performed well on the unseen clean data samples with an accuracy of 95.85%
- We demonstrated that the conventional model is susceptible to adversarial attacks by exposing it to the FGSM attack.
- We proposed the framework for a robust face mask detection model that is resistant to adversarial attacks such as FGSM.

1.3 Paper organization

Following the introduction, Section 2 reviews the literature on face mask detection and adversarial attacks. The proposed dataset and comparison with other standard datasets are described in Section 3. The methodology, which is broken down into three parts, training a face mask classifier, attacking it with FGSM, and improving the robustness of the face mask model, is described in Section 4. Section 5 discusses the Experimental

results and outputs in detail. Section 6 deals with the conclusion. Finally, the paper is concluded with the future scope of the study.

2 Related work

The literature review has been split into two sections. First, we will go through the existing literature on face mask detection systems, and then we will discuss the literature on adversarial attacks on COVID-19 monitoring systems, including face mask detection systems.

2.1 Face mask detection models

Due to the ongoing epidemic, there has been a lot of interest in projects with similar purposes. Most researchers utilize convolutional neural networks (CNN) from all the approaches mentioned in the literature because of their 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.

2.1.1 CNN-based approaches

Contrary to ML approaches, we do not need to manually extract the features in CNN-based methods. 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 [16], 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 [50]. 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 [8], similar to our model, they used MobileNetV2 to classify the face mask and Caffe-based face detector. A small dataset of 4095 images was used. Additionally, the data set has only two classes, masked and unmasked. Hence, the model trained cannot detect the incorrectly masked faces (i.e., having his or her mask below the nose). It achieved a descent f1-score of 0.93.

In [3], 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. In order to overcome the problem of overfitting, they used the concept of transfer learning and fine-tuned the model with the WiderFace dataset [63]. 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 [49], 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 Facemask-Wearing (IFW), Correct Facemask-Wearing (CFW), and No FaceMask-Wearing (NFW). The model was trained using the MMD or Medical Masked dataset [1] and the MobileNetV2 CNN algorithm was adopted. The design was well-organized and effective. However, the dataset used for training was very small and the face mask detection speed was plodding than other algorithms. The model achieved an accuracy of 98.07%.

The approach in [24] 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, then extracts the face from the input, and then applies the face masknet model for classification. It includes extremely limited and regionally specific data. The accuracy of the model was reported to be 98.06%.

In [38], they proposed the novel face mask detection technique using YOLOv2 [52] and ResNet50 [18] together. They used the FMD [35] and MMD [1] datasets to train and test a model. SGDM and ADAM optimizers were used to compare the performance [32]. The model achieved an average precision of 81%.

In [23], the VGG16 architecture was utilized to identify and categorize face expressions [55]. The accuracy of their VGG16 model trained on the KDEP database is 88%.

The transfer learning of the InceptionV3 model [56] was used in [29]. The last layer of the model was removed and five new trainable layers were added. The last layer consists of 2 neurons followed by a softmax activation function where each neuron corresponds to a masked face and unmasked face, respectively. The model obtained an accuracy of 99.91% training and 100% testing accuracy in 80 epochs.

In [42], 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 [44]. They used a similar approach to ours—for face detection, they utilized a single shot multi-box detector [37] and MobileNetV2 is used 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 [36], for face detection, YOLOv3 was used. It was trained on celebi and wider-face [63] databases. The model was later evaluated on the FDDB database [25] and achieved an accuracy of 93.9%.

2.1.2 Hybrid-based approaches

The algorithms for deep learning and machine learning were combined in [39]. 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 [45], 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 [59]. 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 facemasks. The recall was said to be above 95%, and the rate of false positives was less than 5%.

In [4], a haar-cascade-based feature detector was utilized to recognize a nose and mouth from the detected face [58]. 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 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, hair on a face etc.

Principal Component Analysis (PCA) algorithm was implemented in [11] 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%.

2.2 Adversarial attack on COVID-19 monitoring models

Since it was claimed in [57] that adversarial techniques might attack neural networks, the study of such techniques has become a hot topic in the field of artificial intelligence, with researchers continually proposing novel adversarial attack methods and mitigation techniques. A paucity of research has been conducted up until this point to address the challenges posed by adversarial attacks on computer-based COVID-19 combating technologies.

In [19], they conducted both targeted and untargeted universal adversarial perturbation (UAP) attacks against the COVID-Net architecture [60]. COVID-Net is a model developed specifically for classifying COVID-19 patients based on chest X-ray pictures. The outcome revealed that the COVID-net is susceptible to tiny UAPs. In particular, 2% of the UAPs to the average norm of image perturbations for face recognition systems using the image translation techniques achieve success rates of >85% and >90% for nontargeted and targeted attacks, respectively. Due to nontargeted UAPs, most chest X-ray pictures are classified as COVID-19 instances by the DNN models. The targeted UAPs cause DNN models to classify most chest X-ray pictures into a specific target class.

In [51], they studied six different applications used for COVID-19 diagnosis and each adversarial application attack was proposed. They attacked the following applications: 1) recognize whether a subject is wearing a mask from a live camera feed; 2) maintain DL-based QR codes as immunization certificates; 3) add explainability of GRAD-CAM DL algorithms; 4) recognize COVID-19 from CT scan images; 5) detect noninvasive biometrics and identify social distancing from a live camera feed; and 6) recognize COVID-19 from X-ray image analysis. They tested the existing adversarial methods in this study, including FGSM, MI-FGSM, Deepfool, L-BFGS, C&W, BIM, Foolbox, PGD, and JSMA [13].

Using image translation methods, the authors of [30] calculated adversarial perturbation [2] for face recognition systems. To fool the targeted face recognition system, they

would take a source image and distort it to create any desired facial appearance. White-box testing showed a 90% success rate for the attack. With the help of adversarial instances and the dynamic distillation technique, the model was able to reach an 80% success rate in the black-box setting. The translated images were realistic to the human eye and kept the person's identity intact, while the perturbations were significant enough to fool trained defenses.

In [46], they proposed a COVID-19 diagnosis model based on transfer learning. The models were based on the state-of-the-art classification models VGG and the inception model. Models were developed using VGG and the inception model, two of the most advanced classification methods currently available. When the perturbation was increased from 0.0001 to 0.09, the VGG16 model's accuracy plummeted by more than 90% for X-ray images, while the Inception-v3 network's accuracy declined by 30%. The FGSM attack similarly exposed vulnerabilities, such as misdiagnosis, in CT imaging. They also demonstrate that the degree of disturbance has a sizable impact on the degree to which people are aware of attacks.

In [41], machine learning techniques including support vector machine, random forest, logistic regression naïve bayes were implemented to classify chest X-ray images as viral pneumonia, COVID-19 and healthy persons. The model used 1400 images that were collected from the Kaggle public repository. The experimental outcomes of this study confirmed that the supported vector machine technique has high accuracy and excellent performance in the classification of the disease, as reflected by values of 91.8% accuracy, 91.7% sensitivity, 95.9% specificity, 91.8% F1-score, and 97.6% AUC.

3 Dataset description

We uploaded a new dataset called the Sophisticated Face Mask Dataset on the Kaggle public repository on 16 June 2022. This dataset contains many different types of images of people wearing face masks. We collected these images from many different places so that the dataset would be unbiased and diverse. We used some images from other popular datasets like Masked FAcEs (MAFA) 16 and Masked Face Detection Dataset (MFDD), as well as some simulated images to make sure our dataset was complete. We also included some pictures that we found on the internet. Our goal was to create a dataset that could help train a model to recognize different types of face masks from different angles. The model we train with this dataset can categorize images into three groups: a mask on the face, an

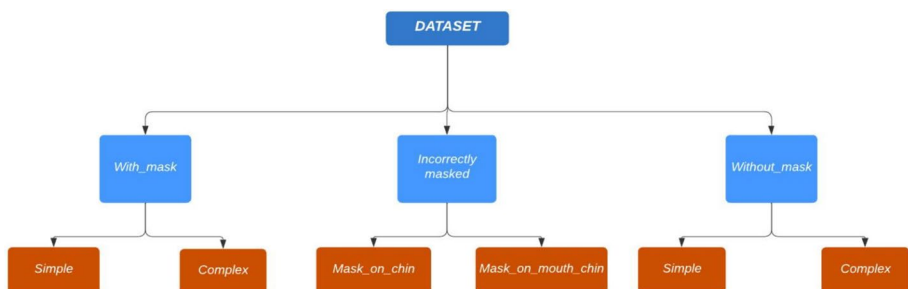


Fig. 2 Dataset's organization structure in a graphical format



Fig. 3 Example of images in the dataset

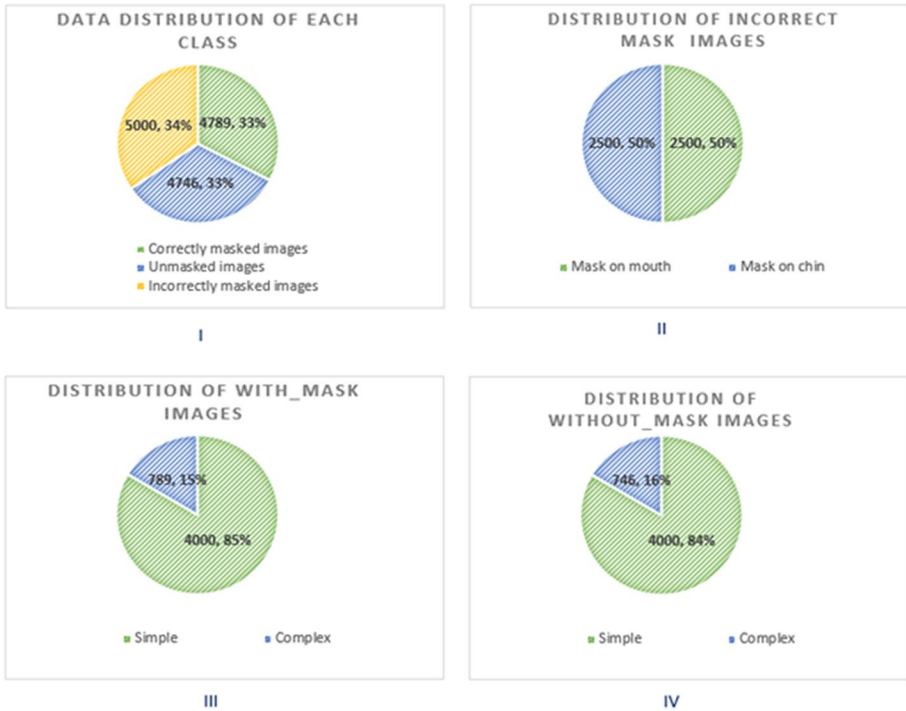


Fig. 4 Data Distribution (a) Distribution of each class. (b) Mask-on chin and Mask-on-face-chin images in the incorrectly masked images. (c) Distribution of simple and complex images in correctly masked images. (d) Distribution of the Unmasked images

incorrectly worn mask, or no mask at all. The organization of the data in the dataset can be seen in Fig. 2.

The dataset contains 14,535 images categorized into three primary categories: with mask(4,789), incorrectly masked(5,000), and without mask(4746). Each category has sub-categories based on the type of image, which makes the dataset useful for other computer vision tasks such as face recognition and occlusion face detection [22]. The images include various depictions of people wearing masks, including those with masks incorrectly placed on the chin or covering only the mouth, simple and complex masks with different designs, and images of people not wearing masks with occlusions like beards, long hair, or hands

covering their faces. Some of the data sample in the dataset are shown in Fig. 3. The dataset's data distribution can be seen in the Fig. 4

Some of the data samples in the dataset are shown in Fig. 3.

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. The data distribution of each class can be seen in Fig. 4.

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

4 Methodology

This study develops a face mask detection model based on transfer learning of MobileNetV2, attacks the face mask model by untargeted FGSM adversarial attack and proposes a framework for a robust face mask detection model that is resistant to adversarial attacks. The proposed face mask model is initially trained on the custom compiled dataset for effective and efficient training. For the proposed model, adversarial images are created by computing the perturbation for each test image using the FGSM technique and combining it with the corresponding clean image. The non-robust face mask model is then modified by using the proposed framework.

4.1 Transfer learning of MobileNetV2

MobileNetV2 is a State-of-the-art CNN model that performs well on devices such as mobile phones. The layers have been trained on the ImageNet dataset [9] and attained an optimal set of values. It is fast, accurate, and lightweight (resources efficient) when dealing with resource-constrained and real-time systems. The face mask detection system is incorporated in surveillance systems with limited resources, such as CCTV cameras. Therefore, it is the perfect model for the proposed method. MobileNetV2 employs a dual-block structure. One is a residual block with a stride of 1 and the other is of stride 2 that is used for downsampling. Both blocks have three layers each. The first layer is 1×1 convolution with linearity. The second layer is the depthwise convolution. The third layer is another 1×1 convolution but without any non-linearity.

For the face mask detection model, we removed the last layer of the MobileNetV2 and added four trainable layers, including Dense 128, Dense 62, Dense 32, and Dense 3. We also used the dropout layers between these layers to avoid overfitting. MobileNetV2 layers are kept frozen so that they are not trainable during the training of the face mask model. The last layer of the model consists of 3 neurons, each corresponding to the required class/output. Transfer learning of MobileNetV2 allowed us to save significant computational costs while improving the result. Figure 5 illustrates the architecture of the face mask detection model.

During the model training, we used 93% of the complete data and 7% for testing a model. The model classifies an image with an accuracy of 98.51% on training data and 95.83% on testing data. During the training, the parameters and hyperparameter used are discussed in Table 2. The learning rate controls how fast the network weights are updated.

Table 1 Comparison of various standard facemask dataset

Paper	Dataset	Composition of Dataset	No. of images	Characteristics	Limitations
[35]	FMDD	Contains only masked face images	853	Available publically	Limited images and only masked images
[21]	MFDD	Solely masked face images	24,471	public	Biased to Chinese faces
[16]	MAFA	Contains 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
[4]	MaskedFaceNet	composed of masked(2,500) and unmasked image(2,500)	5,000	Images are cropped to include the face regions only.	Limited diversity in face mask types and people's facial expressions.
[10]	Wider face Mask	composed of masked(19,272) and unmasked images(12,931)	32,203	It includes different types of masks such as, medical, casual, printed and so on.	Limited diversity in mask types and people's facial expressions. Potential class imbalance as it was collected during the early stage of the pandemic when wearing masks was not yet mandatory in some regions.
[20]	LFW	Composed of celebrity images of different orientation	13,233	Benchmark dataset for face recognition	Does not contain any masked face image
Proposed Dataset	Sophisticated Face Mask Dataset	Composed of the following: 1) simple and sophisticated masked images 2) simple and sophisticated Unmasked images 3) Mask-on-chin incorrectly masked and mask-on mouth-chin incorrectly images.	14,535	It is diverse and unbiased dataset and includes a variety of masks and orientations, addressing the limitations of prior datasets. The resulting model can detect different types of masks and easily recognize occlusion in front of the face.	-

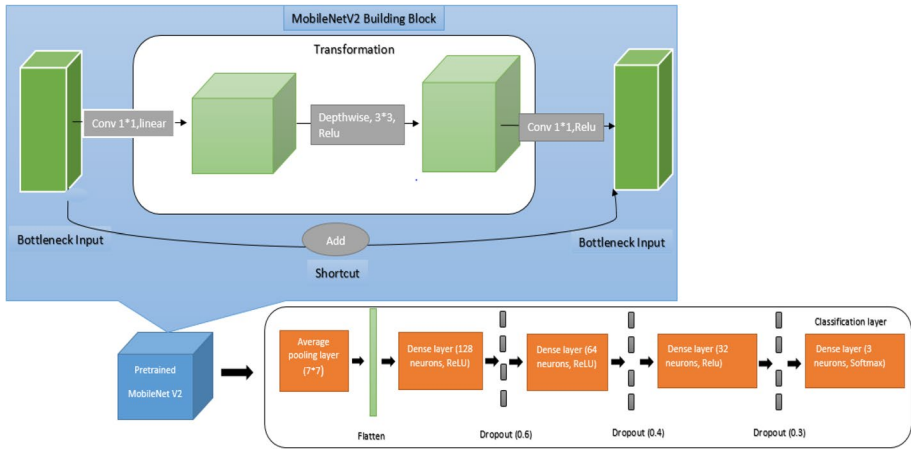


Fig. 5 Face mask Detection model architecture based on transfer learning of MobileNetV2

Table 2 Face mask detection model parameters and hyperparameters

Parameter	Value
Learning rate	1e-3 with decay rate = learning rate / epoch number;
Batch size	32
Epochs	30
Dropout rate	0.6, 0.4, 0.3
Input layer size	224 *224*3
Output layer size	224*224*3
Optimization	ADAM

The batch size controls the number of samples before the network updates its parameters. The number of epochs controls the number of training iterations. The dropout rate determines the percentage of parameters that are dropped while training to prevent over-fitting.

4.2 Untargeted FGSM (fast gradient sign method) attack on the proposed model

We first provide notations and then introduce the formulation of an adversarial attack on the proposed face mask detection system.

Notations: Consider F as a face detection model, x as original input, l as the desired output of a model or class of input, r as a perturbation, t as the target class, and θ as parameters of the face mask detection model.

$x \in$ Sample from the dataset.

$l \in$ With-mask or without-mask or incorrect-mask.

Formulation: Consider x a sample from the sophisticated face mask dataset, for instance, an image with no-mask on a face. We compute perturbation r of . Adding this perturbation r to the original image x is known as an adversarial example i.e.

$x' = x + r$ is an adversarial example.

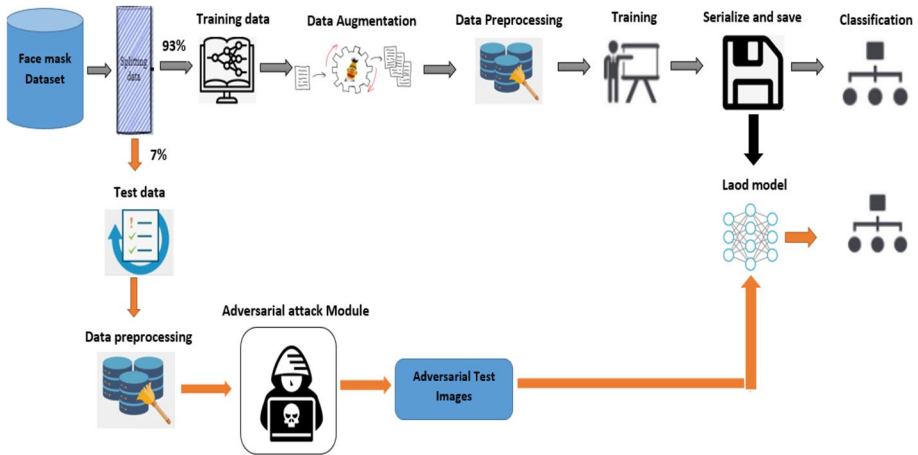


Fig. 6 It demonstrates the adversarial attack on the proposed face mask model

Note: x' should be imperceptible to humans, i.e., it should appear to humans as class l , i.e., no-mask. Given this x' as input to the F , the outcome of F is other than no-mask.

The adversarial attack against the proposed model is carried out by computing the adversarial perturbation for each test image and adding it to the corresponding clean image. We employed the FGSM strategy for adversarial perturbation computation due to its simplicity and attacking efficiency. The framework of the adversarial attack on the face mask model is shown in Fig. 6. The TensorFlow methods and classes used to generate adversarial attacks are discussed in Table 3. FGSM is a white-box attack approach in which gradients are computed with respect to the pixels, and then sign operation is applied to the gradient matrix to get the desired result [54]. Back-propagation is used to calculate the gradient, which determines the perturbation direction. Perturbation of image is accomplished by taking a single large step in increasing the classifier’s loss. Simply stated, for each pixel in an input image, what would happen to the target output neuron if the value of that pixel increased or decreased. As the last step, a small value ‘ ϵ ’ epsilon is multiplied by the threshold gradient matrix and then add it to the input image. Perturbation is given by the Equation (1):

Table 3 Tensorflow methods used for the generation of adversarial examples for the proposed face mask system

Method	Description
Gradient Tape	It is used to record the operation performed on a tensor under the context manager’s watch. We used it to record the operation performed on the image during the forward propagation. This information is required during the computation of the gradient
Gradient	It is a function used to compute the gradient. It is implemented as <i>gradienttape.gradient()</i>
Categoricalcrossentropy	It is a loss function provided by the <i>tensorflow.losses.categoricalcrossentropy()</i> , used to compute the loss between prediction and original label.
Sign	It returns an element-wise indication of the sign of a number.it is provided by <i>tf.sign(x) = -1 if x < 0; 0 if x == 0; 1 if x > 0.</i>

Input: image, label, ϵ (epsilon), face mask model

Output: Adversarial example

STEP 1 for each image in a Testset, do

STEP 1.1 Preprocess the image: Normalizing and resizing (224*224*3)

STEP 1.2 Predict the label of the test image: prediction= face-mask-model.predict(image)

STEP 1.3 compute the loss between actual and predicted label:

$$loss=CategoryCrossEntropy(label,prediction)$$

STEP 1.4 compute the gradient of the loss with respect to the input test image:

$$gradient=gradient(loss, image)$$

STEP 1.5 Apply the sign method and multiply it with the epsilon resulting in the

$$perturbation=(sign(gradient) * \epsilon)$$

STEP 1.6 Add this perturbation to the image: Adversarial image=image + perturbation

STEP 2 Adversarial images are sent to the MobileNetV2 Classifier for classification.

Algorithm 1 Adversarial examples for face mask detection model

$$\delta = \epsilon * \text{sign} (\nabla * \text{loss} (x, l)) \tag{1}$$

where,

l is the true label of clean image x , $\nabla * \text{loss} (x, l)$ calculates the gradient of the loss function around the current value of the model parameters in relation to the ‘ x ,’ ϵ (epsilon) is the magnitude of the perturbation, i.e., it bounds the total number of pixels in x' that can be modified with respect to x . The Adversarial example x' is computed as $x' = x + \delta$.

Since we have applied untargeted FGSM adversarial attacks on the proposed face mask system. The objective function of an untargeted adversarial attack is given in Equation (2):

$$\min_{X'} \|x' - x\|$$

Subject to

$$\begin{aligned} F(x) &= \ell \\ F(x') &= \ell' \\ \ell &\neq \ell' \end{aligned} \tag{2}$$

Here $F(x)$ rightly classifies the input, but $F(x')$ misclassifies the input as some other class. The objective function states that the magnitude of a ‘ r ’ must be small and it must be small enough to make the model misclassify.

Following is the algorithm for generating the adversarial example for the face mask classifier model.

In STEP 1, we iterate through all the images in a test set for perturbation computation. After that, in STEP 1.1, The image selected is first preprocessed before being sent for prediction. The image size is resized into 224*224*3 dimensions and normalized by dividing it by 225.0. The preprocessed image is subsequently forwarded to the trained model for

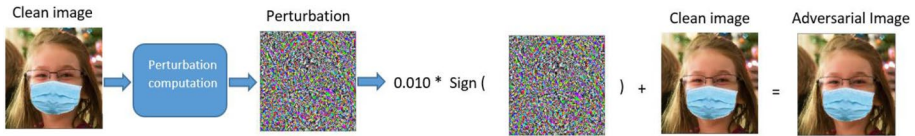


Fig. 7 Illustration of the adversarial example generation

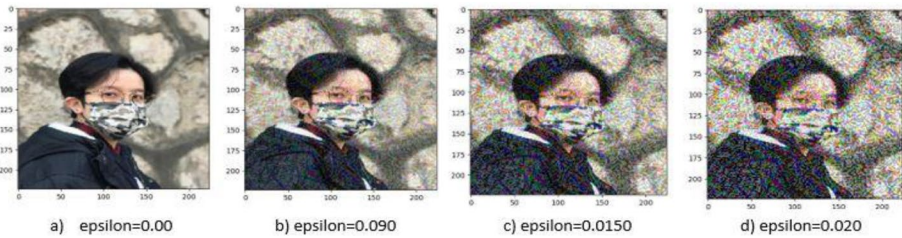


Fig. 8 It illustrates the adversarial image computed on different epsilon values. **a** epsilon=0.00, **b** epsilon=0.090, **c** epsilon=0.0150, **d** epsilon=0.020

prediction in STEP 1.2. After the prediction, the loss between the actual label of the image and the prediction is computed in STEP 1.3. In STEP 1.4, the gradient, which is the rate of change of loss with respect to the input image, is computed. The *sign* operation and epsilon are then applied to the gradient to compute the perturbation IN STEP 1.5. The adversarial image is computed by adding the clean image to the perturbation in STEP 1.6. In STEP 2, the adversarial images are given as input to the face mask detection system for classification.

We have illustrated the generation of adversarial perturbation and adversarial examples in Fig. 7. An adversarial image is produced by setting the epsilon value as 0.010. Although it is clear from the figure that the adversarial image is indistinguishable from the original image, the model still misclassifies it.

The parameter epsilon in FGSM determines the magnitude of the perturbation, i.e., it represents the amount of the change in pixel values in the adversarial image. In Fig. 8, we demonstrated the adversarial image with different epsilon values. We can visualize the impact of epsilon on the clarity of the adversarial image and by increasing epsilon's value, the adversarial images' changes become perceptible.

4.3 Improving the robustness of the proposed face mask detection model

In this section, we designed a framework by which the models' robustness to the adversarial attack is enhanced. We employed an adversarial learning approach in which the dataset is modified by adding adversarial examples and the model is retrained on the modified dataset. However, in our approach, we modified the adversarial learning by introducing an adversarial generator module that generates adversarial examples of the same image on different epsilons and adds them to the sophisticated face mask dataset. The conventional adversarial generator module generated only perturbations on single epsilon values. The reason for producing adversarial images with varying epsilon values is to ensure that the model is resilient to a variety of adversarial perturbations instead of a single perturbation. The framework of the robust model is shown in Fig. 9.

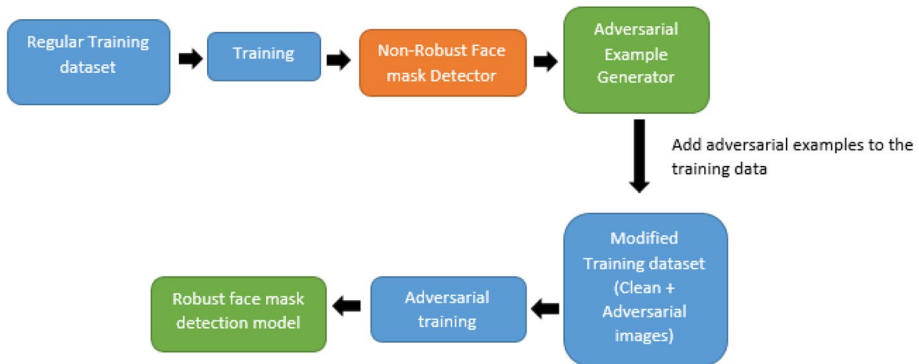


Fig. 9 Framework of the robust face mask detection system

4.3.1 Formulation

The goal is to enhance the robustness of face mask detection model ‘F’ to adversarial attacks. We employ an adversarial learning approach by modifying the dataset say D with adversarial examples and retraining the model F on the modified dataset. We introduced an adversarial generator module ‘A’ that takes an image and an epsilon value as input and outputs an adversarial example. Conventionally, ‘A’ generates adversarial examples by adding perturbations on a single epsilon value. However, we propose a modification to ‘A’ by introducing an adversarial generator module that generates adversarial examples of the same image on different epsilon values and adds them to the dataset D.

Let $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ be n different epsilon values. Then, ‘A’ generates n different adversarial examples of the same image x , denoted as $x'_{11}, x'_{12}, \dots, x'_{1n}$ corresponding to $\epsilon_1, \epsilon_2, \dots, \epsilon_n$, respectively. This modification aims to ensure that the model F is resilient to a variety of adversarial perturbations instead of a single perturbation. We create a modified dataset D’ by adding the adversarial examples $(x'_{11}, x'_{12}, \dots, x'_{1n}), (x'_{21}, x'_{22}, \dots, x'_{2n}), \dots, (x'_{m1}, x'_{m2}, \dots, x'_{mn})$ for each epsilon value to the original dataset D, where m is the number of randomly selected clean samples from D. That is, $D' = \{(x_i, y_i), (x'_{ij}, y_i) : i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}\}$ where (x_i, y_i) is the i th image-label pair in D and x'_{ij} is the adversarial image of the i th clean image corresponding to the epsilon value j . We retrain the model F on the modified dataset D’ to enhance its robustness to adversarial attacks. Algorithm 2 outlines the steps of the robust face mask detection model.

STEP 1 iterates through a bunch of the images in the regular dataset. Instead of computing the perturbation on a single epsilon value, we iterate through a range of epsilons in STEP1.1. For example, $\epsilon_1, \epsilon_2, \dots, \epsilon_N$ are the values of the epsilons that are used to generate the different adversarial examples of same clean image. In STEP 1.1.1, the trained non-robust model predicts the output of the clean test image. STEP 1.1.2 and 1.1.3 computes the loss and the gradient, respectively. In step1.1.4, the perturbation is computed by applying the sign method to the gradient and multiplying it with the current epsilon value, i.e., $\text{perturbation} = \text{sign}(\text{gradient}) * \epsilon_1$. STEP 1.1.5 computes the adversarial image on the current epsilon value, say ϵ_1 . The generator iterates through all epsilon values to generate the adversarial images of the same clean image on different epsilon values. In Step 2, the

Input: regular dataset, label, non-robust-model, ϵ (set of epsilon values)

Output: Robust model

STEP 1 Compute the adversarial for bunch of images in the regular dataset:
for *bunch of images in a regular dataset*, **do**

STEP 1.1 Compute perturbations for a range of epsilon values:
for *range of ϵ* , **do**

STEP 1.1.1 predict the label of the clean test image: prediction= non-robust-model.predict (clean image)

STEP 1.1.2 compute the loss between actual and predicted label:
 $loss = \text{CategoricalCrossEntropy}(label, prediction)$

STEP 1.1.3 compute the gradient of loss with respect to the input test image:
 $gradient = \text{gradient}(loss, image)$

STEP 1.1.4 Apply the sign method and multiply it with the epsilon resulting in the perturbation: $perturbation = (\text{sign}(gradient) * \epsilon)$

STEP 1.1.5 Adversarial image = clean image + perturbation

STEP 1.1.6 Add clean and Adversarial images to the modified training dataset.

STEP 2 Retrain the non-robust-model on the modified dataset.

STEP 3 Resultant model is robust to adversarial attacks.

Algorithm 2 Robust face mask detection model

non-robust model is retrained on a modified dataset consisting of clean and adversarial images. The resultant model is robust to adversarial attacks.

5 Experimental results

We trained the proposed face mask model on google collaboratory and then saved it on a local machine. After that, the model was loaded and the adversarial attack was executed on a local machine with 8 GB RAM and NVIDIA Ge Force 930 max. Finally, the non-robust face mask model was modified using the proposed framework and retrained on google collaboratory.

5.1 Pre attack model performance

The face mask detection model was trained for 20 epochs on 93% of the total data. The remaining 7% were used for validation/testing purposes. The model performed well and classified the images as the *with-mask*, *without-mask*, and *incorrect-masked* faces with an accuracy of 98.51% and 95.83% on training and testing data, respectively. The learning curves of the model are shown in Fig. 10.

The plot makes it clear that as the number of epochs increases, the training and validation accuracy increases while the loss of training and validation data decreases. It is

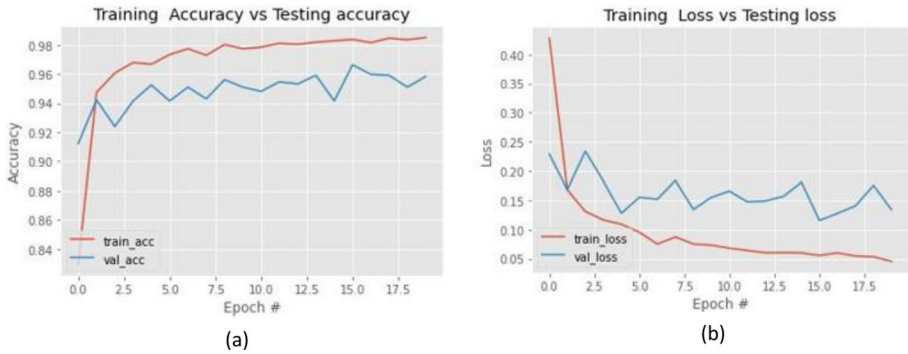


Fig. 10 Learning curves of models accuracy and loss: (a) is the model’s accuracy on training and testing data. (b) illustrates the loss of the model on training and testing data

	precision	recall	f1-score	support
incorrect_mask	0.89	1.00	0.94	353
with_mask	0.98	0.86	0.91	350
without_mask	0.98	0.97	0.98	350
accuracy			0.94	1053
macro avg	0.95	0.94	0.94	1053
weighted avg	0.95	0.94	0.94	1053

Fig. 11 Precision, recall, f1-score of the face mask detection model on clean test data

evident from the plot that the testing and training accuracy are not far from each other, which concludes the model is not overfitting.

Several other metrics were used to assess a model’s performance, including recall, precision, F1-Score, accuracy, macro-average, and weighted average. All these metrics were computed by the *classification_report* method of the SK-learn package. The metrics are defined below:

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

$$F1Score = 2 * (Recall * Precision) / (Recall + Precision)$$

At first, we computed the performance of a face mask detection model on the clean test dataset. The model was evaluated on 1053 unseen/test images, of which 353, 350 and 350

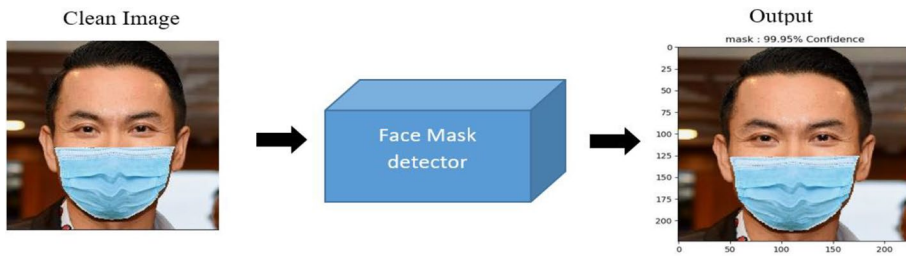


Fig. 12 The model correctly classifies the image as *masked*

Table 4 Accuracy and loss of regular model on various epsilon of FGSM

Epsilon(ϵ)	Accuracy (%)	Loss of a model
0 (clean image)	95.83	0.1339
0.001	75.12	1.2533
0.002	59.54	2.7042
0.003	45.20	3.9079
0.004	33.62	4.8393
0.005	26.59	5.5416
0.006	22.51	6.0692
0.007	18.52	6.4622
0.008	16.33	6.7565
0.009	14.53	6.9780

belong to incorrect-mask, with-mask and without-mask, respectively. The model achieved a precision of 95%, recall of 94%, and f1-score of 94%. The result of the clean test image is shown in Fig. 11.

The output of a face mask detection model is shown in Fig. 12. When given a clean *masked face* image, the model correctly predicts it as a *masked* image with 99.95% confidence.

5.2 Post attack model performance

The classifier achieved excellent training data accuracy and performed well on the test data. However, this was not the case once attacked by the untargeted FGSM. As discussed in Section 4, the model's performance was evaluated on adversarial images generated on different epsilons. The reason is to compare the impact of the different epsilon on the model's performance. It is evident from Table 4 that on increasing the value of epsilon, the model's performance degraded. On epsilon value **0.009**, the accuracy of a model was reduced to 14.53% from 95.83, which is alarming. The plot of accuracy and loss on different epsilon values can be seen in Fig. 13. It is also evident that increasing the value of epsilon resulted in a decrease in the accuracy of the model.

In addition, we have computed the precision, recall, and f1-score on different epsilon values. The results reinforce our conclusion that the performance of the model has degraded. The list of classification reports is shown in Fig. 14.

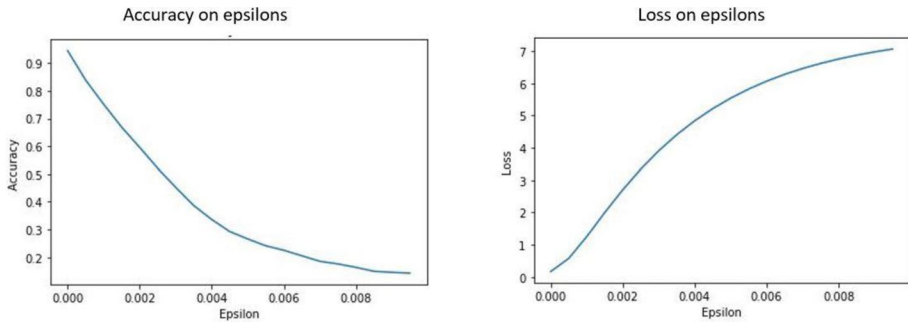


Fig. 13 Accuracy and loss curves on various epsilon values

	precision	recall	f1-score	support
incorrect_mask	0.74	1.00	0.85	353
with_mask	0.88	0.66	0.75	350
without_mask	0.95	0.85	0.90	350
accuracy			0.84	1053
macro avg	0.86	0.84	0.84	1053
weighted avg	0.86	0.84	0.84	1053

	precision	recall	f1-score	support
incorrect_mask	0.66	0.99	0.79	353
with_mask	0.75	0.54	0.63	350
without_mask	0.93	0.72	0.81	350
accuracy			0.75	1053
macro avg	0.78	0.75	0.74	1053
weighted avg	0.78	0.75	0.74	1053

a) epsilon= 0.0005

	precision	recall	f1-score	support
incorrect_mask	0.61	0.96	0.75	353
with_mask	0.59	0.44	0.51	350
without_mask	0.90	0.60	0.72	350
accuracy			0.67	1053
macro avg	0.70	0.67	0.66	1053
weighted avg	0.70	0.67	0.66	1053

b) epsilon= 0.0010

	precision	recall	f1-score	support
incorrect_mask	0.56	0.92	0.70	353
with_mask	0.46	0.35	0.40	350
without_mask	0.85	0.51	0.64	350
accuracy			0.60	1053
macro avg	0.62	0.59	0.58	1053
weighted avg	0.62	0.60	0.58	1053

c) epsilon= 0.0015

	precision	recall	f1-score	support
incorrect_mask	0.16	0.20	0.18	353
with_mask	0.07	0.10	0.08	350
without_mask	0.43	0.13	0.20	350
accuracy			0.14	1053
macro avg	0.22	0.14	0.15	1053
weighted avg	0.22	0.14	0.15	1053

d) epsilon= 0.0020

	precision	recall	f1-score	support
incorrect_mask	0.16	0.20	0.18	353
with_mask	0.07	0.10	0.08	350
without_mask	0.43	0.13	0.20	350
accuracy			0.14	1053
macro avg	0.22	0.14	0.15	1053
weighted avg	0.22	0.14	0.15	1053

e) epsilon= 0.0095

Fig. 14 Show the accuracy, precision, recall, and f1-score on different epsilon values

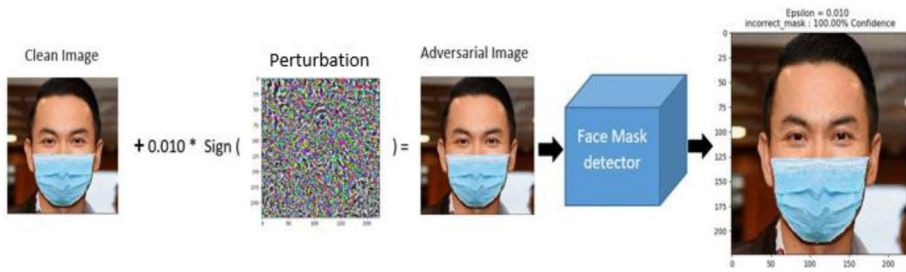


Fig. 15 Model misclassifies an adversarial image as an ‘incorrect mask’ with 100% confidence

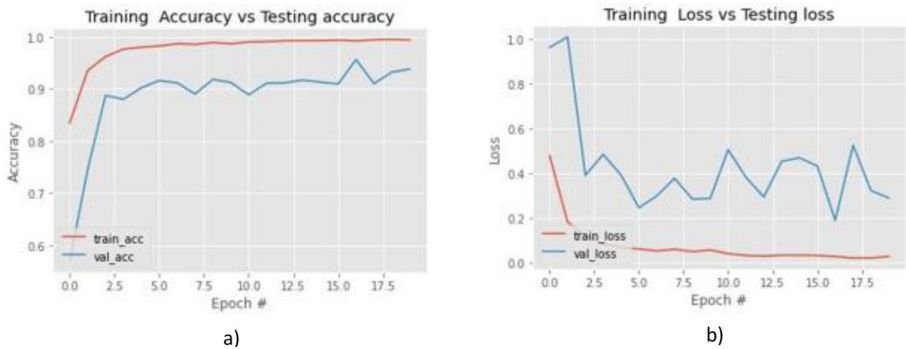


Fig. 16 **a** illustrates the training accuracy vs. the validation accuracy of the robust model on a modified dataset; **b** shows the training and testing loss

The adversarial image of the clean masked image created by FGSM forces the model to classify it wrongly as an incorrect mask with 100% confidence, as shown in Fig. 15. One thing to note here is that the model incorrectly labels the adversarial image and does so with 100% confidence. This confidence level exceeds the correctly classified clean image, which had a 99.95% classification confidence.

5.3 Robust model performance

The proposed robust framework is based on adversarial learning. However, due to the limited resources, including RAM, the adversarial generator selected only a random subset of the training dataset for adversarial example generation and added them to the subset. The model was then retrained on the modified dataset containing a subset of training data and their adversarial examples. Despite using only a portion of the training data, the model generalized well. Figure 16 depicts the robust model’s learning curve. There was a noticeable drop in accuracy when evaluated on the modified dataset from 95.78% to 92.79%, but the defended model performs incredibly on the images that once fooled it. The model maintained an accuracy of 92% on adversarial images computed on different epsilons. Table 5 shows the performance of the robust model on the FGSM attack with different epsilons. Compared to the regular model’s performance attack in Table 4, the robust model performed ideally on adversarial data. e.g., on epsilon 0.009, the non-robust model’s accuracy is 14.53%, whereas the robust model achieved 92% accuracy.

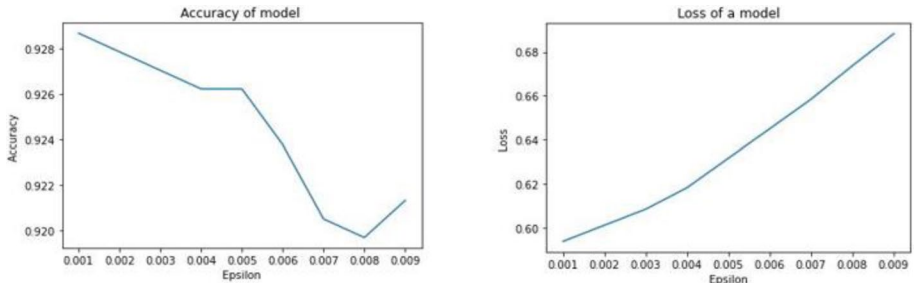


Fig. 17 Illustrates the model’s performance in terms of accuracy and loss on different epsilons values of FGSM

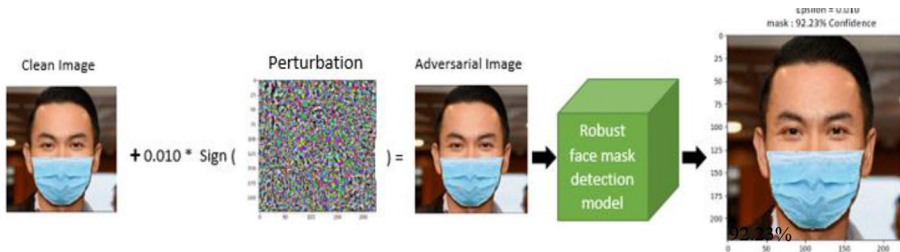


Fig. 18 Robust model correctly classifies an adversarial image of the masked face with 92.23% confidence

The plot of accuracy and loss of the robust model on different epsilon is shown in Fig. 17. The model performed exceptionally well even on varying epsilon of FGSM attack, maintaining its accuracy of about 92% on adversarial data.

The output of a robust model is shown in Fig. 18, where it correctly classifies the adversarial image of the masked face to its intended class, ‘with mask’. Note that the same adversarial image fooled the regular model into misclassifying it into an ‘incorrect mask’, as shown in Fig. 14.

Table 5 Accuracy and loss of robust model on various epsilon of FGSM

Epsilon(ϵ)	Accuracy (%)	Loss of a model
0 (clean image)	93.79	0.5855
0.001	92.87	0.5937
0.002	92.79	0.6011
0.003	92.70	0.6084
0.004	92.62	0.6182
0.005	92.62	0.6317
0.006	92.38	0.6450
0.007	92.05	0.6585
0.008	91.97	0.6738
0.009	92.13	0.6883

6 Conclusion

In this study, we demonstrated the vulnerabilities of the deep learning-based face mask detection model against adversarial examples and designed a framework for a more robust face mask detection system resistant to these attacks.

We began by introducing the custom dataset used to train the face mask identification model. The model was created by applying transfer learning of the state-of-the-art classification model MobileNetV2. The model attained a respectable accuracy of 95.83% when tested on clean inputs, as shown in Figs. 10 and 11. The model was subsequently subjected to adversarial examples computed on various epsilons of the FGSM attack. The results shown in Table 4 revealed that the models' performance declined further as the epsilon's value increased. On epsilon 0.009, the model's accuracy declined to 14.53%, which is alarming. Although we employed extremely small epsilon values to create imperceptible perturbations, the model misclassified the adversarial input with greater certainty. In Fig. 15, we noted that the adversarial image of the masked face was recognized as an incorrect mask face with 100% confidence. During the FGSM attack, the majority of the images in the test data were wrongly classified as incorrectly masked faces.

We employed the adversarial learning strategy to enhance the robustness of the model, which involved modifying the custom dataset with adversarial samples. However, in our defense approach, we modified the conventional adversarial learning approach by using only a portion of the total dataset and generating adversarial examples on the combination of epsilon values. The results in Table 5 and Fig. 17 demonstrate that the model is robust, achieving 92% accuracy on adversarial examples.

Face mask identification is critical for effectively combatting COVID-19. Thus, further study on vulnerabilities and defensive techniques is required before these systems can be implemented. We believe our findings will assist researchers in improving the security of their models and raise awareness of the need to establish face mask detection models with several protection strategies.

7 Future scope

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

- The present study is based on evaluating the model's performance on adversarial images. There is a potential to carry out adversarial attacks on the face mask surveillance systems, forcing the model to misclassify from the live video data.
- In our particular scenario, we attacked the model by utilizing the white-box FGSM attack because the specifics of the model, including its gradient, architecture, and training data, were readily available and easy to access. However, in practical cases, when the model is deployed, all of these specifics are often concealed from the users. However, in that case, there is still a scope to attack the model using Black-box strategies [7].
- In our work, we used the FGSM strategy for adversarial example computation due to its simplicity. However, in the future, we would like to attack the model using other attack strategies such as the Basic Iterative Method (BIM) [34] and Jacobian-based Saliency Map Attack (JSMA) [47].

- As discussed, there are other strategies to generate adversarial perturbation. We aim to design a universal framework that is robust to the adversarial examples generated by any strategies.
- To evaluate the model's performance on adversarial images, we computed the perturbation of each clean image separately and added it to its corresponding clean image. However, there is a method by which we can compute the universal perturbation [43], which is the singular perturbation matrix for all test samples.
- Due to limited resources, the model was retrained on a fraction of the training data and adversarial examples during adversarial learning. Although the robust model performed well on adversarial examples, there is room for improvement by retraining the model on the complete dataset and their adversarial examples.
- In the future, we aim to investigate several different defensive strategies, such as network distillation, which extracts knowledge from deep neural networks to ensure their robustness [48], and adversarial example detection during the testing stage [40].

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 on reasonable request.

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

References

1. Agarwal C, Kaur I, Yadav S (2023) Hybrid CNN-SVM model for face mask detector to protect from COVID-19. In: Artificial intelligence on medical data. Springer, Singapore, pp 419–426
2. Brown TB, Carlini N, Zhang C, Olsson C, Christiano P, Goodfellow I (2018) Unrestricted adversarial examples. arXiv preprint arXiv:1809.08352
3. Bu W, Xiao J, Zhou C, Yang M, Peng C (2017) 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, pp 458–462
4. Deng J, Guo J, Xue N, Zafeiriou S. Arcface-MFnet: Additive angular margin loss for deep face recognition with masked faces. In: 2021 IEEE International Conference on Computer Vision Workshop (ICCVW), pp 4342–4351
5. Chae S, Kwon S, Lee D (2018) Predicting infectious disease using deep learning and big data. Int J Environ Res Public Health 15(8):1596. <https://doi.org/10.3390/ijerph15081596>
6. Chen X, Liu C, Li B, Lu K, Song D (2017) Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526
7. Co KT, Muñoz-González L, de Maupeou S, Lupu EC (2019) Procedural noise adversarial examples for black-box attacks on deep convolutional networks. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, pp 275–289
8. C. (2022) GitHub - chandrikadeb7/face-mask-detection: Face mask detection system based on computer vision and deep learning using OpenCV and Tensorflow/Keras. GitHub <https://github.com/chandrikadeb7/Face-Mask-Detection>

9. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
10. Xie W, Mou W, Zhao F, Feng J (2021) WIDER FACE MASK: A face mask detection benchmark. In: 2021 IEEE International Conference on Computer Vision (ICCV), pp 8828–8837
11. Ejaz MS, Islam MR, Sifatullah M, Sarker A (2019) Implementation of principal component analysis on masked and non-masked face recognition. In: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). <https://doi.org/10.1109/icasert.2019.8934543>
12. Ellis R (2020) WHO changes stance, says public should wear masks. WebMD. <https://www.webmd.com/lung/news/20200608/who-changes-stance-says-public-should-wear-masks>
13. Fei J, Xia Z, Yu P, Xiao F (2020) Adversarial attacks on fingerprint liveness detection. EURASIP J Image Video Process 2020(1):1–11. <https://doi.org/10.1186/s13640-020-0490-z>
14. Ellis R (2020) WHO changes stance, says public should wear masks. WebMD <https://www.webmd.com/lung/news/20200608/who-changes-stance-says-public-should-wear-masks>
15. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2017) Domain-adversarial training of neural networks. In: Domain adaptation in computer vision applications, pp 189–209. https://doi.org/10.1007/978-3-319-58347-1_10
16. Ge S, Li J, Ye Q, Luo Z (2017) Detecting masked faces in the wild with LLE-CNNs. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2682–2690
17. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572
18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). <https://doi.org/10.1109/cvpr.2016.90>
19. Hirano H, Koga K, Takemoto K (2020) Vulnerability of deep neural networks for detecting COVID-19 cases from chest X-ray images to universal adversarial attacks. PLOS ONE 15(12):e0243963. <https://doi.org/10.1371/journal.pone.0243963>
20. Huang GB, Mattar M, Berg T, Learned-Miller E (2008) 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. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, Marseille [Online]. Available <https://hal.inria.fr/inria-00321923>
21. Huang B, Wang Z, Wang G, Jiang K, He Z, Zou H, Zou Q (2021) Masked face recognition datasets and validation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1487–1491
22. Huang B, Wang Z, Jiang K, Zou Q, Tian X, Lu T, Han Z (2022) Joint segmentation and identification feature learning for occlusion face recognition. IEEE Trans Neural Netw Learn Syst 1–14. <https://doi.org/10.1109/tnnls.2022.3171604>
23. Hussain SA, Salim Abdallah Al Balushi A (2020) A real time face emotion classification and recognition using deep learning model. J Phys Conf Ser, 1432(1), 012087. <https://doi.org/10.1088/1742-6596/1432/1/012087>
24. Inamdar M, Mehendale N (2020) Real-time face mask identification using facemasknet deep learning network. SSRN Electronic Journal. <https://doi.org/10.2139/ssrn.3663305>
25. Jain V, Learned-Miller E (2010) FDDB: A benchmark for face detection in unconstrained settings, vol 2, no 6. UMass Amherst technical report
26. Jain A, Duin P, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37. <https://doi.org/10.1109/34.824819>
27. Jamshidi M, Lalbakhsh A, Talla J, Peroutka Z, Hadjilooei F, Lalbakhsh P, Jamshidi M, Spada LL, Mir-mozafari M, Dehghani M, Sabet A, Roshani S, Roshani S, Bayat-Makou N, Mohamadzade B, Malek Z, Jamshidi A, Kiani S, Hashemi-Dezaki H, Mohyuddin W (2020) Artificial intelligence and COVID-19: Deep learning approaches for diagnosis and treatment. IEEE Access 8:109581–109595. <https://doi.org/10.1109/access.2020.3001973>
28. Ji Y, Zhang X, Ji S, Luo X, Wang T (2018) Model-reuse attacks on deep learning systems. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. <https://doi.org/10.1145/3243734.3243757>
29. Jignesh Chowdhary G, Punn NS, Sonbhadra SK, Agarwal S (2020) Face mask detection using transfer learning of inceptionV3. Big Data Anal 81–90. https://doi.org/10.1007/978-3-030-66665-1_6
30. Kakizaki K, Yoshida K (2019) Adversarial image translation: Unrestricted adversarial examples in face recognition systems. arXiv preprint arXiv:1905.03421

31. Khosravy M, Nakamura K, Hirose Y, Nitta N, Babaguchi N (2021) Model inversion attack: Analysis under gray-box scenario on deep learning based face recognition system. *KSII Trans Internet Inf Syst* 15(3). <https://doi.org/10.3837/tiis.2021.03.015>
32. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
33. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. *Commun ACM* 60(6):84–90. <https://doi.org/10.1145/3065386>
34. Kurakin A, Goodfellow IJ, Bengio S (2018) Adversarial examples in the physical world. In: *Artificial intelligence safety and security*. Chapman and Hall/CRC, pp 99–112
35. Larxel's face mask detection dataset: FMDD (2021) Accessed on 17 June 2021. [Online]. Available <https://www.kaggle.com/andrewmvd/face-mask-detection>
36. Li C, Wang R, Li J, Fei L (2019) Face detection based on YOLOv3. *Recent trends in intelligent computing. Communication and Devices* 277–284. https://doi.org/10.1007/978-981-13-9406-5_34
37. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: Single shot multibox detector. *Computer Vision – ECCV 2016*:21–37. https://doi.org/10.1007/978-3-319-46448-0_2
38. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021, February) Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. *Sustain Cities Soc* 65:102600. <https://doi.org/10.1016/j.scs.2020.102600>
39. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021, January) A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement* 167:108288. <https://doi.org/10.1016/j.measurement.2020.108288>
40. Lu J, Issararanon T, Forsyth D (2017) SafetyNet: Detecting and rejecting adversarial examples robustly. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. <https://doi.org/10.1109/iccv.2017.56>
41. Mijwil MM (2021) Implementation of machine learning techniques for the classification of lung X-Ray images used to detect COVID-19 in humans. *Iraqi J Sci*:2099–2109. <https://doi.org/10.24996/ij.s.2021.62.6.35>
42. Militante SV, Dionisio NV (2020) Real-time facemask recognition with alarm system using deep learning. In: *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)*. <https://doi.org/10.1109/icsgrc49013.2020.9232610>
43. Moosavi-Dezfooli SM, Fawzi A, Fawzi O, Frossard P (2017) Universal adversarial perturbations. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/cvpr.2017.17>
44. Nagrath P, Jain R, Madan A, Arora R, Kataria P, Hemanth J (2021, March) SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustain Cities Soc* 66:102692. <https://doi.org/10.1016/j.scs.2020.102692>
45. Nieto-Rodríguez A, Mucientes M, Brea VM (2015) System for medical mask detection in the operating room through facial attributes. *Pattern Recogn Image Anal*:138–145. https://doi.org/10.1007/978-3-319-19390-8_16
46. Pal B, Gupta D, Rashed-Al-Mahfuz M, Alyami SA, Moni MA (2021) Vulnerability in deep transfer learning models to adversarial fast gradient sign attack for COVID-19 prediction from chest radiography images. *Applied Sciences* 11(9):4233. <https://doi.org/10.3390/app11094233>
47. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A (2016) The limitations of deep learning in adversarial settings. In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. <https://doi.org/10.1109/eurosp.2016.36>
48. Papernot N, McDaniel P, Wu X, Jha S, Swami A (2016) Distillation as a defense to adversarial perturbations against deep neural networks. In: *2016 IEEE Symposium on Security and Privacy (SP)*. <https://doi.org/10.1109/sp.2016.41>
49. Qin B, Li D (2020) Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. *Sensors* 20(18):5236. <https://doi.org/10.3390/s20185236>
50. Qingwei Y, Yang S, Xiaodong W, Yu Z (2013) An Improved LLE Algorithm with Sparse Constraint. *J Comput Theor Nanosci* 10(12):2872–2876. <https://doi.org/10.1166/jctn.2013.3293>
51. Rahman A, Hossain MS, Alrajeh NA, Alsolami F (2021) Adversarial examples—security threats to COVID-19 deep learning systems in medical IoT devices. *IEEE Internet Things J* 8(12):9603–9610. <https://doi.org/10.1109/jiot.2020.3013710>
52. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/cvpr.2016.91>
53. Ren K, Zheng T, Qin Z, Liu X (2020) Adversarial attacks and defenses in deep learning. *Engineering* 6(3):346–360. <https://doi.org/10.1016/j.eng.2019.12.012>

54. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) MobileNetV2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/cvpr.2018.00474>
55. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
56. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). <https://doi.org/10.1109/cvpr.2016.308>
57. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan GI, Fergus R (n.d.) Intriguing properties of neural networks. arXiv 2013, arXiv:1312.6199
58. Viola P, Jones M (n.d.) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 2001. CVPR. <https://doi.org/10.1109/cvpr.2001.990517>
59. Wang YQ (2014) An analysis of the viola-jones face detection algorithm. Image Process Line 4:128–148. <https://doi.org/10.5201/ipol.2014.104>
60. Wang L, Lin ZQ, Wong A (2020) COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci Rep 10(1). <https://doi.org/10.1038/s41598-020-76550-z>
61. Sheikh B, Zafar A (2023) RRFMDS: Rapid real-time face mask detection system for effective COVID-19 monitoring. SN Comput Sci 4:288. <https://doi.org/10.1007/s42979-023-01738-9>
62. World Health Organization (2020) 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
63. Yang S, Luo P, Loy CC, Tang X (2016) Wider face: a face detection benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5525–5533

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.