Multimedia Tools and Applications (2024) 83:24749–24775
https://doi.org/10.1007/s11042-023-16439-x

Unlocking adversarial transferability: a security threat towards deep learning-based surveillance systems via black box inference attack- a case study on face mask surveillance

Burhan Ul Haque sheikh[1] · **Aasim Zafar**[1]

Received: 21 January 2023 / Revised: 16 July 2023 / Accepted: 31 July 2023 /
Published online: 10 August 2023

Abstract
With the increasing demand for reliable face mask detection systems during the COVID-19 pandemic, deep learning (DL) and machine learning (ML) algorithms have been widely used. However, these models are vulnerable to adversarial attacks, which pose a significant challenge to their reliability. This study investigates the susceptibility of a DL-based face mask detection model to a black box adversarial attack using a substitute model approach. A transfer learning-based face mask detection model is employed as the target model, while a CNN model acts as the substitute model for generating adversarial examples. The experiment is conducted under the assumption of a black-box attack, where attackers have limited access to the target model's architecture and gradients but access to training data. The results demonstrate the successful reduction of the face mask detection model's classification accuracy from 97.18% to 46.52% through the black-box adversarial attack, highlighting the vulnerability of current face mask detection methods to such attacks. These findings underscore the need for robust defense measures to be implemented in face mask detection systems to ensure their reliability in practical applications.

Keywords Adversarial example · Vulnerability · Adversarial attacks · Face mask detection · Security in DL models

1 Introduction

The global COVID-19 pandemic has brought forth numerous challenges, both in terms of health and economy [6]. In response to the rapid spread of the virus, the World Health Organization declared it a pandemic on March 11, 2020 [47]. Video surveillance has emerged as a crucial tool in combating the spread of COVID-19. One significant application is face mask detection. By utilizing advanced image analysis and machine learning

✉ 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

algorithms, video surveillance systems can accurately identify individuals not wearing masks in public spaces [9, 10, 46]. This technology helps enforce mask mandates, ensuring public health guidelines are followed to reduce virus transmission. Video surveillance with face mask detection plays a vital role in maintaining safety, supporting health measures, and contributing to the overall management of the pandemic. Conventional procedures, such as manually monitoring face mask compliance is impractical and prone to errors, necessitating the development of automated face mask surveillance systems. As a result, there is a growing need for automated face mask surveillance systems that can detect whether individuals are wearing masks or not and alert authorities when they are not.

In the battle against the COVID-19 pandemic, the utilization of artificial intelligence, particularly machine learning (ML) and deep learning (DL), has emerged as a powerful and indispensable tool [3, 7, 20]. The adoption of machine learning and deep learning algorithms in face mask surveillance systems has enabled the development of advanced models capable of accurately identifying face mask violations. These models have achieved impressive detection accuracy and efficiency results by leveraging large-scale datasets and state-of-the-art neural network architectures. Transfer learning, a popular approach, allows researchers to leverage the knowledge and representations learned from pre-trained models, such as ImageNet [8], and apply them to face mask detection tasks. This approach reduces the need for extensive training on limited face mask datasets and enables the model to generalize well to unseen data [37]. Furthermore, researchers have explored novel architectural designs further to improve the face mask recognition model's performance [43].

However, recent research has highlighted a concern surrounding the vulnerability of deep learning models utilized in face mask surveillance systems to adversarial attacks during the inference phase [15, 36]. Adversarial attacks are a technique malicious actors use to mislead a deep learning model by modifying input data samples with a modest perturbation, causing the model to misclassify during inference [33]. This raises concerns about the widespread deployment of surveillance systems based on DL algorithms, especially in insecure situations [45]. Adversarial attacks extend beyond image and video classification models. In [34], they have illustrated instances of attacks on network intrusion detection systems, showcasing the vulnerability of such systems to adversarial manipulation.

This study highlighted that face mask detection systems based on DL algorithms are vulnerable to black-box adversarial attacks. While white-box attacks are easier to implement, they do not always reflect real-world scenarios where attackers may have limited access to the system. Thus, we utilized a black-box strategy, assuming the adversary did not know the target model's architecture, parameters, or gradients. We employed the untargeted FGSM algorithm [12] to craft adversarial examples that caused the face mask model (target model) based on MobileNetV2 [35] to misclassify with high confidence. These examples were generated using a substitute model we trained from scratch on the same dataset. The results showed that the target model's performance significantly degraded, emphasizing the importance of developing robust models that can withstand such attacks in real-world scenarios. Overall, our research provides valuable insights into how attackers might approach these systems and the importance of developing models that can resist such attacks in real-world scenarios.

1.1 Taxonomy

In this section, we present key terminologies related to attacks on DL-based face mask detection systems.

- Adversary: An entity that aims to deceive a machine learning model by causing it to misclassify a valid-looking input. For instance, given an input (I) and its original class (c), the adversary's objective is to make the model predict a modified input (I′) as a different class (c'), while a human annotator would still label I′ as c.
- White-box Attack Approach: In this attack scenario, the adversary possesses complete access to the system, including the gradient, parameters, hyperparameters, and training dataset of the model. This approach allows the adversary to exploit extensive knowledge of the model's internals.
- Black-box Attack: In this case, the attacker does not have access to important information about the model, such as its gradient, parameters, and training data. This method applies to real-world scenarios where the model can only be accessed through a restricted application programming interface (API) that restricts interactions to input queries.
- Targeted Attack Approach: The goal of this attack is to manipulate the system into generating a desired target label. For instance, if the input image depicts a person wearing a mask, the objective is to compel the model to produce an inaccurate label, such as identifying it as an unmasked face.
- Untargeted Attack Approach: In this attack, the primary focus is on causing the model to misclassify the input, irrespective of the specific output label.
- Perturbation: It is a disturbance or noise, denoted by, say $\delta\epsilon$, that is added to the original input I to create perturbed input I,' i.e., $I' = I + \delta\epsilon$. The quantity of disturbance is a difficult task. It must be small enough to be imperceptible to humans yet large enough for the classifier to misclassify it.
- Adversarial Input: It is an input that an adversary has crafted following the perturbation. It forces the model to "jump the classification boundary," leading to misclassifying an adversarial input.

By presenting these definitions, we establish a foundation for discussing various attack strategies employed against deep learning-based face mask detection models, enabling a comprehensive understanding of the adversarial landscape in this domain.

1.2 Contribution

In this study, we made the following contributions to the field of face mask detection using deep learning techniques:

- Development of a Deep Learning-Based Model: We created a face mask detection model using the transfer learning approach with MobileNetV2 as the base architecture. This model achieved a high accuracy of approximately 97% on clean data samples, demonstrating its effectiveness in accurately detecting face masks.
- Identification of Vulnerability to Attacks: Our research revealed that face mask detection models, whether based on conventional machine learning (ML) or deep learning (DL) approaches, are susceptible to attacks even when the attacker has limited access to the model details. This highlights the importance of considering security aspects in the deployment of such models.
- Evaluation of Model Performance under Adversarial Attacks: We conducted experiments to assess the performance of the target model when exposed to various adversarial examples generated using different epsilon values. By evaluating the

model's robustness to these attacks, we gained insights into its vulnerability and the potential impact of adversarial perturbations on classification accuracy.

1.3 Paper organization

After an introduction, our paper progresses as follows: Section 2 offers a comprehensive literature review on face mask detection and adversarial attacks, summarizing previous research and insights in the field. In Section 3, we discussed the dataset utilized for our study. Section 4 presents our methodology, which involves training a target face mask classifier, training a substitute model from scratch, and employing a black box strategy to attack the target model. Section 5 presents the experimental results and detailed analysis of the target model's performance under various attack scenarios. Section 6 explores the potential limitations of our study and outlines future research avenues. Finally, we conclude the paper by summarizing the key findings obtained from our research.

2 Related work

The literature section of our paper is divided into three subsections. The first subsection comprehensively reviews DL-based face mask models. The second subsection focuses on adversarial attacks on DL models. Lastly, we specifically examine the essential works of adversarial attacks on COVID-19 monitoring systems. By organizing the literature review in this manner, we ensure a thorough exploration of both face mask detection models and the vulnerabilities associated with adversarial attacks.

2.1 Face mask detection systems

In [19], they used a Single Shot Multibox detector [22] and InceptionV3 architecture [42] to detect faces and extract features from images. The system is tested on various modeling parameters and achieves 0.98 of average accuracy on two custom datasets, which contain diverse types of masks and unmasked images of humans collected in different environments.

The article [40] combines Efficient-YOLOv3 for mask detection [32] and MobileNet for mask classification. The proposed algorithm can differentiate qualified masks (disposable medical masks and N95) from unqualified masks (sponges, scarves, cotton, etc.) with an average accuracy of 0.9784.

One notable study by [14] introduced a face mask detection model specifically designed for real-time and static videos. Their model was trained and evaluated using a Kaggle dataset comprising approximately 4000 images. The proposed model achieved an accuracy rate of 98%, demonstrating its effectiveness in accurately detecting face masks.

In an effort to address the wider issue of implementing standard operating procedures (SOPs) like wearing face masks and practicing social distancing, a study conducted by researchers [18] introduced a comprehensive dataset comprising 10,000 outdoor images. Their objective was to create an automated system capable of detecting face masks and measuring social distance. To improve real-time performance, they fine-tuned existing object detection networks and optimized the YOLO-v3 architecture. The proposed pipeline demonstrated a notable 5.3% increase in accuracy compared to the baseline version, suggesting its potential effectiveness in enforcing SOPs.

In real-time face mask detection, [4] proposed a solution utilizing an ensemble of three fine-tuned state-of-art classifications: Inception-v3, VGG-16 and ResNet50. Their framework consisted of offline training and online deployment phases, enabling real-time face detection in live videos. Notably, their model achieved exceptional performance, with an average classification accuracy of 0.9997, recall of 99.7,% F1-score of 99.7%, precision of 99.7% and kappa coefficient of 99.4%. These results surpassed several state-of-the-art methods, highlighting the effectiveness and superiority of their approach.

Furthermore, authors of [23] addressed the challenge of recognizing faces with masks and proposed a novel network architecture called the Upper-Lower Network (ULN) for mask-robust face recognition. Their method involved generating facial images with masks and utilizing the ULN along with a specifically designed loss function to efficiently recognize faces with masks. Comparative evaluations demonstrated that the proposed ULN-based approach outperformed other state-of-the-art face recognition methods, showcasing its effectiveness in handling face recognition tasks in the presence of masks.

In a recent study [39], researchers trained two advanced object detection models, Faster R-CNN and YOLOv3, using a dataset that included images of individuals wearing and not wearing face masks. The primary objective was to utilize these models to draw bounding boxes around faces and track the number of people wearing masks daily. The performance of both models was assessed based on precision rate and inference time. The findings revealed that Faster R-CNN demonstrated higher precision, while YOLOv3 showcased the advantage of a faster frame rate and single-shot detection, making it more suitable for real-world surveillance camera applications. The selection between the two models depends on resource availability, as Faster R-CNN requires powerful GPUs, whereas YOLOv3 can be deployed on mobile phones. Additionally, the approach can be fine-tuned to accommodate the specific field of view of the camera, enhancing its adaptability in practical scenarios.

In another research work by [26], an automated approach for detecting violence of face masks and social distancing was proposed to mitigate the transmission of diseases like COVID-19. The system utilized a two-cascaded YOLO architecture, where the first cascade focused on detecting humans and calculating social distance, while the second cascade aimed to detect human faces with or without masks. To extract relevant features, the study employed transfer learning and incorporated a Local Binary Patterns (LBP) layer to capture the task's general and specific characteristics. The implemented system achieved an average precision of 66% in human detection using Resnet50 and 95% average precision in mask detection using Darknet19 + LBP with YOLO.

2.2 Adversarial attack

The literature on adversarial attacks is organized into white-box and black-box attacks.

2.2.1 White box adversarial attack

Since it was proposed in [41] that adversarial techniques could be used to attack neural networks, studying these techniques has become a hot topic in artificial intelligence. Their method proposes an optimization function to find adversarial examples, as shown in Eq. (1). This function computes an additive perturbation that slightly distorts an image and forces a model to classify it incorrectly. Equation (1) can be read as follows: Given an image x, it identifies an alternative image x' that is identical to x under L2 distance but is

labeled differently by the classifier. This is achieved by calculating the lowest value of c > 0 for which the minimizer 'δ' of the following problem achieves the constraint F(x+δ) = ℓ'.

$$\begin{aligned} &\text{Min } c \parallel \delta \parallel_2 + loss(x+\delta, l) \\ &\quad \delta \\ &s.t \quad x+\delta \in [\,0,1\,]^n \end{aligned} \tag{1}$$

Where x is the clean input image, loss(.,.) is the loss of the classifier, l is the associated label and δ is the magnitude of the perturbation.

In the work presented by [33], the FGSM attack approach was introduced, which involves computing gradients in a single step with respect to the pixel values of the input image. These gradients are then subjected to a sign operation, as depicted in Eq. (2):

$$\delta = \varepsilon * \text{sign}\,(\nabla * loss\,(x,\ \ l\,)) \tag{2}$$

Here, ' l 'represents the true label of the clean image ' x ', and x calculates the gradient of the loss function with respect to the current model parameters, specifically concerning the input ' x '. The perturbation's magnitude, denoted as ε, determines the level of noise added to the input image to generate the adversarial example. Ultimately, the adversarial example, denoted as x ', is obtained by adding the perturbation δ to the original input image x. i.e. $x^{'} = x + \delta$.

In [21], the Basic iterative method (BIM) is proposed to compute the adversarial example iteratively. The BIM extends the FGSM by applying the FGSM iteratively on the input image with a small step size in each iteration, as shown in Eq. (3). By doing so, the BIM generates a sequence of adversarial examples that are gradually optimized to be more effective in evading the target model's classification.

$$x_{i+1} = Clip_{x,\varepsilon}\left\{ x_p^i + \alpha\ \text{sign}\left(\nabla * loss\left(x_p^i, l\right)\right)\right\} \tag{3}$$

where, x_p^i signifies the perturbed image at the ith iteration, $Clip_{x,\,\varepsilon}$ conducts per-pixel clipping on the image at ε and α specifies the step size (Usually, α = 1). This method begins with $x_p^0 = x$ and continues until the number of iterations specified by the formula min $\lfloor$ (α +4, 1:25 α) $\rfloor$.

The paper [25] proposed a universal adversarial perturbation (UAP). It is a perturbation designed to be universally applicable to any input image and generated using an iterative optimization process. The optimization process seeks to find a single noise pattern that, when added to any image, causes the model to misclassify it. Once the UAP is generated, it can be added to any input image to create an adversarial example. The dataset μ in Eq. (4) contains all the samples. p denotes probability and is typically $0 < \text{ʒ} << 1$. The objective is to find δ, which could fool F(.) on almost any sample from μ.

$$\begin{aligned} &\text{F}\,(x+\delta\,) \neq \text{F}\,(x\,), \quad \text{for most } x \sim \mu \\ &\text{s.t } \parallel \delta \parallel \text{p} < \xi \\ &p_{x \sim \mu}\ (\text{F}\,(x+\delta\ ) \neq \text{F}\,(x\,)) \geq 1 - \text{ʒ} \end{aligned} \tag{4}$$

The parameter ξ determines the magnitude of the perturbation vector and quantifies the intended fooling rate for each image sampled from the distribution μ.

2.2.2 Black box adversarial attack

In [29], they introduced a new attack method called the "substitute model attack," which involves training a surrogate model to approximate the behavior of the target model and

using it to generate adversarial examples. This approach was effective against various machine-learning models, including image classifiers and spam filters.

The paper [17] introduced a new attack method called the "query-efficient black-box attack," which involves generating a small number of queries to the target model and using them to construct a locally linear approximation of the model. This approach was effective against various machine learning models, including image classifiers and speech recognition systems.

Researchers in [11] directed their attention towards the generation of adversarial examples in the domain of textual data. Their objective was to deceive deep classifiers by introducing novel scoring strategies that identify the most significant words for modification. They proposed an approach known as DeepWordBug, which involved applying simple character-level transformations to the highest-ranked words to minimize the perturbation's edit distance. To evaluate the effectiveness of their approach, they conducted experiments on real-world text datasets, including Enron spam emails and IMDB movie reviews.

Furthermore, the paper by [5] proposed an effective black-box attack leveraging zeroth order optimization (ZOO). Unlike other approaches, the ZOO attack estimates the targeted deep neural network (DNN) gradients without requiring the training of substitute models. The findings showed that the ZOO attack was on par with state-of-the-art white-box attacks and outperformed existing black-box attacks employing substitute models.

2.3 Adversarial attacks on COVID-19 monitoring systems

In a recent study conducted by [16], the COVID-Net architecture, specifically designed for COVID-19 classification based on chest X-ray images, was subjected to untargeted universal adversarial perturbation (UAP) and targeted attacks [44]. The findings revealed the susceptibility of COVID-Net to tiny UAPs, with a success rate of over 85% for untargeted attacks and over 90% for targeted attacks using image translation techniques. The untargeted UAPs led to the misclassification of most chest X-ray images as COVID-19 instances, while targeted UAPs caused the models to classify most chest X-ray images into a specific target class.

In [30], they developed an image-based medical adversarial attack method designed to consistently induce adversarial perturbations on medical images. The adversarial perturbations needed for the proposed method are generated by an iterative process that prioritizes increasing the deviation loss term while simultaneously reducing the loss stabilization term. They continue their investigation by analyzing the KL divergence of the suggested loss function, and they discover that the loss stabilization component causes the perturbations to be updated towards a fixed objective location while simultaneously diverging from the ground truth.

In another study by [27], a transfer learning-based COVID-19 diagnosis model was proposed. The VGG16 and Inception-v3 models, which are widely recognized classification methods, were employed. The results showed that increasing the perturbation from 0.0001 to 0.09 significantly impacted the accuracy of the VGG16 model, causing a decline of over 90% for X-ray images. The Inception-v3 network also experienced a decline in accuracy. The vulnerability of CT imaging to the FGSM attack was also demonstrated, revealing the potential for misdiagnosis.

Furthermore, [31] investigated various applications employed for diagnosing COVID-19 and proposed adversarial attacks tailored to each application. These applications included DL-based QR codes for immunization certificates, mask recognition from live

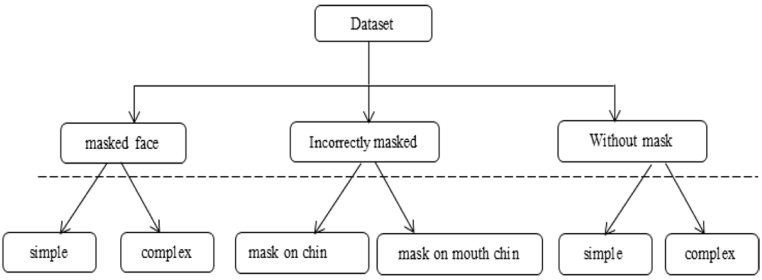


Fig. 1 It illustrates the dataset's structure in a visual format, with a dashed line separating the upper and lower sections. The upper section is specifically designed for detecting face masks, while the lower section contains subcategories that can be applied to other computer vision tasks. For example, the without mask subcategory in the lower section can be used for more complex applications such as occlusion detection

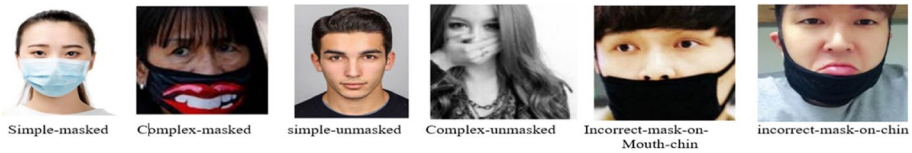


Fig. 2 Example of images in the dataset

camera feeds, explainability of GRAD-CAM DL algorithms, noninvasive biometrics detection, COVID-19 recognition from CT scan images, social distancing identification from live camera feeds, and COVID-19 recognition from X-ray image analysis. The existing adversarial methods, such as MI-FGSM, FGSM, Deepfool, C&W, L-BFGS, BIM, Foolbox, JSMA and PGD and, were tested in this study.

The literature demonstrates that many attacks on COVID-19 monitoring systems rely on a white-box adversarial approach. However, in real-world scenarios, the assumption of having access to model details is often impractical. For instance, the model may be accessible through an Application Programming Interface (API) that conceals the underlying model information. The proposed work offers more feasible insights and solutions in such practical scenarios.

3 Dataset description

We used the Sophisticated Face Mask Dataset from https://www.kaggle.com/datasets/shiekhburhan/face-mask-dataset to train our target and substitute models. The dataset is organized into three categories: with mask, incorrectly masked, and without mask, with each primary category further divided into subcategories based on specific characteristics. Figure 1 depicts the dataset's organization, while Fig. 2 displays a sample of the images included in the dataset.

The dataset contains different images, including mask-on-chin images where the mask is only placed on the chin, exposing both the nose and lips. Mask-on-chin-mouth images show the mask covering only the chin and mouth, leaving the nose uncovered.

Simple-with-mask images contain samples of plain face masks without any logos or textures, while complex-with-mask images consist of elaborate face masks with designs, patterns, or logos. Simple-without-mask images do not have any occlusion, while complex-without-mask images include people's faces obscured by features like beards, long hair, or hands covering their faces. The dataset has a total of 14,535 images and Table 1 shows the distribution of the dataset in each class.

4 Methodology

We developed a face mask detection model (target model) based on the transfer learning of MobileNetV2. Then, we launched a black box attack, assuming the adversary did not know the target model's details, such as architecture, weights, and gradients. To accomplish this, we built a substitute CNN model from scratch that closely mimicked the target model. We trained and fine-tuned the substitute model until it achieved satisfactory accuracy and approximated the target model. With full access to the substitute model, we employed the white-box attack strategy (FGSM) to generate an adversarial example for it. When these adversarial examples began to misclassify the substitute model, we used them to attack the target model, leveraging the transferability property of adversarial examples. Our evaluation of the target model's performance on both clean and adversarial data showed a significant degradation in the face mask detection model's performance due to the proposed black box attack. The overall framework of the black box adversarial attack on the face mask model is shown in Fig. 3.

4.1 Transfer learning of MobileNetV2 (Target model)

Transfer learning is a strategy by which knowledge learned by a model from the large dataset is transferred to the new model. It improves the new model's performance even when trained on insufficient data. We developed a face mask detection model by employing the transfer learning of the state-of-the-art model MobileNetV2 that has learned its weights on the ImageNet Dataset [8].

Why MobileNet-V2? The architecture of MobileNetV2 is designed to be lightweight and efficient, which makes it ideal for devices with limited resources, such as low memory and processing. One of the key features of MobileNetV2 is depthwise separable convolutions, which greatly reduces the number of parameters and computations required by the model compared to traditional convolutional layers. This allows the model to run more efficiently on devices with limited resources and power. Additionally, MobileNetV2 uses linear

Table 1 Distribution of data in each class

	Incorrect mask images	With mask	Without mask
Simple	–	4000	4000
Complex	–	789	746
Mask on mouth	2500	–	–
Mask on mouth chin	2500	–	–
Total	5000	4789	4746

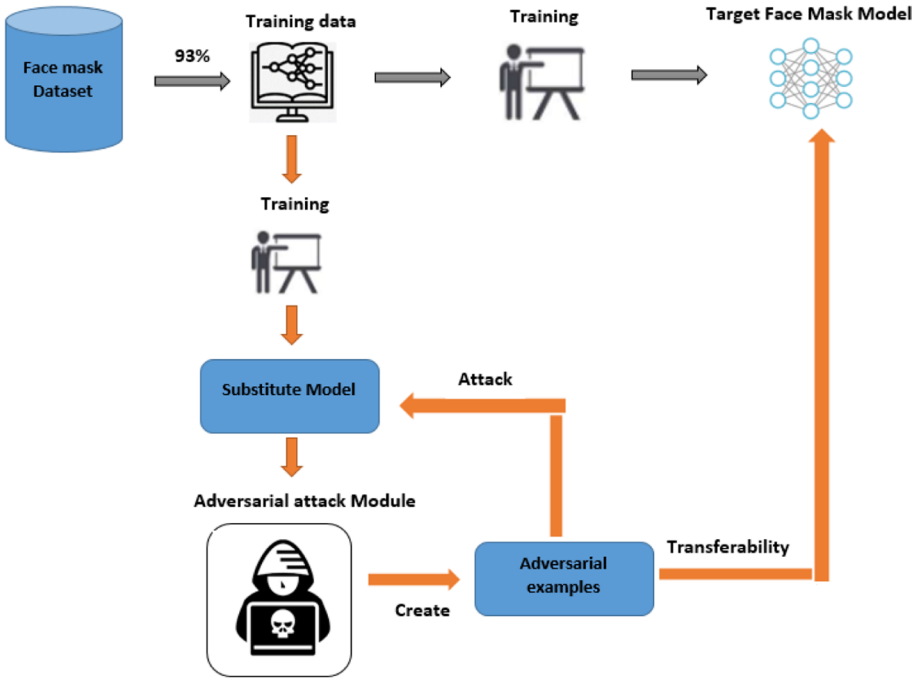


Fig. 3 It illustrates the overall design of the black box attack on the developed face mask detection model (target model)

bottlenecks and inverted residual blocks, which further reduce the computational cost of the model while maintaining accuracy. This makes MobileNetV2 a good choice for image classification tasks on mobile, embedded devices and CCTV surveillance systems, where computational resources are limited.

Transfer learning in our approach facilitated the utilization of pre-trained weights as a starting point for training. The target model underwent fine-tuning by removing the last layer of MobileNetV2 and incorporating four trainable layers: Dense 128, Dense 64, Dense 32, and Dense 3. To mitigate overfitting, a dropout layer was inserted between these layers. Moreover, the base layers were kept frozen, rendering them non-trainable. The final layer of the model consisted of three neurons representing the respective output classes: incorrect mask, masked faces, and unmasked faces. The architecture of the face mask detection model is depicted in Fig. 4.

4.2 Substitute model architecture

The CNN architecture of the substitute model consists of 9 trainable layers. Dropout and pooling layers are between these trainable layers to reduce overfitting and dimensions. The architecture of the substitute model is shown in Fig. 5 and contains the following:

a. Input layer: The input layer defines the shape of the input images that the model will receive. The input shape of the image for the substitute model is set to 224*224*3, the

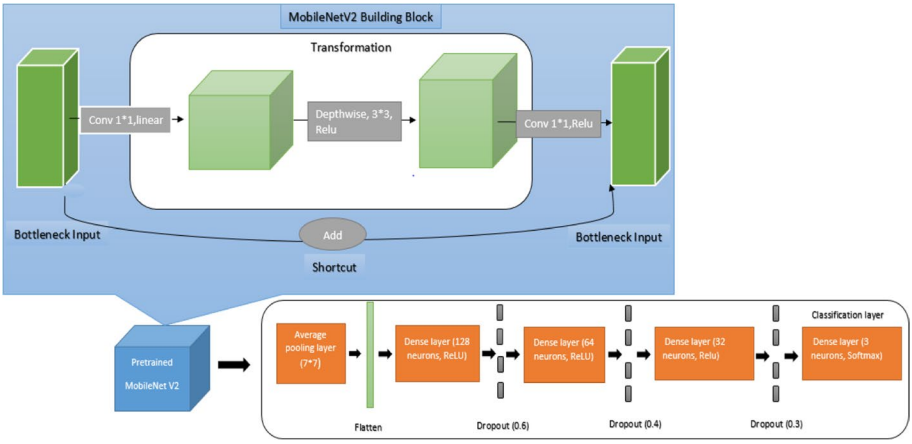


Fig. 4 It illustrates the architecture of the face mask model with Mobilenet-V2 as the backbone

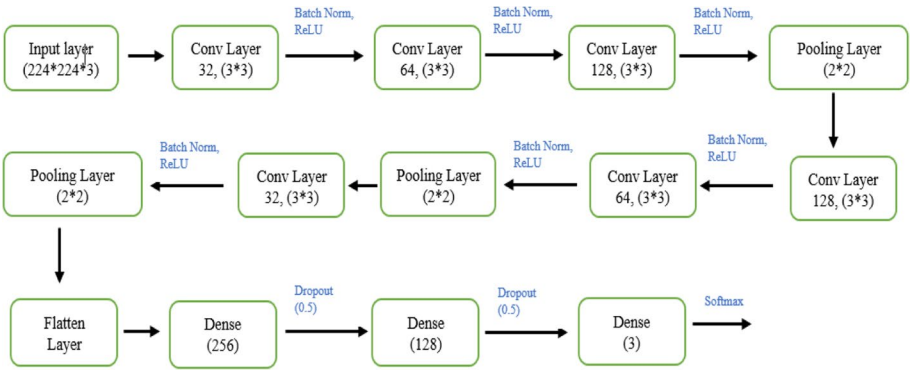


Fig. 5 illustrates the architecture of the substitute model

same as the target model. It is because both models should be compatible in terms of the size of an image they accept as input. Since we use a substitute model to generate an adversarial example for the target model, the input image size for which adversarial perturbation is computed should be the same as the input accepted by the MobileNetV2.

b. Convolutional layer: The first convolutional layer has 32 filters with a size of (3,3) and uses the 'same' padding mode to preserve the spatial dimensions of the input image. The activation function used in this layer is the Rectified Linear Unit (ReLU).
c. Batch Normalization layer: Batch normalization is a technique to improve training stability and speed up the training process. It normalizes the previous layer's output by subtracting the batch mean and dividing it by the batch standard deviation.
d. Convolutional layer: The second convolutional layer has 64 filters with a size of (3,3) and uses the 'same' padding mode. The activation function used in this layer is ReLU.
e. Batch Normalization layer: This is another batch normalization layer that follows the second convolutional layer.
f. Convolutional layer: The third convolutional layer has 128 filters with a size of (3,3) and uses the 'same' padding mode. The activation function used in this layer is ReLU.

g. Batch Normalization layer: This is another batch normalization layer that follows the third convolutional layer.
h. Max Pooling layer: The first max pooling layer has a pool size of (2,2) and reduces the spatial dimensions of the feature maps produced by the previous convolutional layers.
i. Convolutional layer: The fourth convolutional layer has 128 filters with a size of (3,3) and uses the 'same' padding mode. The activation function used in this layer is ReLU.
j. Batch Normalization layer: This is another batch normalization layer that follows the fourth convolutional layer.
k. Convolutional layer: The fifth convolutional layer has 64 filters with a size of (3,3) and uses the 'same' padding mode. The activation function used in this layer is ReLU.
l. Batch Normalization layer: This is another batch normalization layer that follows the fifth convolutional layer.
m. Max Pooling layer: The second max pooling layer has a pool size of (2,2) and reduces the spatial dimensions of the feature maps produced by the previous convolutional layers.
n. Convolutional layer: The sixth convolutional layer has 32 filters with a size of (3,3) and uses the 'same' padding mode. The activation function used in this layer is ReLU.
o. Batch Normalization layer: This is another batch normalization layer that follows the sixth convolutional layer.
p. Max Pooling layer: The third max pooling layer has a pool size of (2,2) and reduces the spatial dimensions of the feature maps produced by the previous convolutional layers.
q. Flatten layer: It flattens the previous layer's output into a 1-dimensional vector that the fully connected layers can process.
r. Fully connected layer: The first fully connected layer has 256 units and uses the ReLU activation function. A dropout layer with a rate of 0.5 is added after this layer to prevent overfitting.
s. Fully connected layer: The second fully connected layer has 128 units and uses the ReLU activation function. Another dropout layer with a rate of 0.5 is added after this layer to prevent overfitting further.
t. Output layer: The output layer has three units, one for each output and uses the softmax activation function.

4.3 Training of target and substitute model

We utilized 93% of the entire dataset during the training process of both the substitute and target models. We reserved 7% for testing the models. Notably, the target model attained a high level of accuracy during both the training and testing phases, achieving rates of 98.41% and 97.18%, respectively. These impressive results indicate that the model performed well on previously unseen data. To develop the substitute model, we trained it until it achieved an acceptable level of accuracy that was similar to the target model. Specifically, we stopped training the model when it reached an accuracy rate of 98.13% on the training data and 96.83% on the testing data. This outcome suggests that the substitute model performed well and could accurately predict outcomes similar to those produced by the target model. The specific hyper-parameters used to develop both models can be found in Table 2. Furthermore, we have compared the complexity of both models in terms of size, layers and number of trainable parameters in Table 3.

Table 2 Hyperparameter settings used for the face mask detection model (Target model) and a substitute model

Hyperparameter	Target Model	Substitute Model
Learning rate	1e-3 with decay rate = learning rate/epoch number;	1e-4 with decay rate = learning rate/epoch number;
Batch size	32	32
Epochs	30	30
Loss	CategoricalCrossentropy	CategoricalCrossentropy
Optimization	ADAM	ADAM
Dropout rate	0.6, 0.4, 0.3	0.5, 0.5
Output layer size	224*224*3	224*224*3
Input layer size	224 *224*3	224 *224*3

Table 3 Complexity of the proposed models regarding memory and trainable parameters

Model	Layers	Size	Number of trainable parameters
Target Model	53 + 4 new trainable	18.4 MB	~ 4.6 million
Substitute Model	18 (15 trainable)	27.44 MB	~ 7.2 million

4.4 Generation of adversarial examples via substitute model

Since we assume that the training dataset is available during the attack, we don't have to collect a set of input-output pairs from the target model to train the substitute model. Next, a substitute model is trained using the same dataset. The substitute model has a complexity similar to the target model and is trained to predict the same outputs as the target model. Then, we used the trained substitute model to create adversarial examples by selecting an input that the target model correctly classifies and finding a small perturbation that causes the substitute model to misclassify it. The perturbation is then added to the original input to obtain an adversarial example. The generated adversarial example is tested on the target model. If the target model misclassifies the example, then the attack is successful. This process is repeated for multiple inputs to evaluate the effectiveness of the attack.

In the context of adversarial attacks on the DL-based face mask detection model, we introduce a set of notations to represent the key components of the problem. Specifically, the face detection model is denoted as F, the original input as x, the class of input or desired output of a model as l, the target class as t, the perturbation as r and the parameters of the F as θ.

Given a sample x from the sophisticated face mask dataset, for example, an image with no mask on a face, we compute a perturbation r for x, which, when added to the original image x, generates an adversarial example $x' = x + r$. The objective is to generate an adversarial example x' that is indistinguishable from the original input x to humans, i.e., x' should appear to humans as class l .

Formally, we can express the above as follows:

Let $x \in X$ be a sample from the sophisticated face mask dataset, where X is the dataset of all possible samples. The desired output of the face detection model is denoted as $l \in$, where L is the set of possible output classes (with-mask, without-mask, incorrect-mask). Let r be the perturbation added to the original input x. The adversarial example x

'is obtained as x '$= x + r$. To ensure that x 'appears imperceptible to humans, we enforce the constraint that the distance between x and x 'is small. Specifically, we minimize the distance $\| r \|$ subject to the condition that $F(x') = l$, where F is the face detection model.

Finally, we evaluate the effectiveness of the adversarial attack by computing the success rate, which is the probability that the face detection model misclassifies x 'as a target class t, where $t \neq l$. Formally, we can express the success rate as Eq. (5).

$$F(x') = (t \mid \quad \|r\| \leq \varepsilon) \tag{5}$$

where ε is a small positive constant representing the maximum allowable perturbation.

Given that untargeted adversarial attacks have been employed on the developed face mask detection systems, the objective function is defined as Eq. (6).

$$\begin{aligned} &\underset{X'}{\text{Min}} \left\| x' - x \right\| \\ &\textit{Subject to}: \\ &\qquad F(x) = \ell \\ &\qquad F(x') = \ell' \\ &\qquad \ell \neq \ell' \end{aligned} \tag{6}$$

The objective is to minimize the Euclidean distance between the original input x and the adversarial example x ', subject to the conditions that the model classifies x correctly as ℓ, while misclassifying x 'as a different class ℓ'. The objective function emphasizes the importance of keeping the perturbation r small, ensuring that it is imperceptible to humans while still causing the model to misclassify the input.

Adversarial examples were generated by calculating the perturbation for each test image and adding it to the corresponding clean image. To ensure simplicity and effectiveness, the FGSM was employed to compute the perturbation. Instead of using a single epsilon value, a range of values was utilized to generate diverse adversarial examples from the same image. This approach allows for an analysis of the target model's performance across various epsilon values, rather than just one fixed value. The specific methods and classes from the TensorFlow library used to generate these adversarial attacks are provided in Table 4. The process of developing black box attacks is outlined in Algorithm 1.

In Step 1, the input test image is preprocessed by normalizing its pixel values and resizing it to a fixed size (224*224*3). In Step 2, the face mask detection model is used to predict the label of the preprocessed test image. This gives the baseline result that the algorithm will attempt to subvert. In Step 3, the loss is calculated as the difference between the true label of the input image and the predicted label from the face mask detection model. This is the metric that the algorithm will try to minimize by generating adversarial examples. In Step 4, the gradient of the loss with respect to the input image is calculated using backpropagation. This gradient represents the direction of maximum increase in the loss and will be used to generate the adversarial perturbation. For each epsilon value in the set of epsilon values (E), the algorithm generates an adversarial example in Step 5 by adding a perturbation to the input image. The perturbation is scaled by the epsilon value and its direction is determined by the sign of the gradient. In Step 5.1, the signed gradient is scaled to limit the size of the perturbation that will be added to the input image. This scaling ensures that the perturbation is within a reasonable range and does not introduce noise that is too large. In Step 5.2, the perturbation is added to the input image to generate the adversarial example. The epsilon

Table 4 Tensorflow methods are used to generate adversarial examples for the proposed face mask system

Method	Description
Gradient Tape	It is a feature provided by TensorFlow that enables automatic differentiation and gradient computation. It functions as a recording mechanism during the execution of operations on tensors, allowing for the calculation of gradients with respect to variables. Using the Gradient Tape context manager, TensorFlow records the operations performed on tensors within the context, creating a graph for subsequent gradient computation. We used it to record the operation performed on the image during the forward propagation.
Gradient	It is a function used to calculate the gradients of a tensor with respect to other variables or tensors. It enables efficient computation of gradients for tasks like backpropagation and parameter optimization in machine learning. The function returns the gradients by providing the tensor and the list of variables/tensors, facilitating automatic differentiation and gradient-based optimization methods.it is implemented as *gradienttape.gradient()*
Categoricalcrossentropy	It is employed to compute the loss between the predicted output and the original label. This function allows us to measure the dissimilarity between the predicted and true labels.Implemented as *tensorflow.losses.categoricalcrossentrpy*().
Sign	Returns the element-wise sign of a number. In our case, we used tf.sign(x), where -1 is assigned if $x < 0$, 0 if $x = 0$, and 1 if $x > 0$. This function is useful for obtaining the sign of the gradients during the adversarial perturbation computation.

Algorithm 1 Black Box adversarial attack on face mask detection model

Input: Test image (x), true label (y_true), target_model (F), substitute_model (S) set of epsilon values (E)

Output: Adversarial examples (x_adv) for each epsilon value in E

Step 1: Normalize and resize test image
x = normalize_resize(x)

Step 2: Predict label for test image using face mask model
y_pred = S.predict(x)

Step 3: Calculate loss between true label and predicted label
loss = categorical_cross_entropy(y_true, y_pred)

Step 4: Calculate gradient of loss with respect to input image
grad = gradient(loss, x)

Step 5: For each epsilon value in E, generate adversarial example
for e in E:
STEP 5.1: Scale the signed gradient to limit the perturbation: scaled_signed_gradient = signed_gradient / abs(signed_gradient).max()
STEP 5.2: Add the epsilon multiplied by the scaled signed gradient to the image:
x_adv= image + e * scaled_signed_gradient
STEP 5.3: Clip the pixel values of the adversarial image to the valid range:
x_adv= clip(adversarial_image, min_pixel_value, max_pixel_value)

Step 6: Return set of adversarial examples for each epsilon value to the target model
return [x_adv1, x_adv2, ..., x_advn]

value determines the strength of the perturbation, with larger values producing more noticeable changes to the image. In Step 5.3, the pixel values of the adversarial image are clipped to ensure that they fall within the valid range of values for the image. This step prevents the adversarial example from being too different from the original image and ensures that it is still recognizable as a face mask image. Finally, in Step 6, the algorithm returns a set of adversarial examples for each epsilon value in the set of epsilon values.

The overall time complexity of algorithm 1 is approximately O(|E|), where |E| is the number of epsilon values. This complexity analysis considers the operations performed for each epsilon value, which includes normalizing and resizing the input image, predicting the label using the face mask model, calculating the loss and gradient, scaling the signed gradient, adding the perturbation to the image, and clipping the pixel values. These operations have a constant time complexity, so the total complexity is proportional to the number of epsilon values. However, if we consider the algorithm's complexity with respect to the number of images, the time complexity is O (N * |E|). This is because Steps 1–4 are performed once per image, and Step 5 is performed N times for each image.

In Fig. 6, we have illustrated the generation of adversarial perturbation and adversarial examples from the substitute model. An adversarial image was produced by setting the epsilon value as 0.010. It is clear from the figure that both adversarial and pristine images have the same appearance. The adversarial images are computed so that they are imperceptible to the human eye yet able to fool the classifier.

4.4.1 Transferability of Adversarial examples (cross model, cross dataset generalization)

Adversarial attack generalizes from one data set to another dataset and from one model to another. The nature of the perturbation is not a random learning artifact; DNN trained on very different subsets of the dataset and using separate architecture can both misclassify the same input due to the same perturbation. In simple terms, adversarial examples are relatively robust and are shared by DNNs trained on various subsets of the training data with varying layers and activations. In many cases, adversarial examples for DNN 'A' will also fool DNN 'B'. This property of an adversarial example makes it possible to design a black box attack for some target model. As a result, an adversarial example derived for the substitute model may also mislead the face mask detection model, even though they were trained on distinct datasets and have different architectures, respectively. Figure 7 demonstrates the concept of transferability and black box attack in the context proposed work.

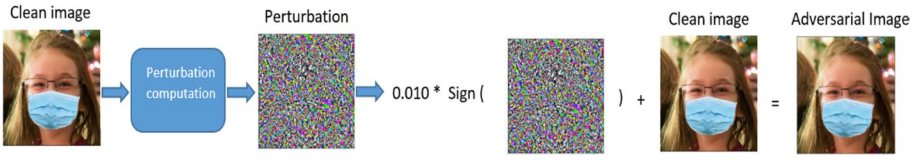


Fig. 6 Illustration of the adversarial example generation for substitute model

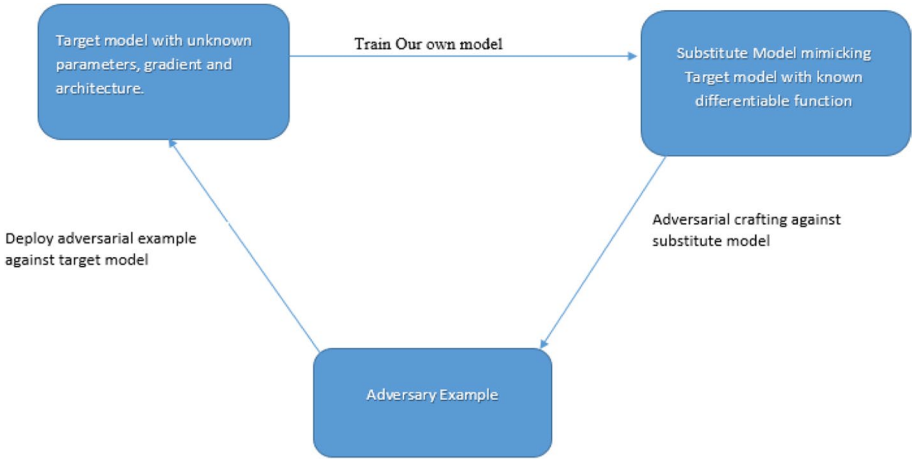


Fig. 7 We have a target model face mask detection system for which we want to design an adversarial example, but the internal details of a model are assumed to be unknown. In that case, we develop a substitute model which mimics the Target model and craft an adversarial example for the substitute model. The transferability property of adversarial examples makes it possible to deploy them successfully against the target model. It is important to note that the black-box adversarial attack occurs during the testing phase. It means that both the substitute and target models should be trained before the attack. After training the substitute model, we utilized the untargeted FGSM strategy to generate adversarial samples for the target model. We first provide some notations and introduce the formulation of black-box adversarial attacks on the developed face mask detection systems

5 Experimental results and discussion

The target model was initially trained on Google Colaboratory and then saved onto a local machine. Subsequently, the model was loaded onto an HP laptop equipped with 8 GB RAM and an NVIDIA GeForce 930 Max graphics card, where the adversarial attack was conducted. The entire experiment was conducted within a Python environment. For the implementation of the face mask detection system, a convolutional neural network (CNN) was designed using a combination of Keras and Tensorflow. The SKlearn library was utilized to calculate various performance metrics such as accuracy, precision, recall, and F1 score. To mitigate potential issues of overfitting and underfitting, data augmentation and image preprocessing techniques were employed using the ImageDataGenerator module. Finally, the accuracy and loss learning curves during the model's training were visualized using Matplotlib.

5.1 Performance evaluation metrics

In the evaluation of the models, various metrics were used to assess their performance. These metrics included accuracy, precision, recall (sensitivity), F1-Score, weighted average and macro-average. Let's briefly define each of these metrics:

- Accuracy: Accuracy is a metric that measures the ratio of correctly classified instances to the total number of instances. It provides an overall assessment of the model's accuracy in predicting the correct class labels.
- Precision: Precision quantifies the ratio of correctly predicted positive observations to the total number of predicted positive observations. It evaluates the model's ability to minimize false positives, indicating how precise the model is in identifying positive instances.
- Recall or Sensitivity: Recall calculates the proportion of true positive observations out of all the actual positive instances. It assesses the model's ability to correctly identify all positive instances, indicating its sensitivity to detecting positive cases.
- F1-Score: The F1-Score is a metric that combines both precision and recall into a single measure. It is the harmonic mean of precision and recall, providing a balanced assessment of the model's performance. This metric is particularly useful when dealing with imbalanced datasets.

$$\text{F1Score} = 2 * (\text{Recall} * \text{Precision})/(\text{Recall} + \text{Precision})$$

- Macro-Average: Macro-Average calculates the average performance across all classes, assigning equal weight to each class. It gives an overall evaluation of the model's performance without considering the class distribution.
- Weighted Average: Weighted Average calculates the average performance across all classes, taking into account the class distribution. It provides a performance measure that considers the importance of each class based on its representation in the dataset.

5.2 Performance of the target model and a substitute model

Both the target and substitute models underwent training on the same dataset for 30 epochs each. The models were trained to detect masked, unmasked, and incorrectly masked faces, achieving an average testing accuracy of 97.18% and 96.83%, respectively. Figure 8 presents the learning curves for both models, illustrating their accuracy and loss during the training and testing phases. Subplot (a) depicts the accuracy and loss of the target face mask detection model on both training and testing data, while subplot (b) shows the accuracy and loss of the substitute model.

The plot clearly demonstrates that as the number of epochs increases during training and validation, the accuracy of both the training and validation data improves, while the loss decreases. Notably, the testing accuracy closely aligns with the training accuracy, indicating that the model does not suffer from overfitting. These findings validate the models' performance and indicate their ability to generalize well to unseen data. To further evaluate the models, a classification report is provided in Table 5, which offers detailed insights into their performance across different metrics.

The target model successfully classified the clean input image to its desired output. In Fig. 9, the model was tested on an image of an unseen masked person and confidently classified it as masked with a 99.99% accuracy score. This result indicates that the model can perform well on previously unseen and untested clean samples, suggesting that it has reliable classification capabilities. In other words, the model is able to generalize well beyond the training data and make accurate predictions on new and varied inputs.

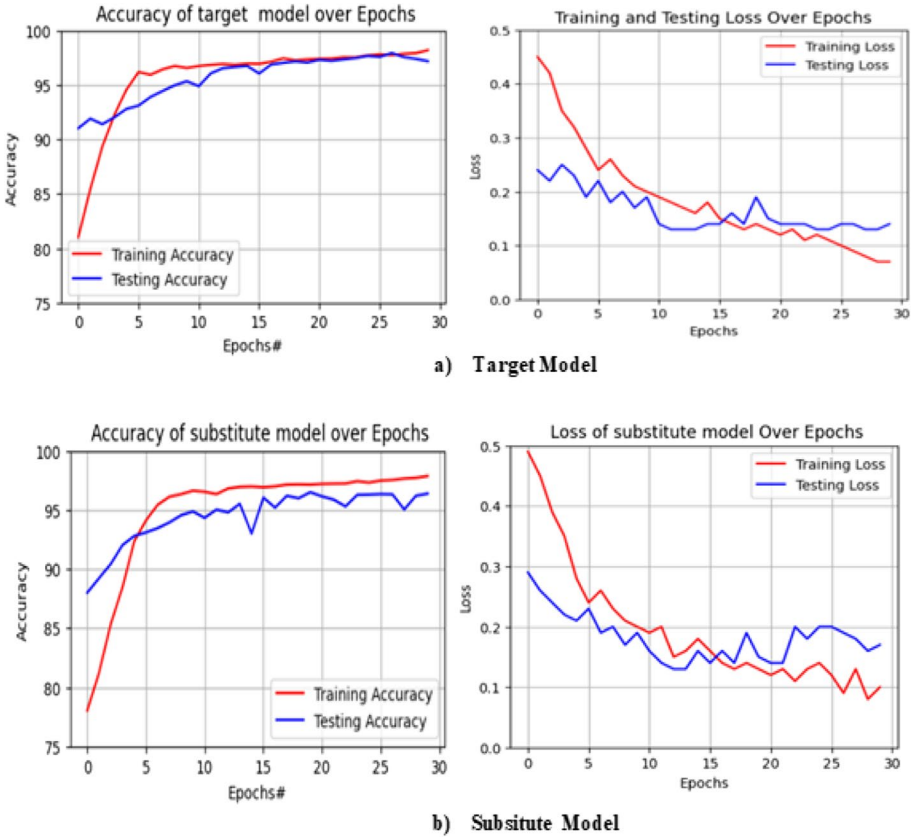


Fig. 8 Illustrates the learning curves of models over 30 epochs: Subplot (a) depicts the accuracy and loss of the target face mask detection model on both training and testing data, while subplot (b) shows the accuracy and loss of the substitute model

5.3 Post-attack performance of target model

The performance of the target model was found to be satisfactory during training and testing phases on the clean data. However, when subjected to the proposed black-box adversarial attack, its accuracy significantly decreased. The impact of the Fast Gradient Sign Method (FGSM) attack, applied at various epsilon values, is presented in Table 6. The accuracy of the target model was evaluated on adversarial images generated using different epsilon values by a substitute model. The results showed a decline in accuracy as the epsilon value increased, ranging from a high of 97.18% for clean images (epsilon = 0) to a low of 46.52% at an epsilon value of 0.009. The corresponding loss values were also recorded, indicating the negative effect of black-box adversarial attacks on the model's performance. A visual representation of the accuracy and loss variation with respect to different epsilon values is shown in Fig. 10, illustrating the inverse relationship between epsilon and model accuracy. Additionally, Fig. 11 provides insights into the number of correct and incorrect predictions made by the target model for each epsilon value.

Table 5 Precision, recall, f1-score of the target model (MobileNetV2-based face mask model) and a substitute model on clean test data

Target Model (MobileNetV2 Based face mask detection)				
	precision	recall	f1-score	support
Incorrect_mask	0.92	1.00	0.97	353
With_mask	0.98	0.93	0.96	350
Without_mask	0.98	0.97	0.98	350
Average Accuracy			0.97	1053
Macro_Average	0.96	0.95	0.96	1053
Weighted_Average	0.96	0.96	0.96	1053
Substitute Model (CNN Model from scratch)				
Incorrect_mask	0.92	1.00	0.94	353
With_mask	0.97	0.91	0.94	350
Without_mask	0.99	0.97	0.98	350
Average Accuracy			0.96	1053
Macro_Average	0.95	0.95	0.95	1053
Weighted_Average	0.95	0.95	0.95	1053

Fig. 9 It showcases a significant result, demonstrating the accurate classification of a clean masked image by the target model. The model correctly identifies the presence of a mask on the face, categorizing it as a masked face with high confidence

Table 6 Accuracy and loss on various epsilon of FGSM

	Total correct	Total Incorrect	Average accuracy (%)	Loss
Clean image (0)	1023	30	97.18	0.1256
0.001	884	169	84.34	0.9742
0.002	792	261	75.23	1.7405
0.003	756	297	71.87	1.2090
0.004	715	338	68.56	1.7345
0.005	621	432	59.02	2.9956
0.006	581	472	56.86	3.0018
0.007	556	497	52.87	3.8929
0.008	508	545	48.33	4.1344
0.009	490	563	46.52	4.7897

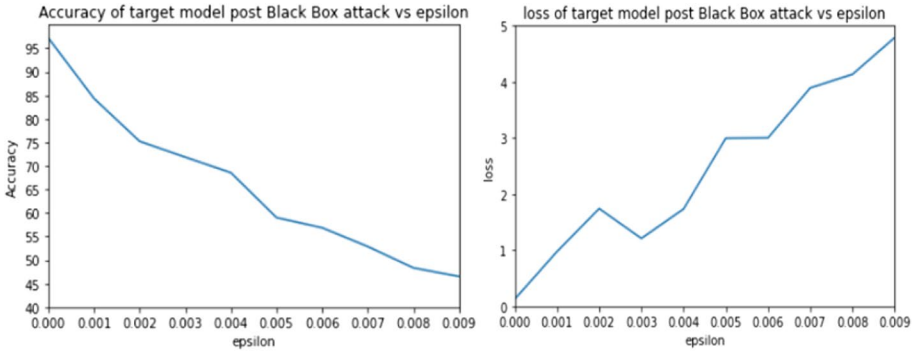
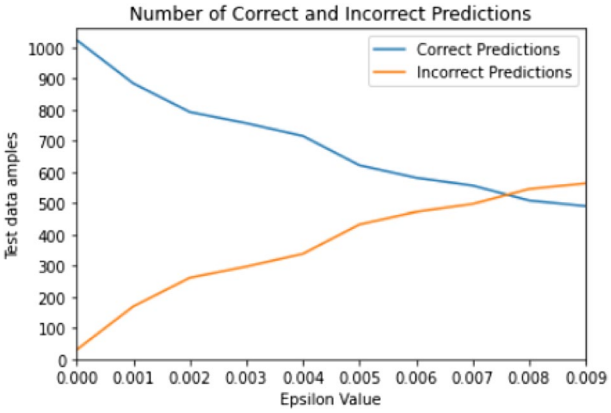


Fig. 10 Accuracy and loss curves of target model on various epsilon values

Fig. 11 It illustrates that as the value of epsilon increases, the number of incorrect predictions also increases while the number of correct predictions decreases. This is because higher values of epsilon lead to more random or noisy predictions from the model



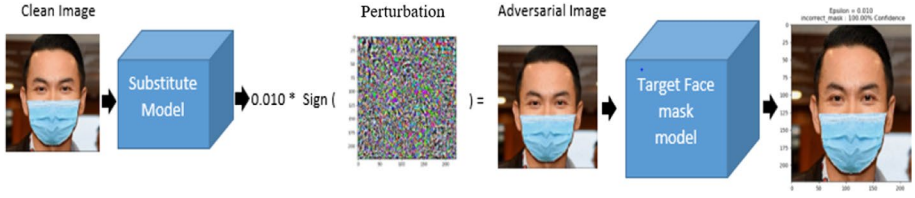


Fig. 12 It illustrates a significant outcome of the study, where the target model consistently misclassifies an adversarial image, generated by the substitute model, as an incorrect mask with a confidence level of 100%. This finding highlights the effectiveness of the adversarial attack in deceiving the target model, leading it to confidently label the adversarial image as an incorrect mask despite the presence of a mask in the original image. The high confidence score further emphasizes the model's certainty in its misclassification, revealing the potential vulnerability of face mask detection systems to adversarial attacks

When presented with the adversarial version of the input image depicted in Fig. 3, the target model exhibited a significantly erroneous classification of the masked face image, labeling it as an incorrect mask with 100% confidence, as shown in Fig. 12. This misclassification by the model is noteworthy, as it was done with absolute confidence, highlighting

Table 7 Comparison of adversarial attacks and defense strategies used in previous works

Article	Models	Dataset	Approach	Attack Method	Attack performance
[16]	COVID-Net	X-Ray	White Box	UAP	>90%
[8]	ResNet,YOLO, DarkNet, GRAD-CAM	CT-scan X-ray	White Box	FGSM,MIFGSM, C&W DF, JSMA, LBFGS, ,BIM, PGD, FB, BD, MS,poisoning	91%
[27]	VGG 16 InceptionV3	X-ray, CT scan	White Box	FGSM	83.3
[30]	U-Net	CT-scan	White Box	Stabilized Medical Image Attack(SMIA)	60.82%
[38]	Mobile-Net V2	Face mask dataset	White-Box	FGSM	90%
Proposed work	Black Box (Substitute model)	Sophisticated face mask dataset	Black Box	FGSM on Substitute model	>50%

the model's vulnerability to adversarial attacks. This confidence level surpasses the correct classification confidence of the clean image, which had a confidence score of 99.95%. These results indicate that the target model's classification performance is susceptible to slight perturbations in the input, resulting in significant and erroneous misclassifications with overconfident predictions.

In addition, we compare the previous work of the COVID-19 battling tools regarding their vulnerabilities against adversarial attacks in table 7. All the attempts to attack the tool used to combat COVID-19 were based on white box attacks, which are relatively easy to perform. These attacks yielded better results than the proposed work, likely because the internal workings of the model were available to the attackers. However, it's important to note that white box attacks are not practical in real-world situations because the model details are not publicly available. In such scenarios, the proposed work is applicable and achieves a good misclassification accuracy of over 50%.

6 Future scope

Initially, we identified and outlined the limitations of our study, followed by a thorough discussion on potential future scope and solutions to mitigate these limitations.

- Limited scope: The study only focused on one specific type of face mask detection model (MobileNetv2) and one type of adversarial attack (FGSM). This may not be representative of all types of models and attacks.
- Lack of real-world testing: The study was conducted in a controlled environment using test images. The performance of the models in real-world scenarios may differ.
- Lack of exploration of defense strategies: The study focused primarily on attacking the face mask detection model without exploring defense strategies to mitigate adversarial attacks.

Despite the limitations mentioned above, this study opens up several avenues for future research in the field of face mask detection and adversarial attacks:

- We plan to expand our current study to real-world scenarios where the face mask model could be attacked in a system that recognizes face mask violations in real-time video data from CCTV cameras.
- Additionally, while our study utilized the FGSM (Fast Gradient Sign Method) strategy for adversarial example computation, we acknowledge the importance of exploring other attack strategies further to assess the robustness of face mask detection systems. In future studies, we plan to investigate the efficacy of attack strategies such as projected gradient descent [24] and Jacobian-based Saliency Map Attack [28]. By exploring a wider range of attack strategies, we can gain a more comprehensive understanding of the vulnerabilities of face mask detection models and devise effective defense mechanisms.
- Further research can focus on developing more advanced defense mechanisms against adversarial attacks. Adversarial training techniques can be explored, involving training models with clean and adversarial examples to improve their resilience [38]. Additionally, investigating input preprocessing methods, such as image denoising or filtering [1], can help mitigate the impact of adversarial perturbations on detection accuracy.We

would like to apply the image forgery detection technique to detect the adversarial input during the inference phase of the model [2].

- Furthermore, we aim to design a universal framework that can effectively counter adversarial examples generated by any adversarial attack strategy. Currently, most defense strategies are tailored to specific attack methods, which limits their applicability in real-world scenarios where multiple attack strategies may be employed. Therefore, we plan to develop a robust defense framework that can mitigate the impact of adversarial examples, regardless of the attack strategy used. This universal framework would enhance the overall resilience of face mask detection systems against adversarial attacks.
- There is a need to evaluate the performance of face mask detection models in diverse real-world scenarios and environments. This includes studying the impact of lighting conditions, occlusions, camera angles, and mask types and design variations. By conducting experiments in these real-world settings, researchers can better understand the limitations of the current models and identify areas for improvement.
- Furthermore, exploring ensemble methods can enhance the reliability of face mask detection systems. Ensemble methods involve combining multiple models or classifiers to make collective decisions. By leveraging the diversity of multiple models, ensemble techniques can improve detection accuracy and robustness.
- Additionally, it would be beneficial to investigate techniques for mitigating adversarial attacks in the training data itself. By incorporating adversarial examples during the training process and applying techniques such as data augmentation, model regularization, or generative adversarial networks (GANs) [13], it may be possible to improve the generalization capability of face mask detection models and reduce their vulnerability to adversarial attacks.
- Lastly, exploring the transferability of defense approaches across different face mask detection models is equally important. It is necessary to investigate whether defense strategies that prove effective for one model can be successfully applied to protect other models as well. By studying the transferability of defense approaches, researchers can identify robust and generalizable methods that can mitigate adversarial attacks across a variety of face mask detection models.

Overall, the future scope of this study involves a deeper exploration of defense mechanisms, evaluation in real-world scenarios, ensemble methods, data augmentation techniques, and the transferability of attacks. By addressing these areas, researchers can advance the development of more robust and reliable face mask detection systems resistant to adversarial attacks and adaptable to diverse environments.

7 Conclusion

The findings of this study highlight the susceptibility of face mask detection models to adversarial attacks in black box scenarios. By utilizing the substitute model and leveraging the transferability property of adversarial examples, we successfully attacked the target face mask model. Despite achieving a remarkable accuracy of 97.18% on clean inputs, as demonstrated in Figs. 8 and Table 5, the target model's performance significantly deteriorated when subjected to the proposed FGSM attack with adversarial examples generated by the substitute model. Table 6 and Fig. 10 clearly depict the decline in accuracy, with

a drastic drop to 46.52% at epsilon 0.009. Notably, Fig. 12 exposes the complete misclassification of the adversarial image of a masked face as an incorrect mask face with 100% confidence during the FGSM attack. Additionally, a substantial number of images in the test dataset were misclassified as incorrectly masked faces during the attack.

These results emphasize the vulnerability of face mask detection models to adversarial attacks, calling for the implementation of robust defensive techniques before deploying them in real-world scenarios. The accurate identification of face masks is crucial in effectively combating COVID-19, necessitating further research, consideration, and implementation of protective measures. By raising awareness of the vulnerabilities present in DL based face mask classifiers, we aim to motivate researchers to enhance the security of their models and promote the development of face mask detection models with multiple protection strategies.

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

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

Declarations

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

References

1. Ahmad M, Khursheed F (2022) A novel image tamper detection approach by blending forensic tools and optimized CNN: Sealion customized firefly algorithm. Multimed Tools Appl 81(2):2577–2601
2. Ahmad M, Khursheed F (2022) Detection and localization of image tampering in digital images with fused features. Concurr Comput Pract Exp 34:7191
3. Alrashed S, Min-Allah N, Ali I, Mehmood R (2022) COVID-19 outbreak and the role of digital twin. Multimed Tools Appl 81(19):26857–26871. https://doi.org/10.1007/s11042-021-11664-8
4. Bania RK (2023) Ensemble of deep transfer learning models for real-time automatic detection of face mask. Multimed Tools Appl 82:25131–25153. https://doi.org/10.1007/s11042-023-14408-y
5. Chen PY, Zhang H, Sharma Y, Yi J, Hsieh CJ (2017, November 3) ZOO. Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. https://doi.org/10.1145/3128572.3140448
6. "Coronavirus disease (COVID-19)." (n.d.) https://www.who.int/news-room/q-a-detail/coronavirus-disease-covid-19. Accessed 11 Apr. 2021
7. Das D, Biswas SK, Bandyopadhyay S (2022) Perspective of AI system for COVID-19 detection using chest images: a review. Multimed Tools Appl 81(15):21471–21501. https://doi.org/10.1007/s11042-022-11913-4
8. Deng J, Dong W, Socher R, Li LJ, Li K, Li F-F (2009, June) ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2009.5206848
9. 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
10. Feng S, Shen C, Xia N, Song W, Fan M, Cowling BJ (2020, May) Rational use of face masks in the COVID-19 pandemic. The Lancet. Respir Med 8(5):434–436. https://doi.org/10.1016/s2213-2600(20)30134-x
11. Gao J, Lanchantin J, Soffa ML, Qi Y (2018, May) Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. 2018 IEEE Security and Privacy Workshops (SPW). https://doi.org/10.1109/spw.2018.00016
12. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572

13. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, ..., Bengio Y (2014) Generative adversarial nets. Advances in neural information processing systems, 27
14. Goyal H, Sidana K, Singh C, Jain A, Jindal S (2022) A real time face mask detection system using convolutional neural network. Multimed Tools Appl 81(11):14999–15015. https://doi.org/10.1007/s11042-022-12166-x
15. Haque SBU, Zafar A, Roshan K (2023) Security vulnerability in face mask monitoring system. In: 2023 10th International conference on computing for sustainable global development (INDIACom). New Delhi, India, 231–237
16. 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
17. Ilyas A, Engstrom L, Athalye A, Lin J (2018, July) Black-box adversarial attacks with limited queries and information. In International conference on machine learning. PMLR, pp. 2137–2146
18. Javed I, Butt MA, Khalid S, Shehryar T, Amin R, Syed AM, Sadiq M (2022) Face mask detection and social distance monitoring system for COVID-19 pandemic. Multimed Tools Appl 82:14135–14152. https://doi.org/10.1007/s11042-022-13913-w
19. Jayaswal R, Dixit M (2022) AI-based face mask detection system: a straightforward proposition to fight with Covid-19 situation. Multimed Tools Appl 82:13241–13273. https://doi.org/10.1007/s11042-022-13697-z
20. Kuchana M, Srivastava A, Das R, Mathew J, Mishra A, Khatter K (2020) AI aiding in diagnosing, tracking recovery of COVID-19 using deep learning on Chest CT scans. Multimed Tools Appl 80(6):9161–9175. https://doi.org/10.1007/s11042-020-10010-8
21. Kurakin A, Goodfellow IJ, Bengio S (2017) Adversarial examples in the physical world. In: Proceedings of the 5th International Conference on Learning Representations (ICLR) Workshop Track, pp. 1–14
22. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: Single Shot MultiBox Detector. Comput Vis – ECCV 2016:21–37. https://doi.org/10.1007/978-3-319-46448-0_2
23. Lu H, Zhuang Z (2022) ULN: An efficient face recognition method for person wearing a mask. Multimed Tools Appl 81(29):42393–42411. https://doi.org/10.1007/s11042-022-13495-7
24. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2017) Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083
25. Moosavi-Dezfooli SM, Fawzi A, Fawzi O, Frossard P (2017, July) Universal Adversarial Perturbations. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2017.17
26. Oztel I, Yolcu Oztel G, Akgun D (2022, October 21) A hybrid LBP-DCNN based feature extraction method in YOLO: An application for masked face and social distance detection. Multimed Tools Appl 82(1):1565–1583. https://doi.org/10.1007/s11042-022-14073-7
27. 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. Appl Sci 11(9):4233. https://doi.org/10.3390/app11094233
28. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A (2016, March) The Limitations of Deep Learning in Adversarial Settings. 2016 IEEE European Symposium on Security and Privacy (EuroS&P). https://doi.org/10.1109/eurosp.2016.36
29. Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A (2017, April 2) Practical Black-Box Attacks against Machine Learning. Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. https://doi.org/10.1145/3052973.3053009
30. Qi G, Gong L, Song Y, Ma K, Zheng Y (2021) Stabilized medical image attacks. arXiv preprint arXiv:2103.05232
31. Rahman A, Hossain MS, Alrajeh NA, Alsolami F (2021, June 15) 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
32. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767
33. 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
34. Roshan K, Zafar A, Haque SBU (2023) A novel deep learning based model to defend network intrusion detection system against adversarial attacks. In: 2023 10th international conference on computing for sustainable global development (INDIACom). New Delhi, India, 386-391
35. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520

36. Sheikh B, Zafar A (2023) Beyond accuracy and precision: a robust deep learning framework to enhance the resilience of face mask detection models against adversarial attacks. Evolving Systems. https://doi.org/10.1007/s12530-023-09522-z
37. 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
38. Sheikh BUH, Zafar A (2023) Untargeted white-box adversarial attack to break into deep learning based COVID-19 monitoring face mask detection system. Multimed Tools Appl:1–27. https://doi.org/10.1007/s11042-023-15405-x
39. Singh S, Ahuja U, Kumar M, Kumar K, Sachdeva M (2021) Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. Multimed Tools Appl 80(13):19753–19768. https://doi.org/10.1007/s11042-021-10711-8
40. Su X, Gao M, Ren J, Li Y, Dong M, Liu X (2021) Face mask detection and classification via deep transfer learning. Multimed Tools Appl 81(3):4475–4494. https://doi.org/10.1007/s11042-021-11772-5
41. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.
42. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826
43. Ullah N, Javed A, Ali Ghazanfar M, Alsufyani A, Bourouis S (2022) A novel DeepMaskNet model for face mask detection and masked facial recognition. J King Saud Univ - Comput Inf Sci 34(10):9905–9914. https://doi.org/10.1016/j.jksuci.2021.12.017
44. 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):19549. https://doi.org/10.1038/s41598-020-76550-z
45. Wani MH, Faridi AR (2022) Deep learning-based video action recognition: A Review. In: 2022 international conference on computing, communication, and intelligent systems (ICCCIS). Greater Noida, India, 243–249. https://doi.org/10.1109/ICCCIS56430.2022.10037736
46. World Health Organization. (2020) Advice on the use of masks in the context of COVID-19: interim guidance, June 5 2020 (No. WHO/2019-nCoV/IPC_Masks/2020.4). World Health Organization
47. "WHO Director-General's opening remarks at the media briefing on COVID-19 - March 11 2020." https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19%2D%2D-11-march-2020. Accessed 11 April 2021