International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Face Mask Image Classification Using Fine-Tuning and the Effect of FGSM and PGD Attacks

Kouassi Joshua Caleb Micah[1], Lou Qiong[2]

Zhejiang University of Science and Technology, Hangzhou 310023, China.

Abstract

This research seeks to use a fine-tuning method for automatic diagnostics and classification of face masks without face mask images. We trained Deep Convolutional Neural Network models on the face mask dataset obtained from the Mendeley data repository to determine whether individuals will adhere to policies that face mask reduces the spread of SARS-COV-2. The proposed architectures used in this study include the VGG19, MobileNetV3, and InceptionV3 models. These models are known for their role in classifying images. They were trained using the fine-tuning approach and their respective outputs were compared. After training the face mask dataset, it can be said that the fine-tuned InceptionV3 model performed extremely well against the other models obtaining an accuracy of 99.21%, and was able to predict 99.63% of the test dataset. However, the robustness of the fine-tuned model was tested against fast gradient sign method (FGSM) and projected gradient descent (PGD) attacks which generate adversarial images using the gradients of the model. Additionally, the classification report shows after the FGSM attack that the model's accuracy was reduced by 43%. Also, after the PDG attack, the accuracy of the model was reduced by 19%. We assessed the models using many performance metrics such as precision, recall, F1 score, and accuracy before subjecting them to two common adversarial attack techniques: FGSM and PDG. Finally, we demonstrated how the proposed robust method improved the model's defense against adversarial attacks. The findings emphasize the critical need to increase awareness about adversarial assaults on SARS-COV-2 monitoring systems and to advocate for proactive steps to protect healthcare systems from comparable threats prior to practical deployment.

Keywords: Convolutional Neural Network, Face Mask, Fine-Tuning, FGSM attack

Introduction

Many theories have emerged about how to prevent the spread of the novel coronavirus (SARS-COV-2) outbreak since the World Health Organization (WHO) declared it a global pandemic on March 11, 2020 [1]. Derailing economies, especially developing ones of their yearly economic target, by heavily hitting their service sectors. Entirely every part of the world economy struggled with it corresponding to current struggles in developing countries. Origin of SARS-COV-2 as an infectious disease resulting from the SARS-COV-2 virus. The virus could be spread by droplets from a contaminated person [2,3]. Many proposed theories, but face masks proved to be very helpful in

controlling and preventing the spread of SARS-COV-2.

Although the discovery of vaccines has controlled the spread of SARS-COV-2, face mask as prevention against the outbreak has been very beneficial in the absence of immunization. The use of face masks was so helpful that the World Health Organization advised its use across the world [4,5]. With the ongoing spread of COVID-19 and the presence of emerging variants, various regulatory authorities have emphasized the importance of wearing masks, particularly in public places. Unfortunately, there are still individuals who disregard health recommendations. Carelessness and lack of awareness have been the major cause of SARS-COV-2 [6,7] Unmasked people can be detected and alerted using an autonomous system. Several face mask detection machines have already been developed and are in use by some organizations.

K. Bhambani et al [8] employ YOLO object detection with an accuracy of 94.75%. It concentrated on face mask recognition as well as keeping a certain distance in crowded areas. The developed model is used in public places because it has high precision, few heavy components, and is time efficient. Convolutional neural networks (CNN) are employed in a cascading fashion to recognize masked faces [9]. In terms of modern approaches, the retina face mask network is intended to be a one-of-a-kind framework for recognizing face masks correctly and effectively [10]. Several experiments [11,12,13] were carried out in order to develop a technique that can detect whether or not a person is wearing a face mask. To detect facial masks in real-time, the YOLOv3 technique, and the haar cascading classifier are used [14]. In face recognition, four reducing deep learning approaches, namely VGGFace, FaceNet, OpenFace, and DeepFace, are compared [15]. Deep neural network (DNN) methods [16] also employ a visual geometry group-16 (VGG-16) for face mask detection. To create real-time facial mask identification with an alert system, the VGG-16 architecture for real-time face mask recognition is used [17].

A lot of transfer learning approaches can also be used to solve the problem of face mask detection in the real world. A pre-trained InceptionV3 model is used as a transfer learning method to detect people wearing or not wearing masks [18]. For face mask detection, the hybrid deep transfer learning model, which combines deep learning methods with traditional computer learning, is used [19]. The model was also created using a transfer learning concept developed on the pre-trained MobileNetV2 model for real-time face mask detection and localization [20].

Fine-tuning increases model precision relative to transfer learning, as demonstrated by Enoch Binney et al. [22]. In their research [22], the authors compared the impact of transfer learning and fine-tuning on the JMuBEN2 dataset. They discovered that fine-tuning the model resulted in more training and accuracy than transfer learning. In their study, they utilized the VGG-19, ResNet50, and DenseNet121 designs. The Densenet-121 model was superior to the others after training on the JMuBEN2 dataset with the aforementioned models, achieving an accuracy of 95.44 percent after transfer learning and 99.36 percent after fine-tuning. This article served as a reference for this study, which aims to apply fine-tuning above other training methods due to its superior accuracy rate attainment.[32] looked on the robustness of three face mask detection models based on cutting-edge convolutional neural networks (CNNs), namely MobileNetV2, ResNet50, and EfficientNet-B2.The findings showed that adversarial attacks reduced model accuracy significantly, with MobileNetV2 dropping from 95.83-14.83% to 0% (under FGSM and PGD attacks, respectively), ResNet50 dropping from 96.48-13.97% to 0% (under FGSM and PGD attacks, respectively), and EfficientNet-B2 dropping from 95.56-15.53% to 0% (under FGSM and PGD attacks, respectively).

In this study, we compared and analyzed the prediction accuracy amongst three CNN models (MobileNet-V3, VGG-19, and Inception-V3) to predict the FaceMask dataset [21]. To our knowledge, this dataset has not been utilized for computer vision research; hence, we recommend the aforementioned models by comparing their performance on the dataset. In this paper, we develop fast and accurate architectures for mask detection-driven facial image categorization using fine-tuned convolutional neural networks. Moreover, FGSM and PGD attacks are both applied to the best-performing model among the three Deep Neural Networks used in this study. These attacks are applied to examine the robustness of the best model and how these attacks can fool this model.The remaining portions of this work are organized as follows: The Research Design and Dataset used in the research are discussed in Section II. Section III delves into the Models and Theories of the Individual Deep CNN used in this study. Section IV reports the findings, followed by Section V's conclusion.

Research Design and Dataset

This study proposes using Deep Convolutional Neural Networks to construct a feasible and reasonably accurate approach for recognizing and classifying photos of individuals with and without face masks. By fine-tuning, the performance of the algorithms utilized in this work will be enhanced. If this investigation is successful, it will provide a comprehensive and non-invasive way to distinguish images with and without face masks. However, the robustness of the model will be tested against the Fast Gradient Sign Method (FGSM) and the Projected Gradient Descent (PGD) attacks where perturbed adversarial images will generated and compared against the original face-masked images. The research will be conducted utilizing a variety of image classification algorithms coupled with FGSM and PGD attacks.

Research Design

To solve the challenge of detecting and categorizing two sets of faced masked image datasets, it will be necessary to build methods based on models of deep learning. This method of design allows the proposed models to be trained and validated data. This will allow us to obtain quantifiable findings that can be compared to previous studies that employed deep learning to address a similar issue. Fine-tuning models were developed with Keras and Tensorflow and implemented with Python for image detection in this study. The selected models have been trained using Mendeley Data Public Access data retrieved online. The outcomes are assessed in terms of accuracy and prediction. Afterward, FGSM and PGD attacks are employed to check for the best model's robustness.

Dataset

The face mask dataset 2022 [21] was collected by the Islamic University Department of Computer Science and Engineering. Its main purpose was to develop a model that can differentiate between individuals wearing masks and those not. This dataset contains a total of 20,347 images belonging to two classes thus with mask and without mask categories.

Table 1: distribution of dataset

Condition of Leaf	Number of images
Masked	10240
No Mask	10107

As seen below, the dataset contains images where individuals have masks (with masks) and others do not (without masks).

Figure 1: sample of the dataset

MODELS AND THEORIES

This paper takes into consideration architectures that are based on Convolutional Neural Networks. Some of the networks used are VGG19 [24], MobileNet V3 [23], and Inception V3 [25].

Deep learning

The architectures used are based on the concept of deep learning. Deep learning derives its name from the basis of neural networks which are composed of various layers. Each Convolutional Neural Network has at least consisted of one convolutional layer (CL). The more the number of layers the deeper the Convolutional Neural Network. However, in conjunction with convolutional layers are pooling and fully connected layers. Additionally, an activation function ends each convolutional layer.

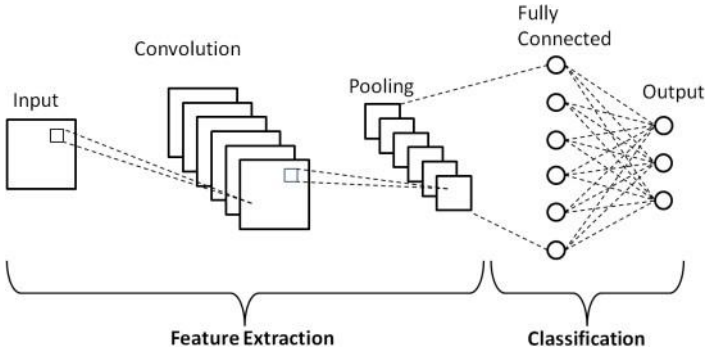

Figure 2: Convolutional neural network

The main building component of the architecture is the convolutional layer. This is where most of the network's rigorous operations occur. The primary CL applies a set of filters to the input image [height, width, depth]. Each filter, small in size, works to the full depth dimension of the image by processing each dimension individually.

Fine Tuning

The concepts of fine-tuning and transfer learning are very closely related. We forward propagate our image dataset through the network using this feature extractor, extract the activations at a specific layer (considering the activations as a feature vector), and then save the values to disk. On top of the CNN features, a common machine-learning classifier was developed. When we apply the knowledge, we learn from solving one problem to another that is unrelated but still challenging, this process is known as transfer learning. For instance, the ability to identify shoes might be used to solve the issue of identifying football boots. We refer to this kind of transfer learning as feature extraction.

Applying or making use of transfer learning involves fine-tuning. In particular, fine-tuning is the process of refining or tweaking an architecture that has already been trained to perform one specific task in order to have it execute a second task that is similar. Using an artificial neural network that has already been constructed and trained allows us to benefit from what the model has previously learned without having to develop it from the start, assuming the original task is similar to the new one. We frequently have to use trial and error to explore various ways while creating a model from scratch. We must decide, for instance, how many layers, what kind of layers, what order to arrange the layers in, how many nodes to include in each layer, how much regularization to employ, what learning rate to utilize, etc.

Depending on the data we're training our model on, developing and validating it might be a laborious process in and of itself. This is what appeals to people about the fine-tuning strategy. We can utilize all of the model's prior knowledge and apply it to our particular work if we can locate a trained model that has already performed one task well and that task is at least tangentially comparable to ours.

Inception v3

The Inception V3[25] is an improved version of the Inception V1, a basic model first released in 2014 as Google Net. As the name implies, it was designed by a Google team. The Inception V3 is a Convolutional Neural Network-based deep learning model for image classification. Several techniques were used by the Inception V3 model to optimize the network for better model adaptation. The Inception V3 model is more efficient and has a deeper network than the Inception V1 and V2 models, but it is faster, cheaper to compute, and employs auxiliary Classifiers as regularizes. The inception V3 model includes 42 layers in total, which is significantly greater than the inception V1 and V2 models. However, the efficiency of this model is pretty impressive.

TYPE	PATCH / STRIDE SIZE	INPUT SIZE
Conv	3×3/2	299×299×3
Conv	3×3/1	149×149×32
Conv padded	3×3/1	147×147×32

TYPE	PATCH / STRIDE SIZE	INPUT SIZE
Pool	3×3/2	147×147×64
Conv	3×3/1	73×73×64
Conv	3×3/2	71×71×80
Conv	3×3/1	35×35×192
3 × Inception	Module 1	35×35×288
5 × Inception	Module 2	17×17×768
2 × Inception	Module 3	8×8×1280
Pool	8 × 8	8 × 8 × 2048
Linear	Logits	1 × 1 × 2048
Softmax	Classifier	1 × 1 × 1000

Table 2: Inception V3 model composition

Softmax activation is used over sigmoid since the current challenge requires multiple class classifications of images, whereas sigmoid excels at binary classification.

MobileNet V3

MobileNetV3[23] comes in two architectures: MobileNetV3-Large and MobileNetV3-Small. These models are intended for scenarios with high and low resource utilization. The models are built by incorporating network enhancements and employing platform-aware NAS and NetAdapt for network search. Complementary search techniques, new efficient versions of nonlinearities ideal for mobile applications, and new efficient network architecture are examples of advancements. The network architecture employs hard swish activation and squeeze-and-excite modules in the MBConv blocks. MobileNetV3-Large is 3.2% more accurate on ImageNet classification than MobileNetV2 while reducing latency by 15%. When compared to MobileNetV2, MobileNetV3-Small is 4.6 percent more accurate and has a 5% lower latency.

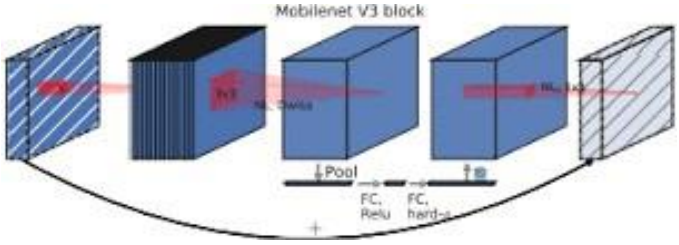

Figure 3: Mobilenet V3 building block

VGG19

The VGG-19 network [24], as the name implies, is made up of 19 Convolutional Neural Network layers and three fully connected layers. The VGG-19 network's architecture is depicted in the

diagram below.

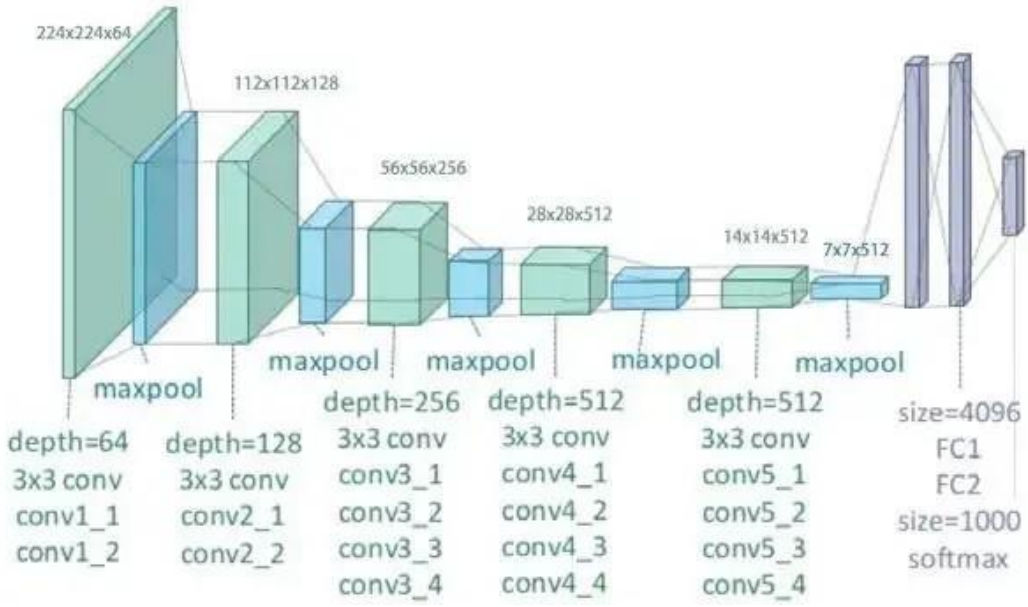

Figure 4: VGG model building process

The architecture is basically composed of three types of layers: the Convolution layer, which extracts the feature from the image by employing various numbers and types of filters, the Max-pooling layer, which decreases the image size and extracts the feature from the feature map obtained from the Convolution layer, the flattened layer, which converts batches of feature maps into a 1D tensor, and finally three fully connected layers, the first two of which have a dense unit of 4096 layers. The final classification layer is classified in 1000 ways and so has 1000 channels (one for each class). The soft-max layer is the final layer.

Fast Gradient Sign Method (FGSM) Attack

Adversarial examples are inputs designed specifically to confuse a neural network, resulting in the misclassification of a given input. These renowned inputs are indistinguishable to the human eye, but they prevent the network from recognizing the image's contents. The focus of this section is on the quick gradient sign method attack, a white-box attack whose objective is to cause misclassification. In a white box attack, the attacker has entire access to the target model. This work contains one of the most renowned instances of an adversarial image, which is displayed below. Goodfellow et al. [31] initially suggest the FGSM, an efficient untargeted attack, to generate adversarial samples in the benign samples' L∞ neighbor, as seen in Fig.4.

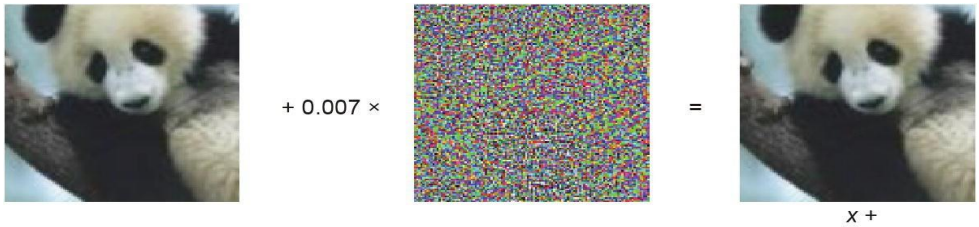

Figure 5: An example of an adversarial sample generated by using FGSM on GoogleNet [5]. FGSM's undetectable disturbance fools GoogleNet into misidentifying the image as a gibbon.

FGSM is a typical one-step attack algorithm that executes a one-step update along the gradient of the adversarial loss $J\theta$, x, y direction (i.e., the sign), to maximize the loss in the steepest direction. Technically, the adversarial sample created by the FGSM is written as follows:

$$x' = x + \epsilon \cdot sign[\nabla_x J(\theta, x, y)] \quad (1)$$

where ϵ is the magnitude of the perturbation, x' is the adversarial image, x is the original input image, y is the input label, θ model parameters, and J is the loss.

By reducing the gradient $J\theta, x, y,$ FGSM can be easily extended to a focused attack algorithm (targeted FGSM). If cross-entropy is used as the adversarial loss, this update process can reduce the cross-entropy between the predicted probability vector and the target probability vector. The following is the updated rule for targeted FGSM:

$$x' = x - \epsilon \cdot sign[\nabla_x J(\theta, x, y)] \quad (2)$$

Projected Gradient Descent (PGD) Attack

The PGD attack is a white-box attack, which means that the attacker gets access to the model gradients, i.e. a copy of your model's weights. This threat model provides the attacker with significantly greater power than black box attacks since they may tailor their attack to mislead your model without relying on transfer attacks, which frequently result in human-visible perturbations. PGD is the most "complete" white-box adversary because it removes all limitations on the amount of time and effort an attacker may devote to discovering the optimum attack. The key to understanding the PGD attack is to think of it as a constrained optimization problem in search of an adversarial example. PGD seeks the perturbation that minimizes a model's loss on a specific input while keeping the size of the perturbation smaller than a predefined amount known as epsilon. This constraint is commonly stated as the perturbation's L^2 or $L\infty$ norm, and it is applied so that the content of the adversarial example is the same as the unperturbed sample — or even so that the adversarial example is indistinguishable from humans.

$$x_{i+1}^{`} = \Pi(FGSM(x_i^{`})) \quad (3)$$

Where FGSM (x') represents an FGSM update of x'as in (2), and the outer product function 1 1 keeps x' $I+1$ within a predefined perturbation range. PGD can also be interpreted as an iterative algorithm to the following problem:

$$x': \max_{||x'-x||\infty<\alpha} L(x', y; \theta) \quad (4)$$

According to Madry et al. [33], the local maxima of the cross-entropy loss obtained by PGD with 10^5 random starts are diverse, but all have similar loss values for both normally and adversarial-trained networks. Based on this concentration phenomenon, they claim that PGD is a universal adversary across all first-order adversaries, implying that attacks exclusively use first-order information.

Image Preprocessing

Pre-processing aims to improve the image's quality so the goal of pre-processing is to improve the image's quality so that we can examine it more effectively. We can minimize unwanted distortions and increase specific features that are required for image classification by preprocessing.

For this research, the input image size was 224x224. As a result, all of the images were downsized to the 224x224 resolution. The batch size is set at 100. Each training cycle results in 171 steps. The training steps per epoch are calculated by dividing the total number of training objects by the batch size. Before training the network, the model must be compiled after preprocessing. The optimizer, loss function, and metrics are three parameters that must be declared during training. The optimizer and loss function are the key components that allow the network to cope with data. Simply, an optimizer governs the rate at which a neural network learns [30]. The Stochastic Gradient Descend method was used to optimize the aforementioned models.

Environment

We shorten neural network training time by compressing the original images to 224 by 224 pixels. The training was done on Google Colab. TensorFlow 2.4.1 [26], Keras 2.4.3 [27] contained in the TensorFlow library for the development and training of the neural network, sci-kit-learn 0.24.1 [28] for the accuracy analysis, and matplotlib 3.3.4 [29] for graph modeling were used to implement the methods.

RESULTS

Throughout the dataset's training, the metrics Accuracy and Loss were measured. These measures were measured using both training and validation data. Training the model was fast because we employed the use of GPUs. Tables I, II, and III show the accuracy and loss for the training and validation data sets, respectively. The accuracy and loss during fine-tuning are displayed in Figures 4, 5, and 6. Each loss graph demonstrates that our models were not overfitting and were effectively trained.

Table 3: Accuracy and Loss Indicators of Inception V3.

State	Metrics	Values
Fine-Tuning	Training Accuracy	0.9953
	Training Loss	0.0218
	Validation Accuracy	0.9921
	Validation Loss	0.0256

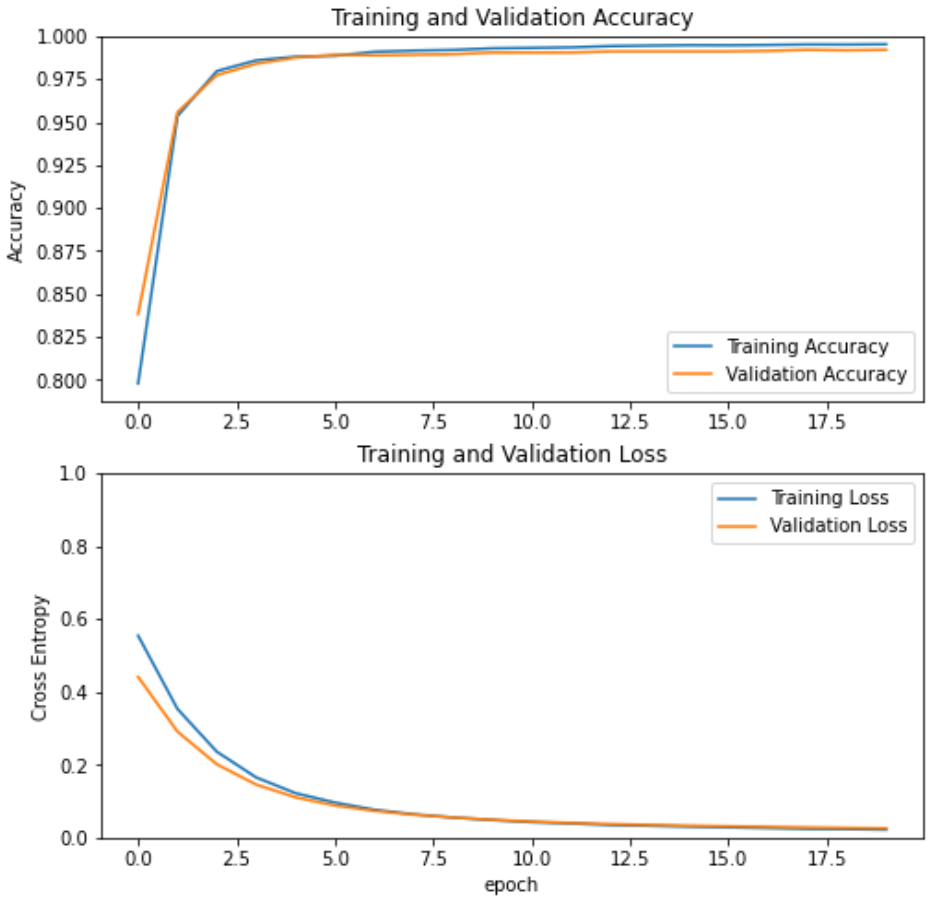

Figure 6: Graph of Accuracy and Loss during training

The fine-tuned Inception V3 architecture achieved a validation accuracy of 99.21% with a loss of 2.56% and a training accuracy of 99.53% with a loss of 2.18%. The Inception V3 fine-tuned model accurately predicted around 100% of the test data, showing that the model is doing effectively.

Table 4: Accuracy and Loss indicators of MobileNetV3.

State	Metrics	Values
Fine-Tuning	Training Accuracy	0.9336
	Training Loss	0.1592
	Validation Accuracy	0.9230
	Validation Loss	0.1756

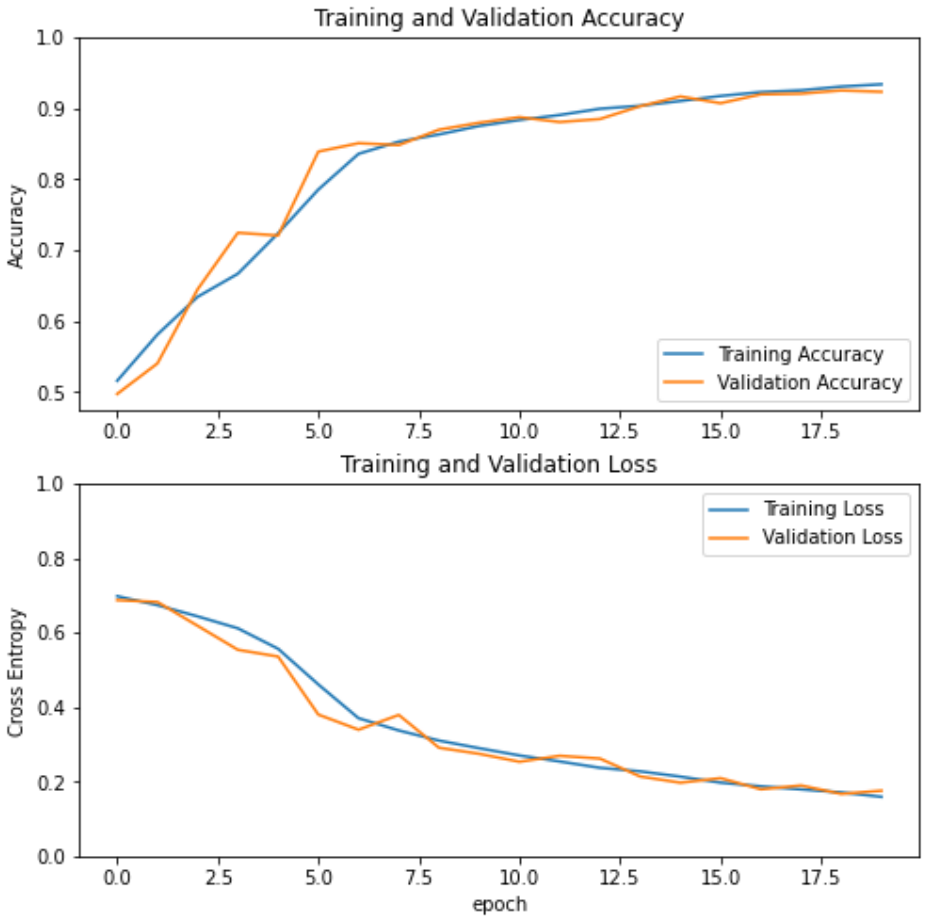

Figure 7: Graph of Accuracy and Loss during training

The fine-tuned MobileNet V3 small architecture achieved a validation accuracy of 92.30% with a loss of 17.56% and a training accuracy of 93.36% with a loss of 15.92%. The Inception V3 fine-tuned model accurately predicted around 92.20% of the test data, showing that the model is doing effectively.

Table 5: Accuracy and Loss Indicators of VGG19

State	Metrics	Values
Fine-Tuning	Training Accuracy	0.9109
	Training Loss	0.2385
	Validation Accuracy	0.9092
	Validation Loss	0.2362

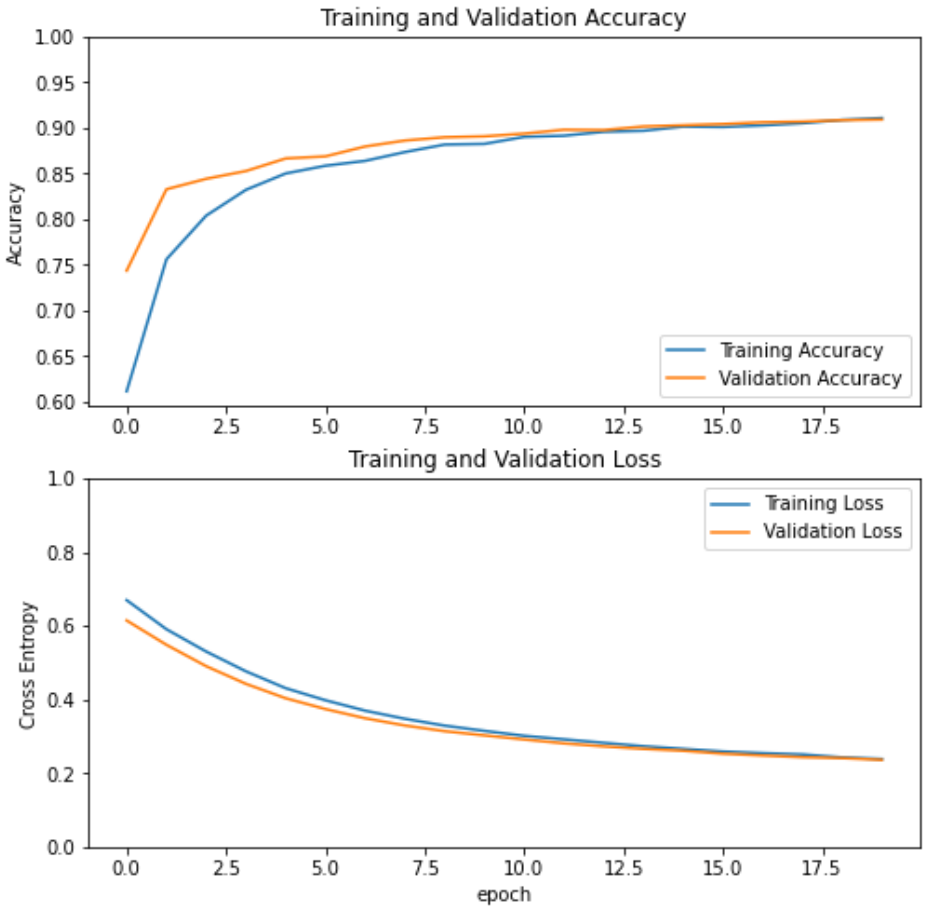

Figure 8: Graph of Accuracy and Loss during training

The fine-tuned VGG-19 architecture achieved a validation accuracy of 90.92% with a loss of 23.62% and a training accuracy of 91.09% with a loss of 23.85%. The VGG-19 fine-tuned model accurately predicted around 89.27% of the test data, showing that the model is doing effectively.

FGSM Attack

The fast gradient sign method generates an adversarial example using the neural network's gradients. The method employs the gradients of the loss with respect to the input image to generate a new image that maximizes the loss for an input image. This new image is known as the adversarial image. Here, beginning with a photograph of an input image, the attacker adds minor perturbations (distortions) to the original image, causing the model to confidently categorize this image as a different image. A similar procedure for adding these perturbations is described in detail below.

	precision	recall	f1-score	support
Without_Mask	1.00	1.00	1.00	2047
With_Mask	1.00	1.00	1.00	2023
accuracy			1.00	4070
macro avg	1.00	1.00	1.00	4070
weighted avg	1.00	1.00	1.00	4070

	Without_Mask	With_Mask
Without_Mask	2039	8
With_Mask	7	2016

Figure 9: Report on the performance of the best-fine-tuned model

As seen in the above classification report, the model was able to predict the images both with and without, with a perfect score. Moreover, out of all the without mask images that the model predicted would be images without masks, 100 percent were actually right. Additionally, out of the without mask images that were without mask images, the model only predicted this outcome correctly in all of those images. Since the F-score is 1, it tells us that the mode does a great job of predicting whether or not someone wears a mask.

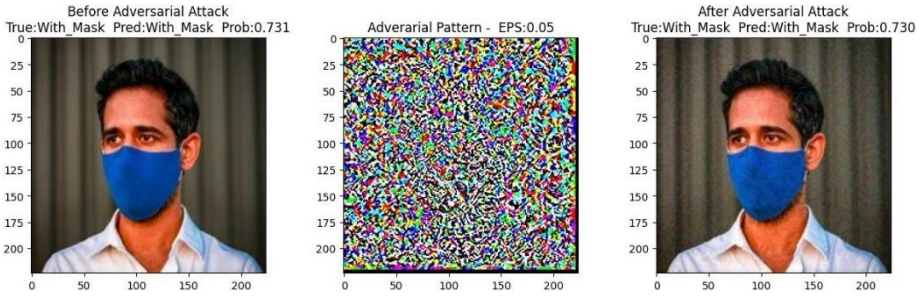

Figure 10: image after FGSM attack on the model

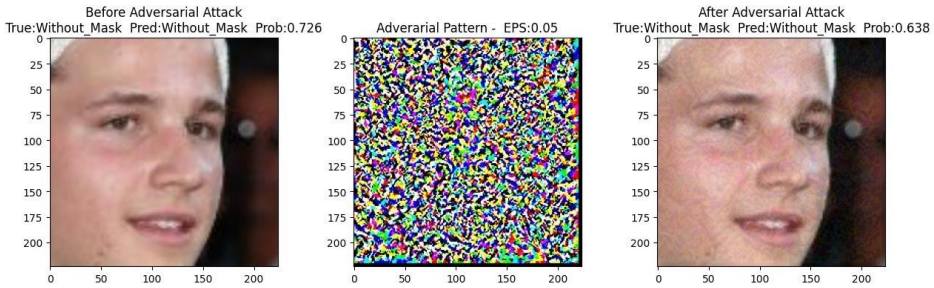

Figure 11: image after FGSM attack on the model

From both Figures 10 and 11, the distortion created with an epsilon of 0.05 was able to fool the model. An instance in Figure 10 shows that before the adversarial attack, the model was able to predict the image correctly with a probability of 0.726 and after the model's distortion, the model was able to predict the image with a probability of 0.638.

	precision	recall	f1-score	support
Without_Mask	0.57	0.59	0.58	2047
With_Mask	0.57	0.55	0.56	2023
accuracy			0.57	4070
macro avg	0.57	0.57	0.57	4070
weighted avg	0.57	0.57	0.57	4070

	0	1
Without_Mask	1211	836
With_Mask	911	1112

Figure 12: Classification report after FGSM attack

However, the classification report above shows after the FGSM attack that, the accuracy of the model was reduced by 43 percent. Moreover, out of all the without-mask images that the model predicted would be images without masks, only 57 percent were right. Additionally, out of the without mask images that were without mask images, the model only predicted this outcome correctly for 36 percent of those images. Since the F-score is not close to 1, it tells us that the mode does a poor job of predicting whether or not someone wears a mask.

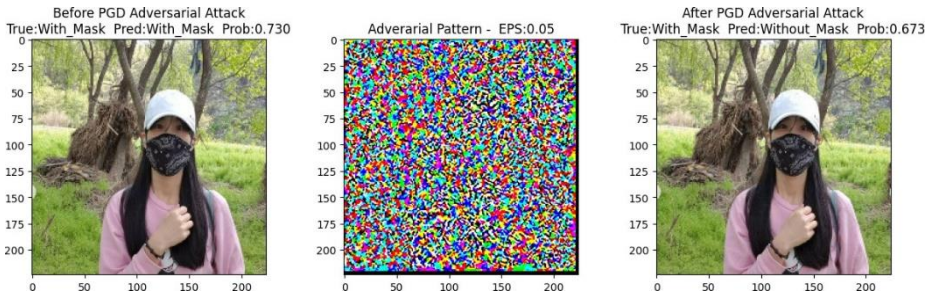

Figure 12: image after PGD attack on the model

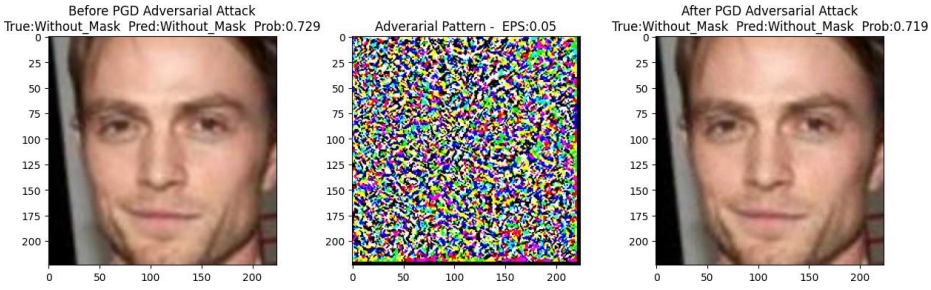

Figure 13: image after PGD attack on the model

From both Figure 12 and 13, the distortion created with an epsilon of 0.05 was able to fool the model. An instance in Figure 12 shows that before the adversarial attack, the model was able to predict the image correctly with a probability of 0.730 and after the model's distortion, the model was able to predict the image with a probability of 0.673.

	precision	recall	f1-score	support
Without_Mask	0.80	0.81	0.81	2017
With_Mask	0.81	0.81	0.81	2053
accuracy			0.81	4070
macro avg	0.81	0.81	0.81	4070
weighted avg	0.81	0.81	0.81	4070

	0	1
Without_Mask	1626	391
With_Mask	394	1659

Figure 14: Classification Report After PGD Attack On The Model

However, the classification report shows after the PGD attack that, the accuracy of the model was reduced by 19 percent. Moreover, out of all the without mask images that the model predicted would be images without masks, only 81 percent were right. Additionally, out of the without mask images that actually were without mask images, the model only predicted this outcome correctly for 81 percent of those images. Since the F-score is close to 1, it tells us that the mode does a good job of predicting whether or not someone wears a mask.

CONCLUSION AND FURTHER WORKS

Fine-tuning the classification of Face Mask pictures has shown positive results. According to the output accuracy, models trained in 20 epochs with 171 steps per epoch seem to be a good fit. After fine-tuning, the model that performed the best on the Face Mask dataset was Inception-V3, followed by MobileNet-V3 and VGG-19. In addition to the Deep CNN variations employed in this study, additional models, including GoogLeNet, ResNet, DenseNet, and others, should be applied to the Face Mask picture dataset. Using the ResNet or DenseNet model could perhaps enhance accuracy. In addition, utilizing various types of designs utilized in this investigation with greater layers may improve accuracy and decrease loss.

However, upon obtaining the best model, we applied the Fast Gradient Sign Method(FGSM) and Projected Gradient Descent(PGD) attacks to the model to check for its robustness. Both the FGSM and PGD attacks method after applying to the model generated adversarial examples using the model's gradients. Using the gradients of the loss concerning the input image, the method generates a new image that maximizes the loss for the input image. As shown in the study, the model although slightly robust, its accuracy reduced by 43% when attacked by FGSM indicating that the model was quite confused between the original image and the adversarial image. However, the model showed robustness when attacked by a PGD attack. The model produced an accuracy of 81%. These results indicate that FGSM attack was able to deceive or fool the best model as compared to the PDG attack. However, the model was also robust in handling these attacks.

To produce adversarial samples in this work, we used an efficient untargeted attack (FGSM and PGD). Other adversarial attacks, such as the L-BFGS algorithm, BIM attacks, Momentum Iterative

Attacks, and Deep Fool, can be used in future studies. In addition, it may be argued that fine-tuning had a favorable impact on the models because the data supported the decision. Yet, fine-tuning a model to execute a certain task more effectively can lead to improved results. On the other side, improper tuning might have a catastrophic effect on a previously trained model.

References

1. Cucinotta D., Vanelli M. WHO declares SARS-COV-2 a pandemic. Acta Biomed. 2020;91:157–160. doi: 10.23750/abm.v91i1.9397. - DOI - PMC - PubMed
2. S.M. Nagashetti, S. Biradar, S.D. Dambal, C.G. Raghavendra, B.D. Parameshachari Detection of Disease in Bombyx Mori Silkworm by Using Image Analysis Approach 2021 IEEE Mysore Sub Section International Conference (MysuruCon), IEEE (2021), pp. 440-444 View PDF CrossRefView Record in Scopus
3. R.K. Kodali, R. Dhanekula Face Mask Detection Using Deep Learning 2021 International Conference on Computer Communication and Informatics (ICCCI) (2021),pp. 1-5, 10.1109/ICCCI50826.2021.9402670
4. D.L. Vu, T.K. Nguyen, T.V. Nguyen, T.N. Nguyen, F. Massacci, P.H. Phung A convolutional transformation network for malware classification 2019 6th NAFOSTED conference on information and computer science (NICS), IEEE (2019), pp. 234-239 View PDF CrossRefView Record in Scopus
5. P. Khamlae, K. Sookhanaphibarn, W. Choensawat An Application of Deep-Learning Techniques to Face Mask Detection During the COVID-19 Pandemic 2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech) (2021), pp. 298-299, 10.1109/LifeTech52111.2021.9391922 View PDF View Record in Scopus
6. P. Kiran, B.D. Parameshachari, J. Yashwanth, K.N. Bharath Offline signature recognition using image processing techniques and back propagation neuron network system SN Computer Science, 2 (3) (2021), pp. 1-8
7. M.H. Rusli, N.N.A. Sjarif, S.S. Yuhaniz, S. Kok, M.S. Kadir Evaluating the Masked and Unmasked Face with LeNet Algorithm 2021 IEEE 17th International Colloquium on Signal Processing & Its Applications, CSPA (2021),pp.171-176,10.1109/CSPA52141.2021.937 7283 View PDF View Record in Scopus
8. K. Bhambani, T. Jain, K.A. Sultanpure Real-time Face Mask and Social Distancing Violation Detection System using YOLO 2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC) (2020), pp. 1-6, 10.1109/B-HTC50970.2020.9297902 View PDF View Record in ScopusGoogle Scholar
9. Bu, W.; Xiao, J.; Zhou, C.; Yang, M.; Peng, C. A cascade framework for masked face detection. In Proceedings of the 2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), Ningbo, China, 19–21 November 2017; IEEE: Piscatway, NJ, USA, 2017; pp. 458–462.
10. Jiang, M.; Fan, X.; Yan, H. Retinamask: A face mask detector. arXiv 2020, arXiv:2005.03950.
11. Chen, Y.; Hu, M.; Hua, C.; Zhai, G.; Zhang, J.; Li, Q.; Yang, S.X. Face mask assistant: Detection of face mask service stage based on mobile phone. IEEE Sensors J. 2021, 21, 11084–11093.
12. Eyiokur, F.I.; Ekenel, H.K.; Waibel, A. A computer vision system to help prevent the transmission of COVID-19. arXiv 2021, arXiv:2103.08773.

13. Christa, G.H.; Jesica, J.; Anisha, K.; Sagayam, K.M. CNN-based mask detection system using openCV and MobileNetV2. In Proceedings of the 2021 3rd International Conference on Signal Processing and Communication (ICSPC), Coimbatore, India, 13–14 May 2021; IEEE: Piscatway, NJ, USA, 2021; pp. 115–119.
14. Vinh, T.Q.; Anh, N.T.N. Real-time face mask detector using YOLOv3 algorithm and Haar cascade classifier. In Proceedings of the 2020 International Conference on Advanced Computing and Applications (ACOMP), Quy Nhon, Vietnam, 25–27 November 2020; IEEE: Piscatway, NJ, USA, 2020; pp. 146–149.
15. Chandra, Y.B.; Reddy, G.K. A comparative analysis of face recognition models on masked faces. Int. J. Sci. Technol. Res. 2020, 9, 175–178.
16. Zhang, E. A Real-Time Deep Transfer Learning Model for Facial Mask Detection. In Proceedings of the 2021 Integrated Communications Navigation and Surveillance Conference (ICNS), Virtual, 20–22 April 2021; IEEE: Piscatway, NJ, USA, 2021; pp. 1–7. [Google Scholar]
17. Militante, S.V.; Dionisio, N.V. Real-time facemask recognition with alarm system using deep learning. In Proceedings of the 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 8 August 2020; IEEE: Piscatway, NJ, USA, 2020; pp. 106–110.
18. Jignesh Chowdary, G.; Punn, N.S.; Sonbhadra, S.K.; Agarwal, S. Face mask detection using transfer learning of inceptionv3. In Proceedings of the International Conference on Big Data Analytics, Sonipat, India, 15–18 December 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 81–90.
19. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement 2021, 167, 108288.
20. Mercaldo, F.; Santone, A. Transfer learning for mobile real-time face mask detection and localization. J. Am. Med. Inform. Assoc. 2021, 28, 1548–1554.
21. Bala, Diponkor, "Face Mask Dataset 2022", 2022 Mendeley Data, V1, doi:10.17632/7bt2 d592b9.1
22. Enoch Binney, Dongxiao Ren. Coffee Leaf Diseases Classification and the Effect of Fine-tuning on Deep Convolutional Neural Networks. IJFMR Volume 4, Issue 5,
23. September-October 2022. DOI 10.36948/ijfmr.2022.v04i05.861
24. Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam. 2019, Abs/1905.02244. Searching for MobileNetV3. arXiv:1905.02244
25. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. http://arxiv.org/abs/1409.1556. 4 Sept. 2014
26. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. 2015, abs/1512.00567. arXiv:1512.00567v3.
27. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Man e, R. Monga,
28. S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vi egas, O. Vinyals, P. Warden, M. Wattenberg, M.

Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," https://www.tensorflow.org/, 2015.
29. F. Chollet et al., "Keras," https://keras.io, 2015.
30. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion,O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
31. J. D. Hunter, "Matplotlib: A 2d graphics environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.
32. Reyes AK, Caicedo JC, Camargo JE. Fine-tuning Deep Convolutional Networks for Plant Recognition. CLEF (Working Notes). 2015 Sep 8;1391.
33. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. 2014. arXiv:1412.6572.
34. 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
35. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.