Computers, Materials & Continua
DOI:10.32604/cmc.2022.020111
Article

Tech Science Press

Deep Image Restoration Model: A Defense Method Against Adversarial Attacks

Kazim Ali[1,*], Adnan N. Qureshi[1], Ahmad Alauddin Bin Arifin[2], Muhammad Shahid Bhatti[3], Abid Sohail[3] and Rohail Hassan[4]

[1]Department of Information Technology, University of Central Punjab, Lahore, 54000, Pakistan
[2]Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, University Putra Malaysia, Salengor, 43400, Malaysia
[3]Department of Computer Science, Comsats University Islamabad, Lahore Campus, 54000, Pakistan
[4]Othman Yeop Abdullah Graduate School of Business, University Utara Malaysia, Kuala Lumpur, 50300, Malaysia
*Corresponding Author: Kazim Ali. Email: kazimravian2003@gmail.com

Received: 10 May 2021; Accepted: 27 July 2021

Abstract: These days, deep learning and computer vision are much-growing fields in this modern world of information technology. Deep learning algorithms and computer vision have achieved great success in different applications like image classification, speech recognition, self-driving vehicles, disease diagnostics, and many more. Despite success in various applications, it is found that these learning algorithms face severe threats due to adversarial attacks. Adversarial examples are inputs like images in the computer vision field, which are intentionally slightly changed or perturbed. These changes are humanly imperceptible. But are misclassified by a model with high probability and severely affects the performance or prediction. In this scenario, we present a deep image restoration model that restores adversarial examples so that the target model is classified correctly again. We proved that our defense method against adversarial attacks based on a deep image restoration model is simple and state-of-the-art by providing strong experimental results evidence. We have used MNIST and CIFAR10 datasets for experiments and analysis of our defense method. In the end, we have compared our method to other state-of-the-art defense methods and proved that our results are better than other rival methods.

Keywords: Computer vision; deep learning; convolutional neural networks; adversarial examples; adversarial attacks; adversarial defenses

1 Introduction

When Artificial Neural Networks (ANNs) consists of more than one hidden layer, then it is called deep learning. Deep learning (DL) is the subfield of Machine Learning (ML), and ML is the subfield of Artificial Intelligence (AI). These deep learning models have gained tremendous success in object recognition, object detection, speech recognition, and drug discovery. Convolutional Neural Networks

(CNNs) are state-of-the-art models for doing different tasks in the computer vision domain [1]. The CNN is a deep learning model used in image processing and the computer vision field to do various tasks. These tasks are image classification, segmentation, object detection, object tracking, video classification, text classification, speech recognition, language translation, autonomous vehicles, robotics, network security, safety-critical system, face recognition, medical science, mobile applications, and other utilities [2].

The adversarial examples are input images imperceptible by the human visual system. A human eye recognizes or classifies correctly without any hesitation, but a deep learning model like CNN can misclassify with high probability or confidence [3]. It is required to take necessary actions against adversarial example attacks because deep learning algorithms are not only limited to the laboratory but are much used in real-world fields [4] such as image recognition, speech recognition, and medical diagnostic, etc. It is possible to attack deep learning models deployed in the physical world, and a model can capture incorrect data through sensors [5]. In the presence of an adversarial example, it raises a big question of the robustness of deep learning models [6]. Recent research shows that DL algorithms cannot give correct results due to adversarial attack [7]. The researcher has gained great success in modern deep learning algorithms but is less concentrated on a robust and security perspective [8]. There are currently two active research areas on adversarial attacks. The first concentrates on creating adversarial example attacks, and the second is on developing defense methods against these attacks. There is a similarity between these two groups of researchers [9]. Szegedy et al. [10] presented the concept of adversarial examples for the first time in 2014 in their paper titled "intriguing properties of neural networks". The authors stated and proved that adversarial examples are a significant threat to deep learning algorithm security, especially in the computer vision domain [10].

In this work, we will propose a defense method to restore adversarial examples to get back the correct prediction of a deep learning model in the computer vision field. The paper is structured as follows. Section 1 presents the introduction of the research area. Section 2 contains the related work, which provides well-known adversarial example attacks and defense methods against these attacks. Section 3 presents our proposed deep-image restoration model that reconstructs adversarial examples to restore a model's performance. Section 4 contains our experiments and results to prove that our proposed method works effectively and performs better when compared with the other state-of-the-art defense methods. Sections 5 and 6 present the discussion and conclusion of this research work.

Our main contributions can be summarized as follows:

- We present a novel method that recovers adversarial examples from the different types of adversarial attacks.
- We propose a deep image restoration model that eliminates the perturbation from adversarial examples to restore in almost original examples, and restored samples are classified correctly.
- Our method does not require changing the internal structure of the model like hidden layers, activation functions, and output layers to remove adversarial noise. We only need the original input image and its adversarial version to get the correct pattern again for classification, unlike the existing methods.
- Our method does not need any detector method to detect adversarial noise or adversarial attack because our method will start its work after a successful adversarial attack. A successful attack means the target model misclassifies the test images.
- Our baseline technique is cGAN ((Conditional Generative Adversarial Network)) defense which uses the power of cGAN to destroy the adversarial noise from the adversarial examples.

Our proposed method is inspired by this but consisted of a simple structure and gives good results.

2 Related Works

This section consists of two subsections. The first section describes some well-known adversarial attack methods, and the second section has consisted of defense methods.

2.1 Adversarial Attacks

There are two types of adversarial attacks that create adversarial examples; (1) gradient-based attacks, in these attacks. The attacker has complete knowledge and access to the model, including structure and parameters, and (2) decision-based attacks, in these types of attacks; the attacker has only observed the output or decision of the underlying model. We will describe these two types of attacks and restore adversarial examples created due to these two types of adversarial attacks in the experiments section to restore the prediction accuracy of deep learning models.

Fast Gradient Sign Method (FGSM) (gradient-based attack) [11], the adversarial example can be created from the original image in a single step through the following Eq. (1):

$$x' = x + \varepsilon \ . \ sign(\Delta x.L(x, y) \tag{1}$$

where x is the original image, $sign\Delta x.L(x,\ y)$ represents the sign of the gradient of the loss with respect to x, ε is a small constant which controls the adversarial perturbation and x' is the adversarial example.

Basic Iterative Method (BIM) (gradient-based attack) [12] is a variant of FGSM [11]. BIM creates an adversarial example through the following Eq. (2):

$$x'_{i+1} = Clip_{\epsilon}\{x'_i + \alpha \ . \ sign(\nabla_x \theta(x'_i, y))\} \ for \ i = 0 \ to \ n, \ and \ x'_0 \ = x \tag{2}$$

Here n represents iterations, α shows step size, and the Clip (.) function has clipped the values of pixel intensities in the range 0−255 in case of an 8-bit image.

Projected Gradient Descent Attack (PGD) (gradient-based attack) [13], which is also an iterative method that crafts adversarial samples by using FGSM [11] on clean example x_0 iteratively, which is created by adding a random noise of quantity α in the original image x. After this, the adversarial example is projected on applicable limits. The projection is searched for the nearest matching sample from the original images, away from the boundary of the original sample. It is explained by the following Eq. (3):

$$x^{i+1} = Proj_{x+S} \ (x^i + \alpha sign(\nabla_{x^i} J(\theta, x^i, t))) \tag{3}$$

where x^{i+1} is the perturbed input at iteration $i + 1$ and S denotes the set of feasible perturbations for x.

Deep Fool Attack (DFA) (gradient-based attack) [14] is a non-targeted attack that is based on l_2 norms. The adversarial examples are produced by the following Eq. (4):

$$r(x_0 = \arg min||r||_2) \tag{4}$$

$$such \ that \ f(x + r) \neq f(x) \ where \ r \ is \ a \ minimum \ perturbation.$$

Carlini and Wanger Attack (CWA) (gradient-based attack) [15] develops three types of adversarial attacks based on the $l_1, l_2, and\ \ l_\infty$ norms. These three attacks especially failed the defensive distillation network, which is a defensive method to increase deep learning algorithms' robustness.

The Spatial Attack (SPA) (decision-based-attack) [16], a classifier is easily fooled using simple image processing techniques like transformation and rotation, and the input image slightly rotate or transforms so that the human visual system classifies it correctly. However, a model misclassified it with high confidence.

2.2 Defense Methods

There are mainly three types of adversarial defense methods in the current literature. First involves the preprocessing of the input data during learning or testing by a model. Second, the defense changes the internal structure of the model by modifying or adding, or dropping any layer in the model's structure. Third, the defenses in which external models are used to destroy adversarial noise.

In adversarial training defensive techniques [11], a model's robustness is increased by adding adversarial examples in the training data and then retrains the model. After retraining the model on the adversarial examples, it will correctly classify the adversarial example to increase the model's robustness. The objective function is given as follows:

$$\alpha L(I, y) + (1 - \alpha)L(I', y) \tag{5}$$

where $L(I,\ y)$ is the objective function, I' is the adversarial example of the original input I and α are constant whose purpose is to balance the cost value between original and adversarial images, which has a constant value of 0.5.

Ensemble Adversarial Training (EAT) [17] is called a new version of the old AE method [11]. The classifier is retraining on adversarial samples which are created for other existing classifiers. The combination of classifier and training adversarial examples prevents over-fitting problems in the old method. EAT approximates inner maximization because of adversarial samples' universal property, among other models.

The Defensive Distillation Method [18] consists of two networks. The first neural network is called the student network, and the second neural network is called a teacher network. The teacher network uses the predicted labels of the first network as inputs and then approximates the first network, increasing the network's robustness. However, this method fails to defend against CWA-based attacks [15].

Mag-Net [19] is a defense method to increase a model's robustness, consisting of two auto-encoders. One is called the detector, and the other is called the reformer. Both auto encoders reconstruct the original input image. The detector is used to detect adversarial perturbation, and the reformer is used to remove that perturbation to increase the robustness of the deep neural network model.

The defense GAN method [20] also consists of a generative model trained on clean images to remove adversarial noise. The Defense GAN uses the GAN model with Wasserstein's loss. The GAN defense method tries to reconstruct the adversarial examples into clean examples used as an add-on. The result of reconstruction is fed to the classifier and aims to reduce adversarial perturbation.

The conditional GAN-Defense method [21] uses the power of the conditional generative adversarial network, which is a variant of the classic generative adversarial network. This method tries to minimize the adversarial perturbation from adversarial examples and then fed reconstructed examples to the target classifier, aiming to restore the predicted accuracy of the underlying model. It is also our baseline technique but our proposed method has simple layers structure to remove adversarial perturbation from adversarial images and gives better results.

3 Proposed Defensive Method

This section will present our proposed defense method, which improves the robustness of CNN models against adversarial attacks, which we have already discussed in the related work Section 2.1. The overall structure of our defense mechanism is shown in Fig. 1.

Our proposed defense method has five phases as follows:

Phase I: We will use the CNN models, e.g., Mobile-Net and Dense-Net, for the cifar10 dataset; M1 and M2 models for the MNIST dataset, their structures are described in Tab. 3; these are used as target models. These models are under the threat of adversarial attacks because CNNs are much weaker under the threat of adversarial attacks. Adversarial attacks decrease the performance of these models. Therefore, we will work towards the robustness of the CNN models against adversarial attacks, so that performance of the model is not degraded on the adversarial images.

Phase II: In this phase, we intentionally apply our adversarial attacks box to create different types of adversarial samples. Our adversarial attack engine creates adversarial examples by using FGSM, BIM, PGD, DFA, CWA, and SPA methods discussed in the related work Section 2.1.

Phase III: We feed our adversarial examples to our target CNN models. The models predict the wrong label of an adversarial image, e.g., the output (soft-max) layer of the model predicts the label or class of the adversarial example seven is three, which is the wrong label. Its mean attack is successful and spoils the correct prediction of the target model.

Phase IV: Now in this important phase; we will feed the adversarial examples created in phase III; into our proposed deep image restoration model which will be already trained to remove adversarial perturbation. For example, we feed adversarial example seven to the proposed restoration model and as a result, we get a reconstructed image that is clean and adversarial-free.

Phase V: In the end, we will feed restored adversarial examples to our target models, which are generated in phase IV by using our proposed deep image restoration models, and then checks their prediction and observed that prediction is correct to measure and evaluate the effectiveness of our method.

The structures of our proposed deep image restoration model for the datasets MNIST and cifar10 are shown in Tabs. 1 and 2 respectively.

Tabs. 1 and 2 present the structures of our proposed deep image restoration model for datasets MNIST and CIFAR-10, respectively. There is a slight difference between the two structures due to the different dimensions of images of the two datasets, but the concept is the same. Our image restoration model is specific to remove adversarial noise from adversarial examples which are created due to the adversarial attacks. Our proposed model consists of two parts: encoder and decoder but works as a single network. The encoder part reduces the dimensionality of the adversarial images by learning the necessary features. Thus, when we are fed adversarial or perturbed example/image into the encoder, it only learns the critical and necessary information of the perturbed image. Since the encoder learns only the important and necessary information to represent the image, it learns that adversarial noise or perturbation is unwanted information and removes the representations of adversarial noise or perturbation from the learned features. Encoder learns 2048 features from the MNIST dataset and

4096 features from the cifar10 dataset shown in Tabs. 1 and 2. Thus, now we will have learned features of the encoder, that is, a representation of the image without any adversarial noise information. When this learned representation of the encoder, the features or intensities, is fed to the decoder. The decoder restores the adversarial image into the clean image from the encodings produced by the encoder. Since the encodings have no noise, the restored image will not contain any adversarial noise.

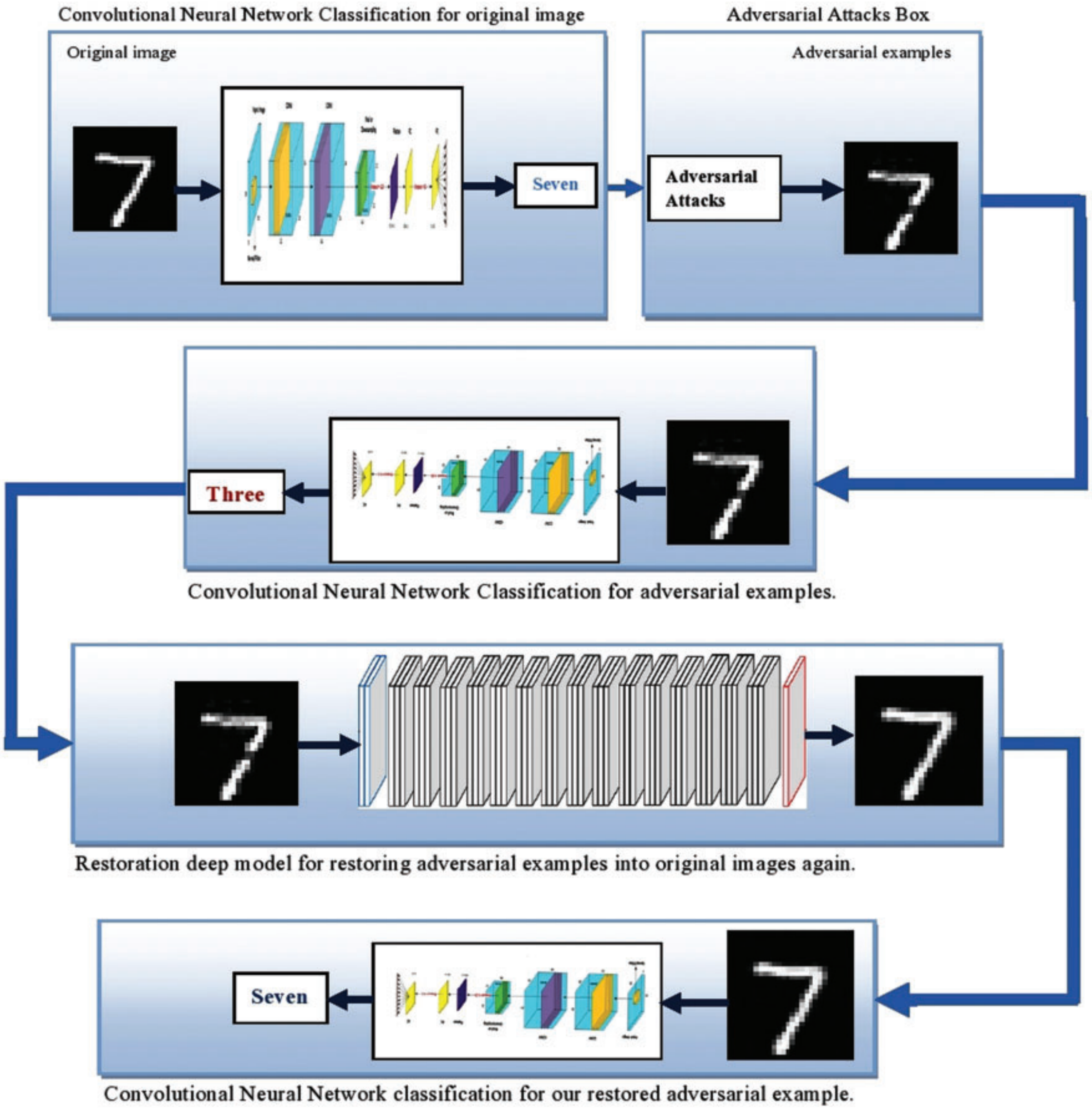
Convolutional Neural Network Classification for original image
Original image
Seven
Adversarial Attacks Box
Adversarial examples
Adversarial Attacks
Three
Convolutional Neural Network Classification for adversarial examples.
Restoration deep model for restoring adversarial examples into original images again.
Seven
Convolutional Neural Network classification for our restored adversarial example.

Figure 1: The overall structure of our proposed defense method. It contains five main phases; (a) Apply CNN as an image classifier, (b) Adversarial attack box to create different types of adversarial images, (c) Feed adversarial image to CNN model, (d) A deep image restoration model to restore adversarial examples into clean examples again and (e) Feed the restored adversarial examples/images to CNN model again

Table 1: The structure of the proposed deep image restoration model to restore adversarial examples into clean or original examples for the MNIST dataset

Encoder	Decoder
Input-Layer (28, 28, 1)	Conv2DTranspose (512, 3 × 3, 2, same) + ReLU
Conv2D (64, 3 × 3, 2, same) + ReLU	Conv2DTranspose (512, 3 × 3, 1, same) + ReLU
Conv2D (64, 3 × 3, 1, same) + ReLU	Conv2DTranspose (256, 3 × 3, 2, same) + ReLU
Batch Normalization ()	Batch Normalization ()
Conv2D (128, 3 × 3, 2, same) + ReLU	Conv2DTranspose (128, 3 × 3, 2, same) + ReLU
Conv2D (128, 3 × 3, 1, same) + ReLU	Conv2DTranspose (128, 3 × 3, 1, same) + ReLU
Batch Normalization ()	Batch Normalization ()
Conv2D (256, 3 × 3, 2, same) + ReLU	Conv2DTranspose (64, 3 × 3, 2, same) + ReLU
Conv2D(256, 3 × 3, 1, same) + ReLU	Conv2DTranspose(64, 3 × 3, 1, same) + ReLU
Batch Normalization ()	Batch Normalization ()
Conv2D (512, 3 × 3, 2, same) + ReLU	Conv2D (1, 3 × 3, 1, same) + ReLU
Conv2D (512, 3 × 3, 1, same) + ReLU	Conv2D (1, 3 × 3, 1, same) + ReLU
Batch Normalization ()	Conv2D (1, 3 × 3, 1, same) + ReLU
Flatten ()	Batch Normalization ()
Dropout (0.2)	Output-Layer (28, 28, 1)
Dense (2048)	
Reshape (2, 2, 512)	Depth of model: 32 layers (from input-layer to output-layer)
	Total parameters: 15.93 million
	Trainable parameters: 15.92 million
	Non-trainable parameters: 0.003 millions
	Size of Model: 15.19 MB

Our proposed model is different from the traditional auto-encoder model because the auto-encoder loses low-level information [21]. Therefore they cannot restore images corrupted by the adversarial attack. The proposed model performs two operations which are encoding and decoding as a single network. This design has several advantages. First, we do not require two networks like traditional auto-encoder and adversarial generative network (GAN), which significantly improves the computational complexity. Second, we do not use the max-pooling layer for the encoding process because it does not maintain low-level information, extracts high-level information, and reduces the dimension, which does not help decode operation. It is also not a trainable layer. Third, we use only the convolution layer for extracting both low-level and high-level information, which decreases the dimension in the encoding process, and it is also trainable. Finally, we also use the convolution-transpose layer instead of the up-sampling layer in the decoding process because the convolution-transpose layer works well in the decoding or restoration process due to its trainability nature and effectiveness adversarial examples with perturbation free like the original image. The visual results of the proposed restoration model of adversarial examples are shown in Figs. 2 and 3 for the MNIST and CIFAR10 datasets, respectively.

Table 2: The structure of the proposed deep image restoration model for the restoration of adversarial examples into clean or original examples for the CIFAR10 dataset

Encoder	Decoder
Input-Layer (32, 32, 3)	Conv2DTranspose (256, 3 × 3, 2, same) + ReLU
Conv2D (64, 3 × 3, 2, same) + ReLU	Conv2DTranspose (256, 3 × 3, 1, same) + ReLU
Conv2D (64, 3 × 3, 1, same) + ReLU	Batch Normalization ()
Batch Normalization ()	Conv2DTranspose(128, 3 × 3, 2, same) + ReLU
Conv2D (128, 3 × 3, 2, same) + ReLU	Conv2DTranspose (128, 3 × 3, 1, same) + ReLU
Conv2D (128, 3 × 3, 1, same) + ReLU	Batch Normalization ()
Batch Normalization ()	Conv2DTranspose (64, 3 × 3, 2, same) + ReLU
Conv2D (256, 3 × 3, 2, same) + ReLU	Conv2DTranspose (64, 3 × 3, 1, same) + ReLU
Conv2D (256, 3 × 3, 1, same) + ReLU	Batch Normalization ()
Batch Normalization ()	Conv2DTranspose(3, 3 × 3, 1, same) + ReLU
Flatten ()	Output-Layer (32, 32, 3)
Dropout (0.2)	
Dense (4096)	Depth of model: 25 layers (from input-layer to output-layer)
Reshape (4, 4, 256)	Total parameters: 19.667 millions
	Trainable parameters: 19.665
	Non-trainable parameters: 0.001 Size of Model: 18.75 MB

Table 3: The structure of the target models M1 and M2 for the MNIST dataset

M1	M2
Conv2D (32, 3 × 3) + ReLU	Flatten ((28, 28))
MaxPooling2D ((2, 2))	Dense (56)) + ReLU
Conv2D (64, 3 × 3) + ReLU	Dense (56) + ReLU
MaxPooling2D ((2, 2))	Dense (10)
Conv2D (128, 3 × 3) + ReLU	Softmax()
MaxPooling2D ((2, 2))	
Flatten ()	
DropOut (0.2)	
Dense (128) + ReLU	
DropOut (0.2)	
Dense (10)	
Softmax ()	

Table 4: Success rate (%) of the Defense system on the MNIST dataset

Target model	FGSM	BIM	PGD	DFA	CWA	SPA
M1	98.96	99.31	99.54	99.3	99.9	97.51
M2	99.12	99.63	99.82	99.51	96.53	97.51

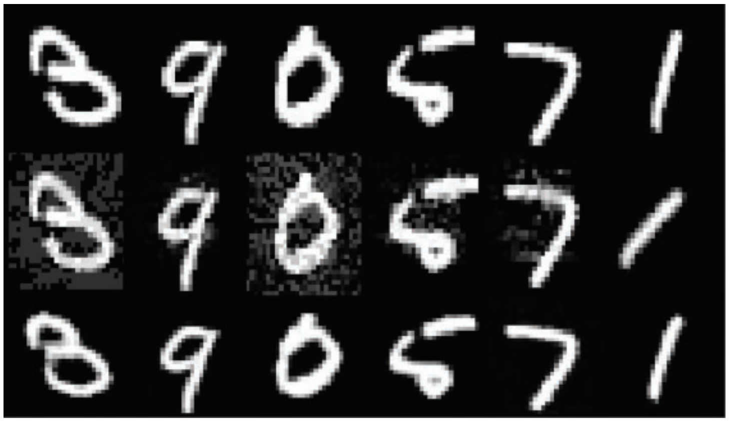

Figure 2: The first row shows the original images of the MNIST dataset, the second row shows the adversarial examples, and the third row represents the restored adversarial examples into original images

4 Experiments and Results

The datasets used in our experiments and evaluations are given by.

4.1 MNIST

The MNIST [22] dataset consisted of 70,000 handwritten digits from 0 to 9 grayscale images, 60,000 images are used for training and 10, 000 images are used for testing the model. The dimension of each image is 28 × 28 × 1. It is a simple dataset and used as a benchmark in computer vision for many years.

4.2 CIFAR-10

CIFAR-10 [23] is considered an alternative standard benchmark dataset for image classification in the computer vision and machine learning literature. CIFAR-10 consists of 60,000 32 × 32 × 3 (RGB) images resulting in a feature vector dimensionality of 3072. As the name suggests, CIFAR-10 consists of 10 classes, including airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

4.3 Evaluation Metrics

The performance or evaluation of the proposed method is measured through the following evaluation metrics:

$$original\_ccuraccy = \frac{Number\ of\ \ Correctly\ classified\ test\ images}{Total\ Number\ of\ \ test\ images} * 100 \tag{6}$$

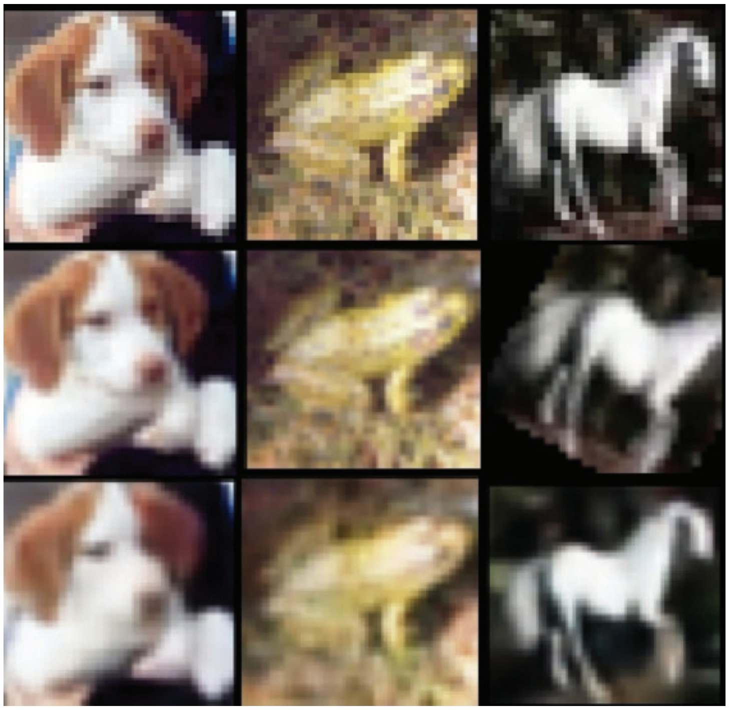

Figure 3: The first row shows the original images of the CIFAR10 dataset, the second row shows the adversarial examples, and the third row represents the restored adversarial examples into original images

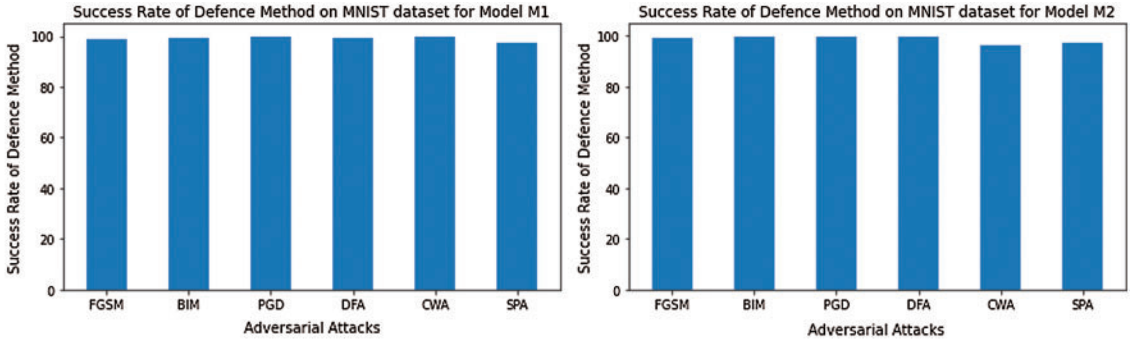

Figure 4: The success rate (%) of the defense system for the MNIST dataset on the target models M1 and M2

And

$$\mathit{adversarial\_ccuraccy} = \frac{\mathit{Number\ of\ Correctly\ classified\ restored\ adversarial\ examples}}{\mathit{Total\ Number\ of\ adversarial\ examples}} * 100 \tag{7}$$

Also

$$\mathit{sucess\_of\_\ restoration\_model} = \frac{\mathit{adversarial\_accuracy}}{\mathit{original\_accuracy}} * 100 \tag{8}$$

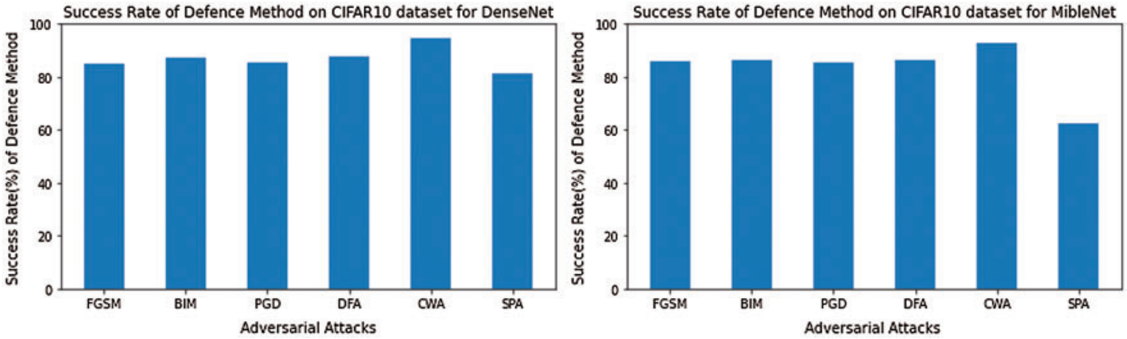

Figure 5: The success rate (%) of the defense system for the CIFAR10 dataset on the target model DenseNet and MobileNet

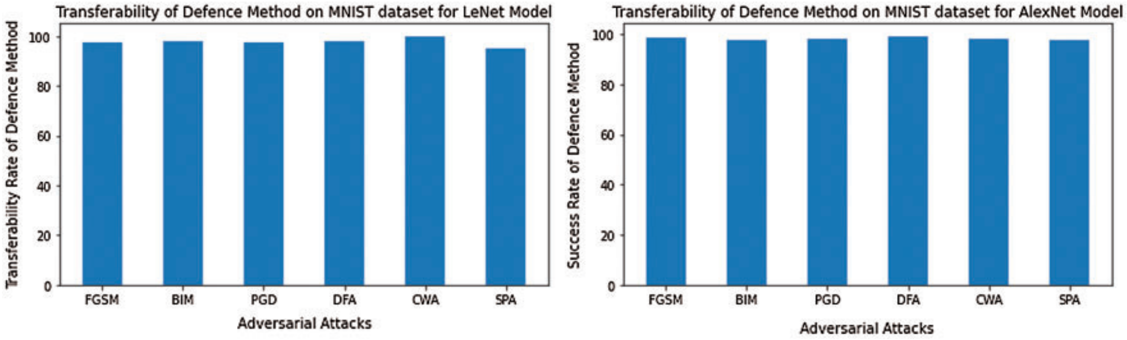

Figure 6: The transferability of the defense system on the MNIST dataset for LeNet and AlexNet models

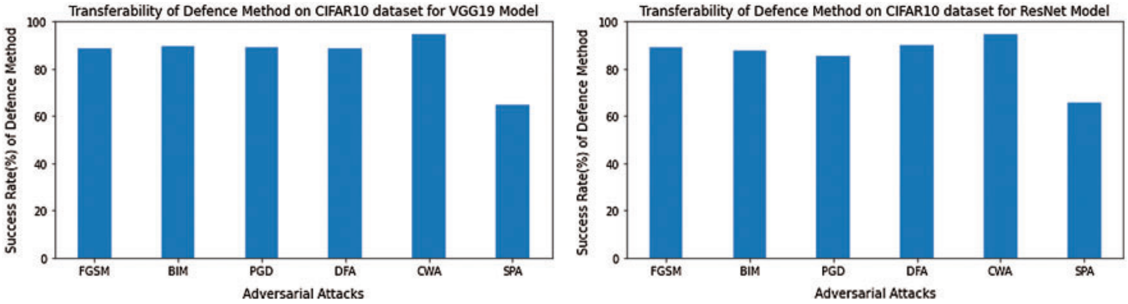

Figure 7: The transferability of the defense system on the CIFAR10 dataset for VGG19 and ResNet models

4.4 *Training CNN Models*

We will train four target models for the above two datasets, the structure of target models trained on the MNIST dataset as shown in Tab. 3. We named these two target models M1 and M2.

M1 has an accuracy of 99%, and M2 has 97.4% on the MNIST dataset. We have used pre-trained models Dense-Net [24] and Mobil-Net [25] for the CIFAR-10 dataset, which has also attained good accuracy of above 80%. CIFAR10 is a complex dataset, so it is challenging to give accuracy more remarkable than a simple dataset like MNIST. We create adversarial examples by using five adversarial attacks, namely, the FGSM, BIM, PGD, DFA, CWA, and SPA, from test set images correctly classified by our target models. CWA and DFA attacks are more robust than other attacks. Attacks robustness means they need small perturbation to create an adversarial example. However, our defense mechanism is performed well on all the above six types of attacks. Our defense system gives a high success rate on the MNIST dataset than the CIFAR-10 dataset, a complex dataset, but we also get a better success rate on the cifar10 dataset. Tab. 4 and Fig. 4 present the results of the proposed method for dataset MNIST. The results for the CIFAR10 dataset are described in Tab. 5 and Fig. 5.

4.5 Transferability of Defense Method

The transferability of defense mechanism means the performance of the defense system trained for target models, now test on other models that have no defense system. Therefore, we check the transferability of our proposed defense method on LeNet [26] and AlexNet [27] models for the MNIST dataset. Alternatively CIFAR10 dataset, we check transferability on pre-trained ResNet [28] and VGG19 [29] models. The results of the transferability of the defense system for models LeNet and AlexNet are given in Tab. 6 and Fig. 6. Also, results for models ResNet and VGG19 are presented in Tab. 7 and Fig. 7.

4.6 Comparison with Other Defense Methods

This section will present the comparison of our proposed defense method with the other well-known and state-of-the-art defense methods. The comparison results are given in Tabs. 8 and 9 for MNIST and CIFAR10 datasets, respectively.

Table 5: Success rate (%) of the defense system on the CIFAR10 dataset

Target model	FGSM	BIM	PGD	DFA	CWA	SPA
Dense-Net	84.95	87.38	85.56	87.62	94.66	81.31
Mobile-Net	86.13	86.23	85.34	86.29	93	62.53

Table 6: Transferability (%) of the defense system on MNIST dataset

Target model	FGSM	BIM	PGD	DFA	CWA	SPA
LeNet	97.45	97.96	97.66	97.96	100	94.91
AlexNet	98.47	97.75	98.26	99.28	98.26	97.51

4.7 Comparative Analysis

We have compared our proposed method with the other state-of-the-art methods such as adversarial training [11], MagNet [19], Defense-GAN [20], and cGan-Defence [21]. The adversarial training uses the adversarial examples as part of the training data to make a model robust. The MagNet uses two auto-encoders; one is called a detector to detect adversarial noise, and the other is called a reformer to remove adversarial noise. The Dense-GAN and cDefesce-GAN also use two networks called generator and discriminator to restore adversarial examples. The results of comparative analysis are shown in Tabs. 7 and 8. Our proposed defense method is better than the above method; (i) it gives better results, (ii) it restores adversarial examples created by more attacks, (iii) this method is simple because it uses a single network to restore adversarial images into clean images (iv) it gives better results on two datasets MNIST and cifar10 (v) our method can be used for different datasets and adversarial attacks by slightly changing or updating its layer structure.

Table 7: Transferability (%) of defence system on CIFAR10 dataset

Target Model	FGSM	BIM	PGD	DFA	CWA	SPA
VGG19	88.64	89.39	89.02	88.76	94.65	64.9
ResNet	89.29	87.57	85.58	89.95	94.58	65.87

Table 8: Comparisons of success rate (%) with the other adversarial defence techniques on the MNIST dataset

Attack	Defense-GAN	MagNet	Adv. Tr	cGan-defence	Our method
FGSM	98.8	67	78.5	98.7	**98.96**
BIM	-	-	-	-	**99.31**
PGD	-	-	-	-	**99.54**
DFA	-	-	-	-	**99.3**
CWA	98.9	38.2	28.3	96.5	**99.9**
SPA	-	-	-	-	**97.51**

Table 9: Comparisons of success rate (%) with the other adversarial defense techniques on the CIFAR10 dataset

Attacks	Defense-GAN	Mag-Net	Adversarial training	cGan-defence	Our method
FGSM	-	-	-	84.19	**84.95**
BIM	-	-	-	-	**87.38**
PGD	-	-	-	-	**85.56**
DFA	-	-	-	85.10	**87.62**
CWA	-	-	-	81.63	**94.66**
SPA	-	-	-	-	**81.31**

5 Discussions

In general, our proposed deep image restoration model, which is used as a defense method against adversarial attacks, gives promising results. It performs reasonably well on the MNIST dataset and achieving outstanding results on the MNIST dataset than the CIFAR10 dataset. This remarkable achievement is due to the complexity of the CIFAR10 dataset but attaining many convincing results.

The exact reasons for adversarial attacks are not yet confirmed because different researchers have given different reasons for attacks. However, the common thing is that all adversarial attacks decrease the performance of a model. In our experiments, we see that CWA and PGD attacks are the most robust. Attack's robustness means it needs small perturbation and has a significant negative effect on decreasing the accuracy by almost 0%. However, our method gives a high success rate against CWA and PGD attacks.

Our deep image restoration model works in two steps. First, get the low and high-level features and remove the adversarial perturbation by encoding the features layer by layer. Second, we restore the clean image without perturbation with the help of features that we get during the encoding part. Our approach is somewhat different from the traditional auto-encoder model because the auto-encoder loses low-level information. Therefore they cannot restore images that are corrupted by the adversarial attack.

Our proposed model performs two operations which are encoding and decoding as a single network. We do not use two networks like traditional auto-encoder and adversarial generative network (GAN) [30]. Furthermore, we do not use the max-pooling layer for the encoding process because it only extracts high-level information, reduces the dimension, and does not maintain low-level information, which is not helpful in decoding operation. It is also not a trainable layer. Our method only uses the convolution layer to extract low-level, high-level information. It has the benefit to decrease the dimension in the encoding process, and it is also trainable. Similarly, we also used the convolution-transpose layer. Instead of the up-sampling layer in the decoding process because the convolution-transpose layer works effectively in the decoding or restoration process due to its trainability nature and restores adversarial examples with perturbation free like the original image as demonstrated by our results on the MNIST and CIFAR10 datasets (Figs. 2 and 3).

6 Conclusions

In this research paper, we have proposed an easy defense method against adversarial attacks. Our defense method consists of an image restoration model responsible for removing adversarial noise from adversarial examples created due to different adversarial attacks. Our method improves the robustness of CNNs models. We have validated our defense method on MNIST and CIFAR10 datasets and prove that it gives promising results. We have also validated the transferability of the deep image restoration model on other models and restored the adversarial examples into clean examples created on these models due to adversarial attacks and restored successfully. After this, we have evaluated our method to other well-known defense methods and proved that our results are better than other techniques.

Funding Statement: This research study is funding by University Putra Malaysia, Salengor, 43400, Malaysia.

Conflicts of Interest: The authors claim no conflict of interest to report the present study.

References

[1] L. Yann, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] S. Lu, M. Tan and Z. Zhou, "A survey of practical adversarial example attacks," *Cyber Security*, vol. 1, no. 1, pp. 1–9, 2018.

[3] C. Anirban, M. Alam and V. Dey, "A survey on adversarial attacks and defences," in *CAAI Transactions on Intelligence Technology*, vol. 6, no. 1, pp. 25–45, 2021.

[4] A. Kurakin, I. Goodfellow and Y. Bengio, “Adversarial attacks and defences competition,” in *The NIPS’17 Competition: Building Intelligent Systems*, Springer, Cham, pp. 195–231, 2018.

[5] R. Kui, T. Zheng and Z. Qin, “Adversarial attacks and defenses in deep learning,” *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.

[6] A. Constantin, E. Poll and J. Visser, “Adversarial examples-a complete characterisation of the phenomenon,” in arXiv preprint arXiv, 1810.01185, 2018.

[7] R. R. Wiyatno, A. Xu, O. Dia, A. J. a. p. a. de Berker, “Adversarial examples in modern machine learning: A review,” in arXiv preprint arXiv, 1911.05268, 2019.

[8] Z. Jiliang and C. Li, “Adversarial examples: Opportunities and challenges,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2578–2593, 2019.

[9] N. Akhtar and A. Mian, “Threat of adversarial attacks on deep learning in computer vision: A survey,” *IEEE Access*, vol. 6, no. 1, pp. 14410–14430, 2018.

[10] C. Szegedy, W. Zaremba and D. Erhan, “Intriguing properties of neural networks,” in arXiv preprint arXiv 1312.6199, 2013.

[11] I. Goodfellow, J. Shlens and C. Szegedy, “Explaining and harnessing adversarial examples,” in ArXiv Preprint ArXiv, 1412.6572, 2014.

[12] A. Kurakin, I. Goodfellow and Y. Bengio, “Adversarial examples in the physical world,” in *Artificial Intelligence Safety and Security*, vol. 1, no. 3, pp. 99–112, 2018.

[13] A. Madry and A. Makelov and L. Schmidt, “Towards deep learning models resistant to adversarial attacks,” in arXiv preprint arXiv, 1706.06083, 2017.

[14] M. Dezfooli, S. M. Fawzi and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in *Proc. 2016 of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 2574–2582, 2016.

[15] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *Proc. of 2017 IEEE Symp. on Security and Privacy (sp)*, San Jose, CA, vol. 2, no. 6, pp. 39–57, 2017.

[16] S. Swetha, D. Mishra and S. Sai, “Scale and rotation corrected cnns (src-cnns) for scale and rotation invariant character recognition,” in *Proc. of the 11th Indian Conf. on Computer Vision, Graphics and Image Processing (ICVGIP 2018)*, Association for Computing Machinery, New York, NY, USA, Article 20, pp. 1–8, 2018.

[17] T. Florian, A. Kurakin and N. Papernot, “Ensemble adversarial training: Attacks and defenses,” in arXiv preprint arXiv:1705.0720, 2017.

[18] N. Papernot, P. McDaniel and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in *Proc. of 2016 IEEE Symposium on Security and Privacy (SP)*, California, USA, vol. 3, no. 1, pp. 582–597, 2016.

[19] D. Meng and H. Chen, “Magnet: A two-pronged defense against adversarial examples,” in *Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security*, Dallas, TX, USA, vol. 1, no. 5, pp. 135–147, 2017.

[20] P. Samangouei, M. Kabkab and R. Chellappa, “Defense-gan: Protecting classifiers against adversarial attacks using generative models,” in arXiv preprint arXiv, 1805.06605, 2018.

[21] C. Xie, J. Wang and Z. Zhang, “The defense of adversarial example with conditional generative adversarial networks,” *Security and Communication Networks*, vol. 1, no 3, pp. 140–152, 2020.

[22] D. Li, “The mnist database of handwritten digit images for machine learning research, “*IEEE Signal Processing Magazine*,” vol. 29, no. 6, pp. 141–142, 2012.

[23] R. Benjamin, R. Roelofs, L. Schmidt and V. Shankar, “Do cifar-10 classifiers generalise to cifar-10,” in ArXiv Preprint ArXiv:1806.00451, 2018.

[24] I. Forrest, M. Moskewicz and S. Karayev, “Densenet: Implementing efficient convnet descriptor pyramids,” in ArXiv Preprint ArXiv:1404.1869, 2014.

[25] Q. Zheng, Z. Zhang and X. Chen, “Fd-mobilenet: improved mobilenet with a fast downsampling strategy,” in *Proc. of 25th IEEE Int. Conf. on Image Processing (ICIP)*, Dallas, TX, USA, pp. 1363–1367, 2018.

[26] L. Yann, “Gradient-based learning applied to document recognition,”*IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[27] L. Yann, Y. Bengio and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[28] H. Kaiming, X. Zhang, S. Ren and J. Sun, “Identity mappings in deep residual networks,” in *Proc. of European Conf. on Computer Vision*, pp. 630–645, Springer, Cham, 2016.

[29] H. Kaiming, X. Zhang, S. Ren and J. Sun, “Developing deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proc. of the IEEE Int. Conf. on Computer Vision*, Santiago, Chile, pp. 1026–1034, 2015.

[30] I. Goodfellow, P. Jean, M. Mirza, B. Xu, D. Warde-Farley *et al.,* “Generative adversarial networks,” in arXiv preprint arXiv:1406.2661, 2014.