Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.028810
Article

Tech Science Press

Defending Adversarial Examples by a Clipped Residual U-Net Model

Kazim Ali[1,*], Adnan N. Qureshi[1], Muhammad Shahid Bhatti[2], Abid Sohail[2] and Mohammad Hijji[3]

[1]Department of Computer Science, Faculty of Information Technology, University of Central Punjab Lahore, 54000, Pakistan
[2]Department of Computer Science, COMSAT University Islamabad, Lahore Campus, Lahore, 54000, Pakistan
[3]Faculty of Computers and Information Technology, Computer Science Department, University of Tabuk, Tabuk, 47711, Saudi Arabia
*Corresponding Author: Kazim Ali. Email: kazimravian2003@gmail.com
Received: 18 February 2022; Accepted: 29 March 2022

Abstract: Deep learning-based systems have succeeded in many computer vision tasks. However, it is found that the latest study indicates that these systems are in danger in the presence of adversarial attacks. These attacks can quickly spoil deep learning models, e.g., different convolutional neural networks (CNNs), used in various computer vision tasks from image classification to object detection. The adversarial examples are carefully designed by injecting a slight perturbation into the clean images. The proposed CRU-Net defense model is inspired by state-of-the-art defense mechanisms such as MagNet defense, Generative Adversarial Network Defense, Deep Regret Analytic Generative Adversarial Networks Defense, Deep Denoising Sparse Autoencoder Defense, and Condtional Generattive Adversarial Network Defense. We have experimentally proved that our approach is better than previous defensive techniques. Our proposed CRU-Net model maps the adversarial image examples into clean images by eliminating the adversarial perturbation. The proposed defensive approach is based on residual and U-Net learning. Many experiments are done on the datasets MNIST and CIFAR10 to prove that our proposed CRU-Net defense model prevents adversarial example attacks in WhiteBox and BlackBox settings and improves the robustness of the deep learning algorithms especially in the computer vision field. We have also reported similarity (SSIM and PSNR) between the original and restored clean image examples by the proposed CRU-Net defense model.

Keywords: Adversarial examples; adversarial attacks; defense method; residual learning; u-net; cgan; cru-et model

1 Introduction

Deep learning (DL) models transform linear data patterns into nonlinear ones [1] and efficiently extract complex features from the data and information [2]. Deep learning solves many complex problems [3] that are almost impossible in the past [4] or challenging to solve in machine learning (ML) [5]. These days, the availability of massive data and high computational power [6] enable deep learning algorithms to make a considerable contribution in different fields of machine learning. For example, in vision systems [7],

linguistic communication process [8], edge computing [9], services computing [10] and spreads excellent applications of artificial intelligence (AI) in the world of real-life [11].

Deep learning algorithms have gained excellent success in their outstanding achievement, and different computing applications also raised questions on the interpretability of the deep learning field [12]. In such circumstances, we cannot reasonably explain the way of prediction by a DL algorithm. Therefore, AI applications based on DL-baed algorithms can face severe security threats [13]. These days, different research studies have proven that DL algorithms are not secured in the environment of membership logical thinking attack [14] and attribute logical thinking attack [15]. These days, the most dangerous threats facing deep learning algorithms are adversarial (negative) examples that Szegedy developed in 2014 [16]. The adversarial examples can easily be crafted by adding a small amount of adversarial perturbation into a clean example by fooling a deep learning model such as CNN in computer vision applications. This perturbation is non-noticeable for humans. The methods used to produce adversarial examples are called adversarial attacks. In the presence of these attacks, a high-performance DL algorithm gives incorrect prediction results and degrades the overall prediction of the algorithm [17]. Finally, the adversarial attacks decrease the robustness of deep learning applications such as image classification, face recognition, visual systems, self-driving vehicles, and many more [18].

When we carefully study the area of adversarial example attacks and defense methods, the following trends: The speed of developing various methods to create adversarial examples are increasing rapidly. It is found that there are two types of attacks from a classification point of view in computer vision, such as targeted attack and non-target attack. In a targeted adversarial attack, the attacker tries to misclassify a model to a specific class, e.g., enforce a model to classify a dog as a horse. These attacks are JSMA [19], EAD [20], and C&W [21]. In non-targeted Attacks, the aim of the attacker, a classifier to misclassify a class other than its actual class or ground truth class, e.g., classify a dog as any other class in the ground truth list. BIM [22], FGSM [23], PGD [24], and DeepFool [25] are some examples of non-targeted attacks.

Since the robustness of creating negative (adversarial) examples increases, detecting and defending these examples is challenging. The computation cost of building an adversarial attack also decreases due to the active area of research. The transferability property of negative examples makes it a significant challenge to defend adversarial examples. Transferability means the adversarial examples created for one model also fool another [26]. Reinforcement and recurrent learning models are also easily made fooled by adversarial examples. Therefore, these attacks are not limited to computer vision. The adversarial examples are also a serious security issue in text processing [27] and speech-related applications [28]. That is why developing a proper defense method against adversarial attacks is vital, but we are limited to defending adversarial examples in the computer vision field in this research study.

Usually, there are two types of defense approaches are reported in recent literature; First, the defense systems strengthen the neural network during the training process by adjusting learnable parameters (weights and biases) such as adversarial training [29] and distillation network defense technique [30]. Second, the defensive approaches to protect neural networks from removing adversarial perturbation or features from the adversarial examples and restore into clean examples. These approaches are used after training target neural networks and adversarial attacks. For example, LID [31], Defense-GAN [32], MagNet [33], ComDefend [34], DRAGAN-Defense [35], DDSA-Defense [36], and cGan-Defense [37].

However, each defense technique has some limitations. First, defensive distillation prevents attacks based on the gradient of the target model but fails against the CWA attack. Second, adversarial training is computational hard because it is needed to retrain the model for adversarial attacks again and again. Third, the adversarial perturbation is imperceptible, and it is challenging to differentiate adversarial examples from clean (original) examples. Hence, it is hard for defensive approaches to remove adversarial noise from adversarial examples to map into clean examples. Finally, we have proposed a novel defense framework to restore adversarial examples into clean examples by eliminating the adversarial noise.

This research work will propose a defense framework based on residual network (ResNet) and the U-Net model. We mix the structures and properties of ResNet and U-Net to develop a defense system named as CRU-Net (Clipped-Residual-U-Net) defense model. The proposed CRU-Net model maps the adversarial examples into clean examples (original images) by removing adversarial perturbation or noise and restoring the original input features. The uncluttered images are now correctly classified by the target image classifier. After a successful adversarial attack, we add our proposed CRU-Net model as a preprocessor block between the adversarial attack (attacker) and the target model. The overall working of our defense system is shown in Fig. 1. We describe our contribution as follows:

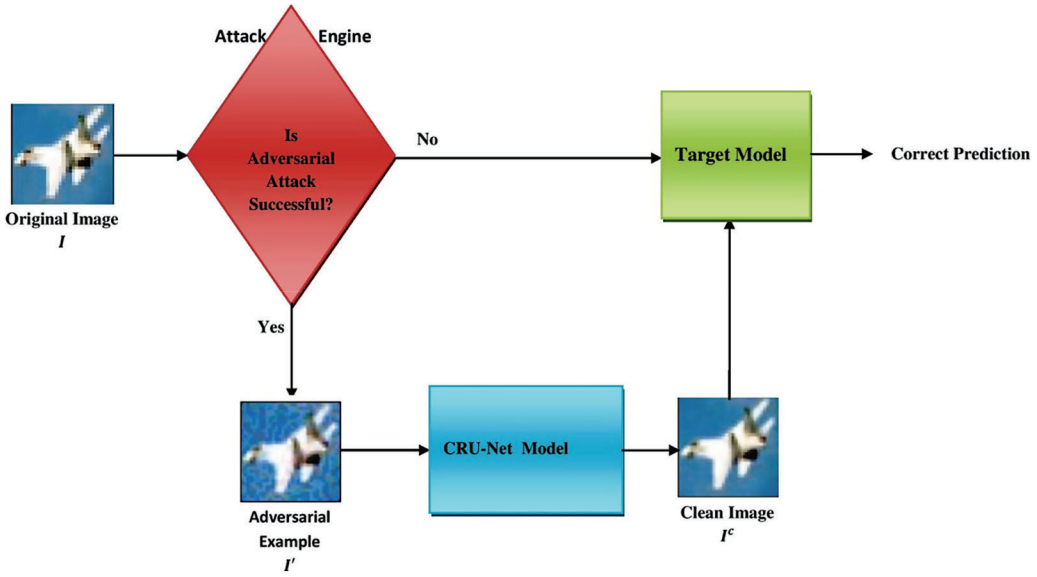

Figure 1: Represents the whole process of our proposed CRU-Net defense system against adversarial attack examples. For example, suppose the adversary has successfully attacked the target model and created an adversarial image example. In that case, the adversarial image examples are passed to our proposed CRU-Net Defense Model (substituted between the attacks and target model) to regenerate the adversarial examples into clean examples by removing adversarial features and feeding them to the target model for correct prediction

- Our primary goal is to develop a defense system responsible for removing adversarial features from adversarial examples before feeding the target model.
- Our proposed defense system is inspired by state-of-the-art methods such as MagNet-Defense, Defense-GAN, DRAGAN-Defense, DDSA-Defense, and cGan-Defense and improved their results. These defense methods remove adversarial noise from negative examples to protect target model.
- Our CRU-Net defense system is model-independent, which means there is no need to know about the structure of the target model. Therefore, the target model will remain independent, and there is no need to change its internal structure.

The remaining paper is organized as follows; Section 2 will describe the related or background study about the adversarial examples and defense frameworks. Section 3 will present the residual and U-Net learning theories because our proposed method is based on them. Section 4 will consist of our proposed defense method and its detail. Finally, Section 5 will present the experimental results, which we will drive during this research study, and Sections 6 and 7 will present the discussion and conclusion.

2 Related Works

This section will discuss some related work about adversarial attacks and defense methods.

2.1 Adversarial Attacks

In recent literature, there are two types of adversarial attacks are investigated, which are given as under:

- *White-box Attacks*: The attacks where the attacker has complete knowledge about the target model, like the internal structure of the target model.
- *Black-box Attacks*: The attacker does not know the model structure in these attacks. The adversary has only known the information of the output of the model.

FGSM (Fast Gradient Sign Method) [23] is a single step (need no iteration) white-box adversarial attack, developed by Goodfellow et al. in 2014. The following relation gives the mathematical form of FGSM:

$$I' = I + \varepsilon.sign(\nabla_I L(\theta,\ I,\ \ y) \tag{1}$$

where I is the original image, ε is the small constant which controls the magnitude of the adversarial perturbation, $sign(\nabla_I L(\theta,\ \ I,\ \ y)$ is the sign of the gradient of the loss of the target model w.r.t I, θ represents the parameters (weights and biases) of the target model, y is the actual label and $I^{'}$, shows the required adversarial example.

R-FGSM (Random Fast Gradient Sign Method) [38] is an advanced variant of FGSM [23], which decreases the robustness of the adversarial training (AT) defense technique [27] by adding some random noise in the input image. The following relation gives the R-FGSM:

$$I^{\wedge} = I + \varnothing.sign(X) \tag{2}$$

where $\varnothing$ represents a constant and X shows a vector taken from multivariate Gaussian Distribution and then applying Eq. (1) of FGSM as under:

$$I' = I^{\wedge} + (\epsilon - \varnothing)\nabla_{I^{\wedge}} L(I^{\wedge}\ \ y),\ \text{with}\ \varnothing < \ \varepsilon \tag{3}$$

PGD (Projected Gradient Descent) is proposed by Madry et al. [24]. In a PGD attack, the process of creating adversaries is taken as a bounded optimization problem and optimizing the following relation:

$$min_\theta\ \rho(\theta),\ with\ \rho(\theta) = E_{(I,\,y)} \sim\ D[max_{\delta\ \epsilon\ S} L_\theta(I + \delta,\ \ y)] \tag{4}$$

where E represents an objective function, and δ represents the adversarial noise.

DFA (Deep Fool Attack) [25] was launched by Moosavi-Dezfooli et al. as a repetition attack based on l_2 distance metric. The closest distance from the original input to the decision limit is determined in DFA. Decision limits distinguish the different classes on a hyperplane made by a classifier. Adversarial perturbation is created in a way that suppresses the negative pattern (adversarial example) outside the boundary, resulting in being classified as any other category or class.

CWA (Carlini and Wanger Attack) [21] has proposed producing negative examples surpassing many defense systems, especially the distillation defensive technique. CWA approach sees the target model as flexible and fulfills two conditions to create adversarial examples. (1) to decrease the distance (difference) between the adversarial sample and the original sample, (2) the adversarial sample should increase the error rate of the decision boundary of the target model. The authors have developed three types of attacks to minimize l_0, l_1, *and* l_2 distance metrics between original and adversarial examples.

SPA(Spatially Transformed Attack) [39] produces adversarial examples by transforming the original image around x and y coordinates in its frame of reference. This attack change location of the pixels

instead of changing the value. The SPA creates an adversarial example by optimizing the following relation, which is given by

$$L_{flow}(f) + \alpha.L_{adv}(x^t + t) \tag{5}$$

where α is the weight constant, $L_{adv}(x^t + t)$ is the prediction loss of the model, and $L_{flow}(.)$ is the variant loss required to produce adversarial perturbation for creating transformed adversarial examples.

2.2 Defense Method Against Adversarial Attacks

Now we will describe the defense systems against adversarial attacks such as Adversarial Training (AT) defense, MagNet-Defense, Defense-Gan, DRAGAN-Defense, DDSA-Defense, and cGAN-Defense.

AT (Adversarial Training) [29], the adversarial examples created due to different adversarial attacks are mixed in the original dataset and then retraining the classifier. In this way, the target model is trained on both actual and adversarial samples and predicts correct results on adversarial examples. The cost function of AT is given by Eq. (6):

$$\alpha J(x,\ y) + (1 - \alpha)L(x',\ y) \tag{6}$$

Mag-Net [33] is a security system against adversarial example attacks that enhances the robustness of a model by using two encoders. One is a detector, and the other is a re-constructor or reformer. The first auto-encoder is trained to detect the adversarial examples, and the second auto-encoder is trained for cleaning the adversarial perturbation from adversarial examples. The output of the Mag-Net system is a clean image without adversarial noise.

Defense-GAN [32] is similar to Mag-Net, but this defense technique uses a Generative Adversarial Network (GAN) instead of the traditional auto-encoder. Defense-GAN trains WGAN (Wasserstein's loss) on uninterrupted images and re-creates adversarial images before allowing the classifier to conflicting de-noising examples.

DRAGAN-Defense (Deep Regret Analytic Generative Adversarial Network) [35] is inspired by Defense-GAN [32] and claims to improve its results. DRAGAN is an updated version of GAN that offers fast and stable training and the chance to overcome mode collapse. Mode collapse occurs when the GAN produces the same output or small group of outputs repeatedly.

DDSA-Defense (Deep De-noising Sparse Auto-encoder) [36], in this technique, the authors first retrained the target model with adversarial examples produced by PGD [24] attack. They then used a pre-trained sparse auto-encoder as a preprocessing block to remove adversarial perturbation. Sparse means they apply a constraint on selecting features from the latent space of the encoder. Features with high-level pieces of information are selected, and others are skipped.

The conditional GAN-Defense (cGAN) [37] approach uses the power of a conditional GAN, unlike the old-fashioned GAN. This approach attempts to minimize adversarial features from negative examples and provides reconstructed images to the target identifier, aiming to restore the predicted accuracy of the target model.

3 Residual and U-Net Learning

This section introduces the residual learning (ResNet model) and U-Net learning used in the proposed CRU-Net defense model shown in Fig. 2.

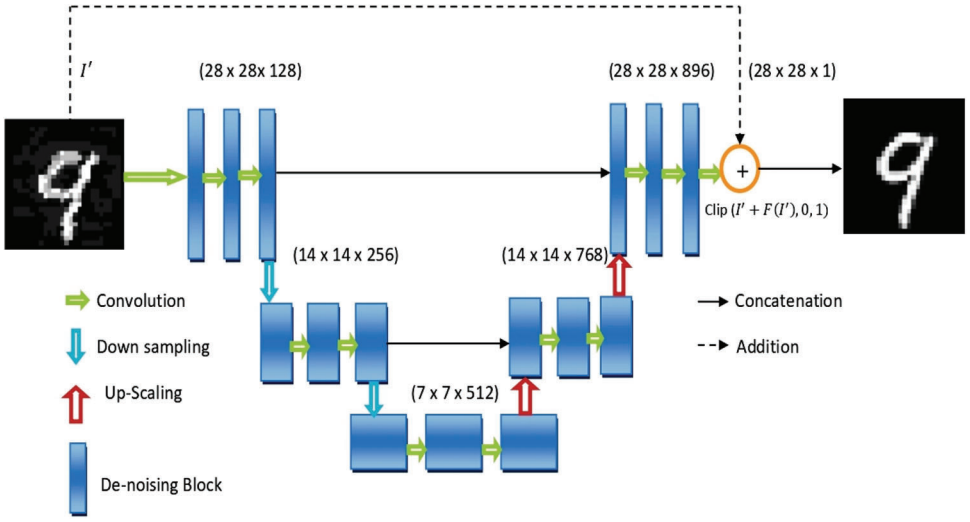

Figure 2: The structure of the CRU-Net model is to restore adversarial examples into clean examples. The restored adversarial example is cleaned from adversarial perturbation, and the target model gives the correct classification result on the restored clean example

The residual neural networks (ResNets) are proposed by He et al. [40] to deal with network damage as the depth of the network increases. The ResNet learning network developed by Microsoft can quickly solve the vanishing gradient problem. Residual learning reuse maps of each feature generated as inputs from subsequent combinations within the same block, and this model structure is an artistic image classification method. Therefore we will use the residual learning concept in our CRU-Net defense model, which will be used to remove adversarial perturbation from adversarial examples.

The U-Net [41] architecture was designed mainly for image segmentation, especially in medical imaging science. However, the U-Net model is now widely used in image encoding and decoding. The encoding phase contains several combinations of convolutional layers and many varieties of the max-pooling layer that reduce the size of the feature maps at each level while doubling the number of feature maps. The decoding section restores the size of the feature maps and keeps symmetric forms concerning the encoding section. In addition, the U-Net model enables the feature maps to concatenate simultaneously and reduces the loss of information during the encoding process.

In the proposed CRU-Net defense model, we use U-Net for the encoding and decoding process and finally make a clipped residual network block of input, and output obtained from the U-Net decoding process as shown in Fig. 2.

4 The Proposed CRU-Net Defense System

The proposed CRU-Net defense model is a defense system against adversarial example attacks. We have briefly described adversarial attacks in the related work Section 2.1. The process of the CRU-Net defense is to restore adversarial examples into clean examples to increase the robustness of the target or attacked models against adversarial attacks. It takes an adversarial image example as input and gives us a clean image example without adversarial noise, as shown in Fig. 2.

The proposed CRU-Net defense is based on U-Net and residual learning (ResNet model). Suppose that I is an original image and δ is the adversarial perturbation, then the adversarial example is created by the following Eq. (7)

$$I' = I + \delta \tag{7}$$

Then the restoring process of the adversarial example I' into the clean image is approximating the following parametric function, which is given as under:

$$\boldsymbol{I^r = F(I';\ \theta)} \tag{8}$$

where I^r is the restored image, and θ represents the training parameters of the proposed CRU-Net model. To minimize trainable parameters θ, we solve the following optimization problem:

$$\boldsymbol{\theta^* = \arg\, min_\theta \frac{1}{N} \sum_{i}^{N} L(F(I_i',\ \ \theta),\ \ I_i)} \tag{9}$$

where $(I_i',\ \ I_i)$ is the training set mapped from I_i' to I_i by the proposed CRU-Net. $L(\ .\ )$ is the loss function which is given by Eq. (10):

$$\boldsymbol{L = \frac{1}{N} \sum_{i}^{N} (I_i - I_i^r)^2} \tag{10}$$

The final output residual block of the CRU-Net model, which is our required cleaned example, is written as:

$$I^r = Clip(F(I';\ \ \theta) + I',\ \ \ 0,\ \ 1\ or\ 0,\ \ 255) \tag{11}$$

where *Clip*(.) is a function that controls the intensities of the restored clean examples from 0 to 1 or 0 to 255, and the model has consisted of three de-noising blocks at each level according to the size of the images, e.g., $28 \times 28 \times 1$ in our case in Fig. 2. These blocks show three consecutive convolution layers with the number of filters increased at each level by factor 2, the kernel size is 3×3, having the same padding, and stride is 1. Two blocks are used in the encoding and decoding phase at each level. The output of each block in the encoding phase is downscaled by 2 with the help of the convolution layer with stride two instead of max pooling. The number of feature maps is doubled on every downscaling to decrease the loss of information due to downscaling. The up-scaling is done using a transpose convolution layer in the decoding process. The shortcuts are managed in the encoding and decoding phases by using the concatenation of the same level. After upscaling and concatenating the same level blocks, a 3×3 convolution is performed to smooth and restore important information in the decoding process. We have used LeakyReLU() as an activation function in each layer of the CRU-Net model. This activation function is more flexible and gives good accuracy for our model than other activation functions such as ReLu(). The de-noising blocks are based on ResNet50 and reuse the feature maps.

In de-noising blocks, first, we perform a 3×3 convolution operation to decrease the feature maps by half, and after this, two 3×3 convolution operations are performed. In the end, a 3×3 convolution block combines all the feature maps of the first and last blocks. Then we reduce the output depth of the last concatenated blocks by $28 \times 28 \times 1$. Now, this block is used globally residual learning, which means to add the input and the output of the CRU-Net model. Finally, we clipped the residual function to regularize the intensities of our restored clean image from 0 to 1 or 0 to 255. Therefore we have named this proposed method as the Clipped Residul U-Net model (CRU-Net). The restoration loss of the CRU-net model during the restoration process of adversarial examples into clean examples is shown in Figs. 3–5.

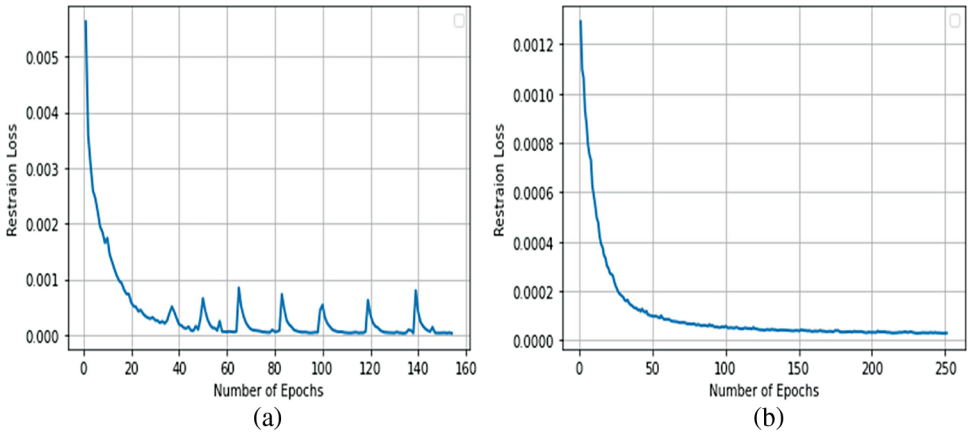

Figure 3: (a) The restoration loss of the CRU-Net model to restoring adversarial examples into clean examples was created due to the FGSM attack (b) the restoration loss of the CRU-net model to restoring adversarial examples into clean examples was created due to the R-FGSM attack

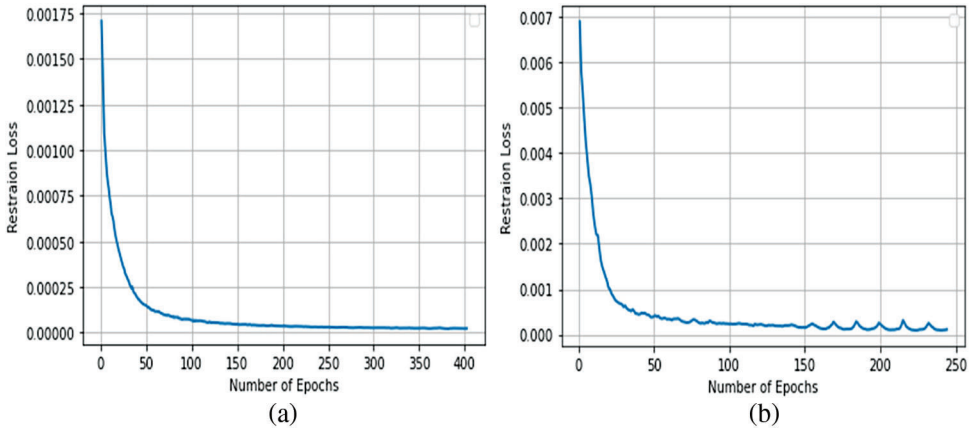

Figure 4: (a) The restoration loss of the CRU-Net model to restoring adversarial examples into clean examples was created due to the PGD attack (b) the restoration loss of the CRU-Net model to restoring adversarial examples into clean examples was created due to the DFA attack

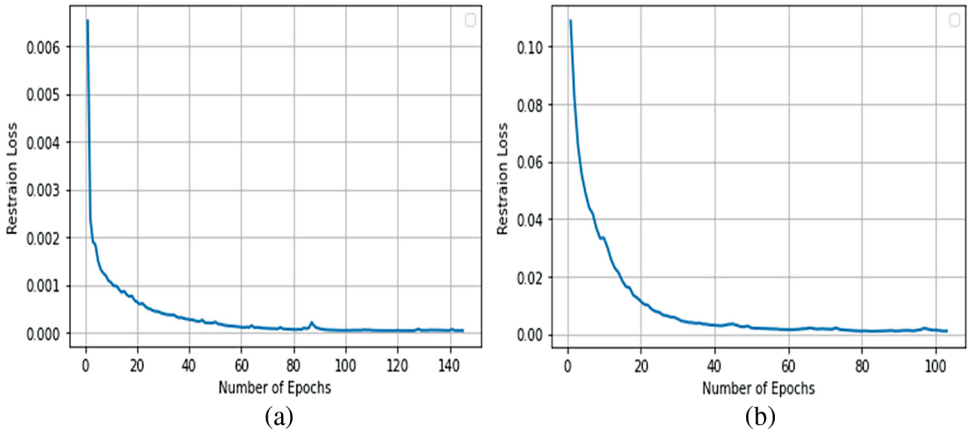

Figure 5: (a) The restoration loss of the CRU-Net model to restoring adversarial examples into clean examples was created due to the CWA attack (b) the restoration loss of the CRU-Net model to restoring adversarial examples into clean examples was created due to the SPA attack

The CRU-Net defense method is worked as a preprocessor block between the adversarial attack and target model. When the attacker successfully creates an adversarial example, it is fed into the model to restore the adversarial example into the clean example. The CRU-Net model is already trained to map adversarial examples into clean examples, and the structure of the model is shown in Fig. 2 for cleaning adversarial examples created from the MNIST dataset. Similarly, we develop this structure for different datasets of images of different sizes, e.g., CIFAR10. This model is beneficial when substituting between the adversary and the target model. The restoring loss is much slight during the training of the model on adversarial examples after each epoch, and the evidence of slight loss can be shown in Figs. 3–5 for different adversarial example attacks. The restoration loss is slight; therefore, the images generated by the model are almost identical to the original images shown in Figs. 6 and 7. So the target model has no issue predicting the generated clean images correctly.

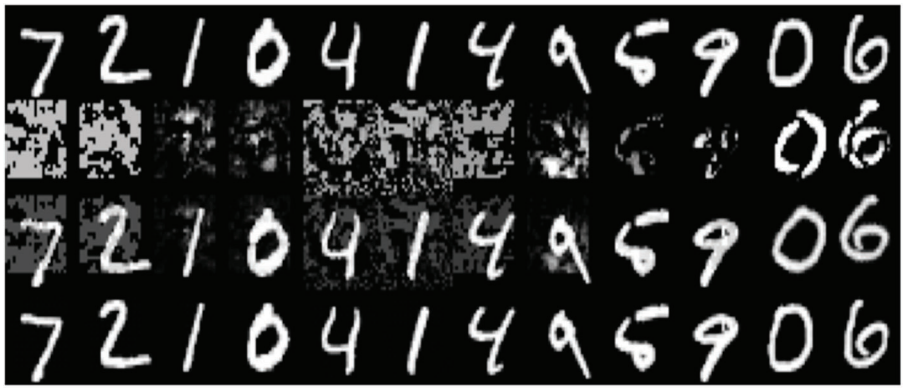

Figure 6: The first row shows the original images of the MNIST dataset. The second row represents the adversarial perturbations produced by the adversarial attacks (FGSM, R-FGSM, PGD, DFA, CWA, and SPA). The third row shows the adversarial examples, two for each attack. Finally, the fourth row demonstrates the restored clean examples generated by the proposed CRU-Net defense model

5 Experiments and Results

This section will evaluate how to defend our defense system against adversarial example attacks. Finally, we will use two datasets, MNIST and CIFAR10, for all our experiments.

The MNIST dataset contains 70000 handwritten digits, and we will use 60000 images as training set and 10000 images at test set. It is publicly available at http://yann.lecun.Com/exdb/mnist/. We have trained two models for the MNIST dataset; their structures and names (A-net and B-net) are the same as described in cGAN-Defense [37], which is one of our baseline techniques. The structures of A-net and B-net are described in Tab. 1. The accuracies of models A-net and B-net are 99.35% and 99.39%, respectively. The CIFAR10 contains 60000 RGB images of 10 categories, where we will use 50000 images for training purposes and 10000 images for testing purposes. It can be publicly found at https://www.cs.toronto.edu/~kriz/cifar.html. In addition, we have used pre-trained DenseNet and MobileNet models for the CIFAR10 dataset named D-net and M-net. The models D-net and M-net achieve an accuracy of 83.3% and 82.3% on the CIFAR10 dataset, respectively.

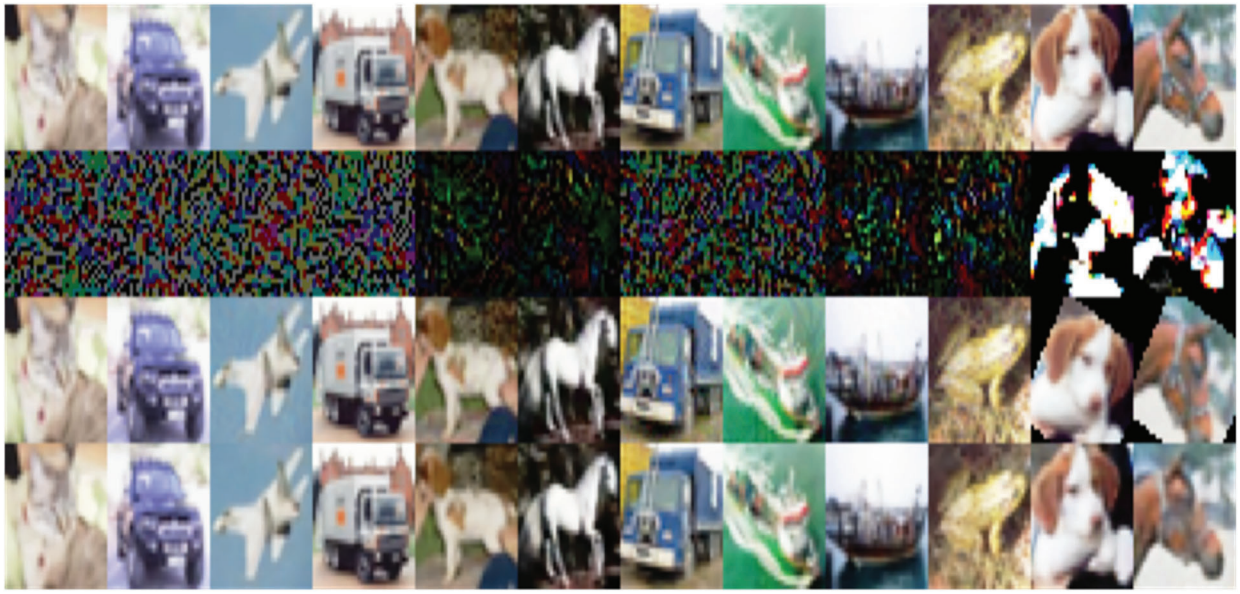

Figure 7: The first row shows the original images of the CIFAR10 dataset. The second row represents the adversarial perturbations produced by the adversarial attacks (FGSM, R-FGSM, PGD, DFA, CWA, and SPA). The third row shows the adversarial examples, two for each attack. Finally, the fourth row demonstrates the restored clean examples generated by the proposed CRU-Net model

Table 1: The structures of the target model A-net and B-net for MNIST Dataset

A-net	B-net
Conv(64, 5 × 5, 1) + ReLu	Conv(64, 3 × 3, 1) + ReLu
Conv(64, 3 × 3, 2) + ReLu	Conv(64, 3 × 3, 1) + ReLu
DropOut(0.5)	MaxPoling2D()
FC(10) + Relu	Conv(64, 3 × 3, 1) + ReLu
Softmax	Conv(64, 3 × 3, 1) + ReLu
	MaxPoling2D()
	FC(200) + ReLu
	DropOut(0.5)
	FC(200) + ReLu
	DropOut(0.5)
	FC(10) + Relu

5.1 Performance Metrics for CRU-Net Defense

The performance of the proposed CRU-Net defense model is evaluated through the following metrics.

$$\boldsymbol{accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN}} \tag{12}$$

$$\textbf{\textit{restored\_acc}} = \frac{\textbf{\textit{correctly classified restored images}}}{\textbf{\textit{a total number of images}}} \tag{13}$$

$$\textbf{\textit{success\_rate}} = \frac{\textbf{\textit{restored\_acc}}}{\textbf{\textit{accuracy}}} \tag{14}$$

The Eq. (11) is used to determine the accuracy of A-net and B-net for MNIST, similarly D-Net and M-Net for CIFAR10 datasets, respectively. Eq. (12) evaluates the accuracy of the target models (A-net, B-net, D-net, M-net) on restored adversarial examples by the proposed CRU-Net model. Eq. (13) shows the success rate of CRU-Net, which is used when we compare our CRU-Net results with other state-of-the-art defense systems in comparison Section 5.5.

We have measured similarity between the original image and clean image examples by the CRU-Net model by using Eqs. (14) and (15):

$$\textbf{\textit{SSIM}}(\boldsymbol{I},\ \boldsymbol{I^{res}}) = \frac{(2\mu_I\mu_{I^{res}} + \boldsymbol{C_1})(2\sigma_I\sigma_{I^{res}} + \boldsymbol{C_2})}{(\mu_I^2 + \mu_{I^{res}}^2 + \boldsymbol{C_1})(\sigma_I^2 + \sigma_{I^{res}}^2 + \boldsymbol{C_2})} \tag{15}$$

where μ_I and $\mu_{I^{res}}$ compare the luminance, σ_I^2 and $\sigma_{I^{res}}^2$ measures the contrast, and $\frac{\sigma_I\sigma_{I^{res}}}{\sigma_I^2 + \sigma_{I^{res}}^2}$ shows the structural similarity of images I and I^{res}, respectively.

$$PSNR = 10\ log_{10}\frac{(L-1)^2}{MSE} \tag{16}$$

$$Where\ MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I_{(i,j)} - I_{(i,j)}^{res})^2$$

Here L is the maximum number of intensity levels in an image (in the case of image number of intensity levels are 256, ranging from $0-255$), m and n represent the number of rows and columns in image matrix respectively, $I_{(i,\ j)}$ and $I_{(i,j)}^{res}$ are the corresponding intensity value of the original and restored clean image by our proposed defense methods.

5.2 Implementation Details

We have used fool-box [42], a library to check the robustness of a model. We have developed six types of adversarial attacks using the fool-box library, e.g., FGSM, R-FGSM, PGD, DFA, CWA, and SPA. We have used different values of ε for developing above various adversarial attacks. We have developed our proposed CRU-Net model for restoring original images with the help of the TensorFlow, Keras, and NumPy libraries.

5.3 Results of CRU-Net Defense Model in White-Box and Black-Box Setting

As discussed previously, an attacker fully knows the target model in the white-box attack setting. To confirm the effectiveness of the CRU-Net model in the white-box attack setting, we have validated results on two datasets, MNIST and CIFAR10. We have trained two CRU-Net models for each dataset to restore adversarial examples. In addition, we are made adversarial examples from the test set data. We send adversarial examples to a pre-trained CRU-Net model to generate clean examples in our defense system. Then we feed the restored clean examples to the target model for classification, as shown in Fig. 1. Some visual results or restored clean examples by the CRU-Net defense model are shown in Figs. 6 and 7.

The results of the CRU-Net defense model are shown in Tab. 2 and Fig. 8 on the MNIST dataset. Tab. 3 and Fig. 9 represent the results of the CRU-Net defense model on the dataset CIFAR10. In the Black-Box setting, the attacker does not know about the structure of the target model. The results of the CRU-Net defense model in the Black-Box settings are shown in Tabs. 4 and 5, Figs. 10 and 11.

Table 2: Our Proposed Defense Framework CRU-NET results on the MNIST dataset in a WHITE-BOX setting

Attacks	Target models	No attack	No defense	CRU-net defense
FGSM	A-Net	0.9935	0.2244	0.9935
	B-Net	0.9939	0.1844	0.9939
R-FGSM	A-Net	0.9935	0.1315	0.9937
	B-Net	0.9939	0.1011	0.9940
PGD	A-Net	0.9935	0.1264	0.9933
	B-Net	0.9939	0.1378	0.9937
DFA	A-Net	0.9935	0.4297	0.9934
	B-Net	0.9939	0.4097	0.9938
CWA	A-Net	0.9935	0.5050	0.9930
	B-Net	0.9939	0.5050	0.9935
SPA	A-Net	0.9935	0.0512	0.9931
	B-Net	0.9939	0.0732	0.9939

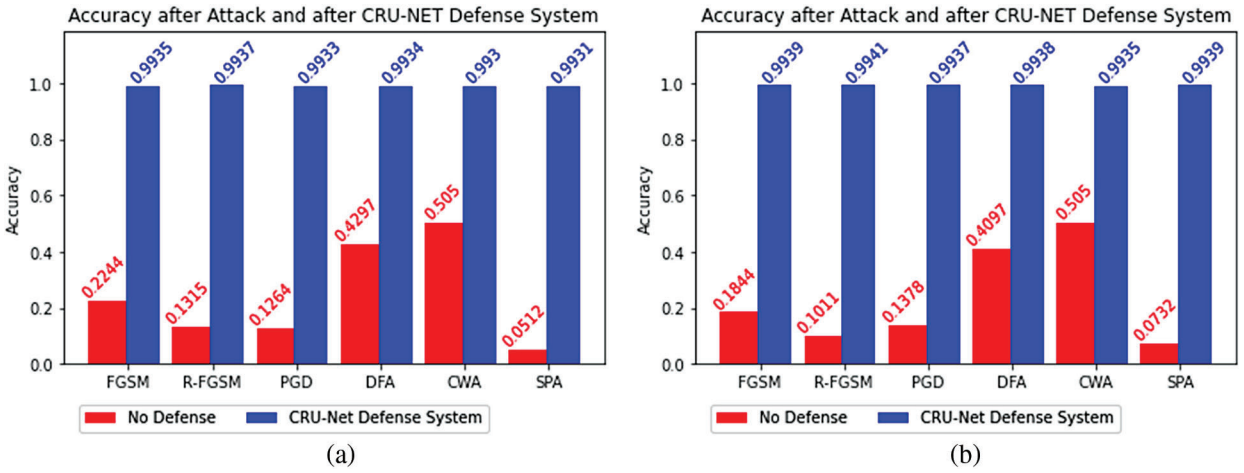

Figure 8: (a) The degraded adversarial accuracy (red bar) due to adversarial attacks and restored accuracy (blue bar) of the target model **A-net** after applying CRU-Net Defense model (b) The degraded adversarial accuracy (red bar) due to adversarial attacks and restored accuracy (blue bar) of the target model **B-net** after applying CRU-Net Defense model

Tabs. 4 and 5 describe the performance of CRU-Net in the BlackBox attacks setting. It is also called the transferability property of a defense system. A-net/B-net means that the adversarial examples are generated from the target model A-net and restored by using CRU-Net trained for B-net.

Table 3: Our proposed defense framework CRU-NET results on the MNIST dataset in a white-box setting

Attacks	Target models	No attack	No defense	CRU-net defense
FGSM	D-Net	0.8330	0.1370	0.8310
	M-Net	0.8230	0.1820	0.8170
R-FGSM	D-Net	0.8330	0.1520	0.8360
	M-Net	0.8230	0.3360	0.7970
PGD	D-Net	0.8330	0.1260	0.7990
	M-Net	0.8230	0.4200	0.7990
DFA	D-Net	0.8330	0.2610	0.8340
	M-Net	0.8230	0.2310	0.8100
CWA	D-Net	0.8330	0.5470	0.8100
	M-Net	0.8230	0.6500	0.7880
SPA	D-Net	0.8330	0.0720	0.8230
	M-Net	0.8230	0.0540	0.8040

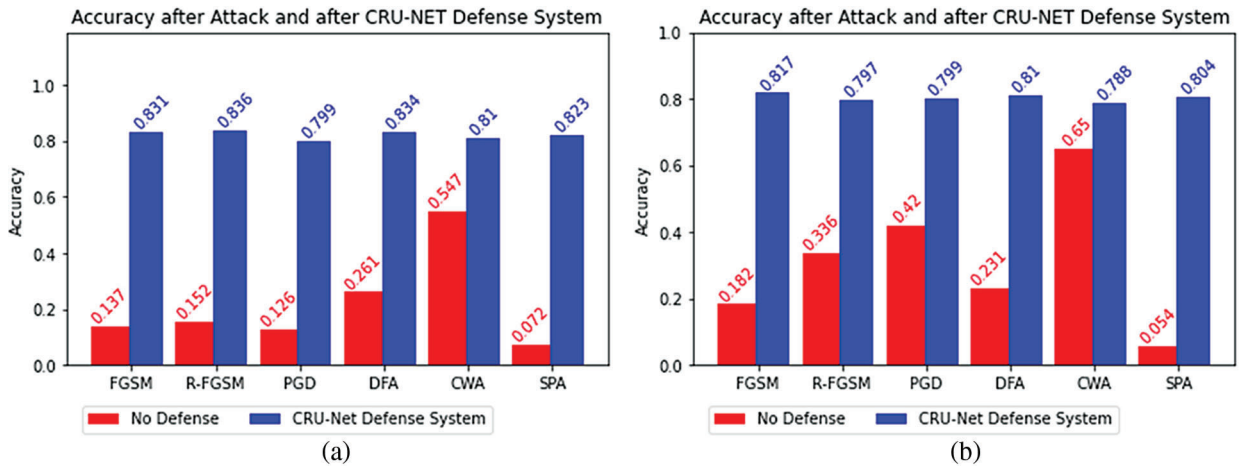

Figure 9: (a) The degraded adversarial accuracy (red bar) due to adversarial attacks and restored accuracy (blue bar) of the target model **D-net** after applying CRU-Net Defense model (b) The degraded adversarial accuracy (red bar) due to adversarial attacks and restored accuracy (blue bar) of the target model **M-net** after applying CRU-Net Defense model

5.4 Similarity Between the Original Image Examples and Restored Image Examples by the CRU-Net Defense Model

The primary purpose of our proposed CRU-Net defense method is to restore adversarial examples into clean examples. Therefore, the similarity (SSIM and PSNR) between clean examples restored by our method and the original examples are described in Tabs. 6 and 7 for target models A-Net, B-Net on the MNIST, and D-Net, M-Net on the CIFAR10 dataset, respectively. We have used two similarity metrics, SSIM and PSNR, to check whether the restored adversarial examples are near to the original examples or not.

Table 4: Our proposed defense framework CRU-NET results on the dataset MNIST in the black-box setting

Attacks	Target models	No attack	No defense	CRU-net defense
FGSM	A-net/B-net	0.9935	0.2244	0.9866
	B-net/A-net	0.9939	0.1844	0.9890
R-FGSM	A-net/B-net	0.9935	0.1315	0.9911
	B-net/A-net	0.9939	0.1011	0.9923
PGD	A-net/B-net	0.9935	0.1264	0.9799
	B-net/A-net	0.9939	0.1378	0.9811
DFA	A-net/B-net	0.9935	0.4297	0.9922
	B-net/A-net	0.9939	0.4097	0.9925
CWA	A-net/B-net	0.9935	0.5050	0.9821
	B-net/A-net	0.9939	0.5050	0.9830
SPA	A-net/B-net	0.9935	0.0512	0.9913
	B-net/A-net	0.9939	0.0732	0.9915

Table 5: Our proposed defense framework CRU-NET results on the dataset CIFAR10 in the black-box setting

Attacks	Target models	No attacks	No defense	CRU-net defense
FGSM	D-net/M-net	0.8330	0.1370	0.8230
	M-net/D-net	0.8230	0.1820	0.7857
R-FGSM	D-net/M-net	0.8330	0.1520	0.8278
	M-net/D-net	0.8230	0.3360	0.7767
PGD	D-net/M-net	0.8330	0.1260	0.7893
	M-net/D-net	0.8230	0.4200	0.7854
DFA	D-net/M-net	0.8330	0.2610	0.8921
	M-net/D-net	0.8230	0.2310	0.7953
CWA	D-net/M-net	0.8330	0.5470	0.7999
	M-net/D-net	0.8230	0.6500	0.7689
SPA	D-net/M-net	0.8330	0.0720	0.8045
	M-net/D-net	0.8230	0.0540	0.7989

5.5 Comparison of Proposed CRU-Net Defense with Other State-of-the-Art Defense System

We have compared our CRU-Net with the start-of-the-art defense systems such as Adversarial Training (AT) defense technique, MagNet-Defense, Defense-Gan, DRAGAN-Defense, DDSA-Defense, and cGAN-Defense. We have used four models named A, B, C, and D. The structures of these models were first described in Defense-GAN and used for experiments for both white-box and black-box attacks set. The structure of models A, B, D, and C are given in Tab. 8. We will use the success rate as a comparison metric with the above defense systems, presented in Eq. (12). The reason for using success rate metrics is that the accuracy of these models in our setting is different from the other defense techniques. Therefore

the accuracy metric for comparison is not suitable here. Therefore, the comparison results for the MNIST and CIFAR10 datasets are shown in Tabs. 9 and 10.

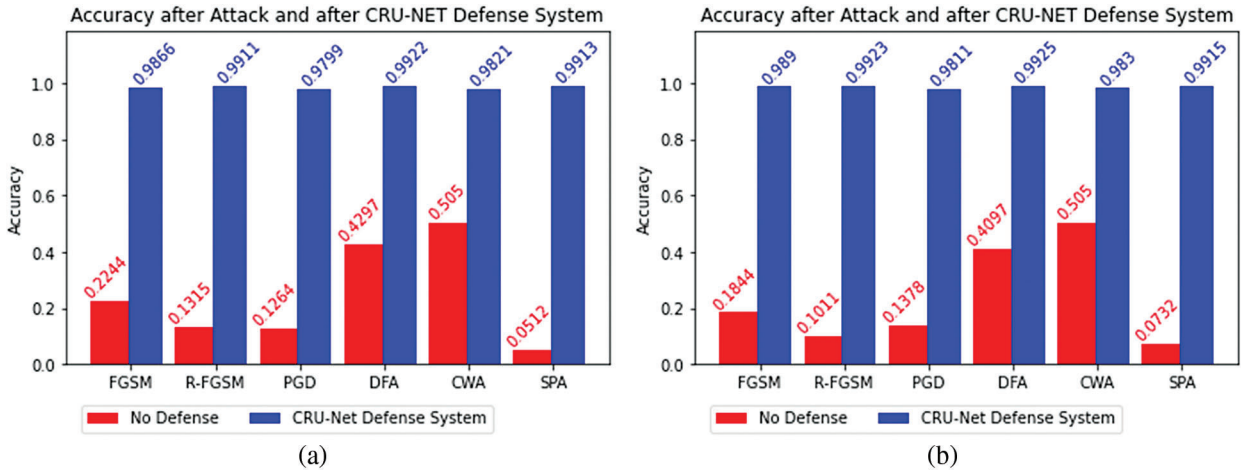

Figure 10: (a) The degraded adversarial accuracy (red bar) due to adversarial attacks and restored accuracy (blue bar) of the target model **A-net/B-net** after applying CRU-Net Defense model (b) The degraded adversarial accuracy (red bar) due to adversarial attacks and restored accuracy (blue bar) of the target model **B-net/A-net** after applying CRU-Net Defense model

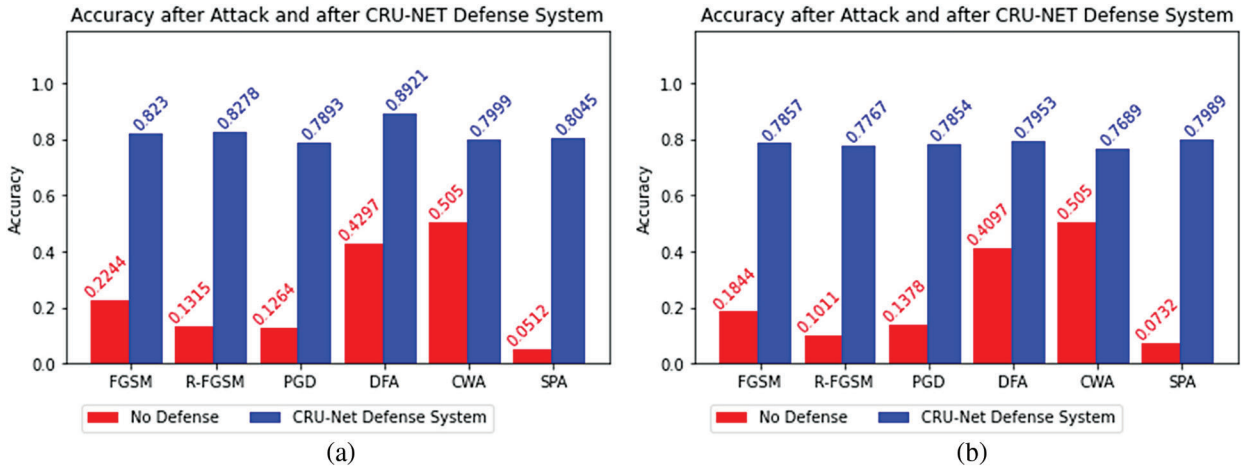

Figure 11: (a) The adversarial (red bar) and restored accuracy (blue bar) of the target model **D-net/M-net** after CRU-Net Defense model, (b) The adversarial (red bar) and restored accuracy (blue bar) of the target model **M-net/D-net** after CRU-Net Defense model

The results of state-of-the-art defense approaches MagNet, Defense-GAN, DRAGAN-Defense, DDSA-Defense, and cGAN-Defense are taken from the original research sources.

Table 6: The SSIM and PSNR similarity between original and restored examples for tested models A-Net, B-Net on the MNIST dataset

Target model	Average SSIM & PSNR	Similarity between restored FGSM & original examples	Similarity between restored R-FGSM & original examples	Similarity between restored PGD & original examples	Similarity between restored DFA & original examples	Similarity between restored CWA & original examples	Similarity between restored S. A. & original examples
A-Net	SSIM	0.9996	0.9998	0.9997	0.9998	0.9995	0.9996
	PSNR	94.31	97.98	95.67	95.89	94.89	94.31
B-Net	SSIM	0.9995	0.9996	0.9996	0.9998	0.9990	0.9997
	PSNR	94.31	97.99	95.50	95.91	94.85	94.35

Table 7: The SSIM and PSNR similarity between original and restored examples for tested models D-Net, and M-Net on the CIFAR10 dataset

Target model	Average SSIM & PSNR	Similarity between restored FGSM & original examples	Similarity between restored R-FGSM & original examples	Similarity between restored PGD & original examples	Similarity between restored DFA & original examples	Similarity between restored CWA & original examples	Similarity between restored S. A. & original examples
D-Net	SSIM	0.9996	0.9998	0.9752	0.9631	0.947	0.967
	PSNR	101.11	105.13	82.48	81.91	80.18	80.59
M-Net	SSIM	0.9996	0.9999	0.9667	0.9612	0.9476	0.970
	PSNR	101.12	105.32	81.34	81.89	80.20	81.67

Table 8: The Structures of models A, B, C, and D, are for comparison purposes

Model A	Model B	Model C	Model D
Conv(64, 5 × 5, 1)	Dropout (0.2)	Conv(128, 3 × 3, 1)	FC (200)
ReLU	Conv(64, 8 × 8, 2)	ReLU	ReLU
Conv(64, 5 × 5, 2)	ReLU	Conv(64, 3 × 3, 2)	Dropout (0. 5)
ReLU	Conv(128, 6 × 6, 2)	ReLU	FC (200)
DropOut(0.25)	ReLU	Dropout(0.25)	ReLU
FC(128)	Conv(128, 5 × 5, 1)	FC (128)	Dropout (0.5)
DropOut(0.5)	ReLU	ReLU	FC (10)
FC (10)	Dropout (0.5)	Dropout (0.5)	Softmax
Softmax	FC (10)	FC (10)	
	Softmax	Softmax	

Table 9: Comparison of the success rate of proposed CRU-Net Defense System with other Defense Systems for the MNIST dataset. Dash (-) means the corresponding Defense System does not deal with or defend that specific attack

Adversarial attacks	Target models	AT defense success_rate	MagNet defense success_rate	Defense-GAN success_rate	DRAGAN-Defense success_rate	DDSA-Defense success_rate	cGAN-Defense success_rate	Proposed CRU-Net success_rate
FGSM	A	0.6529	0.1915	0.9909	-	0.9120	0.9889	1.0000
	B	0.0600	0.0852	0.9937	-	-	0.9958	1.0000
	C	0.7891	0.1636	0.9929	-	-	0.9906	1.0000
	D	0.7379	0.0947	0.9879	-	-	0.9909	1.0000
R-FGSM	A	-	-	0.9614	-	0.8989	-	1.0000
	B	-	-	0.9537	-	-	-	1.0000
	C	-	-	0.9726	-	-	-	1.0000
	D	-	-	0.9671	-	-	-	1.0000
PGD	A	-	0.9940	-	-	0.9211	-	1.0000
	B	-	-	-	-	-	-	0.9999
	C	-	-	-	-	-	-	1.0000
	D	-	-	-	-	-	-	0.9998
DFA	A	-	-	-	-	-	-	1.0000
	B	-	-	-	-	-	-	1.0000
	C	-	-	-	-	-	-	1.0000
	D	-	-	-	-	-	-	1.0008
CWA	A	0.0772	0.0381	0.9919	0.9967	0.8503	0.9679	1.0000
	B	0.2910	0.0353	0.9521	0.9941	-	0.9604	1.0000
	C	0.0311	0.0251	0.9929	0.9856	-	0.9718	1.0000
	D	0.0100	0.0211	0.9909	0.9846	-	0.9738	0.9990
SPA	A	-	-	-	-	-	-	1.0000
	B	-	-	-	-	-	-	1.0000
	C	-	-	-	-	-	-	0.9990
	D	-	-	-	-	-	-	1.0000

Table 10: Comparison of the success rate of the proposed CRU-Net Defense System with other Defense Systems for the CIFAR10 dataset. Dash (-) means the corresponding Defense System does not deal with or defend that specific attack

Adversarial attacks	AT defense success_rate	MagNet defense success_rate	Defense-GAN success_rate	DRAGAN-defense success_rate	DDSA-defense success_rate	cGAN-defense success_rate	Proposed CRU-Net success_rate
FGSM		**0.9990**	-	-	0.7494	0.8858	0.9976
R-FGSM	-	-	-	-	0.6671	-	1.0036
PGD	-	-	-	-	0.7506	-	0.9592
DFA	-	0.9340	-	-		0.8954	1.0012
CWA	-	0.9370	-	-	0.4508	0.8589	0.9724
SPA	-	-	-	-	-	-	0.9880

6 Discussion

In general, our proposed CRU-Net model, which is used as a defense method against adversarial examples, gives excellent results. It performs reasonably well on the MNIST and CIFAR10 datasets and achieves outstanding results than other defense techniques, shown in Tabs. 9 and 10. The exact reasons for negative attacks are not yet confirmed because different researchers have given various reasons. However, the common thing is that all adversarial attacks decrease the performance of a model. Our experiments show that CWA and PGD attacks are the most robust. The robustness of the attack means it needs small perturbation and has a significant negative effect on decreasing the accuracy of the target model. However, our method gives a high success rate against CWA and PGD attacks.

Our CRU-Net model works based on the structures of two popular learning algorithms, Residual and U-Net models. First, get the low and high-level features and remove the adversarial perturbation by de-noising blocks in the encoding part of the U-Net model. Second, we upscaled and restored the clean image with the help of features that we got during the decoding part of the U-net Model. In the end, we add the input layer and output layer to merge the attributes by using residual learning (ResNet). This way, we get the output as a clear example that the target model correctly predicts. Finally, it restores adversarial examples with adversarial perturbation-free like the original image, as demonstrated by our results on the datasets MNIST and CIFAR10, shown in Figs. 6 and 7, respectively. Finally, our proposed defense method is limited to removing adversarial perturbation produced by the six types of adversarial attacks such as FGSM, R-FGSM, PGD, DFA, CWA, and SA. The results of the proposed defense methods have been presented in Tabs. 2–5.

7 Conclusions

This study proposed a defensive technique to prevent the adversarial example attack named the CRU-Net defense model. The structure of the proposed CRU-Net model is inspired by the famous ResNet and U-Net learning, as shown in Fig. 2. The CRU-Net restores adversarial examples into clean examples, almost like original examples or images for the correct prediction of the target models. The proposed CRU-Net defense central is the sequel of well-known state-of-the-art methods such as Magnet-Defense, Defense-GAN, DRAGAN-Defense, DDSA-Defense, and cGan-Defense. In addition, we tested the CRU-Net defense model on the adversarial samples produced by the many adversarial attacks such as FGSM, R-FGSM, PGD, DFA, CWA, and SPA from the MNIST CIFAR10 datasets. In addition, we have shown excellent results on selected datasets and target models. Finally, our CRU-Net defense system gives a high success rate to restore adversarial examples into clean examples than other latest and state-of-the-art defensive approaches.

Acknowledgement: We shall be very thankful to Dr. Adnan N. Quereshi for his kind supervision throughout this study.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

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

[2] B. Wu, Z. Chen, J. Wang and H. J. N. Wu, "Exponential discriminative metric embedding in deep learning," *Neurocomputing*, vol. 290, pp. 108–120, 2018.

[3] M. M. Zhang, K. Shang and H. J. N. Wu, "Learning deep discriminative face features by customized weighted constraint," *Neurocomputing*, vol. 332, pp. 71–79, 2019.

[4] C. Liu and H. J. S. P. Wu, "Channel pruning based on mean gradient for accelerating convolutional neural networks," *Signal Process*, vol. 156, pp. 84–91, 2019.

[5] X. Li and H. J. I. W. C. L. Wu, "Spatio-temporal representation with deep neural recurrent network in MIMO CSI feedback," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 653–657, 2020.

[6] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan *et al.,* "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2019.

[7] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi *et al.,* "BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2019.

[8] Y. Zhang, C. Yin, Q. Wu, Q. He, H. J. I. T. O. S. Zhu *et al.,* "Location-aware deep collaborative filtering for service recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3796–3807, 2019.

[9] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang *et al.,* "Trust-oriented IoT service placement for smart cities in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2019.

[10] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang *et al.,* "Efficient query of quality correlation for service composition," *IEEE Transactions on Services Computing*, vol. 14, pp. 695–709, 2018.

[11] Y. Zhang, W. Kaibin, H. Qiang, C. Feifei, D. Shuiguang *et al.,* "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, vol. 14, no. 5, pp. 1333–1344, 2019.

[12] Q. S. Zhang, S. C. J. F. O. I. T. Zhu and E. Engineering, "Visual interpretability for deep learning: A survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 27–39, 2018.

[13] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEE Symp. on Security and Privacy (SP)*, CA, pp. 3–18, 2017.

[14] S. Yeom, I. Giacomelli, M. Fredrikson and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *IEE 31st Computer Security Foundations Symp. (CSF)*, Oxford, UK, pp. 268–282, 2018.

[15] M. Fredrikson, S. Jha and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. of the 22nd ACM Sigsac Conf. on Computer and Communications Security*, Denver, USA, pp. 1322–1333, 2015.

[16] C. Szegedy, Z. Wojciech, S. Ilya, B. Joan, E. Dumitru *et al.,* "Intriguing properties of neural networks," arXiv preprint arXiv:1312.6199, 2013.

[17] X. Yuan, P. He, Q. Zhu, X. J. I. Li and l. systems, "Adversarial examples: Attacks and defenses for deep learning," *IEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[18] M. Sharif, S. Bhagavatula, L. Bauer and M. K. J. Reiter, "Adversarial generative nets: Neural network attacks on state-of-the-art face recognition," arXiv preprint arXiv:1801.00349, vol. 2, no. 3, 2017.

[19] R. Wiyatno and A. J. Xu, "Maximal jacobian-based saliency map attack," arXiv preprint arXiv:1808.07945, 2018.

[20] P. Y. Chen, Y. Sharma, H. Zhang, J. Yi and C. J. Hsieh, "Ead: Elastic-net attacks to deep neural networks via adversarial examples," in *Proc. of the AAAI Conf. on Artificial Intelligence*, California, USA, vol. 32, no. 1,. 2018.

[21] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symp. on Security and Privacy (SP)*, San Jose, USA, pp. 39–57, 2017.

[22] A. Kurakin, I. Goodfellow and S. Bengio, "Adversarial examples in the physical world," ed, 2016.

[23] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv:1412.6572, vol. abs/1412.6572, 2015.

[24] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. J. A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1312.6199, vol. abs/1706.06083, 2018.

[25] S. M. Moosavi-Dezfooli, A. Fawzi, P. J. I. Frossard and P. Recognition, "DeepFool: A simple and accurate method to fool deep neural networks," in *IEEE Conf. on Computer Vision Pattern Recognition*, Las Vegas, NV, USA, pp. 2574–2582, 2016.

[26] F. Tramèr, N. Papernot, I. J. Goodfellow, D. Boneh and P. J. A. Mcdaniel, "The space of transferable adversarial examples," arXiv preprint arXiv:1312.6199, vol. abs/1704.03453, 2017.

[27] M. Alzantot, Y. Sharma, A. Elgohary, B. J. Ho, M. B. Srivastava *et al.,* "Generating natural language adversarial examples," arXiv:1804.07998, 2018.

[28] Y. Qin, N. Carlini, I. J. Goodfellow, G. Cottrell and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," arXiv:1903.10346, 2019.

[29] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh and P. J. A. Mcdaniel, "Ensemble adversarial training: Attacks and defenses," arXiv preprint arXiv:1312.6199, vol. abs/1705.07204, 2018.

[30] N. Papernot, P. Mcdaniel, X. Wu, S. Jha and A. J. I. S. Swami, "Distillation as a defense to adversarial erturbations against deep neural networks," in *IEEE Symp. on Security Privacy*, Florida, USA, pp. 582–597, 2016.

[31] X. Ma, Y. Wang, S. Erfani, M. Wijewickrema, S. Schoenebeck *et al.,* "Characterizing adversarial subspaces using local lntrinsic dimensionality," arXiv preprint arXiv:1312.6199, vol. abs/1801.02613, 2018.

[32] P. Samangouei, M. Kabkab and R. J. A. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," arXiv preprint arXiv:1312.6199, vol. abs/1805.06605, 2018.

[33] D. Meng, H. J. P. Chen and C. Security, "MagNet: A two-pronged defense against adversarial examples," in *Proc. of the ACM SIGSAC Conf. on Computer Communications Security*, Dallas, USA, 2017.

[34] X. Jia, X. Wei, X. Cao, H. J. I. Foroosh and P. Recognition, "ComDefend: An efficient image compression model to defend adversarial examples," in *IEEE/CVF Conf. on Computer Vision Pattern Recognition*, Long Beach, CA, USA, pp. 6077–6085, 2019.

[35] A. ArjomandBigdeli, M. Amirmazlaghani, M. J. Khalooei and I. Systems, "Defense against adversarial attacks using DRAGAN," in *6th Iranian Conference on Signal Processing Intelligent Systems*, Tehran, Iran, pp. 1–5, 2020.

[36] Y. Bakhti, S. A. Fezza, W. Hamidouche and O. J. I. A. Déforges, "DDSA: A defense against adversarial attacks using deep denoising sparse autoencoder," *IEEE Access*, vol. 7, pp. 160397–160407, 2019.

[37] F. Yu, L. X. Wang, X. Fang and Y. J. S. C. N. Zhang, "The defense of adversarial example with conditional generative adversarial networks," *Security and Communication Networks*, vol. 2020, pp. 3932584:1–3932584:12, 2020.

[38] Y. Liu, S. Mao, X. Mei, T. Yang and X. J. Zhao, "Sensitivity of adversarial perturbation in fast gradient sign method," in *IEEE Symp. Series on Computational Intelligence*, Orlando, Florida, USA, pp. 433–436, 2019.

[39] S. Baluja and I. S. J. A. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," arXiv preprint arXiv:1312.6199, vol. abs/1703.09387, 2017.

[40] K. He, X. Zhang, S. Ren, J. J. I. C. O. C. V. Sun and P. Recognition, "Deep esidual earning for image recognition," in *IEEE Conf. on Computer Vision Pattern Recognition*, Honolulu, HI, USA, pp. 770–778, 2016.

[41] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional networks for ciomedical image segmentation," arXiv:1505.04597, 2015.

[42] J. Rauber, W. Brendel and M. Bethge, "Foolbox: A python toolbox to benchmark the robustness of machine learning models," arXiv preprint arXiv:1707.04131, 2017.