DNDNet: Reconfiguring CNN for Adversarial Robustness

Akhil Goel[1], Akshay Agarwal[1], Mayank Vatsa[2], Richa Singh[2], and Nalini K. Ratha[3]
[1]IIIT-Delhi, India; [2]IIT Jodhpur, India; [3]IBM TJ Watson Research Center, USA
[1]{akhil15126, akshaya}@iiitd.ac.in; [2]{mvatsa, richa}@iitj.ac.in; [3]ratha@us.ibm.com

Abstract

Several successful adversarial attacks have demonstrated the vulnerabilities of deep learning algorithms. These attacks are detrimental in building deep learning based dependable AI applications. Therefore, it is imperative to build a defense mechanism to protect the integrity of deep learning models. In this paper, we present a novel "defense layer" in a network which aims to block the generation of adversarial noise and prevents an adversarial attack in black-box and gray-box settings. The parameter-free defense layer, when applied to any convolutional network, helps in achieving protection against attacks such as FGSM, L_2, Elastic-Net, and DeepFool. Experiments are performed with different CNN architectures, including VGG, ResNet, and DenseNet, on three databases, namely, MNIST, CIFAR-10, and PaSC. The results showcase the efficacy of the proposed defense layer without adding any computational overhead. For example, on the CIFAR-10 database, while the attack can reduce the accuracy of the ResNet-50 model to as low as 6.3%*, the proposed "defense layer" retains the original accuracy of* 81.32%.

1. Introduction

Modern machine learning algorithms generally utilize deep learning architectures to achieve state-of-the-art (SOTA) results. However, these algorithms are vulnerable due to the singularities of deep learning. Szegedy et al. [34] have shown that the deep learning algorithms misclassify an input image if some *adversarial noise* is added. As shown in Figure 1, the adversarial attacks can "learn" a noise pattern such that when it is embedded in the input image, it can misclassify the sample with high confidence.

The adversarial examples can be generated in a number of ways. Researchers have used techniques such as box-constrained optimization, simple signed gradient addition, evolutionary algorithms, and minimization of logit layer representation to fool a healthy system [6, 13, 15, 22, 23]. These attacks highlight the singularities of DNN models towards unseen data distribution. With advancements in attack generation algorithms, adversarial samples have achieved translation, scaling, and rotation invariance [29]. Athalye and Sutskever[3] have shown the existence of 3D adversarial examples that can fool the DNN models in the physical world. The adversarial attack can be in the form of black-box and white-box scenarios. In the black-box case, the attacker generates adversaries that fool the model without actually having any information about it. In contrast, in the white-box case, the attacker holds complete information about the architecture and weights of the model.

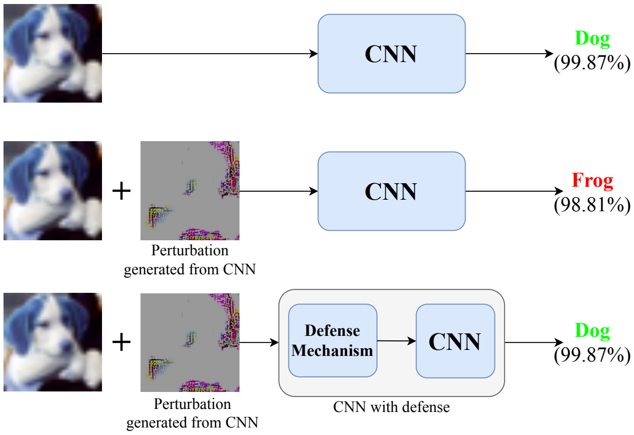

Figure 1: An illustration of an adversarial attack and proposed defense mechanism. A CNN model (e.g. VGG16) correctly classifies a clean input as *Dog* with confidence value of 99.87%; however, after adding adversarial noise, the same model classifies it as *"Frog'* with a very high confidence of 98.81%. The proposed defense algorithm adds a pre-processing layer to defend against adversarial perturbations and the CNN model correctly classifies the image with a confidence of 99.87%. The perturbations here are magnified 6 times for clarity.

As shown in Figure 1 (last row), a defense mechanism can be incorporated to defend CNN against adversarial attacks. Defense algorithms proposed in the literature can be classified into three groups: (i) detection, (ii) mitigation through data manipulation, and (iii) mitigation through network manipulation [14, 37]. Recently, Theagarajan et al.

[35] introduced a theoretical framework that negates the effects of the adversarial perturbations by levaraging a probabilistic model to project perturbed samples to adversarial-free zones. Adversarial training [36] retrains an entire DNN network with clean and adversarial examples with the hope of increasing robustness towards adversarial examples. Papernot et al. [26] proposed the modification in the network so that the gradient magnitude can be reduced. Most of the existing defense algorithms have potential limitations and fail to defend within their claimed threat model assumptions [2, 5, 6, 10]. A detailed survey of existing adversarial defense algorithms can be found in [27, 38].

Studies to find the cause of adversarial effects have been performed; however, no substantial results have been established so far [8, 9, 22, 30]. Some researchers observed that the linearity [13, 21] of DNN models in the input space could be a reason for adversarial effects. The other school of thought believes that large singular values of internal layer's weights [9] make the models vulnerable to slight modification. Some believe that there might be a potential drawback in the DNN architecture, and these architectural flaws or flaws in the learning of networks might have opened the doors for adversarial examples.

In this research, we propose the reconfiguration of the CNN model through a defensive layer that *"blocks"* the adversary generation process. The proposed concept of defense layer, termed as *'DNDNet' (aka Do not disturb Network)*, is shown in Figure 2. In the existing defense algorithms either the external classifier is trained or CNN model is retrained with adversarial examples or the network is modified with an increased parameter for adversarial robustness [1, 11, 20, 25, 26, 32, 36]. However, the proposed model modifies the architecture of conventional CNN models without *any "trainable parameters"*. This makes the proposed model computationally efficient with the advantage of adversarial robustness in a *'gray-box'* and *'black-box'*[1] settings.

2. CNN and Adversarial Vulnerability

Convolutional neural networks learn by training the weight parameters on (preferably) large databases via gradient descent. For the specific purpose of image classification, the convolution layers are followed by a dense layer (usually a softmax classifier), which has neurons equal to the number of classes. In the case of a softmax classifier, the output of a neuron j from the last dense layer gives the probability of an image belonging to class j; $P(y = j \,|\, I)$. The weight update rule of CNNs can be defined using the following equations:

$$\upsilon_{i+1} = c_1 \cdot \upsilon_i - c_2 \cdot \epsilon \cdot \omega_i - \epsilon \cdot \{|\frac{\partial L}{\partial \omega}|\omega_i\}_{D_i}$$

$$\omega_{i+1} = \omega_i + \upsilon_{i+1} \tag{1}$$

where, c_1 and c_2 are the hyperparameters of the model. i, υ, and ϵ denote the iteration, momentum, and learning rate ingredients of the conventional CNN model. $\{\cdot\}$ is the average of the gradient of the loss function over batch size D. Using a train set, back propagation algorithm helps to train a model M. This model outputs the probability vector values P for a particular image I belonging to the classes present in the test set. The model assigns the image I to the class j due to the maximum value in the vector $P = [p_1, p_2, \cdot\cdot, p_j, \cdot\cdot p_n]$ at the j^{th} location. Suppose, c is the true class label of I; an adversarial attack aims to fool the model such that the predicted label is not c. The minimal adversarial noise N is added into the image with the conditions of two folds (i) visual imperceptibility and (ii) decrease in the probability value of the true class. To find the best adversarial noise N, following generalized optimization is solved:

$$\begin{aligned} minimize_D \;\; ||N|| \\ p = f(I+N) \\ max(p_1, p_2, \cdot\cdot, p_n - p_c) > 0 \end{aligned} \tag{2}$$

such that $Min \leq I + N \leq Max$, where Min and Max represent the lowest and highest intensity value of an image, respectively. f is the classification function resulting probability values from the model. The adversarial optimization formulation for simplicity can be written as follows:

$$\begin{aligned} minimize_D \;\; ||N|| + \epsilon \cdot CE(p, p^a) \\ p = f(I+N) \\ Min \leq I + N \leq Max \end{aligned} \tag{3}$$

where, p^a denotes the probability of the class label in which the adversarial images need to be classified, ϵ is the constant value, and CE is the cross-entropy loss of the model.

In many adversarial generation algorithms, the gradient of the network is used to lead the model to misclassification. For example, in the case of a fast gradient sign method (FGSM) attack [13], the signed gradient is added back in the input image to find the adversarial examples. This increases the error function leading to probable chances of misclassification. The calculation of perturbation vector p can be defined as: $p = \epsilon \cdot sign(\nabla_I J(I))$, where, ϵ is the confidence of the attack and J is the classification loss function. ∇ represents the gradient of the loss function J with respect to input I.

[1] The scenarios where the attacker has limited knowledge or no knowledge of the target model and defense mechanism deployed for its security is referred to as gray-box and black-box setting, respectively.

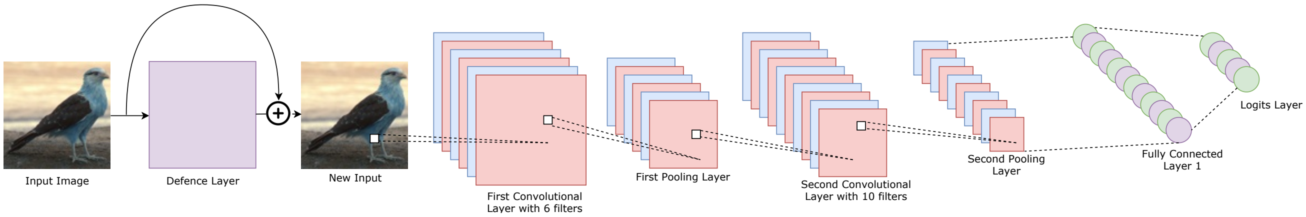

Figure 2: Architecture of the proposed convolutional neural network with defense layer, i.e., DNDNet. Here for illustration purposes we have depicted the concept of defense layer with a shallow network, but as shown in experiments, defense layer can be added to *'any'* CNN models such as VGG, ResNet, and DenseNet.

Carlini and Wagner [6] formulate the fooling system as a trade-off between two terms. The first term minimizes the L_2 norm between the original and perturbed image, and the second term makes sure that the perturbed image gets classified either into the target class or into any class other than the actual class. The formulation of C&W L_2 attack is defined as:

$$\begin{aligned} minimize||\frac{1}{2}(\tanh{(I+p)}+1)-I||_2^2+ \\ c\cdot f(\frac{1}{2}(\tanh{(I+p)}+1)) \end{aligned} \tag{4}$$

where,

$$f(I) = max(max\{Z(I)_i : i \neq t\} - Z(I)_t, -\kappa)$$

is the maximum score corresponding to the non-target class, t is the target class, and Z is the logits layer representation. κ is the parameter that controls the confidence of misclassification and p is the introduced perturbation that minimizes the above expression.

Inspired by the learning of adversarial examples, in this research, the aim is to find a **"defense layer"**, which can counter the adversarial learning process in 'gray-box' and 'black-box' settings. The central concept of the proposed defense layer is to utilize gradient flow to protect against adversarial attacks.

3. DNDNet: Proposed Defense Layer

Krotov and Hopfield [19] argued that changing the nonlinearity might increase a model's robustness. Similarly, Papernot et al. [26] hypothesized that the distillation of deep networks into smaller ones could decrease the sensitivity towards adversarial examples. However, Carlini and Wagner [5] show that distillation based defense is ineffective hence dismissing the claim towards network size on adversarial examples. Inspired by these findings, in this research, we propose a new architecture that shows adversarial robustness over various existing adversarial attacks. DNDNet leverages the fact that a majority of successful adversarial attacks require the knowledge of the flow of gradients within the network to produce useful perturbations. The proposed defense layer conceals the gradient flow without any computational overheads and performance degradation. For simplicity, the proposed defense layer is explained with respect to how the gradient is used to generate an attack.

The learning rule of a network $f(I,\theta)$ with input I and trainable parameters θ can be defined as:

$$\theta \ := \ \theta - \frac{\partial f(I,\theta)}{\partial \theta}$$

where, ∂ represents the partial derivative with respect to the model parameters θ. If the partial derivative of the loss function with respect to model parameter is known then the partial derivative with respect to the input can be defined using chain rule as:

$$\frac{\partial f(I,\theta)}{\partial I} = \frac{\partial f(I,\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial I}$$

This information, although superfluous for model parameter learning, is crucial for conducting a successful gradient-based adversarial attack. Gradient-based attack generation algorithms monitor the effects of the input image on the target network. They do it by either explicitly calculating the rate of change of the model's loss function with respect to the inputs or by implicitly calculating the rate of change of their proposed formulation (which indirectly depends on the model's output) with respect to the input image. The algorithms use this information to guide the adversarial samples for either misclassification to a random or into target class. This shows how small information leaks like this can prove to be fatal for an otherwise healthy and developed DNN.

3.1. Architecture Details

As shown in Figure 3, in this paper, we propose a defense layer L that calculates the k^{th} root (for example 'cube root') of the difference of inputs I and a constant a, as a approaches I. However, the chosen value of k must be greater than 1. Mathematically, the proposed defense layer can be defined as follows:

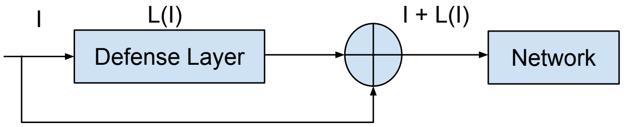

Figure 3: Illustrating the process of addition of new defense layer in the network.

$$L(I) = (I - a)^{\frac{1}{k}}|_{a \to I} \tag{5}$$

$$\frac{dL(I)}{dx} = \frac{1}{k} \cdot \frac{1}{(I-a)^{\frac{k-1}{k}}|_{a \to I}} \tag{6}$$

Consider a single layer network with trainable weights and bias as w and b, respectively and activation function ϕ. From figure 3 output of the network is:

$$f(I, w, b) = \phi(w \cdot (I + L(I)) + b)$$

$$\frac{df(I, w, b)}{dI} = \phi^{'}(w \cdot (I + L(I)) + b) \cdot (w + w \cdot \frac{dL(I)}{dI}) \tag{7}$$

Equation 6 suggests that $\frac{dL(I)}{dI} \to \infty$. This renders equation 7 not defined. The implication of undefined nature shows no generation of adversarial noise to fool the inbuilt defense layer models. The proposed defense layer has **"no trainable parameters"**, and is successfully able to block the generation of adversarial noise. The proposed CNN model with defense layer is termed as *Do not disturb* CNN network (**'DNDNet'**).

For any general network, the **forward pass** output of the first trainable layer O_1, given that the trainable weights and bias are w and b, respectively and the activation function used is ϕ is given by:

$$O_1 = \phi(w \cdot (I + L(I)) + b) \tag{8}$$

The above equation using the notation of proposed layer L defined in equation 6 where the constant (a) equals to the input image (I), can be reduced to the following form:

$$O_1 = \phi(w \cdot I + b) \tag{9}$$

which is essentially the forward pass learning rule of a typical DNN.

Backward pass in the proposed network remains identical to a typical DNN. Here we present the computation of gradients at layer 1. Let the cost function of the network be C, the pre-activation output of layer 1 be z_1 and the weights and bias in the first and the second layer be w_1, b_1, w_2 and b_2, respectively. At layer 1, we have

$$\frac{\partial C}{\partial w_1} = \frac{\partial C}{\partial \phi(z_1)} \cdot \frac{\partial \phi(z_1)}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1} \tag{10}$$

and

$$\frac{\partial C}{\partial b_1} = \frac{\partial C}{\partial \phi(z_1)} \cdot \frac{\partial \phi(z_1)}{\partial z_1} \cdot \frac{\partial z_1}{\partial b_1} \tag{11}$$

where,

$$z_1 = w_1 \cdot (I + L(I)) + b$$

The first two terms of the gradient are known by previous iterations of backpropagation and

$$\frac{\partial z_1}{\partial w_1} = I + L(I) \to I \tag{12}$$

$$\frac{\partial z_1}{\partial b_1} = 1 \tag{13}$$

This shows that the learning rule of the backward pass does not change as well. Hence, the basic functionality of the proposed defended CNN model i.e. classification does not get affected.

4. Experimental setup

We perform experiments to show the effectiveness of the proposed defense layer against various attack generation algorithms on different state of the art models.

4.1. Databases and CNN Models

Databases: The experiments are performed on MNIST[2], CIFAR10 [18] and Point and Shoot Cameras (PaSC) [4] databases. The **MNIST** database contains the handwritten digits of values ranging from 0-9. **CIFAR10** is a popular database of object recognition in color space. The database has images of 10 different objects classes. The MNIST and CIFAR-10 database contains $60,000$ and $50,000$ training images, respectively. Both the databases contain $10,000$ testing images. The databases such as MNIST and CIFAR10 contain low-resolution images of digits and objects. To further demonstrate the performance, experiments with high-resolution **PaSC** face database is also performed. The faces images vary in terms of resolutions, background, illumination, and expression. The database contains $3,224$ images of resolution 224×224 from 293 individuals. We divide the PaSC database into a train set, which consists of 80% images of each person, and the test set consists of the remaining 20% images. Each image in the test set is matched with the train set to form the face identification score (1:N matching) matrix.

Models: The effect of the proposed defense layer is experimented with five existing state-of-the-art architectures including VGG16, VGG19 [31], ResNet50 [16], InceptionV3 [33], and DenseNet121 [17] on several adversarial

[2] http://yann.lecun.com/exdb/mnist/

attacks. Other than VGG models, all models are trained from scratch. The performance of both undefended (i.e., conventional) and defended model is demonstrated with object classification accuracy and robustness under adversarial attacks.

Attack Generation Algorithms: We test the proposed defense layer against FGSM [13], C&W L_2 [6], PGD [22], DeepFool [24], and Elastic-Net (EAD) [7] attacks. Details of these attacks are given in the respective papers.

4.2. Implementation Details

To setup our experiments, we make a custom layer that implements Equation 5 and places it in conjunction with the input layer. The stochastic gradient descent rule with a constant learning rate of 7×10^{-5} is used to learn the model parameters. The training of the models is stopped after the nature of validation loss changes to non-decreasing. In this research, we have used 'cube' root in the defense layer; however, similar performance is observed for other root values as well.

For the FGSM attack, we fix the attack step size parameter (ϵ) to 0.1. For C&W L_2 attack, the default parameter setting is used with a number of binary search steps equal to 9, initial parameter constant set to 0.001, and learning rate set to 0.01. We perform the C&W attack for 5000 iterations with batch size set to 20. Similar to [6], we use Adam optimizer for perturbation optimization. The existing attack algorithms are implemented using SmartBox adversarial toolbox [12].

5. Experimental Results and Analysis

First, the results of object/digit classification and face identification experiments are reported to showcase that the addition of the defense layer does not affect the generic nature of CNN models. Later, the adversarial robustness against the gradient and strong optimization-based attack is reported.

5.1. Classification Performance

The classification performance on MNIST, CIFAR10, and face PaSC database are computed with state-of-the-art CNN models. In this experiment, we first trained the conventional models and then matched their performance with new models with the proposed defense layer. Figures 4, 5, and 6 show the result of each classification task with conventional and defended CNN models. It is evident from the results that the new proposed defense layer has no significant *'negative'* impact on the classification nature of the conventional models (as also concluded from section 3). The analysis of the results can be summarized using the following points:

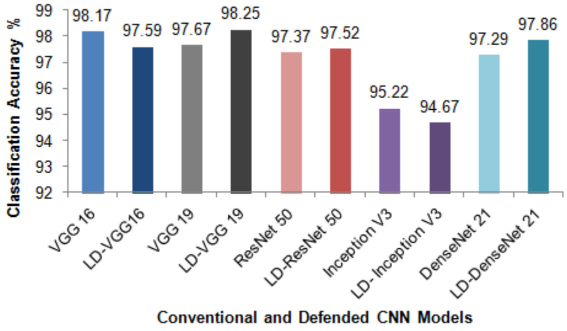

Figure 4: MNIST digit classification performance of the conventional and proposed defense layered CNN models. LD denoted the layer defended version of the network.

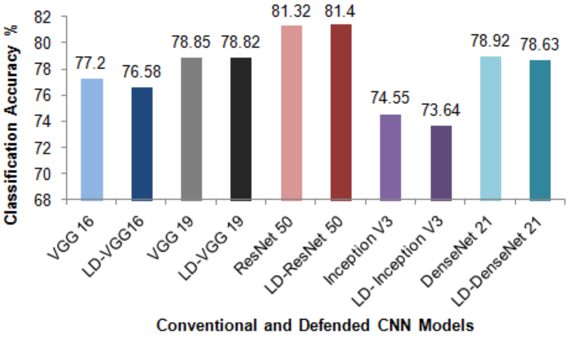

Figure 5: CIFAR10 object classification performance of the conventional and proposed defense layered CNN models. LD denoted the layer defended version of the network.

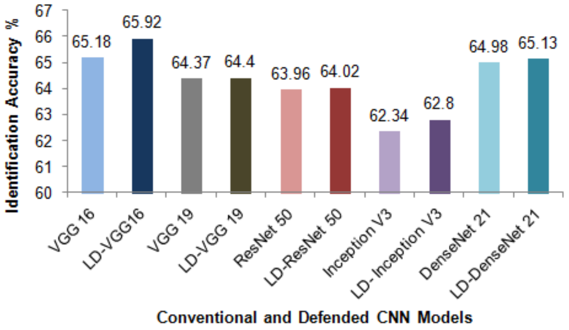

Figure 6: PaSC face identification performance of the conventional and proposed defense layered CNN models. LD denoted the layer defended version of the network.

1. As defined in section 3, the learning of the CNN models with the addition of the new layer is not affected. It is also gets reflected in the classification performance, which shows that the models either retain the accuracy or gets improved.

2. On the MNIST database, the conventional VGG16 model yields 98.17% accuracy, whereas the proposed

defended model shows a slight drop in accuracy with a value of 97.59%.

3. Similarly, when the object recognition task on the CIFAR10 database is performed using the ResNet50 model, the highest classification accuracy of 81.32% is achieved. This shows that the defended models are able to retain the classification performance with respect to base models.

4. Conventional and defended DenseNet models show the face identification accuracy of 64.98% and 65.13%, respectively. The proposed model is not only able to retain the performance on low-resolution databases such as MNIST and CIFAR10 but also on the high-resolution images of PaSC.

5. Even in the case of the conventional ResNet50 model, the defended model shows similar performance with a slight improvement of 1.15%, 0.08%, and 0.16% in digit recognition, object classification, and face identification performance, respectively. Similar performance improvement has been noticed against the DenseNet model for digit and face identification experiments.

In general, it is our observation that the object recognition accuracy of the conventional and defended networks are within the same range. The object recognition experiments across each state-of-the-art (SOTA) CNN model show that modification through defense does not affect the recognition performance. Therefore, the proposed model can be utilized for object recognition with high adversarial robustness.

5.2. Adversarial Robustness

In this paper, several attack generation algorithms implemented in SmartBox [12] are utilized to test the robustness. Based on the way the algorithms are wired, different algorithms react differently to the proposed defended network:

- **FGSM:** The results before and after the defense-related to FGSM adversarial attack are listed in Table 1. In all cases, the proposed algorithm is able to defend with very high accuracy.

- **Optimization Attacks:** While C&W L_2 [6] and Elastic-Net (EAD) [7] produces black images, FGSM [13] returns the images containing no adversarial information. For conventional CNN, the C&W attack is successfully able to reduce the accuracy to 0%; however, for the defended network, they can be discarded before processing. Therefore, with respect to the L_2 and EAD attack, no drop in recognition accuracy is reported. Similar to previous attacks, the proposed defense layer is successfully able to block the adversarial nature of the DeepFool adversary [24].

Table 1: Identification accuracy (%) on the MNIST, CIFAR10, and PASC databases under the FGSM attack, with and without the proposed defense layer.

CNN Model	**MNIST**	**CIFAR-10**	**PaSC**
VGG-16 (attacked)	0.2	4.3	0.0
VGG-16 (with defense)	98.17	77.2	65.18
ResNet50 (attacked)	1.3	6.3	0.1
ResNet50 (with defense)	96.37	81.32	60.27
DenseNet121 (attacked)	0.9	1.40	0.0
DenseNet121 (with defense)	97.86	78.63	65.13

Figure 7 shows the optimization procedure of the C&W L_2 attack before and after the addition of the defense layer and Figure 8 shows the distribution of added perturbations by FGSM before and after addition of the proposed defense layer. Without the defense layer, optimization of C&W L_2 attack proceeds as expected; however, due to blocking the adversarial gradients by the defense layer, the optimization procedure does not proceed further. Similarly, for the FGSM attack, the perturbations calculated using the original network can be represented as a standard Gaussian distribution, but since there are no perturbations generated for a network with the defense layer, they can be at best described using a Dirac delta function.

The accuracy of the VGG16 model on the generated adversarial images from the same network drops to 0.2%, 4.3%, and 0.0% on MNIST, CIFAR10, and PaSC database, respectively. The defended models regain the classification accuracy similar to original images on each database. Similar performance is observed across other secured and conventional CNN models against each adversarial attack.

Similar to recent adversarial defense, including Defense GAN [28], the proposed defended model is tested against recent projective gradient descent (PGD) attack [22]. The attack is performed using default parameters provided with the paper. For example, on the CIFAR10 database, when conventional VGG16 is used for experimentation, the PGD adversarial examples accuracy drops to 0.0%. The proposed defended model improves the accuracy to 51.4%. The slight drop in efficiency is observed because of some random noise added initially in the input images itself.

Other Architectural Modifications: Other than the defense layer L defined in equation 6, we have also explored other configuration as the defense layer and found the robustness against existing attacks. The layer which we also found effective with addition to input layer is defined as the addition of Gaussian (Gaussian(I)) and mean filtered image (Mean(I)) with strength parameters c_1 and c_2. The revised

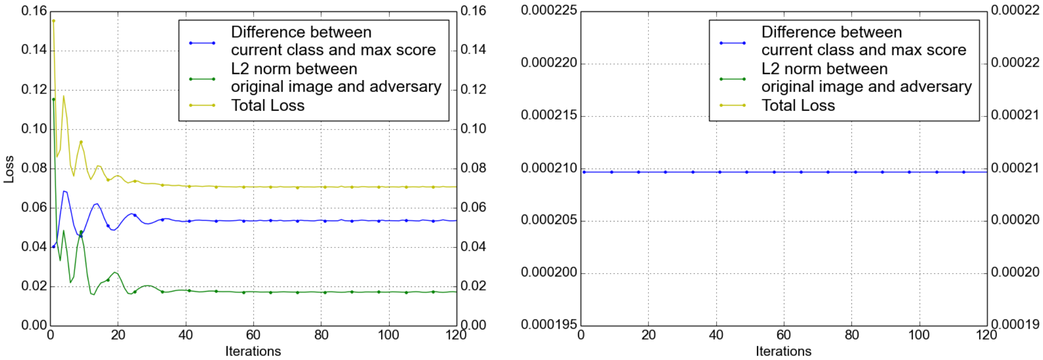

Figure 7: L_2 attack optimization for conventional CNN (left) and DNDNet (right). By blocking the gradients, the attack algorithm does not get a signal and hence it cannot proceed with the optimization process.

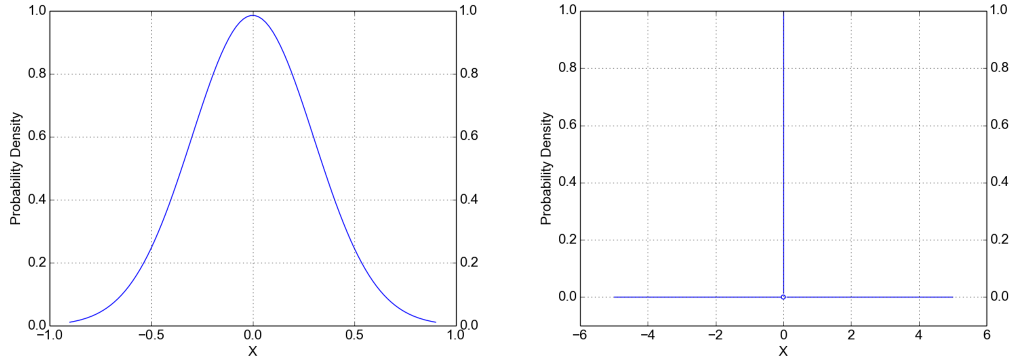

Figure 8: The FGSM perturbation generated using Conventional CNN and DNDNet (i.e., with defense layer). With the addition of the proposed defense layer, the attack fails to produce perturbations.

layer can be written as:

$$L(I) + c_1 \times Gaussian(I) + c_2 \times Mean(I)$$

where, $L(I)$ is the proposed defense layer defined in equation 5. To compensate for the addition of image data in the defense layer, the feedback is weighed down to $1 - c_1 - c_2$ times I. The layer with different input pre-processing helps in reducing the effect of noise present in the input image, such as in the case of PGD attack. In our experiments, we have observed c_1=c_2=0.1.

In the present scenario with the *gray-box* threat model, to the best of our knowledge, none of the existing gradient-based adversarial attack algorithms is able to generate the adversarial noise from the proposed *'DNDNet'*.

6. Conclusion

The existence of adversaries is a significant problem for deep learning algorithms. Existing defense systems either require additional time, computational resources, or knowledge of the system. This research proposes a solution for adversarial defense by reconfiguring CNN. We have developed a **defense layer with no "trainable parameters"**, which successfully shields the target network against the gradient and robust optimization-based adversarial attacks. The effectiveness of the proposed algorithm is demonstrated on three databases using multiple networks such as VGG, ResNet, and DenseNet, and attacks. We observe that the addition of the new layer does not have any time-based or precision-based performance costs. The proposed solution is designed for a black-box and gray-box threat models. In the future, we intend to extend this work and provide a solution for a white-box threat model where the attacker has the knowledge of the target model and the defense mechanism.

7. Acknowledgement

A. Agarwal was partially supported by the Visvesvaraya PhD Fellowship. M. Vatsa is partially supported through the Swarnajayanti Fellowship by the Government of India.

References

[1] A. Agarwal, R. Singh, M. Vatsa, and N. Ratha. Are image-agnostic universal adversarial perturbations for face recognition difficult to detect? *IEEE BTAS*, pages 1–7, 2018.

[2] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ICML*, pages 274–283, 2018.

[3] A. Athalye and I. Sutskever. Synthesizing robust adversarial examples. *ICML*, pages 284–293, 2018.

[4] J. R. Beveridge, P J. Phillips, D. S Bolme, B. A Draper, G. H Givens, Y. M. Lui, M. N. Teli, H. Zhang, W T. Scruggs, K. W. Bowyer, et al. The challenge of face recognition from digital point-and-shoot cameras. In *IEEE BTAS*, pages 1–8, 2013.

[5] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on AISec*, pages 3–14, 2017.

[6] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE S&P*, pages 39–57, 2017.

[7] P. Chen, Y. Sharma, H. Zhang, J. Yi, and C. Hsieh. EAD: elastic-net attacks to deep neural networks via adversarial examples. *AAAI*, 2018.

[8] P. Chiang, R. Ni, A. Abdelkader, C. Zhu, C. Studor, and T. Goldstein. Certified defenses for adversarial patches. *ICLR*, pages 1–17, 2020.

[9] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. In *ICML*, pages 854–863, 2017.

[10] A. Ghiasi, A. Shafahi, and T. Goldstein. Breaking certified defenses: Semantic adversarial examples with spoofed robustness certificates. *arXiv preprint arXiv:2003.08937*, 2020.

[11] A. Goel, A. Agarwal, M. Vatsa, R. Singh, and N. Ratha. DeepRing: Protecting deep neural network with blockchain. *CVPRW*, 2019.

[12] A. Goel, A. Singh, A. Agarwal, M. Vatsa, and R. Singh. Smartbox: Benchmarking adversarial detection and mitigation algorithms for face recognition. *IEEE BTAS*, pages 1–7, 2018.

[13] I. J Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *ICLR*, pages 1–11, 2015.

[14] G. Goswami, A. Agarwal, N. Ratha, R. Singh, and M. Vatsa. Detecting and mitigating adversarial perturbations for robust face recognition. *IJCV*, 127(6-7):719–742, 2019.

[15] G. Goswami, N. Ratha, A. Agarwal, R. Singh, and M. Vatsa. Unravelling robustness of deep learning based face recognition against adversarial attacks. *AAAI*, pages 6829–6836, 2018.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016.

[17] G. Huang, Z. Liu, L. Van D. M., and K. Q. Weinberger. Densely connected convolutional networks. In *IEEE CVPR*, pages 2261–2269, 2017.

[18] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[19] D. Krotov and J. Hopfield. Dense associative memory is robust to adversarial inputs. *Neural Computation*, 30(12):3151–3167, 2018.

[20] X. Li and F. Li. Adversarial examples detection in deep networks with convolutional filter statistics. *ICCV*, pages 5764–5772, 2017.

[21] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao. Foveation-based mechanisms alleviate adversarial examples. *ICLR*, 2016.

[22] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, pages 1–28, 2018.

[23] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. *CVPR*, pages 1765–1773, 2017.

[24] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *IEEE CVPR*, pages 2574–2582, 2016.

[25] T. Na, J. H. Ko, and S. Mukhopadhyay. Cascade adversarial machine learning regularized with a unified embedding. *ICLR*, pages 1–16, 2018.

[26] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE S&P*, pages 582–597, 2016.

[27] K. Ren, T. Zheng, Z. Qin, and X. Liu. Adversarial attacks and defenses in deep learning. *Engineering*, pages 1–15, 2020.

[28] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *ICLR*, 2018.

[29] M. Sharif, S. Bhagavatula, L. Bauer, and M. K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM SIGSAC CCS*, pages 1528–1540, 2016.

[30] I. Shumailov, Y. Zhao, R. Mullins, and R. Anderson. Towards certifiable adversarial sample detection. *arXiv preprint arXiv:2002.08740*, 2020.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, pages 1–14, 2015.

[32] R. Singh, A. Agarwal, M. Singh, S. Nagpal, and M. Vatsa. On the robustness of face recognition algorithms against attacks and bias. *AAAI*, 2020.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE CVPR*, pages 2818–2826, 2016.

[34] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *ICLR*, 2014.

[35] Rajkumar Theagarajan, Ming Chen, Bir Bhanu, and Jing Zhang. Shieldnets: Defending against adversarial attacks using probabilistic adversarial robustness. In *CVPR*, pages 6988–6996, 2019.

[36] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *ICLR*, pages 1–20, 2018.

[37] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. *ICLR*, pages 1–16, 2018.

[38] X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE TNNLS*, 30(9):2805–2824, 2019.