17th International Conference on Machine Vision Applications (MVA)
Fully Online, July 25–27, 2021.

Adversarial Defense Through High Frequency Loss Variational Autoencoder Decoder and Bayesian Update With Collective Voting

Zhixun He
University of California, Merced
Electrical Engineering & Computer Science
Merced, CA 95343
zhe5@ucmerced.edu

Mukesh Singhal
University of California, Merced
Electrical Engineering & Computer Science
Merced, CA 95343
msinghal@ucmerced.edu

Abstract

In recent years, Deep Neural Network (DNN) approaches for computer vision tasks have shown tremendous promise and potential. However, they are vulnerable to data that are carefully crafted with adversarial attacks, which can cause mis-prediction and raise security risk to real-world deep learning systems. To make the DNN-based approaches more robust, we propose a defense strategy based on High Frequency Loss Variational Autoencoder Decoder (VAE) and randomization among multiple post-VAE classifiers' predictions. The main contributions of the proposed defense framework are: 1) a new adversarial defense framework that features randomization process to effectively mitigate adversarial attacks; 2) reconstruction of high-quality images from adversarial samples with the VAE enhanced with spatial frequency loss; 3) use of a Bayesian process to jointly combine the collective voting results and the targeted classifier's prediction for final decision. We evaluate our approach and compare it with existing approaches on CIFAR10 and Fashion-MNIST data sets. The experimental study shows that the proposed method outperforms existing methods.

1 Introduction

In recent years, DNNs have delivered unprecedented results in many computational learning tasks, such as video [9], image [12] and audio [11], etc. However, these methods have been shown to be susceptible to adversarial attacks [5, 26], where a certain number or all pixels in an image are carefully perturbed, such that the classifier is fooled to give wrong prediction.

Various defense mechanisms have been proposed to mitigate the adversarial attacks on images[1, 24, 22, 15]. These mechanisms can be broadly classified into three different categories: (1) the training process mixes adversarial images with original images as the training set to make the classifier more robust to adversarial attacks[5, 26]; (2) use high-level latent features from DNNs for clustering either to do anomaly detection or to classify the category[6, 18]; (3) use DNNs to reconstruct the images using generative networks[21, 25, 7].

Most of the previous work assumes that only the classifier is targeted during an adversarial attack, and the whole defense framework is unknown to the adversary[1, 24, 22, 15]. But if an adversary has an access to the entire defense framework, he can take the advantage of it to attack the entire defense framework. In order to mitigate such attacks, our approach combines both randomization and discretization through VAE and post-VAE collective voting such that an adversary could not easily use back-propagation to attack the entire framework, and at the same time, the quality of reconstructed images is improved by using Spatial Frequency Loss (SFL). At the end, given the targeted classifier's prediction on the reconstructed images and the post-VAE voting result, the final prediction is given by the Bayesian update using the above two results as evidence.

2 The Proposed Method

2.1 Variational Autoencoder Decode (VAE)

VAE features: 1) an **Encoder**, $P(Z|X)$, that maps given input data, X, to the distribution of the unobserved hidden variables (the latent features) Z; 2) a **Decoder**, $P(X|Z)$, that approximates the reverse process, converting the hidden features (with probability distribution $P(Z)$) back to dimension of the observable data. Our method uses parameterized approximators $Q_\theta(Z|X)$, $P_\omega(X|Z)$ and $P_\psi(Z)$ to model the encoder, decoder and hidden feature distribution, respectively. By optimizing the evidence lower bound (ELBO) of VAE, the decoder is encouraged to reconstruct images with similar distribution as the training input:

$$(\theta^*, \omega^*, \psi^*) = \underset{(\theta,\omega,\psi)\in\Theta\times\Omega\times\Psi}{argmax} \mathbb{E}_{z\sim Q_\theta(Z|X)} \log P_\omega(X|Z) - D_{KL}\left[Q_\theta(Z|X) \parallel P_\psi(Z)\right] \quad (1)$$

Maximizing $\mathbb{E}_{z\sim Q_\theta(Z|X)} \log P_\omega(X|Z)$ is equivalent to generating images that are as close as the observable data, given the latent features Z which is generated by the learnt encoder $Q_\theta(Z|X)$. The loss function to measure the difference is denoted as reconstruction loss, $\mathcal{L}_{\mathbf{rec}}$, and the form we adopt is the Mean Square Error (MSE). Thus, the parameters can be calculated [23]:

$$\omega^* = \underset{\omega\in\Omega}{argmin}\mathbb{E}_{z\sim Q_\theta(Z|X)}\left[\frac{(P_\omega(X|Z)-X)^2}{2}\right] \tag{2}$$

In equation (1), the marginal distribution of hidden features, $P_\psi(Z)$, is selected based on different situation of interest [17]. We adopt zero-centered independent Gaussian distribution, in which its mean and variance are the parameters. We empirically choose the dimension k for the hidden feature Z, such that the outputs from $Q_\theta(Z|X)$ (the mean vector and variance vector) have the same dimension as that of $P_\psi(Z)$. We denote $D_{KL}\left[Q_\theta(Z|X) \parallel P_\psi(Z)\right]$ as regularization loss, $\mathcal{L}_{\mathbf{reg}}$.

2.2 Spatial Frequency Loss (SFL) for VAE

To learn high spatial frequency features such as sharper edges in the reconstructed images, it is important to guide the learning with extra loss function that reflects the difference in the sharpness of edges between an input and an output. Laplacian of Gaussian (LoG) is an edge detection technique [20], which not only detects brightness intensity changes in an image but also detects the intensity changes at different scale.

We adopt LoG kernel to extract edges for both X and $X^\sim$:

$$LoG = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

where, x and y are the pixel coordinates of the grayscaled image and σ is the scale that controls the fineness of the edges to be detected.

Thus, in order to achieve the reconstruction of detailed features, a spatial frequency loss, $\mathcal{L}_{\mathbf{SFL}}$, between input X and reconstruction $X^\sim$ is added to the total loss when training the VAE. Overall, the loss functions for training VAE are as follows:

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}_{rec} + \mathcal{L}_{reg} + \mathcal{L}_{SFL}\\
\mathcal{L}_{rec} &= \mathbb{E}_{z\sim Q_\theta(Z|X)}\left[\frac{(P_\omega(X|Z)-X)^2}{2}\right]\\
\mathcal{L}_{reg} &= D_{KL}\left[Q_\theta(Z|X) \parallel P_\psi(Z)\right]\\
\mathcal{L}_{SFL} &= \left\|Conv(X_{gray}, LoG) - Conv({X^\sim}_{gray}, LoG)\right\|_2
\end{aligned} \tag{4}$$

2.3 Post-VAE Classification and Voting

The proposed approach introduces randomization and discretization in decision making process. All post-VAE classifiers have different structures and hyperparameters, and they are trained using reconstructed images $X^\sim$ from VAE. After images are reconstructed from VAE, R out of M post-VAE classifiers are randomly selected to classify $X^\sim$. A majority vote is taken from the R predictions. Thus, although an adversary has access to VAE and all post-VAE classifiers and it can craft strong adversarial samples using the information in $f_c^{(i)}(f_{VAE}(x))$, $i = 1, 2, ..., M$, the adversarial sample crafted using $f_c^{(i)}(f_{VAE}(x))$ does not work as effectively as it will on $f_c^{(j)}(f_{VAE}(x))$ when $i \neq j$, because $f_c^{(i)}$ and $f_c^{(j)}$ have different network structures and parameters, which affect the noise crafted by the adversary.

2.4 Combine VAE Reconstruction and Voting Result Using Bayesian Update

The image reconstructed by the VAE, $X^\sim$, is first fed into the original classifier for prediction, and this prediction is used as evidence e_A for later use in Bayesian update. Then the most likely class given by the collective voting process is used as the second evidence e_B in Bayesian update.

For general multi-evidence cases, we denote the marginal likelihood of evidence with $P(E_A = e_A)$, $P(E_B = e_B)$, ..., $P(E_M = e_K)$, where e_K is the Kth type of evidence we could find from the data. The posterior from the Bayesian update is calculated and is used as the final prediction for an unknown input image:

$$\begin{aligned}
&p(c_j|e_A, e_B, ..., e_M)\\
&= \frac{p(e_A|e_B, ..., e_M, c_j)...p(e_M|c_j)p(c_j)}{\sum_j p(e_A|e_B, ..., e_M, c_j)...p(e_M|c_j)p(c_j)}\\
&\underset{j}{\propto} p(e_A|e_B, ..., e_M, c_j)...p(e_M|c_j)p(c_j)
\end{aligned} \tag{5}$$

3 Experiments and Analysis

3.1 Experimental Setup

To evaluate the performance of the proposed defense mechanism and compare it with other existing methods, we conducted extensive experiments on two data sets, Fashion-MNIST [27] and CIFAR10 [2].

For adversarial attacks, Fast Gradient Sign Method (FGSM)[5], Basic Iterative Method (BIM) [13], Projected Gradient Descent (PGD) [19], Carlini and Wagner (CW) Method [3] were used in evaluating our method.

Various defense mechanisms have been proposed during recent years [1, 18, 21, 24, 22, 16, 4], and we compared our defense mechanism with two state-of-the-art methods that are more relevant to our work, **Defense-VAE** [15] and **Defense-GAN** [24].

The VAE's structure is shown in Table (1) in Appendix A. For post-VAE classifiers, we adopted three different types of networks (Residual-Networks [8], Wide-Residual-Networks [28] and DenseNet [10]). By adjusting the structure parameters, 4 different structures of each type were created (12 in total) and used as post-VAE classifiers for voting. The structures of all 12 post-VAE classifiers are shown in Table (3,4,5) in Appendix A and the implementation of the classifiers is from a public Github repository [14]. The Conditional Probability Tables (CPT) for $P(e_A|x^{\sim})$ and $P(e_B|e_A, x^{\sim})$ for Bayesian update were constructed by calculating the statistics using the entire training set.

3.2 Reconstructed Image Quality using SFL

Because reconstructed images are used for classification, we compare the reconstruction quality of VAEs that are trained with and without $\mathcal{L}_{SFL}$ in Figure (1). Under four different adversarial attacks and various amounts of perturbation, we observed that the images reconstructed from VAE trained with $\mathcal{L}_{SFL}$ have lower L2 norm difference from original clean images for all four adversarial attacks under wide range of perturbation levels, except when $\epsilon \geq 0.15$ from FGSM, which is also observed from a drop in the accuracy of post-VAE classifiers after the $\epsilon = 0.15$ under FGSM attack in Figure (2).

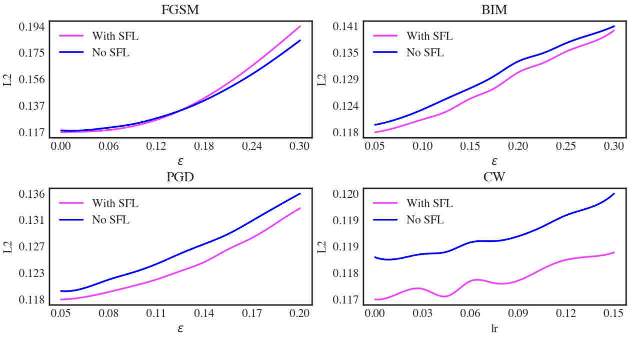

Figure 1: Comparison of VAEs' reconstruction quality of CIFAR10 data set (L2 norm between clean images and reconstructed images from different adversarial attacks and perturbation levels). The magenta line shows the reconstruction quality of VAE trained with $\mathcal{L}_{SFL}$, while the blue line shows that of without $\mathcal{L}_{SFL}$.

3.3 White-box Attacks on Classifiers

To test the proposed defense mechanism and compare the defense performance with other methods over a wide spectrum of adversarial perturbations, we tested the defense mechanisms against white-box attacks on three target classifiers (model A, B and C shown in Table (2) in Appendix A). Four different attacks were used: FGSM, CW, BIM and PGD at various perturbation levels, and the results are shown in Figure (2). For each test, 5000 adversarial samples were crafted using the testing data set and the accuracy for each perturbation level is the average accuracy after defense methods are implemented on classifiers A, B and C.

We observe from Figure (2) that the proposed method outperforms all the other methods by improving extra 25% accuracy on average across a wide range of adversarial perturbation levels for CIFAR10. On Fashion-MNIST, the proposed defense method gives better accuracy (a 5% margin on average) than other existing methods, and its accuracy on Fashion-MNIST remains above 93% under various adversarial attacks and perturbation levels. Figure (2) also shows that our defense method maintains about 91% accuracy on clean images for CIFAR10 and about 99% accuracy for Fashion-MNIST when there is no attack.

3.4 Attacks When Adversary Has An Access to Entire Defense Mechanism

We compared our method with other methods under the attacks on the entire defense framework. For Defense-GAN, we connected a trained generator to a well-trained classifier. By doing so, the input z for the generator can be tampered by an adversary who knows the entire framework $f_c(f_{gen}(z))$. For Defense-VAE, we connected a well-trained VAE to a classifier which has been trained using those reconstructed images from the same VAE. By doing so, the adversary can use the information $f_c(f_{VAE}(x))$ to launch a second wave of attacks. For the proposed method, each post-VAE classifier is connected to the well-trained VAE to form an end-to-end network, and the adversary uses this end-to-end network to craft adversarial samples. During the test time, those adversarial samples that are generated by different end-to-end networks are randomly selected and fed into the proposed defense framework.

Results in Figure (3) show that the proposed defense mechanism achieves an additional 10% better accuracy on CIFAR10 data set, and an additional 10% better accuracy on Fashion-MNIST data set than that of the existing methods on FGSM and CW. All three defense mechanisms' performance gets worse under this new attack, which means that attackers' knowledge about the entire defense framework affects its robustness.

We also observe that, in between the proposed method and Defense-GAN, although the adversary uses the similar information, $f_c(f_{VAE}(x))$, to generate attacks on both of them, the proposed method has a better performance. This extra gain in accuracy largely benefits from the use of collective voting results as explained in the next subsection.

3.5 Collective Voting

The voting of randomly selected post-VAE classifiers (here referred as VNN) gives extra increase in accuracy ($3\% \sim 5\%$) on the top of averaged individual VNN's performance as shown in Figure (2).

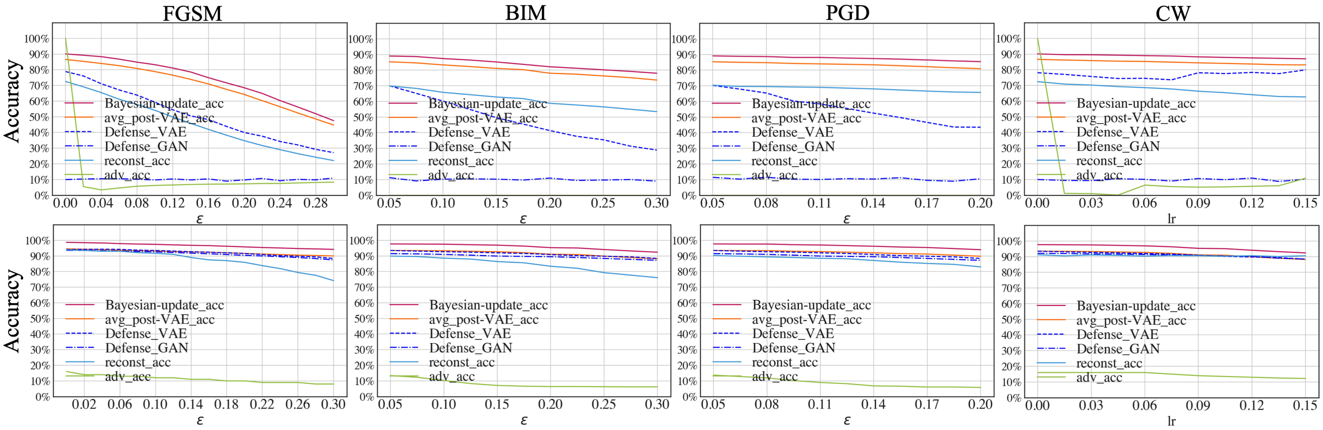

Figure 2: Defense methods comparison under FGSM, BIM, PGD and CW on CIFAR10(Top) and Fashion-MNIST(Bottom). The dash line is for Defense-VAE and dash-dot line is for Defense-GAN. The solid lines show the break-down performance of the proposed methods, including: average individual post-VAE classifier's accuracy(orange), original classifier's accuracy on reconstructed images(blue), accuracy after voting and Bayesian update(dark red) and the accuracy without defense(green).

To take a closer look at how voting benefits the classification, we calculated how much mis-prediction get recovered through voting process. Among each VNN, the total wrong prediction is $n_{wrong\_pred/vnn}$, which contains both recoverable predictions and non-recoverable predictions: $n_{can\_be\_recovered} + n_{not\_recovered}$. The recovery rate is: $\alpha_{recovery} = \frac{n_{can\_be\_recovered}}{n_{wrong\_pred/vnn}}$. Figure (4)(Left) plots the recovery rate along different numbers of VNNs used for voting. Different colors represents different perturbation levels of FGSM attack on CIFAR10 (we observed similar pattern for other attacks, too). We observed that as the number of VNN increases, less extra gain on recovery rate is received from each additional VNN used.

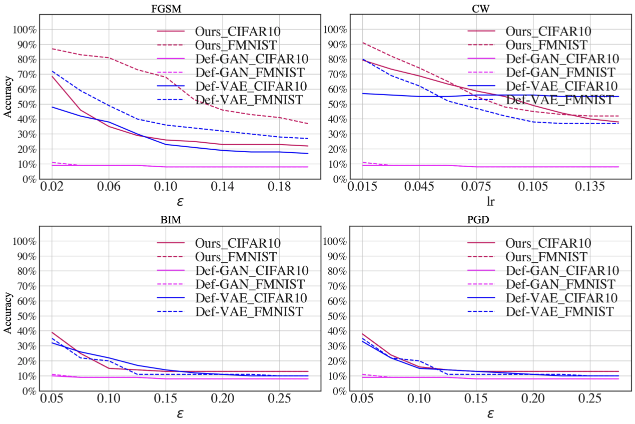

Figure 3: Different defense mechanisms' performance under 4 different adversarial attacks on the entire defense framework.

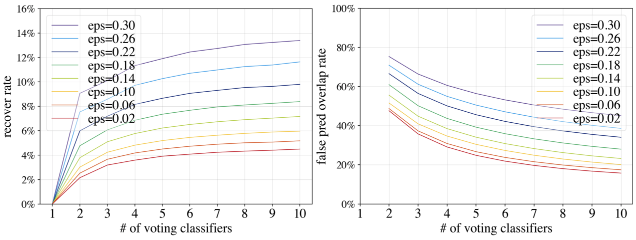

Figure 4: (Left) Recovery rate from voting. (Right) False prediction overlapping rate.

To explain this decreasing effect in extra gain, Figure (4) (Right) plots the VNNs' wrong prediction overlapping rate among VNNs. It shows that the more VNNs used, the less overlapping on their wrong predictions, which indicates that there are more variations in wrong prediction when more VNNs are used in prediction. This is also because with different structures and random initialization of parameters, each VNN forms a slightly different classification boundary. But the slope of overlapping rate is decreasing as the number of VNN increases, which explains why the recovery gain from using multiple VNNs is less prominent as more VNNs are used for voting.

4 Conclusion

An innovative adversarial defense mechanism was proposed, which introduces a process with randomization and discretization, by combining results from SFL VAE and post-VAE collective voting using Bayesian update. We conducted extensive experiments to compare its performance with two well-known existing defense mechanisms against 4 different adversarial attacks. The results show that our method outperforms by additional 25% increase in accuracy on CIFAR10 and by 5% on Fashion-MNIST. The proposed method maintains over 93% accuracy on Fashion-MNIST across a wide range of adversarial perturbation levels. For attacks on the entire framework, our method consistently outperforms the existing methods by about a 10% margin on accuracy for Fashion-MNIST and an average of 5% for CIFAR10.

References

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018. 1, 2

[2] Benchmarks.ai. Cifar-10 benchmarks, 2020. 2

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 2

[4] Partha Ghosh, Arpan Losalka, and Michael J Black. Resisting adversarial attacks using gaussian mixture variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 541–548, 2019. 2

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[6] Gaurav Goswami, Akshay Agarwal, Nalini Ratha, Richa Singh, and Mayank Vatsa. Detecting and mitigating adversarial perturbations for robust face recognition. *International Journal of Computer Vision*, 127(6-7):719–742, 2019. 1

[7] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3, 6

[9] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012. 1

[10] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. 3, 6

[11] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Suk thankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 1

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[13] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 2

[14] Wei Li. cifar-10-cnn: Play deep learning with cifar datasets. `https://github.com/BIGBALLON/cifar-10-cnn`, 2017. 3, 6

[15] Xiang Li and Shihao Ji. Defense-vae: A fast and accurate defense against adversarial attacks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 191–207. Springer, 2019. 1, 2

[16] Qi Liu, Tao Liu, Zihao Liu, Yanzhi Wang, Yier Jin, and Wujie Wen. Security analysis and enhancement of model compressed deep learning systems under adversarial attacks. In *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 721–726. IEEE, 2018. 2

[17] Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-bnn: Improved adversarial defense through robust bayesian neural network. In *International Conference on Learning Representations*, 2018. 2

[18] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 446–454, 2017. 1, 2

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2

[20] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980. 2

[21] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017. 1, 2

[22] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016. 1, 2

[23] Joseph Rocca. Understanding variational autoencoders (vaes), Sep 2019. 2

[24] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 1, 2

[25] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *CoRR*, abs/1805.06605, 2018. 1

[26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[27] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. 2

[28] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. 3, 6

5 Appendix A

5.1 VAE Architecture

The VAE's architecture used in the proposed defense method is shown in Table 1. Except the Decoder needs to be adjusted for the dimension difference between Fashion-MNIST and CIFAR10, the rest of the structure is same for the experiments on both data sets.

5.2 Classifier Structures

The classifier models that were used in comparison experiments between different defense mechanisms are shown in Table 2.

The post-VAE classifiers are created using the implementation from the Github repository [14]. Residual-Networks [8], Wide-Residual-Networks [28] and DenseNet [10] were used to create the post-VAE classifiers. Table 3, 4 and 5 show the parameters that were used to create the 12 different post-VAE classifiers (the rest of the parameters were set with their default values).

Encoder	Decoder
Conv(64,4,2)+BN+ReLU	Dense(4096)+ReLU
Dropout(0.2)	Reshape(4,4,256)
Conv(128,4,2)+BN+ReLU	ConvTrans(512,4,2)+ReLU
Dropout(0.2)	ConvTrans(256,4,2)+ReLU
Conv(256,4,2)+BN+ReLU	ConvTrans(128,4,2)+ReLU
Dropout(0.2)	ConvTrans(64,4,2)+ReLU
Conv(512,4,2)+BN+ReLU	ConvTrans(3,4,2)+ReLU
Dropout(0.2)	
Flatten	
Dense1(1024), Dense2(2014)	

Table 1: Network structure of VAE used for de-noising and image reconstruction.

A		B		C	
F-MNIST	CIFAR10	F-MNIST	CIFAR10	F-MNIST	CIFAR10
Conv(64,5,1)	Conv(32,3,1)	Dropout(0.2)	Conv(32,3,2)	Conv(128,3,1)	Conv(32,3,1)
ReLU	ELU,BN	Conv(64,8,2)	ReLU,BN	ReLU	ELU,BN
Conv(64,5,2)	Conv(32,3,1)	ReLU	Conv(32,3,2)	Conv(64,5,2)	Conv(32,3,1)
ReLU	ELU,BN	Conv(128,6,2)	ReLU,BN	ReLU	ELU,BN
Flatten	Pooling(2)	ReLU	Pooling(2)	Flatten	Pooling(2)
Dropout(0.25)	Dropout(0.2)	Conv(128,5,1)	Dropout(0.2)	Dropout(0.25)	Dropout(0.2)
Dense(128)	Conv(64,3,1)	ReLU	Conv(64,3,2)	Dense(0.25)	Conv(64,3,1)
ReLU	ELU,BN	Flatten	ReLU,BN	ReLU	ELU,BN
Dropout(0.5)	Conv(64,3,1)	Dropout(0.5)	Conv(64,3,2)	Dropout(0.5)	Conv(128,3,1)
Dense(10)	ELU,BN	Dense(10)	ReLU,BN	Dense(10)	ELU,BN
Softmax	Pooling(2)	Softmax	Pooling(2)	Softmax	Pooling(2)
	Dropout(0.2)		Dropout(0.2)		Dropout(0.2)
	Conv(128,3,1)		Conv(128,3,1)		Dense(10)
	ELU,BN		ReLU,BN		Softmax
	Conv(128,3,1)		Conv(128,3,1)		
	ELU,BN		ReLU,BN		
	Pooling(2)		Pooling(2)		
	Dropout(0.2)		Dropout(0.2)		
	Dense(10)		Dense(10)		
	Softmax		Softmax		

Table 2: Detailed network structures of different classifiers used for white-box attacks.

Params	stack_n	epochs	batch_size
ResNet A	3	50	128
ResNet B	5	50	128
ResNet C	10	100	128
ResNet D	18	100	128

Table 3: Four different ResNet architectures used throughout the experiments.

Params	depth	growth_rate	epochs	batch_size
DenseNet A	50	12	100	128
DenseNet B	50	24	100	128
DenseNet C	100	12	100	128
DenseNet D	100	24	80	256

Table 4: Four different DenseNet architectures used throughout the experiments.

Params	depth	wide	epochs	batch_size
WResNet A	8	8	80	128
WResNet B	16	8	80	128
WResNet C	16	10	80	256
WResNet D	28	10	80	256

Table 5: Four different Wide-ResNet architectures used throughout the experiments.