U-Autoencoder: Robust Defence Breakthrough on Adversarial Attack

Juneau Jung†, Geun Hyeong Ham, Woo Seok Song, Seung Chan Moon
Department of ECE, Seoul National University, Korea
{sean2ie, khchvic, cody1129, trinity0309}@snu.ac.kr

Abstract

We present a robust defense model in image classification against any kind of adversarial attack. Our model uses a U-Net structure superinduced on the Autoencoder followed by the general ResNet architecture. By fusing the identity mapping between the encoder and decoder, contextual information reserved from the fine-grained model and coarse model from the decoder makes an entangled robust architecture to deal with ambiguous images. We find that elaborately amalgamating those structures can make the model capable of coping with various adversarial attacks by proposing U-Autoencoder Network. Moreover, this model is also talented in identifying real-world images contaminated by noise or even feculent images with low resolution.

1. Introduction

We address the problem of defending against data poisoning to drop the performance of the deep learning model, commonly referred to as adversarial attacks. [6]

The adversarial attacks work by adding unrecognizable minor perturbations in the input data, causing results in completely different predictions. This performance degradation is fatal in fields directly related to human life, such as autonomous driving and medical imaging based on deep learning. Adversarial attacks are the biggest factor that shakes the trust of deep learning thus, devising ways to overcome those attacks is one of the most significant problems deep learning faces.

Currently, methods to defend against such adversarial attacks are being researched and ways to modify training methods, or input data are mainly proposed, noting the characteristics of attacks that poison data. Brute-force adversarial training by Zhang et al. [1] takes contaminated inputs to learn and Data compression by Jia et al. [2] uses classical machine learning methods like PCA to initially compress the data, and then train the model.

However, deviating from the mathematically designed model of deep learning and training with new methods that avoid accumulated knowledge is often applied only to specific datasets or lose their versatility.

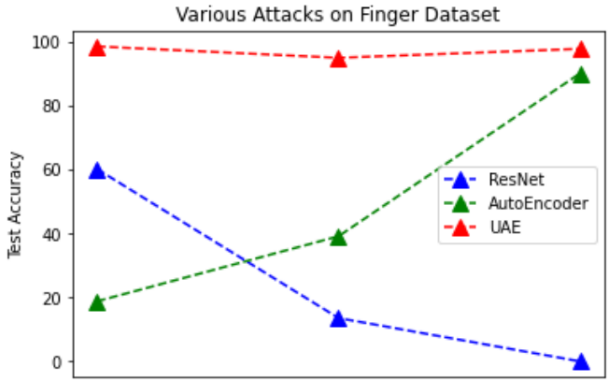

Figure 1: Our U-Autoender performs better than naïve ResNet-18 models and Autoencoders on adversarial attacks. The Finger dataset [27] is used as a benchmark, and the attacks are Gaussian Noise, FGSM, and PGD from left to right.

While other researches focus on manipulating the input of the training data. Luo et al. [3] have demonstrated foveation on CNN to modify the domain on convolution to dodge attacking area on the image and Xie et al. [4] implement random resizing on adversarial examples to avoid the effect of perturbation attacks on the inputs.

Nevertheless, these methods of transforming input not only reduce the expressive power of latent features of input data but also significantly cut off the expressive power of deep learning in response to random attacks.

The above methods suggest a model whose performance has been degraded by avoiding adversarial attacks. On the contrary, methods that can respond to attacks while maintaining performance by strengthening the robustness of the model itself have been recently studied. Gu et al. [5] give rise to the smoothness penalty on contractive networks for supplementation purposes. Similarly, Vivek et al. [8] suggested gradient regularization to the input to penalize variation on perturbed pixels in training.

While those contractive networks have successfully introduced autoencoder networks before the deep neural network to deal with L-BGFS-based attacks [10], we find limitations in two aspects: first, the research could not go further to handle other kinds of perturbation; second,

without adjusting penalties, there is no clear reason why the model becomes robust.

In this work, we propose a new model to practically derive the robustness of the model and apprehend why this model is regarded as powerful.

Context We utilize enhancing the robustness of the encoder network over a variety of image classification datasets. This model even applies to some delicately data-driven trained adversarial attacks, which makes it difficult to train in naïve ways. Our U-Autoencoder model has versatile utility on any method of attack.

Convergence We suggest a way to enhance prediction accuracy in image classification by using the U-Autoencoder model. As image classification tasks are widely used in real-world problems, we propose a network structure for efficiently resolving the currently facing problems with no additional time complexities.

Contribution In summary, in this work, we propose a robust and efficient image classification model to not only cope with various adversarial attacks but also denoising corrupted images. This is possible due to using an autoencoder to figure out the latent variables of the image while identifying the original image from the attack. Our method is relatively robust and, therefore, more accurate than the state-of-the-art methods as illustrated in Figure 1.

2. Related Work

Autoencoder is the representative state-of-the-art method in deep learning-based adversarial defense models. So let us compare it and analyze it with our proposed method.

2.1. Convolutional Network for Image Classification

Convolutional Neural Network(CNN) in image classification has made tremendous progress every year since AlexNet challenged other classical computer vision tasks with this structure in ImageNet Classification [11, 12] in 2012. With an 8-Layer CNN structure, this model became state-of-the-art for a couple of years.

Szegedy et al. [13] proposed GoogLeNet and Simonyan et al. [14] gave rise to VGGNet that have far deeper CNN structures compared to AlexNet. They got the first and second ranks on ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014.

After a year, He et al. [15] suggested ResNet which took the state-of-the-art from GoogLeNet. They used identity mapping on the deep CNN structure which was previously problematic due to gradient vanishing and other miscellaneous reasons. This ResNet architecture was considered the completion of the CNN-based Image Classification, which remained in a state-of-the-art position for quite a long time before the computationally extreme pre-trained model, Transformer, took control of image classification.

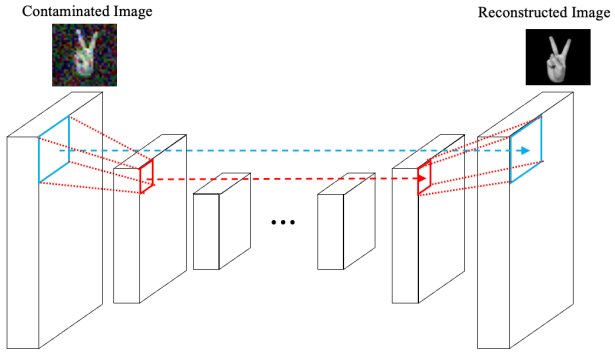

Figure 2: Our Network Structure. We diminish the spaial dimension of the input data using encoder layers (convolutional and non-linearlity). Then reconstruct the data by expanding the shrunk image that contains latent variables with decoder layers. The necessary point is we implement identity mapping.

2.2. Autoencoder

Autoencoder is one of the exciting unsupervised learning methods which is considered the deep neural network version of Principle Component Analysis(PCA), mainly used in machine learning to reduce dimensionality [16, 17]. This structure is divided into two parts: an encoder and a decoder, in which the encoder reduces the dimension of the data and the decoder extends the reduced data again.

Later, Vincent et al. and Nishad [19, 20] found that this structure can be useful in denoising images. This is also good at the compression of data as they can store the coarse representation of the original data. Besides, It is also applied in clustering algorithms that classify data without labels which was previously an area of machine learning tasks [18].

Although this autoencoder model performs great in various fields including machine learning, there are weak points to the autoencoders. It only stores the potential distribution derived from the input, and cannot control this variable, which is inevitable because it is a deep neural network designed for dimensionality reduction.

2.3. U-Net

U-Net is a network that is showing very remarkable performance through simple ideas in deep learning. This structure, designed primarily to preserve the positional information of an image between decreasing coarse images in image segmentation tasks [7], is repeatedly used in dimension reduction stages, similar to ResNet.

Previously, deep learning models in image classification could be used to confirm the presence of an object, but due to the reduced dimensional nature of CNNs, their location information was difficult to preserve. However, the creation of this model has brought innovative discoveries in bio-imaging technology and has established

itself as the most important model in reconstruction and various segmentation fields such as CT and MRI [21].

3. Proposed Method

3.1. Proposed Network

For the purpose of defending against adversarial attacks, we use autoencoders with U-Net inspired by Hinton and Ronneberger [7, 16]. The configuration is emphasized in Figure 2. We use *d* layers for the encoder and the same number of layers in the decoder. We also implemented identity mapping from the encoder layers to the decoder layers.

For example, for the input image size 32 x 32, we first shrink the special dimension of the image into 8 x 8, thus we can obtain some latent vectors of the input image. Then we use decoder layers to extend the spatial dimensions. With the U-Net-based identity mapping structure, the output image is the sum of the encoder input and the decoder output.

3.2. Training

Now we describe the objective of the training method to maintain the robustness of the U-Autoencoder model. As the purpose of our network is to enhance the robustness of the network model itself, we only input the original image into the model illustrated in Figure 3. Without using the corrupted samples of adversarial attack, this model is fairly robust to pull over the corruption stained with the original images.

Identity AE Learning In a naïve autoencoder, the input image merely goes through the encoder and decoder network, but it is widely known that this structure is prone to adversarial attacks but L-BGFS type perturbation [10]. Rather, we superinduce identity mapping between one layer of the encoder to the output of equal spatial size of the decoder with the identity mapping. We refer to this original structure as Identity AE Learning; AE trivially stands for the AutoEncoder.

Given a training dataset $\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}^{N}_{i=1}$ our goal is to learn the original image that can deal with corrupted images. The output of the autoencoder is described as $D_{\phi}\ (E_{\theta}\ (\boldsymbol{X_i}))$ where D_{ϕ} and E_{θ} refers to the network of the decoder and encoder respectively. Our identity autoencoder in the model can be referred to as $D_{\phi,k}\ (E_{\theta,k}\ (\boldsymbol{X_{i,k'}}) + \boldsymbol{X_{i,k'}})$. The sub k identifies the order of the layer of each layer, i.e. the result in k-*th* encoder layer is identically exerted into the k-*th* decoder. This account takes for granted only when the spatial structure between the encoder and the decoder are symmetrical.

Since the deep learning model optimizes the variable of the model through the loss function, the corresponding loss function is also necessary when the structure of the model changes. In order to simply this change of loss function, let us put the naïve autoencoder loss as $\| X_i - D_{\phi}(E_{\theta}(\boldsymbol{X_i})) \|$. This is because the purpose of autoencoder, especially decoder, is to reconstruct the original image even if it had been spatially destructed by the encoder. Then the identity autoencoder model gives the loss in a similar manner as $\| X_i + D_{\phi,k}(X_{i,k'}) - D_{\phi}, k(E_{\theta,k}(\boldsymbol{X_{i,k'}}) + X_{i,k'}) \|$. It seems a bit complicated at a glance, however the implantation is pytorch is rather simple.

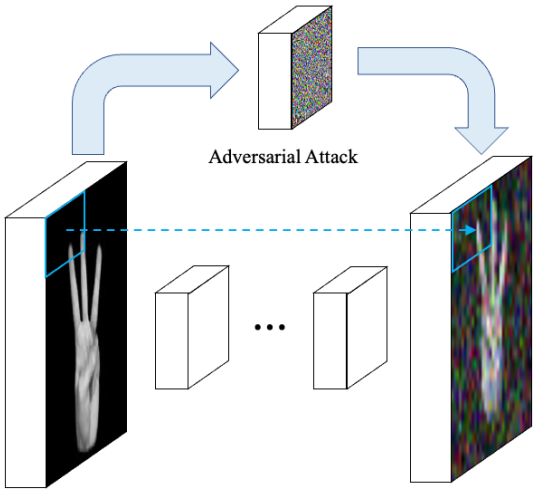

Figure 3: Training Method for U-Autoencoder Network Structure. We first train the original image through the network. Then, we give an adversarial attack to the image which outputs a corrupted image. Our network is robust enough to clean up the filthy perturb and identify the virtually original image.

Adversarial Attacks To clarify this model works good on the adversarial attacks, we contaminate the image or launched an adversarial attack by applying three perturbation: Gaussian Noise, FGSM, and PGD.

Gaussian noise is a very basic noise covered in electrical engineering [22] as they turn up when transmitting the image. When we receive such images through a network that requires signal processing, it is difficult to avoid the noise from being attached to the data. This is the most frequently generated noise when a camera or a data collector, such as a road view cam, uses it to transmit and receive or compress data, and the method of removing Gaussian noise is the most impactful study in engineering.

In fact, a naïve autoencoder has a not-bad performance of denoising against gaussian noise, however, it has an even bad prediction ability when the dataset is not a continuous image. An example of an un-continuous dataset is the 'Traffic-sign' dataset we used as a benchmark, which

we will discuss later.

Unlike the naturally made noise, there are also data-driven noises to purposely attack such neural networks. Goodfellow et al. Fast Gradient Sign Method (FGSM) suggested by Goodfellow et al. [6] to disturb neural networks being too linear. It is, namely, designed to give a pinch of the 'linear' network that is too weak against the attack.

A more powerful attack mechanism Projected Gradient Descent (PGD) is suggested by Madry et al. [23] that implements n-steps of applied FSGM. Currently, this method is known as a universal first-order adversary, and tons of research utilizes it as a baseline attack mechanism.

We show that our proposed model is robust compared to existing structures for the three representative methods of the above-mentioned adversarial attacks.

Datasets We used three 32 x 32 x 3 light-weighted datasets as a benchmark of the model. But, to our knowledge, the spatial scale of the image does not affect the performance much because it is known that any simple model structure can be expanded to large size model as input-256-ResNet is modified to input-28-ResNet to apply in the MNIST dataset, and vice versa. In the same manner, ResNet is applied to 4 K-resolution images with SRCNN [24]. Therefore, although we used a light-sized dataset, the overall structure is simple and thus can be applied to large datasets without complication.

Each of the three datasets is Typeface MNIST(TMNIST) provided by Magre et al. [9, 25], the Fingers dataset offered by Davis et al. [26, 27], and the Traffic-sign prediction dataset [28].

Inference In contrast to the training process using only the original image, the inference process measures the performance of the above adversarial attack on the contaminated data applied to the original image. As shown in Figure 3, the prediction accuracy is measured through whether the image subjected to adversarial attacks on three channels of RGB in the original image is received as input and classified with the same value compared with the labels.

Code Description Our code implementation is available in ItDL_2022_Project_2.zip file submitted via SNU eTL.

We used Add_Gaussian_Noise function to provide gaussian noise, fgsm function for the FGSM adversarial attack, and pgd_attack function for PGD attacks.

ResNet-18 network for the workflow using basicblock structure for 18 layers. UEncoder function is implemented as a total of 4 layers of an autoencoder, and also the important identity mapping.

We then check the visualization by printing images in the dataset to see if the proper noise or perturbation is added to the inference image and finally measure the accuracy. We also saved the pre-trained model and datasets we used, so the reviewers can easily check whether our overall training and inference process is done properly.

Models	ResNet	Autoencoder + ResNet	U-Autoencoder + ResNet(ours)
Without Noise	99.35	99.26	99.21
Gaussian (σ = 0.225)	74.28	87.04	91.44
FGSM (ε = 0.2)	26.39	92.53	93.80
PGD (ρ= 0.1)	0.00	98.14	98.26

(a) Prediction accuracy in TMNIST dataset

Models	ResNet	Autoencoder + ResNet	U-Autoencoder + ResNet(ours)
Without Noise	100		
Gaussian (σ = 0.125)	60.19	18.78	98.53
FGSM (ε = 0.1)	13.56	39.14	94.97
PGD (ρ= 0.1)	0.00	90.11	97.78

(b) Prediction accuracy in Finger dataset

Models	ResNet	Autoencoder + ResNet	U-Autoencoder + ResNet(ours)
Without Noise	70.65	96.43	96.53
Gaussian (σ=0.06)	85.86	89.13	91.17
FGSM (ε = 0.01)	41.04	88.67	91.83
PGD (ρ= 0.1)	0.00	60.85	67.99

(c) Prediction accuracy in Traffic-sign dataset

Table 1: Performance of U-Autoencoder network compared to naïve ResNet architecture and autoencoder in terms of prediction accuracy in image classification.

4. Experimental Results

ResNet perfectly classifies noise-free images, but its performance drops to 60% in classifying images containing Gaussian noise, and it rarely classifies data when data-driven attacks such as FSGM and PGD are applied. While Autoencoder has considerably more ability to defend against attacks than naïve ResNet but shows far less performance to apply to the real-world problem.

Our proposal U-Autoencoder model has a compliant defense performance of both Gaussian noise, FGSM, and PGD, in contrast to existing models' inability to cope with attacks. Even in the case of the Fingers dataset, where the performance of the autoencoder is rather poor, our model complements its weakness very well and shows performance in the 90% range.

5. Conclusion

In this work, we presented a robust defense model against adversarial attacks with a U-Net structure attached to the autoencoder following the ResNet network. We have demonstrated that our proposed model outperforms the existing methods in image classification problems by a large gap in benchmark performances. The model is robust against not only adversarial attacks but also Gaussian noises likely to face in the real world. We believe our approach is ad hoc and applicable to other image processes using a neural network that needs to deal with noises and adversarial attacks.

6. Contribution

Juneau Jung: Team workspace environment setting, Problem raise, research direction, ResNet and Autoencoder backbone implementation, paper writer.

Geun Hyeong Ham: Gaussian Noise augmentation, Dataset Determination.

Woo Seok Song: FGSM adversarial attack augmentation, Dataset Determination.

Seung Chan Moon: PGD adversarial attack augmentation.

References

[1] Sicong Zhang, Xiaoyao Xie, and Yang Xu. A Brute-Force Black-Box Method to Attack Machine Learning-Based Systems in Cybersecurity. In *IEEE Xplore*, 2020

[2] Xiaojun Jia, and Xiaochun Cao. ComDefend: An Efficient Image Compression Model to Defend Adversarial Examples. *In CVPR,* 2019.

[3] Yan Luo, Xavier Boix, and Gemma Roig. Foveation-based Mechanisms Alleviate Adversarial Examples. In *CBMM Memo No.44,* 2016

[4] Cihang Xie, Jianyu Wang, and Zhishuai Zhang. Mitigating Adversarial Effects Through Randomization, In *ICLR*, 2018

[5] Shixiang Gu, and Luca Rigazio, Towards Deep Neural Network Architectures Robust to Adversarial Examples, In IEEE, 2017

[6] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *ICLR* , 2015

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI* ,2015

[8] Vivek B S, Arya Babura, and R. Venkatesh Babu. Regularizer to Mitigate Gradient Masking Effect During Single-Step Adversarial Training, In *CVPR*, 2019

[9] Nimish Magre, and Nicholas Brown. Typography-MNIST (TMNIST): an MNIST-Style Image Dataset to Categorize Glyphs and Font-Styles. In *arXiv*:2202.08112, 2022

[10] Dong C. Liu, and Jorge Nocedal. On the limited memory BFGS method for large scale optimization, In *Mathematical Programming*, 1989

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. in *NIPS*, 2012

[12] Jia Deng, Wei Dong, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE*, 2009

[13] Christian Szegedy, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE*, 2014

[14] Karen Simonyan, and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015

[15] Kaiming He, and Jian Sun. Deep Residual Learning for Image Recognition, In *CVPR*, 2015

[16] G. E. Hinton, and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. In *Science*, 2006

[17] Jonathon Shlens. A Tutorial on Principal Component Anlysis. In *The Royal Society*, 2016

[18] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016

[19] P. Vincent, H. Larochelle, I. Lajoie, Yoshua Bengio, and P. -A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. In *JMLR*, 2010

[20] G. Nishad. Reconstruct corrupted data using Denoising Autoencoder. In *Medium*, 2020

[21] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. In *IEEE TIP*, 2017

[22] Ajay K. Boyat, and Brijendra K. Joshi. Noise Models in Digital Image Processing. In *SIPIJ*, 2015

[23] Aleksander Madry, Aleksander Makelov, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks, In *ICLR*, 2018

[24] Chao Dong, Chen C. Loy, Kaiming He, and Xiaoou Tang. Image Super-Resolution Using Deep Convolutional Networks. In *IEEE*, 2014

[25] TMNIST (Typeface MNIST) https://www.kaggle.com/datasets/7a2a5621ee8c66c1aba046f9810a79aa27aafdbbe5d6a475b861d2ba8552d1fc

[26] Sergio Davis, Alexander Lucas, Carlos Ricolfe-Viala, and Alessandro D. Nuovo. A Datatbase for Learning Numbers by Visual Finger Recognition in Developmental Neuro-Robotics, In *Frontiers*, 2021

[27] Fingers Dataset, https://www.kaggle.com/datasets/koryakinp/fingers

[28] Traffic-sign Prediction Dataset, https://www.kaggle.com/datasets/fedesoriano/traffic-prediction-dataset