



A Defense and Detection Against Adversarial Attack Using Denoising Autoencoder and Super Resolution GAN

By –
Subroto Karmokar
ID: 163-35-1734

A thesis submitted in partial fulfillment of the requirement for the degree
of Bachelor of Science in Software Engineering

Department of Software Engineering
Daffodil International University

Summer-2020
Copyright © 2020 by Daffodil International University

APPROVAL

This thesis titled on “**A Defense and Detection Against Adversarial Attack Using Denoising Autoencoder and Super Resolution GAN**”, submitted by **Subroto Karmokar (Student ID: 163-35-1734)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



Dr. Imran Mahmud
Associate Professor and Head In-Charge
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman



Dr. Md. Asraf Ali
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1



Md. Anwar Hossen
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

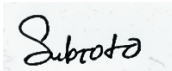


Prof. Dr. Mohammad Abul Kaashem
Professor
Department of Computer Science and Engineering
Dhaka University of Engineering and Technology, Gazipur

External Examiner

DECLARATION

I hereby declare that I have taken this thesis under the supervision of **Md. Maruf Hassan**, Assistant Professor, **Department of Software Engineering**, **Daffodil International University**. I also declare that neither whole document nor any part of this thesis have been submitted elsewhere for award of any degree.



.....
Subroto Karmokar
ID: 163-35-1734
Batch: 21st Batch
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Certified by:



.....
Md. Maruf Hassan
Assistant Professor
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

ACKNOWLEDGEMENT

At first I want to give thanks my almighty God. God gives me the ability to successfully received to the final year. I have learnt ethics, manners and so many things during my university life. I so thankful all my university teachers who are train me well. Without them I never learn all of these things. I want to give thanks my supervisor honorable Md. Maruf Hassan, Assistant Professor, DIU for giving me such a wonderful opportunity and continuously guided as well as supported me to continue this thesis on the topic of ‘A Defense and Detection Against Adversarial Attack Using Denoising Autoencoder and Super Resolution GAN’ which helped me in doing research and writing of this thesis paper. Therefore, I have learnt so many new things from my supervisor which help me for higher study. Beside my supervisor, I would like to thanks my senior brothers and sisters who give support mentally. Most importantly, I thank to my parents, they are always inspiring me.

TABLE OF CONTENTS

DECLARATION	i
ACKNOWLEDGEMENT.....	ii
TABLE OF CONTENTS	iii
TABLE OF FIGURE	iv
ABSTRACT	v
CHAPTER 1.....	1
INTRODUCTION.....	1
1.1 Background	1
1.2 Motivation of the Research	2
1.3 Problem Statement	3
1.4 Research Questions	3
1.5 Research Objectives	3
1.6 Research Scope.....	4
1.7 Thesis Organization.....	5
CHAPTER 2.....	6
LITERATURE REVIEW	7
2.1 Modifying Data	5
2.2 Modifying Model	7
2.3 Using Auxiliary Tool.....	9
CHAPTER 3.....	11
RESEARCH METHODOLOGY	11
3.1 Detector layer	12
3.2 Denoiser layer.....	13
3.3 Super resolution GAN	14
3.4 Main Classifier	16
3.5 Attack generator	18
3.6 Algorithm	19
3.7 Implementation.....	23
CHAPTER 4.....	25
RESULT AND DISCUSSION.....	25
4.1 Discussion	25
4.2 Details of Effectiveness	26
4.3 Performance comparision	28
CHAPTER 5.....	30
CONCLUSION AND FUTURE RECOMMENDATION	30
5.1 Findings and Contributions	30
5.2 Recommendations for Future Works.....	30
REFERENCES.....	31

TABLE OF FIGURE

Figure 1: System Architecture of ADDA-Adv. tool.....	12
Figure 2: MobileNetV2 Workflow	17
Figure 3: MobileNetV2 Architecture	17
Figure 4: Algorithm of ADDA-Adv.....	22
Figure 5: Test Application Details	24
Figure 6: The Impact hyper parameter	25
Figure 7: Effectiveness of ADDA-Adv.....	26
Figure 8: Effectiveness Of Noise Image Detection	28
Figure 8: Performace matrix.....	29

ABSTRACT

Neural network usages are being popular in different sectors that make life easier and automated; so that security of neural network is a big concern. Against adversarial attack every Deep Neural Network are vulnerable. Adversarial attack is technique to specially design an image sample with adversarial noise. Several number of adversarial attack technique are found in recent research, which are fool neural network with high misclassify accuracy. There is also various defense mechanism are proposed and build with Deep Neural Network to defend and increase robustness of the main classifier neural network model. However, there are very few model can work with high resolution image data and work with pre-trained neural network classifier. The main objective of this research was to proposed and developed a model which can integrate with any existing trained Neural network and more generic defense against adversarial attack tool called ADDA-Adv. Based on proposed model detect highly distorted image and reject those sample to avoid misclassification. Additionally, this work is intended to work with high resolution adversarial image sample. ADDA-Adv. tool restore the adversarial sample and provide 89.23 percent accuracy.

Keywords: Adversarial Attack; Defense and detect tool; Deep Neural Network; Computer Vision vulnerability.

CHAPTER 1

INTRODUCTION

Adversarial attack is a technique to fool the neural network and misguided the classifier to the wrong direction. An adversarial sample is an image that has been slightly modified and that is almost identical to the original image which is why it is nearly undetectable to the human eye. There are several methods out there to generate adversarial example which can successfully misclassify any neural network with high confidence level. This kind of adversarial image can fool the neural network without prior knowledge of neural network. Deep neural network is vulnerable to the adversarial attack. Successful adversarial attack could lead to serious security breaches for both the organization and the user.

1.1 Background

Over the years, artificial intelligent has become so popular in the many business and organizations. This is most cutting edge technology in computer science field. Most importantly, AI based solution is most effective and efficient. Specially, Deep Neural Network (DNNs) incontestable wonderful performance within the field of image classification, object detection, speech recognition etc. Based on this solution many computer vision task are highly dependent like bio metric security, self-driving car, medical science, crime investigations. Statista says computer vision market worth USD 48 Billion by 2023. (<https://www.globenewswire.com/news-release/2019/04/09/1799533/0/en/Computer-Vision-Market-Worth-USD-48-Billion-by-2023-The-Emergence-of-Computers-is-Intensifying-Global-Computer-Vision-Market-to-Prosper-with-Major-Developments-in-Virtual-Reality.html>)

But DNN security is still a big issue. For example, change the input slightly and add some noise that made the DNN vulnerable. So, crafted an input sample in a special way to generate a wrong answer from classifier model is called adversarial attack. We can divide adversarial sample in two ways. First one is access the network and the parameters, which called white-box attack. Second one is no knowledge to the neural network architecture but read input and output of neural network, which is called black-box attack.

1.2 Motivation of the Research

Nowadays many organizations extensively depend on computer vision task such as facial recognition security system, self-driving car, traffic control system, medical science, crime investigation. So, the attacker generates adversarial sample to misguide the neural network. Because of this vulnerability many user and organization face serious security breaches. To defend these types of attacks, several defense model are developed with different features.

1.3 Problem Statement

After analyzing the previous research works about adversarial attack detection, no research methodology has work with white-box, gray-box and black-box setting at a time. And also observed that that defense system has performed very poor in iterative strong attack. It is also being noted that they failed defend unknown attack. Most of the defense system are overhead to training model.

1.4 Research Questions

With having this background, motivation and problem statement in mind, I propose the following questions:

- Is our introduced model adversarial defense can effectively defend different kind of adversarial samples?
- Is our implemented tool providing better accuracy as compared to the existing solution?
- Is our model more generic compare others?

1.5 Research Objectives

- To propose a model that perfectly work with pre-trained model. Not overhead to trained model.
- To prepare generic model which can defeat newly generate attack.
- To analysis sample is Real or Fake.
- To recover the sample, whether it is adversarial sample.
- To implement a detector model that can reject highly perturbed image.

1.6 Research Scope

The experimental process has been done in the Convolutional Neural Network (CNN) model using Python, Tensorflow & Keras. Use that technique in different application area like Object Detection, Self-Driving Car, Traffic Control, Bio Metric Security, Crime Branch, Medical Science etc. Our model can easily integrate with any existing

Convolutional Neural Network classifier model. In this case we do not need to developed our own model to conduct out test plan.

1.7 Thesis Organization

In this research, APA referencing technique has been used through this document. The paper has been completed with six chapters which are described below:

Chapter 1: Research background, motivation, problem statement, objectives and scopes are conferred in this chapter.

Chapter 2: This chapter includes discussion of the existing related works and figured out the research gap.

Chapter 3: This chapter contains the research methodology and approaches as it follows for the research.

Chapter 4: Result analysis and evaluation are discussed here.

Chapter 5: The outcome of research and direction of further work is presented here.

CHAPTER 2

LITERATURE REVIEW

This section describes the previous research and findings of Adversarial vulnerabilities / attacks which is focused on CNN classifier model. To do our work cycle in the simplest way we have been split our research process into some particular portion for better understanding.

2.1 Modifying Data

Modifying data is technique which is modify and changing training dataset at stage of training or change the input data at stage of testing. There are different of technique like gradient hiding, data compression, adversarial training, data randomization, transferability blocking etc.

2.1.1 Adversarial Training

Introduced adversarial sample into the training data is increase the robustness of the model and train the model with the adversarial samples. [Szegedy et al.] First of all include the adversarial sample and change label to make the classifier model more robust against adversarial attack. [Goodfellow et al.] reduce misclassification rate of MNIST dataset from 89.4% to 17.9% use of adversarial training. [Huang et al.] improve the robustness of classifier model against adversarial attack by punishing adversarial data. [Tramèr et al.] proposed ensemble adversarial training, that improve the variety of adversarial sample. However, this is impossible to include all unknown attack sample as adversarial training. Which is main limitation of this technique.

2.1.2 Gradient Hiding

A defense against attack like gradient based proposed by [Tramèr et al.] and attack used by adversarial generated method like FGSM. This method generally hides information like gradient. If a classifier model is non differentiable gradient base attack will not work. However, by learning the proxy black-box model with gradient and using the adversarial samples which is generated from this model [Papernot, N et al.], the tactic will simply be fooled during this case.

2.1.3 Data Compression

The JPG compression method found by [Dziugaite et al.]. This method can improve large number of network model recognition accuracy declined in FGSM attack. [Das et al.] also used the same JPEG compression technique to defense against DeepFool and FGSM. But those methods are not effective solution against strong attack such as Carlini & Wagner attacks [Carlini, N.; Wagner, D.]. On the other hand, Display Compression Technology (DCT) technique [Das, N. et al.] used to fight against universal attack [Akhtar, N. at el.] is also ineffective. The main limitation is data large amount of data compression can make the classifier less accuracy and small amount can't remove the adversarial signature.

2.1.4 Data Randomization

The random resizing adversarial sample can reduce the effective of adversarial attack which introduced by [Xie et al.]. On the hand, add some random textures to the adversarial sample increase the effectiveness to the classifier model. [Wang et al.] used a data conversion module separated for defense against possible adversarial attack and conduct

data expansion during the training phase. Such as adding some Gaussian randomization processing which increase the robustness of classifier model against adversarial attack.

2.1.5 Blocking the Transferability

The transferability can hold attribute, if the classifier model has different architecture. The main idea to protecting from black-box attack is to prevent transferability of adversaries. [Hosseini et al.] proposed three step Null labeling method. To protect the adversarial samples from one network to different network. The main idea behind this technique is add a new Null label to dataset and classify all of them to Null label by trained classifier model to resist adversarial attack. This technique includes three steps (i) initial training target classifier, (ii) calculate Null probabilities, (iii) adversarial training. The advantage of this methodology is marking the perturbation input as empty label instead of classifying it the original label. This method is effective against adversarial attack without effecting classification accuracy.

2.2 Modifying Model

Modify and change neural network is another technique to defend against adversarial attack. Such as defensive distillation, regularization, feature squeezing, mask defense and deep contractive.

2.2.1 Defensive Distillation

A defense distillation method is proposed by [Papernot et al.] which is defend the attack based on distillation technology [Hinton, G. et al.]. The distillation method compresses the large-scale model into small-scale. Most importantly, it retains original accuracy to the model. On the other hand, defensive distillation does not modify the scale of model. This

method produces smoother output surface and increase the robustness of the model against attack. Author of paper claim that defensive distillation reduces the adversarial attack rate by 90%. However, this defense technique is not effective against black-attack.

2.2.2 Regularization

This method adds regular term to improve the generalization ability if the target which are known as penalty term to the cost function. It makes the model good resist attack against unknown dataset prediction. [Biggio et al.] used a regularization technique for SVM model during training to limit the vulnerability.

2.2.3 Feature Squeezing

Feature squeezing is enhancement the model [Xu, W el at.]. The main idea of Feature squeezing is reducing the complexity of the data representation. That's why reduce adversarial interference for low sensitivity. There are two methods available (i) reduce the color depth of the pixel, (ii) use smoot filter on image. thus creating the model safer under noise and resistance attack and as well as prevent adversarial attack. However, this technique reduces the accuracy for real sample.

2.2.4 Mask Defense

Mask defense layer is proposed by [Gao et al.]. In this technique adding a mask layer before the classification model. The mask layer trained the original image corresponding adversarial samples. After that encoded the distinguish between those image and output from the previous layer. It is typically believed that the foremost necessary weight within the additional layer corresponds to the foremost sensitive feature within the network. So, in the final classification all feature is masked by the additional layers with initial weight

zero. In this technique, the deviation of classification results caused by adversarial samples is secure.

2.2.5 Deep Contractive Network (DCN)

The deep compression network introduced by [Gu et al.], which reduce the adversarial noise using automatic encoder. This technique adopted a smoothing penalty kind of like a Convolutional Autoencoder (CAE) during the training process. This method effective against attack like L-BGS [Szegedy, et al.]

2.2.6 Parseval Networks

Parseval network is proposed by [Cisse et al.]. This defense network controls the global Lipschitz constant for adopting hierarchical regularization. Considering the network can be viewed as function combination at every layer. They parameterize the spectral norm of the network weight matrix for controlling the spectral norm of the network weight through Parseval frames [Kovačević, et al.]. That's why it was called Parseval Network.

2.3 Using Auxiliary Tool

Auxiliary tools are basically additional tools which is include in a neural network model. Such as Magnet, defense-GAN and high-level representation guided denoiser.

2.3.1 MagNet

MagNet method is proposed by [Meng et al.], which reads the output of the last layer of the classifier as a black-box while not reading any data of the inner layer or changing the classifier. MagNet use a detector layer which can identify whether sample is legal or adversarial. The detector measure and calculate the gap between given sample and manifold. If the sample exceeds threshold it will be rejected. This technique also use a

reformer to reconstruct an adversarial sample to legal sample. However, this method performance decrease in white-box attack scenario because knowing parameters of MagNet. Which is why author use multiple automatic encoder and randomly select one of them to make it difficult to predict which one used.

2.3.2 Defense-GAN

[Samangouei et al.] proposed a defense mechanism which is effective against white-box and black-box both attack scenario. This technique utilizes generative network power. The main idea of this tools is minimizing the constructor error $\|G(z) - x_2^2\|$ of generator G. Then feed the image to the classifier model. This method is effective against adversarial attack and success of the method depend on GAN. However, without proper training performance may decline.

2.3.3 High-Level Representation Guided Denoiser (HGD)

HGD is different from de-noising device like pixel level reconstruction loss function. In that method have problem to amplification error. On the other hand, HGD solve this problem by using a loss function to compare the output between clean image and denoising image. HGD method is proposed by [Liao et al.], which is design to robust target model against black-box and white-box both attack. Author proposed three HGD training method. Another advantage of HGD is that it is trained on a comparatively small dataset and can be used to protect models aside from the one guiding it.

CHAPTER 3

RESEARCH METHODOLOGY

The system architecture of ADDA-Adv. tools and the problems solved by the components are discussed here. Figure 3.1 demonstrates the architecture based on the components and the way they relate with each other. Basically, that tool is preprocessing technique before feed in a classifier neural network. Which is why that tools can be easily implement in any existing model. We divide our tool in different parts. Those are Detector, Denoiser Autoencoder [Bengio, Y. et al.] and Super Resolution Generative Adversarial Network (SRGSN) [Ledig, C. et al.] for make image sharper. At first we use a detector to identify legal and adversarial samples. The detector measures the distance between a given sample under test and the manifold and block the sample if the distance exceeds the threshold. So here that detector will make distinguish between legal and adversarial noisy sample. After that we will send the legal sample to the next layer. The second layer is Denoiser layer which will remove the sample as much as possible and most importantly retain the necessary features. But the problem is denoiser make the sample little bit blur or fuzzy which is why we send it to the next layer. The third layer is SRGAN which make high resolution and sharper. That will help to improve the overall quality of that image. Finally, that sample will send the main classifier. The working flow of each component are discussed below in more detail.

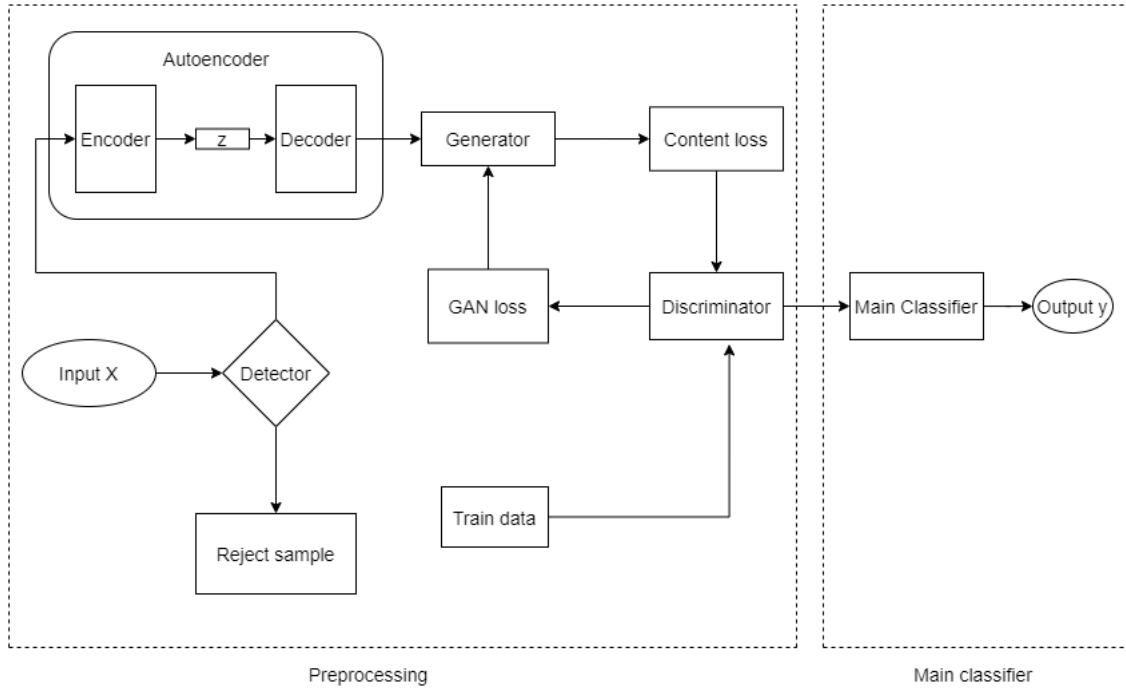


Figure 3.1: System Architecture of ADDA-Adv. tool

3.1 Detector Layer

In this tool, Adversarial noisy image make the main classifier fool in softmax layer. This is very important to detect those noisy image. To implement this detector layer we use “OpenCV” python3 library. Which is make the sample BGR to HSV. Then we take the value and calculate percentage of noise. We set threshold to identify the noisy sample. In that case this is worth to mention higher percentage is closer to legal sample. On the other hand, lower percentage is closer to noisy sample. Based on the thresholder this Detector layer will decide that sample will pass through the layer or rejected. This Detector layer will return a true or false value and percentage of noise pixel of that sample.

3.1.1 Legal image

The legal image means which is original data without any kind of noise or distortion. Which sample retain every necessary feature. Based on Detector layer it will decide whether it legal image or not.

3.1.2 Noisy image

The noisy image is distorted sample. Sometime noisy image which doesn't make any sense to human. But this kind image lead to neural network classifier to wrong direction. So based on Detector layer noisy image will be rejected.

3.2 Denoiser Layer

To Implement this layer, we use Denoising Autoencoder for removing the noise from a adversarial sample. The main idea of Autocoder is input and output is equal feature. This process doesn't reduce or loss number of feature. This is a unsupervised neural network learning feature. Neural network accept image as a input an compress the data. After compress the image reconstruct the sample and learn from ground truth value. There are many types of autoencoder available to do different task like anomaly detection, reformer same, denoise sample etc. We build a Denoiser Autoencoder for our tools. To build this we use tensorflow, keras and deep learning model. The more details of Denoising Autoencoder given below.

3.2.1 Encoder

The first layer of an Autoencoder is input layer. So, Autoencoder take sample as input and send it to the Encoder. In Encoder there couple of layer to compress the sample. So, the

main idea is reduce the pixel and feature. That's how it reduces the sample dimensions and represent the data into encoder formation.

3.2.2 Bottleneck

In any Autoencoder, bottleneck play a very vital role. The data after compression pass through this layer. The most importantly bottleneck layer contain all the compress data and this lowest possible dimension of data. In addition, completing the compression the data send to decoder.

3.2.3 Decoder

This is the last part of the Autoencoder neural network. From bottleneck layer take compressed data and start reconstruct it. During reconstruct the model learn how close to the original input. The whole process goes through a couple of layer which is connected to each other.

3.2.3 Reconstruction Loss

That's method which is measure and calculate the over loss from the whole Autoencoder process. And as well as how good the decoder layer performs in reconstruction process. Most importantly how close to the original input.

3.3 Super Resolution Generative Adversarial Network

When a sample pass through Denoising Autoencoder, the sample loss their clarity or sharpness. Autoencoder make that image little bit blur and fuzzy. Which make difficult classification job for the main classifier. Because of, losing some important feature. To solve the problem we use Super Resolution Generative Adversarial Network (SRGAN). Image super resolution is a technique to reconstruct a higher resolution image from a lower

resolution image. This process neural network observe the neighboring pixel and use a machine learning neural network to reproduce it. The conduct whole operation two main network here Discriminator Network and Generative Network. The main idea of GAN is Generative Network is produce fake image and send it to Discriminator Network. On the other hand Discriminator Network predict whether the sample is real and fake. And again send it Generator Network to reconstruct it. This process continue several time. Gradually, Generative Network improve the sample. This learning process called unsupervised learning.

3.3.1 Notation of GAN

- Pdata: real dataset
- Pz: noise distribution
- D,G: neural network
- D(x): probability of x come from data rather than Pg

3.3.2 Generative Network

In Super resolution GAN, Generative Network take random noise as input. And produce some output which is close to real sample. This network continuously try to fool Discriminator Network. After that Generative Network send it to Discriminator and received feedback. Based on those feedback Generative model update the model.

$$L_G = E_{z \sim p_z}[\log(1 - D(G(z)))]$$

During this process optimizing is that fix value of G to update parameters of D. After that fix the value of D to update parameters of G.

3.3.3 Discriminator Network

In this network data received from Generative Network and predict this whether sample is real or fake. Discriminator Network take input half generated data(fake) and half real data(real). It uses loss function

$$L_D = -E_{x \sim p_{data}}[\log D(x)] - E_{z \sim p_z}[\log(1 - D(G(z)))]$$

The whole process like function $V(G,D)$.

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}}[\log D(x)] - E_{z \sim p_z}[\log(1 - D(G(z)))]$$

3.4 Main Classifier

For the main image classifier, we use here is ‘MobileNetV2’. This is Convolutional Neural Network (CNN) which is very advanced level image classifier. On the other hand, the dataset we used here is ImageNet which is pretty large image 256*256. That ImageNet dataset contain very larger number of data more than 14 million. ‘MobileNetV2’ network architecture is below

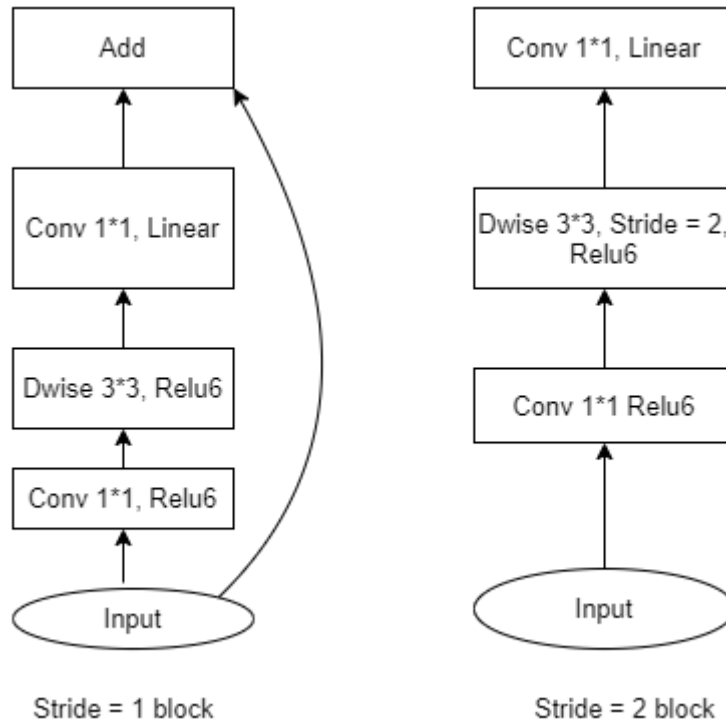


Figure 3.2: MobileNet v2 workflow

Table 3.0 MobileNet v2 architecture

Input	operator	t	c	n	s
$224^2 * 3$	Conv2d	-	32	1	2
$112^2 * 32$	bottleneck	1	16	1	1
$112^2 * 16$	bottleneck	6	24	2	2
$56^2 * 24$	bottleneck	6	32	3	2
$28^2 * 32$	bottleneck	6	64	4	2
$14^2 * 64$	bottleneck	6	96	3	1
$14^2 * 96$	bottleneck	6	160	3	2
$7^2 * 160$	bottleneck	6	320	1	1
$7^2 * 320$	Conv2d 1*1	-	1280	1	1
$7^2 * 1280$	Avgpool 7*7	-	-	1	-
$1 * 1 * 1280$	Conv2d 1*1	-	k	-	-

- Here, c : output channels number, t : expansion factor, n : repeating number, s : stride. 3×3 kernels for using spatial convolution.
- The primary network {width multiplier 1, 224×224 }, computational cost is 300 million multiply-adds. With the uses 3.4 million parameters.
- The network computational cost is up to 585M MAdds, while the model size varies between 1.7M and 6.9M parameters.

3.5 Attack Generator

Adversarial Attack are generated from different type of method. Some of technique are white box attack and some of are black box attack. The basic idea of adversarial attack is perturbation a sample image with some noise. That crafted image make the classifier fool. We used some of attack technique to test our adversarial defense tools.

3.5.1 Fast Gradient Sign Method (FGSM)

Fast Gradient Sign Method (FGSM) is proposed by [Goodfellow et al.]. This method calculate the gradient value of cost function based on the neural network input. The formula of this method is given below:

$$X' = X + \varepsilon * \text{sign}(\nabla_x J(X, y_{true}))$$

In this equation, ε is hyper-parameter. It controls the amplitude. J is the cost function. ∇_x is the gradient of a normal value X . Author claim that this method can misclassify the MNIST dataset 89.4%.

3.5.2 Basic Iterative Method (BIM)

This method is the iterative version of FGSM. The use multiple time of FGSM. The formula of this method is given below:

$$X'_0 = X, X'_{n+1} = \text{Clip}_{X,\epsilon} \{X'_n + \alpha * \text{sign}(\nabla X J X'_n, y_{true})\}$$

Here in this equation, $\text{Clip}_{X,\epsilon}\{A\}$ is element wise clipping of X. And α is the step size. On the other hand, J is the cost function. ∇x is the gradient of a normal value X . This methodology typically doesn't depend on the approximation of the model and produces further harmful adversarial samples once this algorithm runs for a lot of iterations.

3.5.3 Iterative Least-Likely Class Method (ILCM)

Iterative Least-Likely Class Method replace the class variable by using the with the small recognition probity in the distance. Which is get adversarial examples with 99% misclassify accuracy. The formula of this method id given below:

$$X'_0 = X, X'_{n+1} = \text{Clip}_{X,\epsilon} \{X'_n + \alpha * \text{sign}(\nabla X J X'_n, y_{true})\}$$

3.6 Algorithm

This section provides descriptions of the ADDA-Adv. model's central algorithm which is illustrating in figure 3.3.

3.6.1 Noise Detection

This layer detect noise from a sample using 'OpenCv' python3 library. So, detection noisy frame via three steps (i) take input from user and convert the image sample BGR to HSV,

(ii) calculate the percentage of noise, (iii) based on threshold decide whether is legal and noisy.

Step 1: BGR to HSV

After taking input from user 'OpenCV2' convert the image sample BGR to HSV color. Which change view dimension.

Step 2: Calculate noise percentage

We used 'Numpy' python3 library to calculate the percentages of noisy image. Here lower value means image noise level is high and higher value means image noise level is low.

Step 3: Decision make

We set a threshold to identify image state. If the sample is lower than threshold, we consider the sample is noisy and reject the sample.

3.6.2 Denoising Autoencoder

Denoising layer is most crucial part in this work. We used Autoencoder to denoise the sample. These neural network compress and reconstruct our sample. There are three steps to complete the whole process. (i) encoder, (ii) bottleneck, (iii) decoder.

Encoder: encoder take input true image from Detector layer. After taking the input neural network will compress the sample and send it to the next layer.

Bottleneck: bottleneck is the lowest possible dimension of data. This layer contain all the data get from encoder and send it the decoder.

Decoder: from the compressed image data decoder reconstruct the whole image. This reconstructed image is very close to the true image. But this process can reduce noise from image.

3.6.3 Super Resolution GAN

Super Resolution GAN is to make the image high resolution from low resolution fuzzy image. Generative Adversarial Network (GAN) is a continuous process. Which is trained with 'imageNet' dataset. The whole process starts from random noise and tries to generate the ground truth value. The process is a little bit slower but the end result is very impressive. At the end, the neural network returns high resolution image. There are two networks working in SRGAN: (i) Generator, (ii) Discriminator

Generator: This network starts from random noise and tries to make something close to the original image and sends it to the Discriminator for validation. And receives feedback from the Discriminator and updates the sample. Again returns to the Discriminator.

Discriminator: The Discriminator always tries to detect faults of the sample which is returned from the Generator network. This unsupervised learning continuously continues until the Generator returns some impressive result.

Algorithm 1 ADDA-Adv. tools

```

1: Input: image sample X

2: /* Noisy image detector */

3: X := Convert image BGR to HSV
4: Saturation_c := calculate noise
5: if: saturation_c < threshold
6:     return true
7: else:
8:     return false
9: endif

10: /* denoising autoencoder */

11: for loop: range(batch_size)
12:     Encode: compress sample X
13:     Decode: reconstruct sample X
14: endloop
15: Output:  $X_d := X$ 

16: /* super resolution GAN */

17: Input: denoise LR image  $X_d$ 
18: forloop: range(batch_size)
19:     Generator: produce image  $X_g = G(X_d)$ 
20:     Discriminator: map low regulation image  $X_{sr} = D(X_g)$ 
21: endloop
22: Output: super resolution denoise sample  $X_{sr}$ 
23: close

```

Figure 3.2: Algorithm of ADDA-Adv.

3.6.4 Attack Sample Generator

ADDA-Adv. take adversarial sample which can fool image classifier neural network. To test our model ADDA-Adv. tools we implement some of well state-of-the-art attacking algorithm Such as ‘FGSM’, ‘BIM’, ‘ILCM’. All of the attack algorithm we use ‘MobileNetV2’ Convolutional Neural Network(CNN). We use ‘ImageNet’ high resolution image dataset to generate attack. To consider real life scenario, we use those high resolution

image. We provide some high resolution image to the adversarial generator function which will modify little bit and return adversarial sample. That sample can fool any image classifier. Our ADDA-Adv. tools can mitigate those attack if we pass those adversarial sample through our tools.

3.7 Implementation

The adversarial defense tools is developed by several machine learning python3 libraries like 'tensorflow', 'keras', 'numpy', 'scipy', 'openCv' etc. Our tools is preprocessing system so easy to integrate with any classifier neural network model. Here we use 'MobileNetV2' Convolutional Neural Network with 'ImageNet' dataset. We prepare attack environment by using some of Adversarial attack like ILCM, FGSM, BIM attacks. From those attack algorithms we took adversarial sample and pass through our ADDA-Adv. tools.

Test suite setup. We selected 3 attack algorithm to generate adversarial sample. Both attack we consider black-box and white-box scenario. We also used high resolution image dataset. We use 'MobileNetV2' as a main classifier which has high image classification accuracy.

Table 3.1 contains some background information for these applications. We are using Anaconda for environment setup. We are using Jupyter Notebook IDE. Our system configuration 6-core Intel Core i7 4.0 GHz CPU, 16.0 GB RAM, Nvidia GTX1060 GPU 6.0 GB, Windows 10 64bit operating system.

Table 3.1 Test application details

Area	Main classifier model	Attack method	Type	Dataset	Misclassify accuracy
Classification	MobileNetV2	FGSM	White-box	ImageNet	75.37%
Classification	MobileNetV2	FGSM	Black-box	ImageNet	60.70%
Classification	MobileNetV2	BIM	White-box	ImageNet	99.95%
Classification	MobileNetV2	BIM	Black-box	ImageNet	90.00%
Classification	MobileNetV2	ILCM	White-box	ImageNet	98.00%
Classification	MobileNetV2	ILCM	Black-box	ImageNet	80.73%

CHAPTER 4

RESULT AND DISCUSSION

In this section we discuss and analysis the result based on the ADDA-Adv. tool and experiment are also provided.

4.1 Discussion

We provide our experimental result in Table 4.1. In column 1 showed different attack. In column 2 represent the value of ϵ . Which is a hyper parameter. Here the hyper parameter value changes the attack accuracy level. ϵ value impact both black-box and white-box scenario.

Table 4.1 The value of ϵ impact on white-box and black-box attack

Attack	Params.	Accuracy
No Attack	-	79.43%
FGSM	$\epsilon = 0.050$	75.37%
	$\epsilon = 0.200$	34.91%
	$\epsilon = 0.250$	35.93%
BIM	$\epsilon = 0.050$ 2 nd iter.	98.86%
	$\epsilon = 0.100$ 3 rd iter.	99.95%
	$\epsilon = 0.150$ 3 rd iter.	98.22%
ILCM	$\epsilon = 0.016$ 4 th iter.	74.64%
	$\epsilon = 0.020$ 5 th iter.	98.00%

Here, we showed that the effectiveness of our tool ADDA-Adv. in table 4.2. We showed the accuracy of tool in different part. Second column No defense applied. Third and fourth column showed the accuracy of only use of Autoencoder and Super Resolution GAN respectively. And the finally, the combination of AE and SRGAN.

Table 4.2 Effectiveness of ADDA-Adv. tool

Attack	No defense	Auto-Encoder	AE+SRGAN
Clean	89.43%	64.41%	89.23%
FGSM	20.32%	50.12%	83.80%
BIM	16.48%	44.23%	85.90%
ILCM	19.38%	54.00%	80.22%

4.2 Details of effectiveness

In Table 4.2 we showed in different attack scenario. We divided our tool in part by part. We showed the accuracy level. The details are given below

4.2.1 No defense

With clean image sample and no defense technique our classifier model achieves 89.43% classification accuracy with high resolution image. Here main classifier accuracy depend on ‘MobileNet V2’ Convolutional Neural Network. In ‘FGSM’ attack scenario our classifier model accuracy decreased to 20.32%. On the hand, same sample predict as a tools different label which is not match to true label with high accuracy 79.23% in softmax layer. Similarly, in ‘BIM’ attack scenario result is get worse. This method is iterative in every iterative with same hyper parameter accuracy level decease. In our case we get 16.48% accuracy level. In case the case of ‘ILCM’ that is also iterative attack technique. Use of this attack in a without defense Neural Network classifier get accuracy nearly 19.38%. Here, we can result are how decreases. And taking classifier in wrong direction by the softmax layer.

4.2.2 Auto-Encoder

Only use of Auto-encoder sample makes little bit fuzzy. It makes the sample denoise but accuracy level not so high compare to true image sample. In that case 'FGSM' attack scenario achieves more than 50 percent but not that convincing result. But softmax layer retain the prediction in right way. Use of 'BIM' adversarial attack accuracy level is 44.23 %. Again not high accuracy level. Because low resolution fuzzy image. In the case of 'ILCM' attack autoencoder denoise the sample and achieve nearly 54%. Here, autoencoder does the job but make the sample less confident.

4.2.2 AE+SRGAN

Finally, we used Super resolution GAN. After get the value from Auto-Encoder we pass through the SRGAN. Which make the sample high resolution and restore the image close to the true image. Here, we can see pretty good result. In the case of 'FGSM' attack we can see the accuracy is 83.80% with the use of AE+SRGAN. On the other hand, 'BIM' adversarial attack technique our tools ADDA-Adv. work pretty well. It achieves 85.90%. In 'ILCM' iterative adversarial attack method. After use of denoise autoencoder and Super Resolution GAN, classifier model achieve 80.22% accuracy. Here, Auto-Encoder and Super Resolution work pretty well.

4.3 Other Experimental Details

4.3.1 Detect noisy sample

Table 4.3 Effectiveness of noisy image detection

Image size	Success rate
224*224	98.10%
156*156	99.48%
128*128	95.77%
28*28	87.23%

We use noisy sample detection method. Based on image noise and saturation, we set a threshold to identify whether the sample is real or fake. We took some image sample in different resolution. And test the detection rate. In our case high resolution image are easy detectable through this detector. But sometimes comparatively low resolution performance is little bit low. We can see the table no 4.3.

4.3.2 Performance compare

In this section, we compare our tools with some other model. Which did some pretty good job. In table 4.4 we make a comparison Metrix. Here, we set some criteria based on our contribution. In this Metrix, we compare our work with five different methods. In our work we design our model ADDA-Adv. which can integrate any pre-trained Deep Neural Network (DNN) but on the other hand, Adversarial Training method we need to train again the classifier model. Because of this, computational costly will be extremely high. Adversarial train method is not generic; it will work those attack based on trained. Our tool ADDA-Adv. tool is generic and work with high resolution image sample. Data Compression method cannot detect noisy sample and not work with high resolution sample. On the other hand, our tool ADDA-Adv. can detect noisy sample and work with high resolution sample. In the case of, Feature Squeezing technique doesn't provide generic

solution against adversarial attack but our method more generic to adversarial attack. MagNet defense method also doesn't work with high resolution image. Def-GAN is another technic which is good defense technique but it has no detection technique. Thus because softmax layer still can produce wrong prediction. On the other hand, our tools does the good job to detect noisy image.

	Pre-train model integration	Generic Solutions	High Resolution data	Detect Noisy sample
Adversarial training	Low	Low	Medium	Low
Data compression	Medium	Low	Medium	Low
MagNet	High	Medium	Medium	High
Feature squeezing	Medium	Low	High	High
Def-GAN	High	High	Medium	Low
DAE+SRGAN(our) ADDA-Adv.	High	High	High	Medium

Table 4.4: Comparison Metrix

CHAPTER 5

CONCLUSION AND FUTURE RECOMMENDATION

5.1 Findings and Contributions

Adversarial attack can easily compromise any deep neural network. Causes for this vulnerability can harm to any computer vision based applications. To make a defense mechanism that mitigate the attack with degrading the performance of the classifier neural network is a big challenge. In this work, we introduce a tool called ADDA-Adv. tool. This tool can detect noisy misguided samples and reject them which are non-recoverable. Furthermore, which samples can pass the detector we use AE and SRGAN to denoise and bring back to the original sample. The ADDA-Adv. solution has been thoroughly tested in different state-of-the-art adversarial attack algorithms and found some impressive results. This tool doesn't have overhead to existing trained neural networks and is easy to implement. Which is why that tool is cost-effective. Tools achieve 89.23% accuracy in high resolution image data. Our defense solution is very generic to this kind of attack.

5.2 Recommendations for Future Works

In the future we will develop an algorithm which can work in real-time image data. Which works with object detection neural networks.

REFERENCES

- Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
- Huang, R., Xu, B., Schuurmans, D., & Szepesvári, C. (2015). Learning with a strong adversary. *arXiv preprint arXiv:1511.03034*.
- Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2017). Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.
- Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017, April). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security* (pp. 506-519).
- Hosseini, H., Chen, Y., Kannan, S., Zhang, B., & Poovendran, R. (2017). Blocking transferability of adversarial examples in black-box learning systems. *arXiv preprint arXiv:1703.04318*.
- Dziugaite, G. K., Ghahramani, Z., & Roy, D. M. (2016). A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*.
- Das, N., Shanbhogue, M., Chen, S. T., Hohman, F., Chen, L., Kounavis, M. E., & Chau, D. H. (2017). Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*.
- Carlini, N., & Wagner, D. (2017, May). Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)* (pp. 39-57). IEEE.
- Akhtar, N., Liu, J., & Mian, A. (2018). Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3389-3398).
- Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., & Yuille, A. (2017). Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1369-1378).

- Wang, Q., Guo, W., Zhang, K., Ororbia II, A. G., Xing, X., Liu, X., & Giles, C. L. (2016). Learning adversary-resistant deep neural networks. *arXiv preprint arXiv:1612.01401*.
- Biggio, B., Nelson, B., & Laskov, P. (2011, November). Support vector machines under adversarial label noise. In *Asian conference on machine learning* (pp. 97-112).
- Lyu, C., Huang, K., & Liang, H. N. (2015, November). A unified gradient regularization family for adversarial examples. In *2015 IEEE International Conference on Data Mining* (pp. 301-309). IEEE.
- Zhao, Q., & Griffin, L. D. (2016). Suppressing the unusual: towards robust cnns using symmetric activation functions. *arXiv preprint arXiv:1603.05145*.
- Rozsa, A., Gunther, M., & Boulton, T. E. (2016). Towards robust deep neural networks with BANG. *arXiv preprint arXiv:1612.00138*.
- Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016, May). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)* (pp. 582-597). IEEE.
- Papernot, N., & McDaniel, P. (2017). Extending defensive distillation. *arXiv preprint arXiv:1705.05264*.
- Xu, W., Evans, D., & Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*.
- Gu, S., & Rigazio, L. (2014). Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*.
- Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*. Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011, January). Contractive auto-encoders: Explicit invariance during feature extraction. In *Icml*.
- Gao, J., Wang, B., Lin, Z., Xu, W., & Qi, Y. D. Masking deep neural network models for robustness against adversarial samples. *arXiv 2017. arXiv preprint arXiv:1702.06763*.
- Cisse, M., Adi, Y., Neverova, N., & Keshet, J. (2017). Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*.
- Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*.

- Meng, D., & Chen, H. (2017, October). Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 135-147).
- Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., & Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1778-1787).
- Bengio, Y., Yao, L., Alain, G., & Vincent, P. (2013). Generalized denoising auto-encoders as generative models. In *Advances in neural information processing systems* (pp. 899-907).
- Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690).

Turnitin Originality Report

Processed on: 06-Oct-2020 12:21 +06
 ID: 1406799943
 Word Count: 7638
 Submitted: 1

163-35-1734 By Subroto Karmokar

Similarity Index

21%

Similarity by Source

Internet Sources: 17%
 Publications: 13%
 Student Papers: 17%

8% match (Internet from 16-Sep-2020)

<https://www.mdpi.com/2076-3417/9/5/909/htm>

2% match (student papers from 09-Apr-2018)

Class: April 2018 Project Report

Assignment: Student Project

Paper ID: [943592609](#)

2% match (Internet from 01-Apr-2020)

<https://www.slideshare.net/RaihanMahmud5/remote-doctor-project-report>

1% match (Internet from 05-Oct-2020)

<https://jivp-urasipjournals.springeropen.com/articles/10.1186/s13640-020-00512-8/tables/1>

1% match (Internet from 20-Sep-2020)

<https://www.globenewswire.com/news-release/2019/04/09/1799533/0/en/Computer-Vision-Market-Worth-USD-48-Billion-by-2023-The-Emergence-of-Computers-is-Intensifying-Global-Computer-Vision-Market-to-Prosper-with-Major-Developments-in-Virtual-Reality.html>

1% match (publications)

[Shilin Qiu, Qihe Liu, Shijie Zhou, Chunjiang Wu. "Review of Artificial Intelligence Adversarial Attack and Defense Technologies", Applied Sciences, 2019](#)

< 1% match ()

<http://eprints.utm.edu.my/23394/>

< 1% match (student papers from 13-Aug-2019)

[Submitted to Masinde Muliro University of Science and Technology on 2019-08-13](#)

< 1% match (Internet from 13-Jul-2020)

<https://deepai.org/publication/random-directional-attack-for-fooling-deep-neural-networks>

< 1% match (Internet from 08-Mar-2020)

http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3808/P15042%20%2824_%29.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 23-Aug-2020)

<http://dSPACE.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3557/P13665%20%2815%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 15-May-2019)

<http://export.arxiv.org/pdf/1904.08444>

< 1% match (Internet from 19-Jun-2020)

<http://www.ijac.net/en/article/doi/10.1007/s11633-019-1211-x>

< 1% match (student papers from 14-Nov-2019)

[Submitted to Universiti Teknologi Petronas on 2019-11-14](#)

< 1% match (student papers from 30-Oct-2017)

[Submitted to United International University on 2017-10-30](#)

< 1% match (student papers from 22-Nov-2019)

[Submitted to University of Glasgow on 2019-11-22](#)

< 1% match (student papers from 04-Jun-2018)

[Submitted to University of Melbourne on 2018-06-04](#)

< 1% match (publications)

[Jianjun Hu, Mengjing Yu, Qingzhen Xu, Jing Gao. "Classifiers Protected against Attacks by Fusion of Multi-Branch Perturbed GAN", Mobile Networks and Applications, 2020](#)

<p>< 1% match (Internet from 09-Sep-2020) https://www.researchgate.net/publication/331499625_Review_of_Artificial_Intelligence_Adversarial_Attack_and_Defense_Technolog</p>
<p>< 1% match (Internet from 06-Jan-2020) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3547/P13646%20%2824%25%29.pdf?isAllowed=y&sequence=1</p>
<p>< 1% match (Internet from 20-May-2019) https://images.template.net/wp-content/uploads/2016/01/18102236/swot-analysis-of-hospital-pdf.pdf</p>
<p>< 1% match (student papers from 13-Nov-2019) Submitted to uva on 2019-11-13</p>
<p>< 1% match (student papers from 05-Jun-2020) Submitted to University of College Cork on 2020-06-05</p>
<p>< 1% match (student papers from 09-Jul-2012) Submitted to Universiti Teknologi Petronas on 2012-07-09</p>
<p>< 1% match (Internet from 01-Aug-2020) https://www.mdpi.com/2079-9292/9/7/1145/html</p>
<p>< 1% match (publications) Peiqing Ni, Dongping Zhang, Kui Hu, Changxing Jing, Li Yang. "Single face image super-resolution using local training networks", 2017 4th International Conference on Systems and Informatics (ICSAI), 2017</p>
<p>< 1% match (Internet from 24-Apr-2020) http://dspace.library.daffodilvarsity.edu.bd:8080/bitstream/handle/20.500.11948/2633/142-15-3540.pdf?isAllowed=y&sequence=1</p>
<p>< 1% match (Internet from 07-Aug-2017) http://ulspace.ul.ac.za/bitstream/handle/10386/755/khoza_nn_2012.pdf?isAllowed=y&sequence=1</p>
<p>< 1% match (publications) Bo-Ching Lin, Hwai-Jung Hsu, Shih-Kun Huang. "Testing Convolutional Neural Network using Adversarial Attacks on Potential Critical Pixels", 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), 2020</p>
<p>< 1% match (Internet from 09-Dec-2019) https://arxiv.org/pdf/1809.08999.pdf</p>
<p>A Defense and Detection Against Adversarial Attack Using Denoising Autoencoder and Super Resolution GAN By – Subroto Karmokar ID: 163-35-1734 A thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Software Engineering Department of Software Engineering Daffodil International University, Summer-2020 Copyright © 2020 by Daffodil International University APPROVAL This thesis titled on "A Defense and Detection Against Adversarial Attack Using Denoising Autoencoder and Super Resolution GAN", submitted by Subroto Karmokar (Student ID: 163-35-1734) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents. BOARD OF EXAMINERS ----- Dr. Imran Mahmud Associate Professor and Head In-Charge Department of Software Engineering Faculty of Science and Information Technology Daffodil International University Chairman ----- Dr. Md. Asraf Ali Associate Professor Department of Software Engineering Faculty of Science and Information Technology Daffodil International University Internal Examiner 1 ----- Md. Anwar Hossen Lecturer (Senior Scale) Department of Software Engineering Faculty of Science and Information Technology Daffodil International University Internal Examiner 2 Dhaka University of Engineering and Technology, Gazipur ----- Prof. Dr. Mohammad Abul Kaashem Professor Department of Computer Science and Engineering External Examiner DECLARATION I hereby declare that I have taken this thesis under the supervision of Md. Maruf Hassan, Assistant Professor, Department of Software Engineering, Daffodil International University. I also declare that neither whole document nor any part of this thesis have been submitted elsewhere for award of any degree. Subroto Karmokar ID: 163 -35- 1734 Batch: 21st Batch Department of Software Engineering Faculty of Science & Information Technology Daffodil International University Certified by: Md. Maruf Hassan Assistant Professor Department of Software Engineering Faculty of Science & Information Technology Daffodil International University. I ACKNOWLEDGEMENT At first I want to give thanks my almighty God. God gives me the ability to successfully received to the final year. I have learnt ethics, manners and so many things during my university life. I so thankful all my university teachers who are train me well. Without them I never learn all of these things. I want to give thanks my supervisor honorable Md. Maruf Hassan, Assistant Professor, DIU for giving me such a wonderful opportunity and continuously guided as well as supported me to continue this thesis on the topic of 'A Defense and Detection Against Adversarial Attack Using Denoising Autoencoder and Super Resolution GAN' which helped me in doing research and writing of this thesis paper. Therefore, I have learnt so many new things from my supervisor which help me for higher study. Beside my supervisor, I would like to thanks my senior brothers and sisters who give support mentally. Most importantly, I thank to my parents, they are always inspiring me. ii TABLE OF CONTENTS DECLARATION i ACKNOWLEDGEMENT.....ii TABLE OF</p>

CONTENTS.....iii TABLE OF FIGUREiv

ABSTRACTv

CHAPTER

1.....1

INTRODUCTION.....1

1.1 Background1

Motivation of the Research2

1.3 Problem Statement3

.....3 **1.4 Research Questions3**

1.5 Research Objectives.....3

1.6 Research Scope.....4

1.7 Thesis Organization.....5

CHAPTER

2.....6

LITERATURE REVIEW.....7

2.1 Modifying Data.....5

2.2 Modifying Model7

.....3 Using Auxiliary Tool.....9

CHAPTER

3.....11

RESEARCH METHODOLOGY.....11

3.1 Detector layer.....12

3.2 Denoiser layer.....13

3.3 Super resolution GAN.....14

3.4 Main Classifier.....16

3.5 Attack generator.....18

3.6 Algorithm.....19

3.7 Implementation.....23

CHAPTER

4.....25

RESULT AND DISCUSSION.....25

4.1 Discussion.....25

4.2 Details of Effectiveness.....26

4.3 Performance comparison28

CHAPTER

5.....30

CONCLUSION AND FUTURE RECOMMENDATION.....30

5.1 Findings and Contributions30

5.2 Recommendations for Future Works.....30

REFERENCES.....31

iii TABLE OF FIGURE Figure 3.1: System Architecture of MobileNer v212

Figure 3.2: ADDA-Adv.....17

Figure 3.3: Algorithm of ADDA-Adv.....22

Figure 4.1: Performace matrix.....29

iv ABSTRACT Neural network usages are being popular in different sectors that make life easier and automated; so that security of neural network is a big concern. Against [adversarial attack](#) every [Deep Neural Network](#) are [vulnerable](#). [Adversarial attack](#) is technique to specially design an image sample with adversarial noise. Several number of adversarial attack technique are found in recent research, which are fool neural network with high misclassify accuracy. There is also various defense mechanism are proposed and build with Deep Neural Network to defend and increase robustness of the main classifier neural network model. However, there are very few model can work with high resolution image data and work with pre-trained neural network classifier. The main objective of this research was to proposed and developed a model which can integrate with any existing trained Neural network and more generic defense against adversarial attack tool called ADDA-Adv. Based on proposed model detect highly distorted image and reject those sample to avoid misclassification. Additionally, this work is intended to work with high resolution adversarial image sample. ADDA-Adv. tool restore the adversarial sample and provide 89.23 percent accuracy. Keywords: Adversarial Attack; Defense and detect tool; Deep Neural Network; Computer Vision vulnerability. v

CHAPTER 1 INTRODUCTION Adversarial attack is a technique to fool the neural network and misguided the classifier to the wrong direction. An adversarial sample is an image that has been slightly modified and that is almost identical to the original image which is why it is nearly undetectable to the human eye. There are several methods out there to generate adversarial example which can successfully misclassify any neural network with high confidence level. This kind of adversarial image can fool the neural network without prior knowledge of neural network. Deep neural network is vulnerable to the adversarial attack. Successful adversarial attack could lead to serious security breaches for both the organization and the user.

1.1 Background Over the years, artificial intelligent has become so popular in the many business and organizations. This is most cutting edge technology in computer science field. Most importantly, AI based solution is most effective and efficient. Specially, Deep Neural Network (DNNs)

incontestable wonderful performance within the field of image classification, object detection, speech recognition etc. Based on this solution many computer vision task are highly dependent like bio metric security, self-driving car, medical science, crime investigations. Statista says <https://www.globenewswire.com/news-release/2019/04/09/1799533/0/en/Computer-Vision-Market-Worth-USD-48-Billion-by-2023-The-Emergence-of-Computers-is-Intensifying-Global-Computer-Vision-Market-to-Prosper-with-Major-Developments-in-Virtual-Reality.html>) But DNN security is still a big issue. For example, change the input slightly and add some noise that made the DNN vulnerable. So, crafted an input sample in a special way to generate a wrong answer from classifier model is called adversarial attack. We can divide adversarial sample in two ways. First one is access the network and the parameters, which called white-box attack. Second one is no knowledge to the neural network architecture but read input and output of neural network, which is called black-box attack.

1.2 Motivation of the Research Nowadays many organizations extensively depend on computer vision task such as facial recognition security system, self-driving car, traffic control system, medical science, crime investigation. So, the attacker generates adversarial sample to misguide the neural network. Because of this vulnerability many user and organization face serious security breaches. To defend these types of attacks, several defense model are developed with different features.

1.3 Problem Statement After analyzing the previous research works about adversarial attack detection, no research methodology has work with white-box, gray-box and black-box setting at a time. And also observed that that defense system has performed very poor in iterative strong attack. It is also being noted that they failed defend unknown attack. Most of the defense system are overhead to training model.

1.4 Research Questions With having this background, motivation and problem statement in mind, I propose the following questions: ? Is our introduced model adversarial defense can effectively defend different kind of adversarial samples? ? Is our implemented tool providing better accuracy as compared to the existing solution? ? Is our model more generic compare others? 1.5 Research Objectives ? To propose a model that perfectly work with pre-trained model. Not overhead to trained model. ? To prepare generic model which can defeat newly generate attack. ? To analysis sample is Real or Fake. ? To recover the sample, whether it is adversarial sample. ? To implement a detector model that can reject highly perturbed image.

1.6 Research Scope The experimental process has been done in the Convolutional Neural Network (CNN) model using Python, Tensorflow & Keras. Use that technique in different application area like Object Detection, Self-Driving Car, Traffic Control, Bio Metric Security, Crime Branch, Medical Science etc. Our model can easily integrate with any existing Convolutional Neural Network classifier model. In this case we do not need to developed our own model to conduct out test plan.

1.7 Thesis Organization In this research, APA referencing technique has been used through this document. The paper has been completed with six chapters which are described below: Chapter 1: Research background, motivation, problem statement, objectives and scopes are conferred in this chapter. Chapter 2: This chapter includes discussion of the existing related works and figured out the research gap. Chapter 3: This chapter contains the research methodology and approaches as it follows for the research. Chapter 4: Result analysis and evaluation are discussed here. Chapter 5: The outcome of research and direction of further work is presented here.

CHAPTER 2 LITERATURE REVIEW This section describes the previous research and findings of Adversarial vulnerabilities / attacks which is focused on CNN classifier model. To do our work cycle in the simplest way we have been split our research process into some particular portion for better understanding.

2.1 Modifying Data Modifying data is technique which is modify and changing training dataset at stage of training or change the input data at stage of testing. There are different of technique like gradient hiding, data compression, adversarial training, data randomization, transferability blocking etc.

2.1.1 Adversarial Training Introduced [adversarial](#) sample [into the training](#) data is increase [the robustness of the model](#) and train the [model with the adversarial samples](#). [Szegedy et al.] First of all include the adversarial sample and change label [to make the classifier model](#) more [robust against](#) adversarial [attack](#). [Goodfellow et al.] reduce misclassification rate of [MNIST dataset from 89.4% to 17.9%](#) use of [adversarial training](#). [Huang et al.] improve [the robustness of classifier model](#) against adversarial attack [by punishing adversarial](#) data. [Tramèr et al.] [proposed ensemble adversarial training](#), that improve [the variety of adversarial](#) sample. [However, this is impossible to include all unknown attack](#) sample as [adversarial training](#). Which is main [limitation of](#) this technique.

2.1.2 Gradient Hiding A defense against attack like [gradient based](#) proposed by [Tramèr et al.] and attack used by adversarial generated method like [FGSM](#). [This method](#) generally [hides information](#) like [gradient](#). [If a classifier model is non differentiable](#) gradient base attack will not work. [However, by learning the proxy black-box model with gradient and using the adversarial samples](#) which is [generated](#) from [this model](#) [Papernot, N et al.], the tactic will simply be fooled during this case.

2.1.3 Data Compression The JPG compression method found by [Dziugaite et al.]. This [method can improve large number of network model recognition accuracy declined in FGSM attack](#). [Das et al.] also [used](#) the same [JPEG compression](#) technique [to defense against](#) DeepFool [and](#) FGSM. But those methods are not effective solution against strong attack such as Carlini & Wagner attacks [Carlini, N.; Wagner, D.]. On the other hand, Display Compression Technology (DCT) technique [Das, N. et al.] used to fight against universal attack [Akhtar, N. at el.] is also ineffective. The main limitation is data large amount of data compression can make the classifier less accuracy and small amount can't remove the adversarial signature.

2.1.4 Data Randomization The [random resizing adversarial](#) sample [can reduce the](#) effective of [adversarial](#) attack which introduced by [Xie et al.]. On the hand, add [some random textures to the adversarial](#) sample increase the effectiveness [to the](#) classifier [model](#). [Wang et al.] [used a data conversion module separated](#) for defense against possible adversarial attack and conduct data expansion during [the training](#) phase. [Such as adding some Gaussian randomization processing which](#) increase [the robustness of](#) classifier [model](#) against adversarial attack.

2.1.5 Blocking the Transferability The transferability can hold [attribute, if the](#) classifier model has different architecture. The main idea to protecting from [black-box attack is to prevent transferability of](#) adversaries. [Hosseini et al.] [proposed three step Null labeling method](#). To protect [the adversarial samples from one network to different network](#). The [main idea](#) behind this technique is add [a new Null label to dataset and classify](#) all of [them to Null label](#) by trained [classifier](#) model [to resist adversarial](#) attack. This technique includes [three steps](#) (i) [initial training target classifier](#), (ii) calculate [Null probabilities](#), (iii) [adversarial training](#). The [advantage of this](#) methodology is [marking the perturbation input as empty label](#) instead of [classifying it the original label](#). This method is effective against adversarial attack without effecting classification accuracy.

2.2 Modifying Model Modify and change neural network is another technique to defend against adversarial attack. [Such as defensive distillation](#), regularization, [feature squeezing](#), mask defense and [deep contractive](#).

2.2.1 Defensive Distillation A defense distillation method is proposed by [Papernot et al.] which is defend the attack based on distillation technology [Hinton, G. et al.]. The distillation method compresses the [large-scale model into small-scale](#). Most importantly, it retains original accuracy to the model. On the other hand, [defensive](#)

distillation does not modify the scale of model. This method produces smoother output surface and increase the robustness of the model against attack. Author of paper claim that defensive distillation reduces the adversarial attack rate by 90%. However, this defense technique is not effective against black-attack.

2.2.2 Regularization This method adds regular term to improve the generalization ability if the target which are known as penalty term to the cost function. It makes the model good resist attack against unknown dataset prediction. [Biggio et al.] used a regularization technique for SVM model during training to limit the vulnerability.

2.2.3 Feature Squeezing Feature squeezing is enhancement the model [Xu, W et al.]. The main idea of Feature squeezing is reducing the complexity of the data representation. That's why reduce adversarial interference for low sensitivity. There are two methods available (i) reduce the color depth of the pixel, (ii) use smooth filter on image. thus creating the model safer under noise and resistance attack and as well as prevent adversarial attack. However, this technique reduces the accuracy for real sample.

2.2.4 Mask Defense Mask defense layer is proposed by [Gao et al.]. In this technique adding a mask layer before the classification model. The mask layer trained the original image corresponding adversarial samples. After that encoded the distinguish between those image and output from the previous layer. It is typically believed that the foremost necessary weight within the additional layer corresponds to the foremost sensitive feature within the network. So, in the final classification all feature is masked by the additional layers with initial weight zero. In this technique, the deviation of classification results caused by adversarial samples is secure.

2.2.5 Deep Contractive Network (DCN) The deep compression network introduced by [Gu et al.], which reduce the adversarial noise using automatic encoder. This technique adopted a smoothing penalty kind of like a Convolutional Autoencoder (CAE) during the training process. This method effective against attack like L-BGS [Szegedy, C et al.]

2.2.6 Parseval Networks Parseval network is proposed by [Cisse et al.]. This defense network controls the global Lipschitz constant for adopting hierarchical regularization. Considering the network can be viewed as function combination at every layer. They parameterize the spectral norm of the network weight matrix for controlling the spectral norm of the network weight through Parseval frames [Kovačević, J et al.]. That's why it was called Parseval Network.

2.3 Using Auxiliary Tool Auxiliary tools are basically additional tools which is include in a neural network model. Such as Magnet, defense-GAN and high-level representation guided denoiser.

2.3.1 MagNet MagNet method is proposed by [Meng et al.], which reads the output of the last layer of the classifier as a black-box while not reading any data of the inner layer or changing the classifier. MagNet use a detector layer which can identify whether sample is legal or adversarial. The detector measure and calculate the gap between given sample and manifold. If the sample exceeds threshold it will be rejected. This technique also use a reformer to reconstruct an adversarial sample to legal sample. However, this method performance decrease in white-box attack scenario because knowing parameters of MagNet. Which is why author use multiple automatic encoder and randomly select one of them to make it difficult to predict which one used.

2.3.2 Defense-GAN [Samangouei et al.] proposed a defense mechanism which is effective against white-box and black-box both attack scenario. This technique utilizes generative network power. The main idea of this tools is minimizing the constructor error $\|G(x) - x\|$ of generator G. Then feed the image to the classifier model. This method is effective against adversarial attack and success of the method depend on GAN. However, without proper training performance may decline.

2.3.3 High-Level Representation Guided Denoiser (HGD) HGD is different from de-noising device like pixel level reconstruction loss function. In that method have problem to amplification error. On the other hand, HGD solve this problem by using a loss function to compare the output between clean image and denoising image. HGD method is proposed by [Liao et al.], which is design to robust target model against black-box and white-box both attack. Author proposed three HGD training method. Another advantage of HGD is that it is trained on a comparatively small dataset and can be used to protect models aside from the one guiding it.

CHAPTER 3 RESEARCH METHODOLOGY The system architecture of ADDA-Adv. tools and the problems solved by the components are discussed here. Figure 3.1 demonstrates the architecture based on the components and the way they relate with each other. Basically, that tool is preprocessing technique before feed in a classifier neural network. Which is why that tools can be easily implement in any existing model. We divide our tool in different parts. Those are Detector, Denoiser Autoencoder [Bengio, Y. et al.] and Super Resolution Generative Adversarial Network (SRGSN) [Ledig, C. et al.] for make image sharper. At first we use a detector to identify legal and adversarial samples. The detector measures the distance between a given sample under test and the manifold and block the sample if the distance exceeds the threshold. So here that detector will make distinguish between legal and adversarial noisy sample. After that we will send the legal sample to the next layer. The second layer is Denoiser layer which will remove the sample as much as possible and most importantly retain the necessary features. But the problem is denoiser make the sample little bit blur or fuzzy which is why we send it to the next layer. The third layer is SRGAN which make high resolution and sharper. That will help to improve the overall quality of that image. Finally, that sample will send the main classifier. The working flow of each component are discussed below in more detail.

Figure 3.1: System Architecture of ADDA-Adv. tool

3.1 Detector Layer In this tools, Adversarial noisy image make the main classifier fool in softmax layer. This is very important to detect those noisy image. To implement this detector layer we use "OpenCV" python3 library. Which is make the sample BGR to HSV. Then we take the value and calculate percentage of noise. We set threshold to identify the noisy sample. In that case this is worth to mention higher percentage is closer to legal sample. On the other hand, lower percentage is closer to noisy sample. Based on the thresholder this Detector layer will decide that sample will pass through the layer or rejected. This Detector layer will return a true or false value and percentage of noise pixel of that sample.

3.1.1 Legal image The legal image means which is original data without any kind of noise or distortion. Which sample retain every necessary feature. Based on Detector layer it will decide whether it legal image or not.

3.1.2 Noisy image The noisy image is distorted sample. Sometime noisy image which doesn't make any sense to human. But this kind image lead to neural network classifier to wrong direction. So based on Detector layer noisy image will be rejected.

3.2 Denoiser Layer To Implement this layer, we use Denoising Autoencoder for removing the noise from a adversarial sample. The main idea of Autocoder is input and output is equal feature. This process doesn't reduce or loss number of feature. This is a unsupervised neural network learning feature. Neural network accept image as a input an compress the data. After compress the image reconstruct the sample and learn from ground truth value. There are many types of autoencoder available to do different task like anomaly detection, reformer same, denoise sample etc. We build a Denoiser Autoencoder for our tools. To build this we use tensorflow, keras and deep learning model. The more details of Denoising Autoencoder given below.

3.2.1 Encoder The first layer of an Autoencoder is input layer. So, Autoencoder take sample as input and send it to the Encoder. In Encoder there couple of layer to compress the sample. So, the main idea is reduce the pixel and feature. That's how it reduces the sample dimensions and represent the data into encoder formation.

3.2.2 Bottleneck In any

Autoencoder, bottleneck play a very vital role. The data after compression pass through this layer. The most importantly bottleneck layer contain all the compress data and this lowest possible dimension of data. In addition, completing the compression the data send to decoder.

3.2.3 Decoder

This is the last part of the Autoencoder neural network. From bottleneck layer take compressed data and start reconstruct it. During reconstruct the model learn how close to the original input. The whole process goes through a couple of layer which is connected to each other.

3.2.3 Reconstruction Loss

That's method which is measure and calculate the over loss from the whole Autoencoder process. And as well as how good the decoder layer performs in reconstruction process. Most importantly how close to the original input.

3.3 Super Resolution Generative Adversarial Network

When a sample pass through Denoising Autoencoder, the sample loss their clarity or sharpness. Autoencoder make that image little bit blur and fuzzy. Which make difficult classification job for the main classifier. Because of, losing some important feature. [To solve the problem we use Super Resolution Generative Adversarial Network \(SRGAN\). Image super resolution is a technique to reconstruct a higher resolution image from a lower resolution image.](#) This process neural network observe the neighboring pixel and use a machine learning neural network to reproduce it. The conduct whole operation two main network here Discriminator Network and Generative Network. The main idea of GAN is Generative Network is produce fake image and send it to Discriminator Network. On the other hand Discriminator Network predict whether the sample is real and fake. And again send it Generator Network to reconstruct it. This process continue several time. Gradually, Generative Network improve the sample. This learning process called unsupervised learning.

3.3.1 Notation of GAN

P_{data} : real dataset ? P_z : noise distribution ? D, G : neural network ? $D(x)$: probability of x come from data rather than P_g

3.3.2 Generative Network In Super resolution GAN

Generative Network take random noise as input. And produce some output which is close to real sample. This network continuously try to fool Discriminator Network. After that Generative Network send it to Discriminator and received feedback. Based on those feedback Generative model update the model. $L_G = \mathbb{E}_{z \sim p_z} [\log(1 - C(G(x)))]$ During this process optimizing is that fix value of G to update parameters of D . After that fix the value of D to update parameters of G .

3.3.3 Discriminator Network

In this network data received from Generative Network and predict this whether sample is real or fake. Discriminator Network take input half generated data (fake) and half real data (real). Its use loss function $L_D = -\mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{z \sim p_z} [\log(1 - C(G(x)))]$ The whole process like [function \$V\(G, D\)\$](#) . $V(G, D) = \mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{z \sim p_z} [\log(1 - C(G(x)))]$ G, C

3.4 Main Classifier For the main image classifier

we use here is 'MobileNetV2'. This is Convolutional Neural Network (CNN) which is very advanced level image classifier. On the other hand, the dataset we used here is ImageNet which is pretty large image 256*256. That ImageNet dataset contain very larger number of data more than 14 million. 'MobileNetV2' network architecture is below Figure 3.2: MobileNet v2 workflow Table 3.0 MobileNet v2 architecture Input operator t c n s [224 * 3 Conv2d - 32 1 2 1122 * 32 bottleneck 1 16 1 1 1122 * 16 bottleneck 6 24 2 2 562 * 24 bottleneck 6 32 3 2 282 * 32 bottleneck 6 64 4 2 142 * 64 bottleneck 6 96 3 1 142 * 96 bottleneck 6 160 3 2 72 * 160 bottleneck 6 320 1 1 72 * 320 Conv2d 1*1 - 1280 1 1 72 * 1280 Avgpool 7*7 - - 1 - 1 * 1280 Conv2d 1*1 - k - ?](#) Here, c: output channels number, t: expansion factor, n: repeating number, s: stride. [3x3 kernels for using spatial convolution. The primary network {width multiplier 1, 224x224}, computational cost is 300 million multiply-adds. With the uses 3.4 million parameters. The network computational cost is up to 585M MAdds, while the model size varies between 1.7M and 6.9M parameters.](#)

3.5 Attack Generator Adversarial Attack

are generated from different type of method. Some of technique are [white box attack](#) and some of are [black box attack](#). The basic idea of adversarial attack is perturbation a sample image with some noise. That crafted image make the classifier fool. We used some of attack technique to test our adversarial defense tools.

3.5 .1 Fast Gradient Sign Method (FGSM)

[Fast Gradient Sign Method \(FGSM\) is proposed by \[Goodfellow et al.\]](#). This method calculate the gradient value of cost function based on the neural network input. The formula of this method is given below: $X' = X + \epsilon * \text{sign}(\nabla_x J(X, x_{trd}))$ In this equation, ϵ is hyper-parameter. It controls the amplitude. J is the cost function. ∇_x is the gradient of a normal value X . Author claim that this method can misclassify the MNIST dataset 89.4%.

3.5.2 Basic Iterative Method (BIM)

This method is the iterative version of FGSM. The use multiple time of FGSM. The formula of this method is given below: $X'_0 = X, X'_{n+1} = \text{Clip}_{X, \epsilon} \{X'_n + \nabla * \text{sign}(\nabla_x J(X'_n, x_{trd}))\}$ Here in this equation, $\text{Clip}_{X, \epsilon} \{A\}$ is element wise clipping of X . And ∇ is the step size. On the other hand, J is the cost function. ∇_x is the gradient of a normal value X . This methodology typically doesn't depend [on the approximation of the model and produces further harmful adversarial samples](#) once [this algorithm runs for a lot of iterations](#).

3.5 .3 Iterative Least-Likely Class Method (ILCM)

[Iterative Least-Likely Class Method](#) replace the class variable by using the with the small recognition probability in the distance. Which is get adversarial examples with 99% misclassification accuracy. The formula of this method is given below: $X'_0 = X, X'_{n+1} = \text{Clip}_{X, \epsilon} \{X'_n + \nabla * \text{sign}(\nabla_x J(X'_n, x_{trd}))\}$

3. 6 Algorithm

This section provides descriptions of the ADDA-Adv. model's central algorithm which is illustrating in figure 3.3.

3.6.1 Noise Detection

This layer detect noise from a sample using 'OpenCv' python3 library. So, detection noisy frame via three steps (i) take input from user and convert the image sample BGR to HSV, (ii) calculate the percentage of noise, (iii) based on threshold decide whether is legal and noisy. Step 1: BGR to HSV After taking input from user 'OpenCV2' convert the image sample BGR to HSV color. Which change view dimension. Step 2: Calculate noise percentage We used 'Numpy' python3 library to calculate the percentages of noisy image. Here lower value means image noise level is high and higher value means image noise level is low. Step 3: Decision make We set a threshold to identify image state. If the sample is lower than threshold, we consider the sample is noisy and reject the sample.

3.6.2 Denoising Autoencoder

Denoising layer is most crucial part in this work. We used Autoencoder to denoise the sample. These neural network compress and reconstruct our sample. There are three steps to complete the whole process. (i) encoder, (ii) bottleneck, (iii) decoder. Encoder: encoder take input true image from Detector layer. After taking the input neural network will compress the sample [and send it to the next layer](#). Bottleneck: bottleneck is the lowest possible dimension of data. This layer contain all the data get from encoder and send it the decoder. Decoder: from the compress image data decoder reconstruct the whole image. This reconstruct image is very close to the true image. But this process can reduce noise from image.

3.6.3 Super Resolution GAN

Super Resolution GAN is make the image high resolution from low resolution fuzzy image. Generative Adversarial Network (GAN) is continuous process. Which trained with 'imageNet' dataset. The whole process start from random noise and try to generate the ground truth value. The process is little bit slower but the end result is very impressive. At the end the neural network return high resolution image. There two network work in SRGAN (i) Generator, (ii) Discriminator Generator: This network start from random noise and try to make something close to the original image and send to the Discriminator for validation. And receive feedback from Discriminator and update sample. Again return to Discriminator. Discriminator: The Discriminator always try detect fault of the sample which is return from Generator network.

This unsupervised learning continuously conduct until Generator return some impressive result. Figure 3.2: Algorithm of ADDA-Adv. 3.6.4 Attack Sample Generator ADDA-Adv. take adversarial sample which can fool image classifier neural network. To test our model ADDA-Adv. tools we implement some of well state-of-the-art attacking algorithm Such as 'FGSM', 'BIM', 'ILCM'. All of the attack algorithm we use 'MobileNetV2' Convolutional Neural Network(CNN). We use 'ImageNet' high resolution image dataset to generate attack. To consider real life scenario, we use those high resolution image. We provide some high resolution image to the adversarial generator function which will modify little bit and return adversarial sample. That sample can fool any image classifier. Our ADDA-Adv. tools can mitigate those attack if we pass those adversarial sample through our tools.

3.7 Implementation The adversarial defense tools is develop by several machine learning python3 libraries like 'tensorflow', 'keras', 'numpy', 'scipy', 'openCv' etc. Our tools is preprocessing system so easy to integrate with any classifier neural network model. Here we use 'MobileNetV2' Convolutional Neural Network with 'ImageNet' dataset. We prepare attack environment by using some of Adversarial attack like ILCM, FGSM, BIM attacks. From those attack algorithms we took adversarial sample and pass through our ADDA-Adv. tools. Test suite setup. We selected 3 attack algorithm to generate adversarial sample. Both attack we consider black-box and white-box scenario. We also used high resolution image dataset. We use 'MobileNetV2' as a main classifier which has high image classification accuracy. Table 3.1 contains some background information for these applications. We are using Anaconda for environment setup. We are using Jupyter Notebook IDE. Our system configuration 6-core Intel Core i7 4.0 GHz CPU, 16.0 GB RAM, Nvidia GTX1060 GPU 6.0 GB, Windows 10 64bit operating system. Table 3.1 Test application details Area Main classifier model Attack method Type Dataset Misclassify accuracy Classification MobileNetV2 FGSM White-box ImageNet 75.37% Classification MobileNetV2 FGSM Black-box ImageNet 60.70% Classification MobileNetV2 BIM White-box ImageNet 99.95% Classification MobileNetV2 BIM Black-box ImageNet 90.00% Classification MobileNetV2 ILCM White-box ImageNet 98.00% Classification MobileNetV2 ILCM Black-box ImageNet 80.73%

CHAPTER 4 RESULT AND DISCUSSION In this section we discuss and analysis the result based on the ADDA-Adv. tool and experiment are also provided. 4.1 Discussion We provide our experimental result in Table 4.1. In column 1 showed different attack. In column 2 represent the value of ϵ . Which is a hyper parameter. Here the hyper parameter value changes the attack accuracy level. ϵ value impact both black-box and white-box scenario. Table 4.1 The value of ϵ impact on white-box and black-box attack Attack Params. Accuracy No Attack - 79.43% FGSM $\epsilon = 0.050$ $\epsilon = 0.200$ $\epsilon = 0.250$ 75.37% 34.91% 35.93% BIM $\epsilon = 0.050$ 2nd iter. $\epsilon = 0.100$ 3rd iter. $\epsilon = 0.150$ 3rd iter. 98.86% 99.95% 98.22% ILCM $\epsilon = 0.016$ 4th iter. $\epsilon = 0.020$ 5th iter. 74.64% 98.00% Here, we showed that the effectiveness of our tool ADDA-Adv. in table 4.2. We showed the accuracy of tool in different part. Second column No defense applied. Third and fourth column showed the accuracy of only use of Autoencoder and Super Resolution GAN respectively. And the finally, the combination of AE and SRGAN. Table 4.2 Effectiveness of ADDA-Adv. tool Attack No defense Auto-Encoder AE+SRGAN Clean 89.43% 64.41% 89.23% FGSM 20.32% 50.12% 83.80% BIM 16.48% 44.23% 85.90% ILCM 19.38% 54.00% 80.22%

4.2 Details of effectiveness In Table 4.2 we showed in different attack scenario. We divided our tool in part by part. We showed the accuracy level. The details are given below 4.2.1 No defense With clean image sample and no defense technique our classifier model achieves 89.43% classification accuracy with high resolution image. Here main classifier accuracy depend on 'MobileNet V2' Convolutional Neural Network. In 'FGSM' attack scenario our classifier model accuracy decreased to 20.32%. On the hand, same sample predict as a tools different label which is not match to true label with high accuracy 79.23% in softmax layer. Similarly, in 'BIM' attack scenario result is get worse. This method is iterative in every iterative with same hyper parameter accuracy level decrease. In our case we get 16.48% accuracy level. In case the case of 'ILCM' that is also iterative attack technique. Use of this attack in a without defense Neural Network classifier get accuracy nearly 19.38%. Here, we can result are how decreases. And taking classifier in wrong direction by the softmax layer. 4.2.2 Auto-Encoder Only use of Auto-encoder sample makes little bit fuzzy. It makes the sample denoise but accuracy level not so high compare to true image sample. In that case 'FGSM' attack scenario achieves more than 50 percent but not that convincing result. But softmax layer retain the prediction in right way. Use of 'BIM' adversarial attack accuracy level is 44.23%. Again not high accuracy level. Because low resolution fuzzy image. In the case of 'ILCM' attack autoencoder denoise the sample and achieve nearly 54%. Here, autoencoder does the job but make the sample less confident. 4.2.2 AE+SRGAN Finally, we used Super resolution GAN. After get the value from Auto-Encoder we pass through the SRGAN. Which make the sample high resolution and restore the image close to the true image. Here, we can see pretty good result. In the case of 'FGSM' attack we can see the accuracy is 83.80% with the use of AE+SRGAN. On the other hand, 'BIM' adversarial attack technique our tools ADDA-Adv. work pretty well. It achieves 85.90%. In 'ILCM' iterative adversarial attack method. After use of denoise autoencoder and Super Resolution GAN, classifier model achieve 80.22% accuracy. Here, Auto-Encoder and Super Resolution work pretty well. 4.3 Other Experimental Details 4.3.1 Detect noisy sample Table 4.3 Effectiveness of noisy image detection Image size Success rate 224*224 98.10% 156*156 99.48% 128*128 95.77% 28*28 87.23%

We use noisy sample detection method. Based on image noise and saturation, we set a threshold to identify whether the sample is real or fake. We took some image sample in different resolution. And test the detection rate. In our case high resolution image are easy detectable through this detector. But sometimes comparatively low resolution performance is little bit low. We can see the table no 4.3. 4.3.2 Performance compare In this section, we compare our tools with some other model. Which did some pretty good job. In table 4.4 we make a comparison Metrix. Here, we set some criteria based on our contribution. In this Metrix, we compare our work with five different methods. In our work we design our model ADDA-Adv. which can integrate any pre-trained Deep Neural Network (DNN) but on the other hand, Adversarial Training method we need to train again the classifier model. Because of this, computational costly will be extremely high. Adversarial train method is not generic; it will work those attack based on trained. Our tool ADDA-Adv. tool is generic and work with high resolution image sample. Data Compression method cannot detect noisy sample and not work with high resolution sample. On the other hand, our tool ADDA-Adv. can detect noisy sample and work with high resolution sample. In the case of, Feature Squeezing technique doesn't provide generic solution against adversarial attack but our method more generic to adversarial attack. MagNet defense method also doesn't work with high resolution image. Def-GAN is another technic which is good defense technique but it has no detection technique. Thus because softmax layer still can produce wrong prediction. On the other hand, our tools does the good job to detect noisy image. Pre-train model integration Generic Solutions High Resolution data Detect Noisy sample Adversarial training Low Low Medium Low Data compression [Medium Low](#) Medium [Low](#) MagNet [High Medium Medium High](#) Feature squeezing [Medium Low High High](#) Def-GAN [High High Medium](#) Low DAE+SRGAN(our) ADDA-Adv. High High High Medium Table 4.4: Comparison Metrix CHAPTER 5 CONCLUSION AND FUTURE RECOMMENDATION 5.1 Findings and Contributions

Adversarial attack can easily compromise any deep neural network. Causes for this vulnerability can harm to any computer vision based applications. To make a defense mechanism that mitigates the attack with degrading the performance of the classifier neural network is a big challenge. In this work, we introduce a tool called ADDA-Adv. tool. This tool can detect noisy misguiding samples and reject them which are non-recoverable. Furthermore, which sample can pass the detector we use AE and SRGAN to denoise and back to the original sample. The ADDA-Adv. solution has been thoroughly tested in different state-of-the-art adversarial attack algorithms and found some impressive results. This tool doesn't add overhead to existing trained neural networks and is easy to implement. Which is why that tool is cost-effective. Tools achieve 89.23% accuracy in high-resolution image data. Our defense solution is very generic to this kind of attack. [5.2 Recommendations for Future Works](#) In future we will develop an algorithm which can work in real-time image data. Which works with object detection neural networks.

REFERENCES Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572. Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199. Huang, R., Xu, B., Schuurmans, D., & Szepesvári, C. (2015). Learning with a strong adversary. arXiv preprint arXiv:1511.03034. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2017). Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017, April). Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security (pp. 506-519). Hosseini, H., Chen, Y., Kannan, S., Zhang, B., & Poovendran, R. (2017). Blocking transferability of adversarial examples in black-box learning systems. arXiv preprint arXiv:1703.04318. Dziugaite, G. K., Ghahramani, Z., & Roy, D. M. (2016). A study of the effect of jpg compression on adversarial images. arXiv preprint arXiv:1608.00853. Das, N., Shanbhogue, M., Chen, S. T., Hohman, F., Chen, L., Kounavis, M. E., & Chau, D. H. (2017). Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. arXiv preprint arXiv:1705.02900. Carlini, N., & Wagner, D. (2017, May). Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (pp. 39-57). IEEE. Akhtar, N., Liu, J., & Mian, A. (2018). Defense against universal adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3389-3398). Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., & Yuille, A. (2017). Adversarial examples for semantic segmentation and object detection. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1369-1378). Wang, Q., Guo, W., Zhang, K., Ororbai II, A. G., Xing, X., Liu, X., & Giles, C. L. (2016). Learning adversary-resistant deep neural networks. arXiv preprint arXiv:1612.01401. Biggio, B., Nelson, B., & Laskov, P. (2011, November). Support vector machines under adversarial label noise. In Asian conference on machine learning (pp. 97-112). Lyu, C., Huang, K., & Liang, H. N. (2015, November). A unified gradient regularization family for adversarial examples. In 2015 IEEE International Conference on Data Mining (pp. 301-309). IEEE. Zhao, Q., & Griffin, L. D. (2016). Suppressing the unusual: towards robust cnns using symmetric activation functions. arXiv preprint arXiv:1603.05145. Rozsa, A., Gunther, M., & Boulton, T. E. (2016). Towards robust deep neural networks with BANG. arXiv preprint arXiv:1612.00138. Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016, May). Distillation as a defense to adversarial perturbations against deep neural networks. In 2016 IEEE Symposium on Security and Privacy (SP) (pp. 582-597). IEEE. Papernot, N., & McDaniel, P. (2017). Extending defensive distillation. arXiv preprint arXiv:1705.05264. Xu, W., Evans, D., & Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155. Gu, S., & Rigazio, L. (2014). Towards deep neural network architectures robust to adversarial examples. arXiv preprint arXiv:1412.5068. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. Rifai, S., Vincent, P., Müller, X., Glorot, X., & Bengio, Y. (2011, January). Contractive auto-encoders: Explicit invariance during feature extraction. In ICML. Gao, J., Wang, B., Lin, Z., Xu, W., & Qi, Y. D. Masking deep neural network models for robustness against adversarial samples. arXiv 2017. arXiv preprint arXiv:1702.06763. Cisse, M., Adi, Y., Neverova, N., & Keshet, J. (2017). Houdini: Fooling deep structured prediction models. arXiv preprint arXiv:1707.05373. Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605. Meng, D., & Chen, H. (2017, October). Magnet: a two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC conference on computer and communications security (pp. 135-147). Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., & Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1778-1787). Bengio, Y., Yao, L., Alain, G., & Vincent, P. (2013). Generalized denoising auto-encoders as generative models. In Advances in neural information processing systems (pp. 899-907). Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4681-4690). Copyright © 2020 by [Daffodil International University](#) Copyright © 2020 by [Daffodil International University](#) Copyright © 2020 by [Daffodil International University](#) Copyright © 2020 by [Daffodil International University](#) Copyright © 2020 by [Daffodil International University](#) 1 Copyright © 2020 by [Daffodil International University](#) 2 Copyright © 2020 by [Daffodil International University](#) 3 Copyright © 2020 by [Daffodil International University](#) 4 Copyright © 2020 by [Daffodil International University](#) 5 Copyright © 2020 by [Daffodil International University](#) 6 Copyright © 2020 by [Daffodil International University](#) 7 Copyright © 2020 by [Daffodil International University](#) 8 Copyright © 2020 by [Daffodil International University](#) 9 Copyright © 2020 by [Daffodil International University](#) 10 Copyright © 2020 by [Daffodil International University](#) 11 Copyright © 2020 by [Daffodil International University](#) 12 Copyright © 2020 by [Daffodil International University](#) 13 Copyright © 2020 by [Daffodil International University](#) 14 Copyright © 2020 by [Daffodil International University](#) 15 Copyright © 2020 by [Daffodil International University](#) 16 Copyright © 2020 by [Daffodil International University](#) 17 Copyright © 2020 by [Daffodil International University](#) 18 Copyright © 2020 by [Daffodil International University](#) 19 Copyright © 2020 by [Daffodil International University](#) 20 Copyright © 2020 by [Daffodil International University](#) 21 Copyright © 2020 by [Daffodil International University](#) 22 Copyright © 2020 by [Daffodil International University](#) 23 Copyright © 2020 by [Daffodil International University](#) 24 Copyright © 2020 by [Daffodil International University](#) 25 Copyright © 2020 by [Daffodil International University](#) 26 Copyright © 2020 by [Daffodil International University](#) 27 Copyright © 2020 by [Daffodil International University](#) 28 Copyright © 2020 by [Daffodil International University](#) 29 Copyright © 2020 by [Daffodil International University](#) 30 Copyright © 2020 by [Daffodil International University](#) 31 Copyright © 2020 by [Daffodil International University](#)

University [32 Copyright](#) © 2020 by [Daffodil International University](#), 33 Copyright © 2020 by Daffodil International University