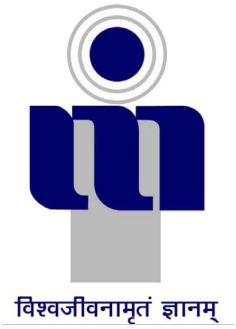
ADVERSARIAL ATTACKS AND DEFENSES ON NEURAL NETWORK

A project report submitted in partial fulfillment of the requirements for B.Tech. Project

B.Tech.

by

Abhishek Kumar (2016IPG-007)
Mayank Yadav (2016IPG-049)
Yash Jaiswal (2016IPG-120)



ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR-474 015

2019

CANDIDATES DECLARATION

We hereby certify that the work, which is being presented in the report, entitled **Adversarial Attacks and Defenses on Neural Network**, in partial fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the institution is an authentic record of our own work carried out during the period *May 2019* to *September 2019* under the supervision of **Dr. Saumya Bhadauria** and **Dr. Yash Daultani**. We also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date: Signatures of the Candidates

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: Signatures of the Research Supervisors

ABSTRACT

Deep learning is at the core of artificial intelligence's present rise. It has become the powerhouse for applications varying from automated-driving cars to computer vision supervision and safety.

While deep neural networks have shown phenomenal achievement in solving complicated issues (often beyond human capacities), latest studies have indicated that these networks are susceptible to adversarial assaults that are in the type of subtle input disturbances that cause a model to predict wrong output. These disturbances are usually too tiny to be perceptible for pictures, yet they become completely successful in fooling the models of deep learning. We have implemented both targeted and non-targeted attacks on different datasets.

Adversarial assaults present a severe danger to the practical achievement of profound learning. This has lately resulted to a substantial increase in number of contributions in this area. We study the projects that have made been designed to analyze the adversarial attacks, evaluate the presence of such attacks, and suggest solutions against them. To illustrate that in practical circumstances adversarial attacks are feasible, we review the contributions in real-world situations that assess adversarial assaults individually. Finally, based on the literature reviewed, we provide a wider perspective on this area of studies.

Keywords: Neural networks, adversarial attacks, datasets, perturbations, defense methods

ACKNOWLEDGEMENTS

We are highly indebted to **Dr. Saumya Bhadauria** and **Dr. Yash Daultani**, and are obliged for giving us the autonomy of functioning and experimenting with ideas. We would like to take this opportunity to express our profound gratitude to them not only for their academic guidance but also for their personal interest in our project and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within us. The nurturing and blossoming of the present work is mainly due to their valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. Our mentor always answered myriad of our doubts with smiling graciousness and prodigious patience, never letting us feel that we are novices by always lending an ear to our views, appreciating and improving them and by giving us a free hand in our project. It's only because of their overwhelming interest and helpful attitude, the present work has attained the stage it has.

Finally, we are grateful to our Institution and colleagues whose constant encouragement served to renew our spirit, refocus our attention and energy and helped us in carrying out this work.

(Abhishek Kumar) (Mayank Yadav) (Yash Jaiswal)

TABLE OF CONTENTS

ABSTRACT **ii**

LIST OF FIGURES **v**

1 INTRODUCTION AND LITERATURE SURVEY **vii**

1.1 INTRODUCTION vii

1.1.1 Deep Neural Networks viii

1.1.2 Vulnerability of DNN viii

1.1.3 Unsupervised Learning ix

1.1.4 Generative Models ix

1.2 Motivation x

1.3 LITERATURE REVIEW x

1.4 PROBLEM STATEMENT x

1.5 OBECTIVE xi

2 DESIGN DETAILS AND IMPLEMENTATION **xii**

2.1 DATASETS xii

2.2 VICTIM DEEP LEARNING MODELS xii

2.3 ADVERSARIAL PERTURBATION xiii

2.4 Threat Model xiii

2.5 ATTACKS xiv

2.5.1 White-Box attack xiv

2.5.2 Black-Box attack xiv

2.6 ADVERSARIAL ATTACK METHODS xiv

2.6.1 BOX-CONSTRAINED L-BFGS xv

2.6.2 FAST GRADIENT SIGN METHOD (FGSM) xv

2.6.3 ITERATIVE - FGSM xvi

2.6.4 SUBSTITUTE BLACK-BOX ATTACK xvi

2.6.5 MOMENTUM BASED ITERATIVE FGSM (*state of the art*) . xvii

2.6.5.1 Description xvii

2.6.5.2 Need xviii

2.6.5.3 Why momentum iterative FGSM demonstrates better transferability? xviii
2.6.5.4 Algorithm xviii
2.7 DEFENSES xix
2.7.1 Adversarial Training xix
2.7.2 Gradient Masking xx
2.7.3 Guided Denoiser xx

3 RESULTS AND DISCUSSION xxii
3.1 Generation of the perturbed image xxii
3.2 Non-targeted attack on a model trained on MNIST datasets xxiii
3.3 Targeted attack on inception model trained on ImageNet datasets xxiii
3.4 Results of Black Box Attack xxiv
3.4.0.1 Performance xxiv
3.4.0.2 Impact xxv

4 CONCLUSION xxviii

REFERENCES xxviii

LIST OF FIGURES

1.1 Left image is actual one and the right image is the perturbed image [1]. . viii

2.1 Attack on Black Box Model [2]. xvii
2.2 Result of adversarial training [3] xx
2.3 this is how denoiser works [4]. xxi

3.1 Inception V3 Model [5] xxii
3.2 Original and generated Adversarial samples on the MNIST dataset [1]. . xxiii
3.3 Graph representing the success rate generated for black box model against white box [6]. xxv
3.4 AWS vision platform recognises the person as Leonaro Dicaprio, which is true. xxvi
3.5 After running MI-FGSM algorithms on the same image, it categorizes the celebrity as Matt Damon. xxvii

CHAPTER 1

INTRODUCTION AND LITERATURE SURVEY

This chapter includes the overview of deep neural network used to implement the project and literature review related to work done in this field. The first section is intended to introduce various definitions related to work done in the report in section 1.1. The next section 1.2 includes the motivation and section 1.3 contains the literature review containing the details about the past papers. It is then followed by the problem statement and objective in section 1.4 and 1.5.

1.1 INTRODUCTION

There was a time when the operating system was in its earlier stage, and security modules were not implicit in the operating system design. Today is the same era for machine learning systems where we are on the verge of making security implicit in machine learning models.

At present stage, AI takes on duties that were earlier the exclusive domain of humans, from surveillance camera footage analysis and image classification to cancer detection, cybercrime investigation and car driving, and much more. Yet AI algorithms can fail spectacularly given their superhuman speed and precision. The modern neural networks are an absolute wonder in providing high accuracies but surprisingly these nets are too vulnerable to adversarial attacks employing providing small perturbations in the image that remains nearly imperceptible to the process of human vision [7].

Existing machine learning models as well as the highly developed and advanced neural networks, are also vulnerable to adversarial noise or perturbations [7]. Either during training the model or during inference phase (predicting the output), the perturbations can be introduced in the model which can cause wrong classification of legit-

imate input. These small perturbations to input images result in an incorrect classification by the models, hence resulting in an arbitrary failure of an otherwise seemingly well-trained system. Neural networks are still quite fragile when attacked by an adversary and can be tricked to give incorrect predictions. It is all about how carefully and precisely, adversarial examples using suitable algorithms can be generated. To show how the attacks work, the perturbed images have been used. These images can be created by adding noise to them and they appear similar to the outside user but the neural network model recognises them as two different images.

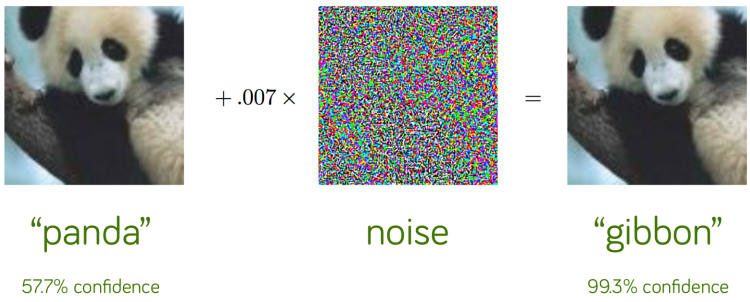


Figure 1.1: Left image is actual one and the right image is the perturbed image [1].

1.1.1 Deep Neural Networks

Deep neural networks are a collection of algorithms intended to acknowledge trends, patterned loosely after the human brain. They understand sensory information by means of a type of pure entry machine perception, marking or clustering. The patterns they acknowledge are numerical, stored in vectors that need to be converted into all real-world information, whether images, noise, writing, or time series.

1.1.2 Vulnerability of DNN

Within the modern world, neural networks are becoming increasingly inescapable and are often applied without considering their prospective security flaws. These networks' existence is highly likely to only improve, and there is a good reason for that [8].

With the increase in computation power and GPUs, deep neural network approaches have been widely accepted for diverse machine learning problems. But, modern profound networks are incredibly vulnerable to adversarial attacks as these tiny disturbances included in the images stay imperceptible to the human vision system, despite their high accuracies [9]. Such attacks can trigger an image prediction to be totally

changed by a neural network classifier. Even worse, on the wrong prediction, the attacked models show strong confidence.

1.1.3 Unsupervised Learning

Data without labels comes under the area of unsupervised learning. Unsupervised learning have goals to learn some underlying hidden structure of the data since we are devoid of the labels. Examples are clustering, association, density estimations, etc.

1.1.4 Generative Models

It belongs to unsupervised domain of machine learning. There are few variations of generative models such as variational auto-encoders and generative adversarial networks. Generative models try to generate new samples from the data distributions. Basic idea is to take a collection of training examples and form some representation of probability distribution that explains where these training examples come from.
Variational auto-encoders are a type of auto-encoder(consists of endoers and decoders, few layers of CNN, commonly used to compress and decompress the data respectively) which learns a latent variable model for its input data. Latent variables(example : underlying objects generating the shadows) are not directly observed but inferred from some other observed variables(example : shadow). The advantage of using latent variable is that it can serve the reduce the dimensionality of the data. Auto-encoders use deterministic variables whereas variational auto-encoders use stochastic variables for sampling.
Generative adversarial networks was pioneered by the research scientist, Ian Goodfellow in 2014 [3]. Yann LeCun described GANs as "the coolest idea in machine learning in the last twenty years". Previously neural networks were able to recognize objects , but now they can create and imagine things . Ever heard about deep dream

In GAN, two neural nets Generators and Discriminators compete for producing worlds eerily similar to our own in domains : images, music, speech, prose, etc. They can mimic any distribution of data. That's the reason, GAN's potential is huge [10].
Images generated by GANs are not only indistinguishable by humans but can also fool the image classifiers [10]. Carefully crafted adversarial images can do massive damage to the real-world systems which uses neural networks to address the problem. Neural networks are hackable. What if natural neural networks, e.g. our brains, are also vulnerable to such attacks? This is the very time to understand the insecurities of neural nets and secure them as early as possible.

1.2 Motivation

The usage of neural networks is becoming more and more popular with each passing day. Nowadays deep neural networks are being used everywhere but their security aspect is not being looked upon before using them. Some researchers have done the work in this field but it is not enough or their work is limited to a particular area of computers like computer vision. We in this paper approach towards a more fundamental and newly developed adversarial attacks and defenses that are currently present and assert them with wide range of practical examples. We adopt a comparable strategy as previous studies but with practical examples, without limiting ourselves to particular apps and also in a more elaborate way. Our main focus is to make everyone aware of the vulnerabilities present in the deep neural network models and not to take benefit from these adversarial attacks.

1.3 LITERATURE REVIEW

We have studied various resources available including online documents and books. The huge collection of reports helped us in our deeper understanding of adversarial attacks and its destructive applications in real life environment. After exploring various areas in machine learning involving security, we realized that neural networks are extensively used in many diverse areas, involving vision domains, speech, text, interesting business problems, understanding and identifying the diseases, and of-course self-driving vehicles. Since the area of deep learning is not very mature and while applying the neural network model to solve the problems, we don't think about the security aspect regarding neural networks. The fact is: neural networks are quite vulnerable and can be easily fooled to go wrong.

1.4 PROBLEM STATEMENT

This section clearly explains the problem we would like to address and solve.

(i) **Non-targeted Adversarial Attack:** The aim is to attack the model in such a way that it wrongly predicts the input image, not specifically belonging to any particular class [11].

(ii) **Targeted Adversarial Attack:** The aim here is to attack the model in such a way that the input image will be wrongly classified as a specific target class.

(iii) **Defenses Against Adversarial Attack:** Although throughout the paper the aim is to find security threats and vulnerabilities but to make the model robust and accurate against adversarial examples, we will be showing few defenses.

Adversarial attacks are used to fool the neural network model. Some of the famous techniques that we have used to perform the adversarial attacks on deep network are listed below : Fast Gradient Sign Method (FGSM), Iterative-FGSM, L-BFGS.
We also implemented the state of the art adversarial form of attack known as momentum based iterative FGSM .

To tackle some of the basic attacks, there are few defenses for the neural network model [1]: such as adversarial training, gradient masking, guided denoising, defense-GAN etc. [10][12].

1.5 OBECTIVE

The goal of this project is to create an adversarial attack on deep learning model by modifying input image such that the image will be classified incorrectly either to a known targeted class or an unknown class.

Further we will show some defenses against adversarial attacks to make the model robust to refrain from such attacks so that it can classify adversarial images correctly. Just to point out, not all the attacks we have shown can be defended by the defenses we have shown. There is a need of more robust training to defend, which is an open area of research and still in very early ages.

In this report, we will be attacking the ImageNet model by carefully generating perturbed images from the standard datasets. Finally, we will show how cloud-based machine learning systems by aws, azure and clarifai.com can be fooled with adversarial examples generated by our model using black-box attacks.

CHAPTER 2

DESIGN DETAILS AND IMPLEMENTATION

This chapter includes the details of some of the key algorithms, models and techniques included in our project. Various algorithms used for adversarial attacks and defense are provided. The first two sections provide the datasets used and the victim deep learning models

2.1 DATASETS

To evaluate adversarial attacks, we are going to use these image datasets :
MNIST : It contains gray-scale images of handwritten digits. Each image in the data set contains 784 (28*28) pixels having value between 0 and 255. There are total of 28000 images.
ImageNet : It is a large database containing more than 14 million images and has more than 20,000 categories.
LFW : Labeled Faces in the Wild (LFW) is a database of face photographs, was created and maintained by researchers at the University of Massachusetts. It contains 13,233 images of 5,749 people were collected from the web. 1,680 of the people pictured have two or more distinct photos in the dataset. The original database contains four different sets of LFW images and also three different types of "aligned" images.

2.2 VICTIM DEEP LEARNING MODELS

Deep learning models such as VGG, ResNet, Inception v3 (trained on ImageNet dataset),facenet can be attacked. These models are highly accurate classifiers and they can achieve maximum accuracy of more than 99%. These deep learning models are still susceptible to the adversarial attacks.

Inception v3 model is trained on ImageNet competition data built upon the previous Inception v1. It has 311 number of layers and 23,885,392 parameters in total.
Facenet model is trained on MS-Celeb-1M dataset and input images must be colored and a square shape of 160 by 160 pixels.

2.3 ADVERSARIAL PERTURBATION

Adversarial perturbation is the noise which on adding to the target image makes it an adversarial example and vulnerable to any form of attack [7]. Perturbations can be introduced with the help of generative model. There are image-dependent perturbations, which can be achieved with the help of FGSM, iterative-FGSM etc. Then there is also universal perturbation which can be applied to most of the images to misclassify a model that is pre-trained on any dataset. It utilizes an algorithm that works by iterating over the target images and by analyzing image-dependent perturbations it normalizes the outcome obtained and slowly generates the universal perturbation.

2.4 Threat Model

Threat modelling involves identifying security threats in some sequential order and rating them according to their severity and their frequency of occurrence. Many threat models have been developed and have been integrated to understand more potential threats.
The most famous threat model is STRIDE, used by MICROSOFT in past(they now use dread), developed in 1999. It evaluates the design of system by building data flow diagrams and identifies system elements, events, and boundaries of the system. Stride stands for spoofing identity, tampering with the data, repudiation, information disclosure, denial of service, elevation of privilege. It can be potentially applied to security systems such as face detection, and for each mnemonics, threat definition can be easily defined.
In the context of defense, the threat model outlines how many different kind of attackers, the defense intends to defend against. Typically, a threat model assumes about the adversary's goals, knowledge and capabilities. In this paper, we assume that attackers can have complete knowledge of target models to be attacked in case of white-box attacks. Whereas gradually increasing towards more severe attacks, the attacker are prohibited from knowing the gradients of the model to knowing absolutely nothing but just outputs.
We focused on the threat models where :

1. Adversary's goals : Targeted as well as non-targeted attacks.
2. Adversary's knowledge : White-box as well as black-box models.
3. Victim's model : Mostly deep neural networks.

2.5 ATTACKS

To delve deeper into some specific attacks, we need to understand some terminologies and high level attacks.

2.5.1 White-Box attack

In this attack, the adversary knows almost everything related to the trained model. They know about the dataset on which the model is trained. They know about the architecture of the model, model weights, no of hidden layers, activation functions. Most of the adversarial attacks that are performed are white box attacks because these are easier to perform than black-box attacks.

2.5.2 Black-Box attack

Contrary to the white-box attacks, here there is no access to the targeted model for the attacker. The only thing known to the world is the output or confidence score of the model. This one can be used for attacking system,which provides machine learning as a service (MLaaS). For example : AWS, Google Cloud AI and many other big organizations provide MLaaS.

2.6 ADVERSARIAL ATTACK METHODS

Let's see a few adversarial attacks in detail :
Notion : The idea is very simple. Just like we train the neural network itself , we will do exactly the same thing to create special perturbations, except that instead of optimizing all the variables in the neural network , we are optimizing the noise overlaying the input image. The first three attack comes under white-box settings and rest two are black-box attacks.

2.6.1 BOX-CONSTRAINED L-BFGS

L-BFGS is an optimization algorithm which recognizes adversarial attack as an optimization problem.

Conventions :

Input example : x
Model : f
Perturbation : r
Input domain : D
Target label : l

Szegedy coined the term adversarial sample and he posed the following minimization problem "":

$$\arg\min f(x + r) = l \text{ s.t. } (x + r)\, \epsilon\, D$$

It is comparable to FGSM, but it does not try to use the measured gradient immediately as an implied disturbance. Instead, to discover an update to an input that significantly improves an objective function, it recognizes adversarial attack as an optimization problem. Designing this as an optimization problem enables us to be flexible in folding more adversarial criteria into the objective function. It is a non-linear gradient based numerical optimization algorithm.

2.6.2 FAST GRADIENT SIGN METHOD (FGSM)

Conventions :

Normal sample : X
Cost function of the trained model : J
Input variation parameter : ϵ
Gradient of the model with respect to X with label Ytrue : ∇x

Goodfellow calculated the cost function gradient with repsect to the neural network input. The following equation is used to generate the adversarial examples [1]:

$$X_{Adversarial} = X + \epsilon.\text{sign}(\nabla_x J(X,Y))$$

Epsilon is the pixel-wise perturbation amount, the value of which must be effective but imperceptible and not too obvious.
This attack is based on the concept of disturbing the input in a manner that changes the model's loss function to the maximum. So, in order to calculate the derivative of the loss function with regard to its input, we need to conduct back propagation. FGSM searches for the manner in which a target machine learning model improves the loss

function as quickly as possible. Here, to conduct back propagation, the attacker seeks to understand the architecture and variables of the framework. Once the gradient is calculated, a tiny amount can be used to push the input towards the adversarial gradient.

2.6.3 ITERATIVE - FGSM

Iteratively apply fast gradient multiple times with a small step size Îś.

The iterative version of FGSM (I-FGSM) can be expressed [**?**]

$$x_0^* = x$$
$$x_{t+1}^* = x_t^* + \alpha.\text{sign}(\nabla_x J(x_t^*, y))$$

Set $\alpha = \epsilon/T$ with T being the number of iterations. It has been shown that iterative methods are stronger white box adversaries than one-step methods at the cost of worse transferability.

Obserations :

Iterative-FGSM or in general algorithms based on optimizations and iterative forms have poor transferability across different models, and therefore they decrease the effect of the black-box attacks [6]. The reason is: iterative FGSM takes greedy approach and hence brings the adversarial image towards the gradient sign in each of the iteration. As a result of which, the adversarial examples fall into poor local maxima and "overfit" the model. So adversarial examples crafted using these algorithms are not likely to transfer across models.

To overcome the dilemma, it is necessary to stabilize the update directions of gradient and escape from local maxima. To achieve this, a factor of momentum is integrated in the existing algorithms, which results in not only good result in white-box attacks but also quite satisfactory result in black-box attacks.

2.6.4 SUBSTITUTE BLACK-BOX ATTACK

This attack requires an approximation of how a model behaves on various inputs. Just like a psychologist asks many questions to a patient to understand the patient's behaviour as per their response. We are essentially approximating the decision boundary of the black-box model that we want to attack [2]. We are going to train a substitute model on a dataset, an artificial one, pretty similar to the dataset on which the black-box model is trained on. We will end up getting the labels for the artificial dataset, which will come from the prediction of black-box model(target model).

But there is a problem, we can't make unlimited query to the target model in the real world. But it can be solved using data augmentation, which involves calculating the

gradients of the label predicted by target model with respect to the artificial inputs in order to generate new augmented additional samples. But we don't know anything about the target model, hence the parameters of the substitute model will be used to calculate the gradients [2]. Now the substitute model is trained on the artificial dataset

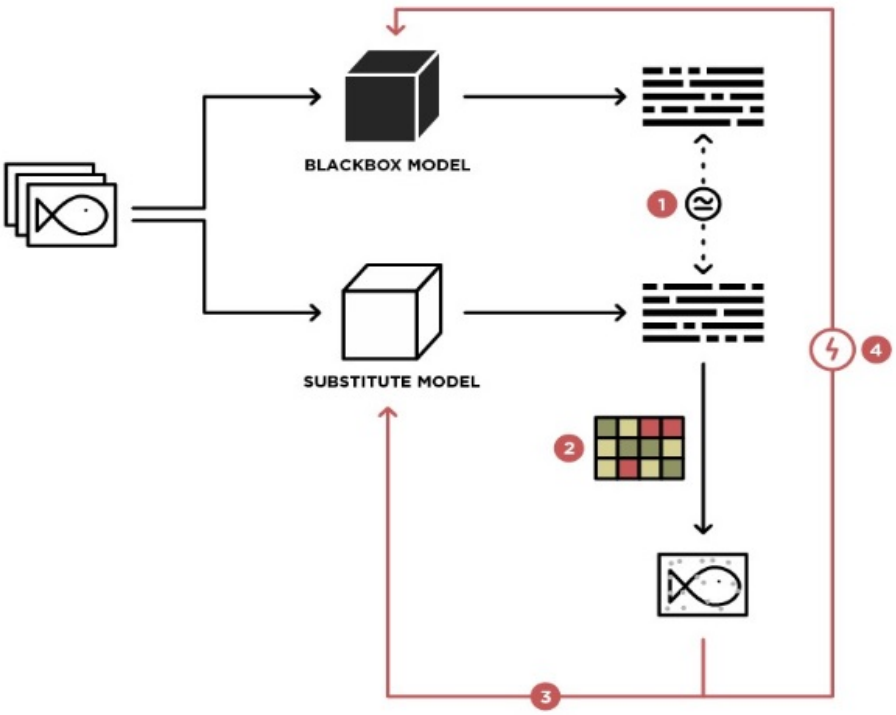


Figure 2.1: Attack on Black Box Model [2].

using label as per the prediction by querying the target model. Once we have trained the substitute model, we generated adversarial examples that can fool the substitute model using white-box attack techniques, since we know all about the substitute model. These adversarial examples areÂăbe used to confuse the black-box model as adversarial examples can be transferred.

Some defenses, such as defensive distillation, just rely on gradient masking. Substitution based attacks are resistant of gradient masking.

2.6.5 MOMENTUM BASED ITERATIVE FGSM (*state of the art*)

2.6.5.1 Description

Contrary to the white-box attacks, here the attacker has no access to the model's structures ,weights and parameters The only thing known to the world is the output or confidence score of the model. For example, attacking a machine learning model via an API is considered a black box attack since we can only do certain things like : provide

different inputs and observe the outputs "".

This technique can be used for attacking system,which provides machine learning as a service (MLaaS). For example : AWS, Google Cloud AI and many other big organizations provide MLaaS.

Consider an interaction between a psychologist(attacker) and a patient (model), where the psychologist queries to patient and asks a variety of questions and analyze the behavior of the patient based on her responses. That's exactly what happens in black-box settings.

2.6.5.2 Need

Using the conventional optimization algorithms like FGMS and Iterative FGMS to attack a machine learning model on which the adversarial examples are not trained produces very low success rates. The reason is : the adversarial examples generated using these algorithms have less transferability across models. The reason is : the adversarial examples fall into poor local maxima and "overfit" the model due to the greedy nature of optimization algorithms. We need to somehow escape the local optima.

2.6.5.3 Why momentum iterative FGSM demonstrates better transferability?

The transferability stems from the concept that the model learns comparable boundaries of choice around a data point. Although the boundaries of the decision are similar, owing to the strong non-linear structures of the DNNs they are highly unlikely to be the same. So some outstanding boundaries of decision may occur around a data point for the model that is difficult to switch to other models. In the optimization phase, these areas correspond to bad local maxima, and iterative methods can readily trap in such areas, leading to a less transferable examples of adversaries [13]. The stable update instructions acquired through the momentum techniques, on the other side, can assist to escape the outstanding areas, leading in better transferability of the adversarial attacks.

2.6.5.4 Algorithm

Input : A loss-function classifier f with the loss function J; a sample example x and ground-truth label y ; the size of the disturbance ϵ ; iterations T and decay factor μ [14][13].

Output : An adversarial example x* with $||x^* - x||_\infty \leq \epsilon$
1. $\alpha = \epsilon/T$;

2. $g_0 = 0$; $x_0* = x$;

3. for t = 0 to T - 1 do
   a. Input x_t* to f and obtain the gradient $\nabla_x J(x_t* , y)$;
   b. Update g_{t+1} by accumulating the velocity vector in the gradient direction as :

$$g_{t+1} = \mu.g_t + \frac{\nabla_x J(x_t*, y)}{\|\nabla_x J(x_t*, y)\|_1} \tag{2.1}$$

   c. Update $x_{t+1}*$ by applying the sign gradient as:

$$x_{t+1}* = x_t * + \alpha.sign(g_{t+1}); \tag{2.2}$$

4. end for

5. return $x* = x_T*$.

We accumulate the gradients of the first t iterations with the decay factor. The adversarial example x_t* until the t-th iteration is perturbed in the direction of the sign of g_t with a step size alpha. If the decay factor equals to 0, MI-FGSM boils down to the iterative FGSM.

2.7 DEFENSES

The defenses against adversarial attacks performed on the model are:

2.7.1 Adversarial Training

Adversarial learning is a defense method used to re-train a model in order to enhance adversarial robustness on adversarial examples [15]. One of the simplest and brutal ways of defending against these attacks is to pretend to be the attacker, create an amount of opposing examples against your own network and train on them.

Naive adversarial learning on a restricted set of adversarial examples can lead to incorrect robustness owing to gradient masking [16]. As a consequence, the model being defended will only be robust against certain kinds of assaults, but can still be circumvented by various techniques of assault.

A better adversarial training approach would be the one that does not depend on gradient masking by training the model only on the strongest adversaries available (i.e.,

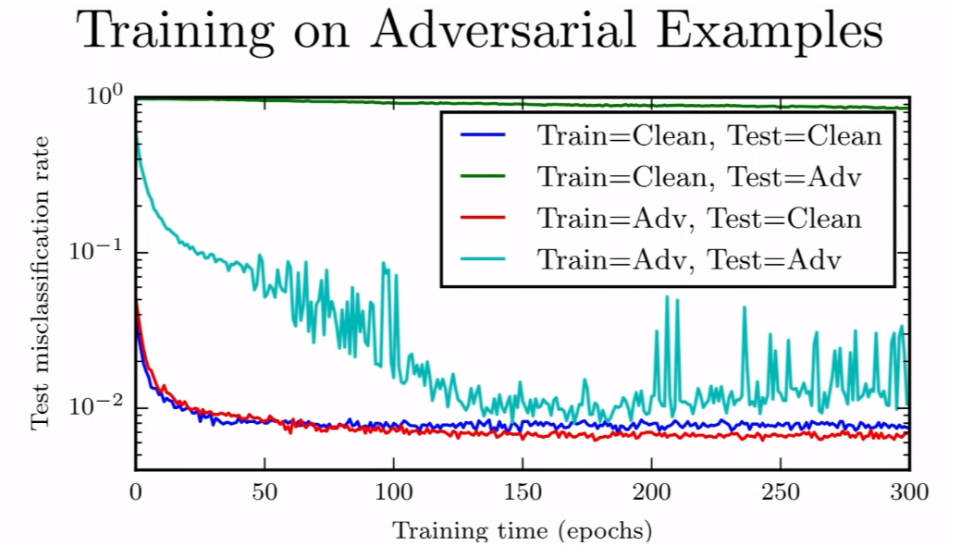


Figure 2.2: Result of adversarial training [3]

.

worst-case adversaries).

2.7.2 Gradient Masking

To create an attack, most adversarial instance generation methods use the model's gradient. For instance, they look at an aeroplane's image, they test which direction in image room increases the likelihood of the "cat" class, and then they offer some disturbance in that direction. False recognition of the fresh, altered picture as a cat. Gradient masking methods require a model that has no helpful gradients, such as using a closest neighbor classifier rather than a profound neural network. Due to the lack of a gradient, such a technique makes it hard to build an adversarial example immediately, but is often still susceptible to the adversarial examples that influence a smooth representation of this same model.

But in defenses, there is a particular defect based on gradient masking. Even if the defender tries to avoid assaults by not releasing the instructions in which the model is vulnerable, it is possible to discover these instructions by other means, in which case the same attack can still succeed. The black-box attack based on a replacement model transfer overcomes the defenses of gradient masking.

2.7.3 Guided Denoiser

This solution is based on the observation that they are amplified all across the network ignoring adversarial disturbances being really low at the pixel stage, creating an adversarial assault. Several higher-level denoisers are suggested to address this challenge: a

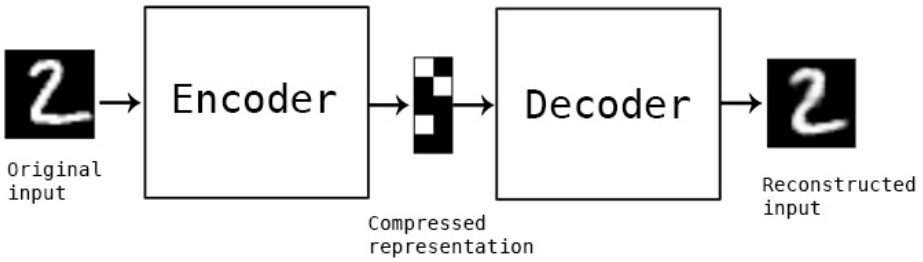


Figure 2.3: this is how denoiser works [4].

directed denoiser function [4], a logits guided denoiser (LGD) and a class label guided denoiser (CGD).

Auto-encoders are good for data denoising and dimensionality reduction. With the help of an encoder network, decoder network and a distance function between the amount of information loss between the compressed representations and decompressed representation of your data. We start by using an auto-encoder to compress the high-dimensional data into low-dimensional data and learn the latent features then decompress again to remove the unnecessary noise.

CHAPTER 3

RESULTS AND DISCUSSION

This chapter includes the results obtained from adversarial attacks using various algorithms. It also provides a table for comparison between these attacks and formulate which one is best suited for different models.

3.1 Generation of the perturbed image

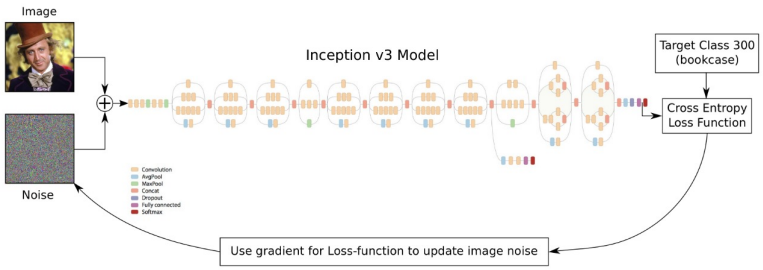


Figure 3.1: Inception V3 Model [5]

To generate perturbations, we add some random noise to the original image.
(Here we are considering noise as the D-dimensional vector, the initial value of which is zero. D is the total dimension of input image.)
We provide the image to the model for training. The gradient from the additional loss function with respect to noisy input image is used to update the noise vector. Gradient tell us how we need to modify the noise in order to move the classification towards the desired class.

We find the most optimal value of gradient for which the perturbed image is neither more imperceptible nor too obvious. Eventually we get the perturbed image (in the center) which is really indistinguishable from the original details, but it contains the specific noise to fool the model.

3.2 Non-targeted attack on a model trained on MNIST datasets

When epsilon is zero, it means no perturbations and the model predicts the correct class for each image. As we increased epsilon, the perturbation increases and model started predicting wrong class for the images [17]. At epsilon = 0.3 the noise became apparent and prediction will be really inaccurate.

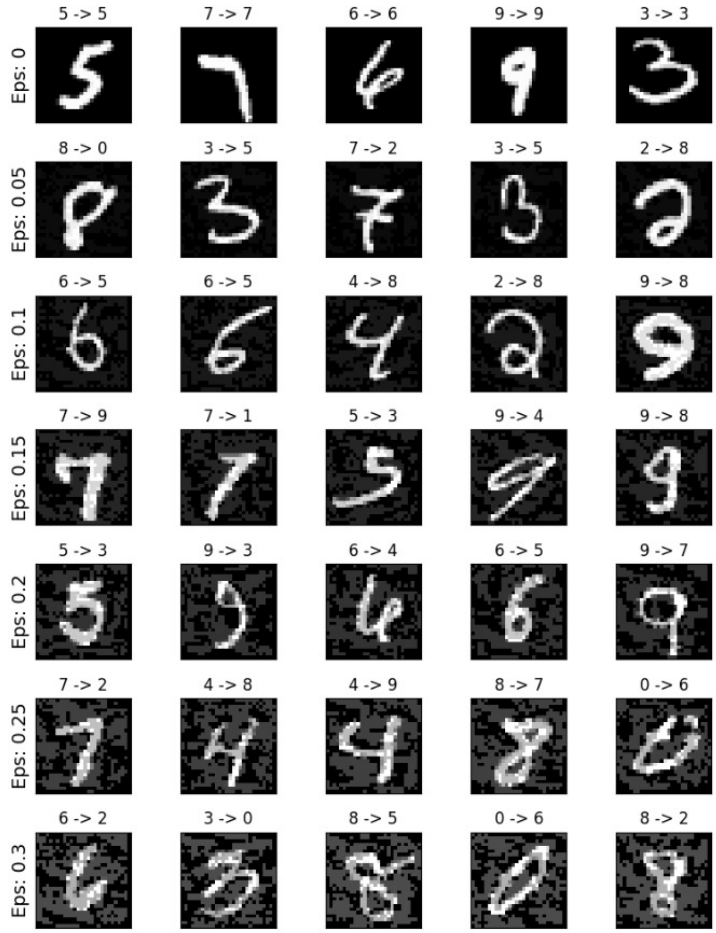
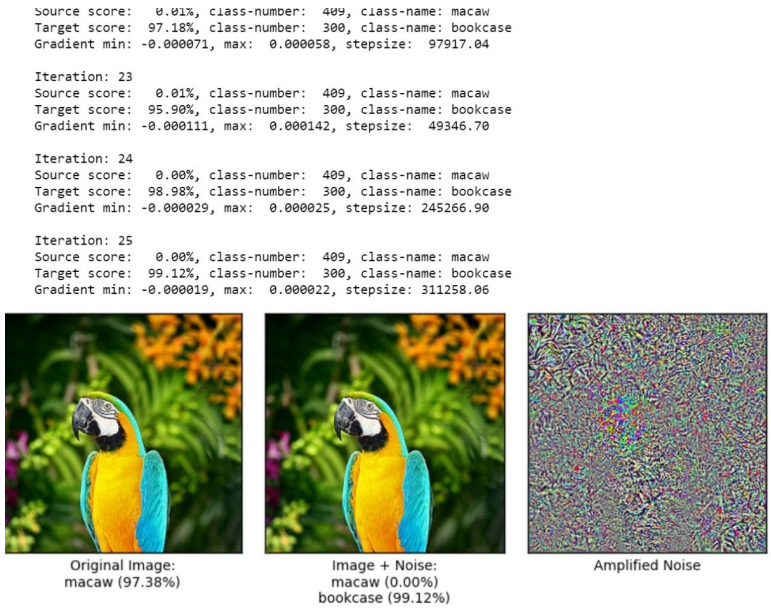


Figure 3.2: Original and generated Adversarial samples on the MNIST dataset [1].

3.3 Targeted attack on inception model trained on ImageNet datasets

It is slightly different from the previous case. Here we are forcing the model to misclassify the input images to a specific class. Initially we started from random noise, and the confidence score was very less for the targeted attack, which is good [7]. After each iteration, the source confidence score gradually decreased and target confidence score

kept on increasing.
The terminating conditions :
a) The training was stopped when the target confidence score crossed 99% barrier.
b) Also maximum number of iterations is 100.
c) Also if the noise level crosses the limit 3.0 , we will stop the iteration.



It means, now the model is almost 0% accurate and more than 99% inaccurate. We can force the model to misclassify the input images to whatever class, we want, apparently. This is the real threat [18].

Observations :

- The attacks based on FGSM algorithm is fast and easy to perform but it is not very effective.
- To enhance the magnitude of attacks, the iterative-FGSM algorithm(applying fast gradient with a small step of size α in a iterative fashion multiple times for certain number of iterations) is used [5]. The success rate of targeted as well as non-targeted attack, that has been achieved is 100%.

3.4 Results of Black Box Attack

3.4.0.1 Performance

Even the most robust models are vulnerable to black-box attacks using MI-FGSM [6]. If we go for ensemble approach, we can substantially increase the success rate for

black-box attacks.

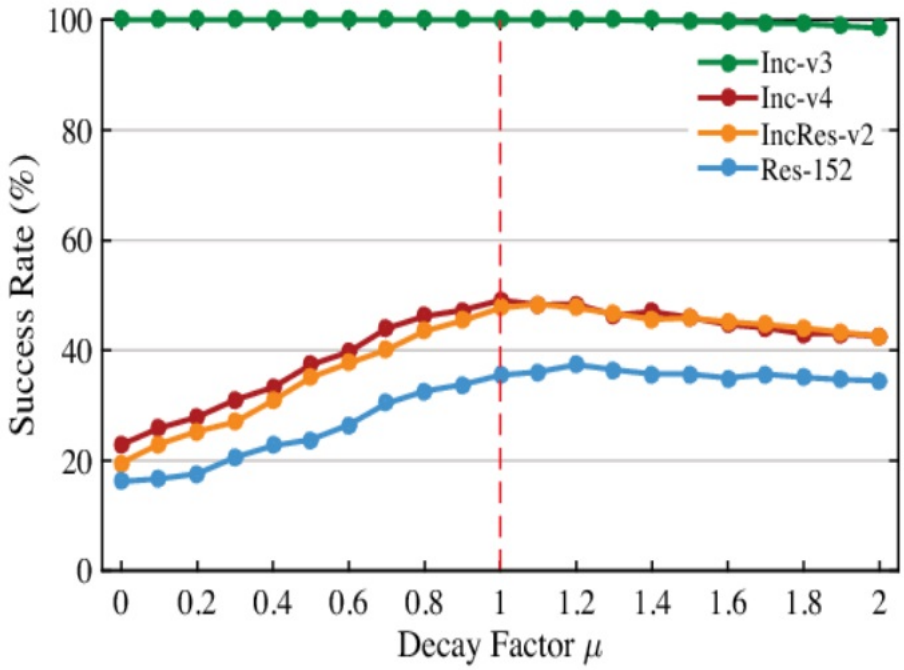


Figure 3.3: Graph representing the success rate generated for black box model against white box [6].

Table 3.1: Success Rates of attacks using various algorithms is depicted below :

Trained On	Attacks	Inc-V3	Res-152	IncRes-v2
Inc-V3	FGSM	72.3%	25.3%	26.2%
	Iterative-FGSM	100%	16.2%	19.9%
	MI-FGSM	100%	35.6%	48%
Res-152	FGSM	35.0%	72.9%	
	Iterative-FGSM	26.7%	98.6%	21.2%
	MI-FGSM	100%	98.5%	44.7%

3.4.0.2 Impact

The adversarial examples generated by MI-FGSM using LFW dataset are able to fool clarifai.com. So as an exercise, using this algorithm, we trained a model on LFW dataset. We are completely unaware of the model used by clarifai.com and another vision platform such as azure, AWS vision platform, but the adversarial examples can fool their predictions [19]. We have demonstrated the same in the result section. Applications involving face detection, such attacks can cause serious security damages. A

more serious training using extensive resources can perform targeted as well as non-targeted attack with much surety [20][19].

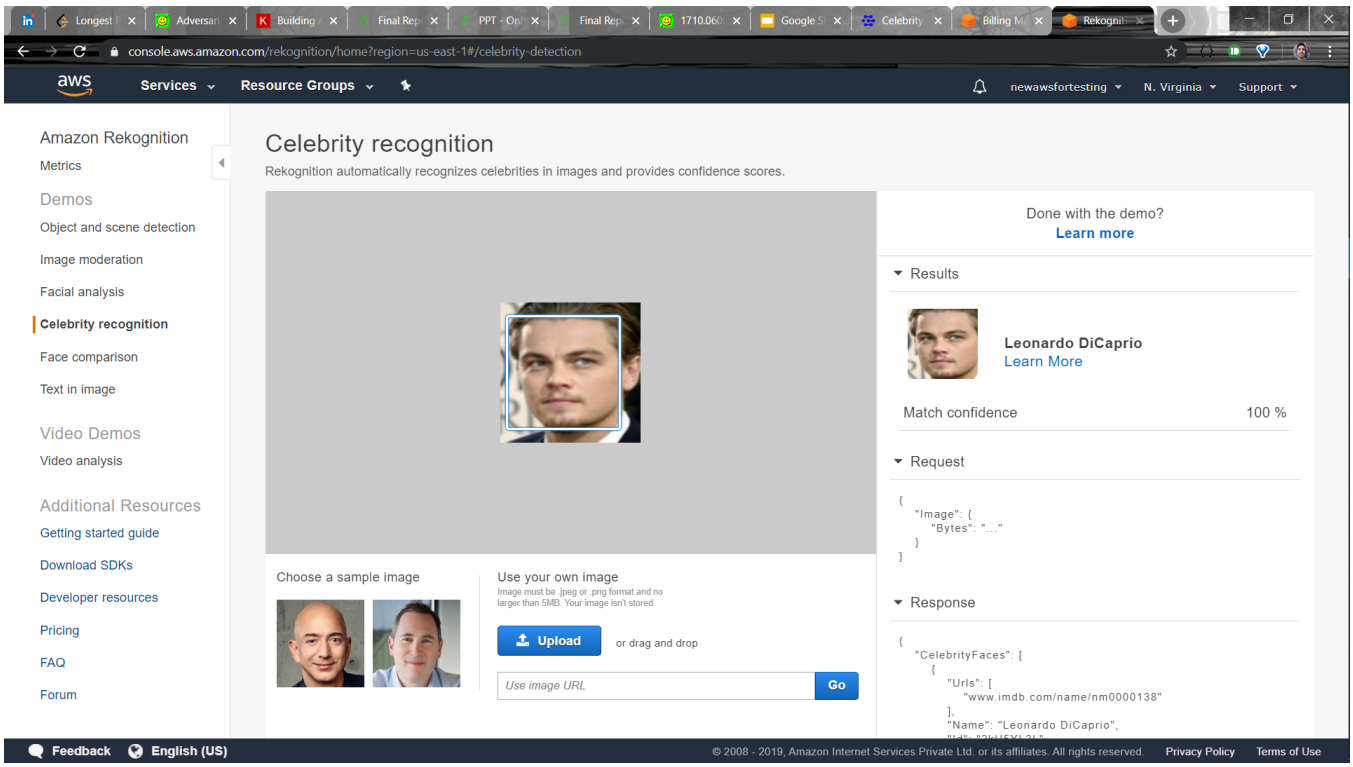


Figure 3.4: AWS vision platform recognises the person as Leonaro Dicaprio, which is true.

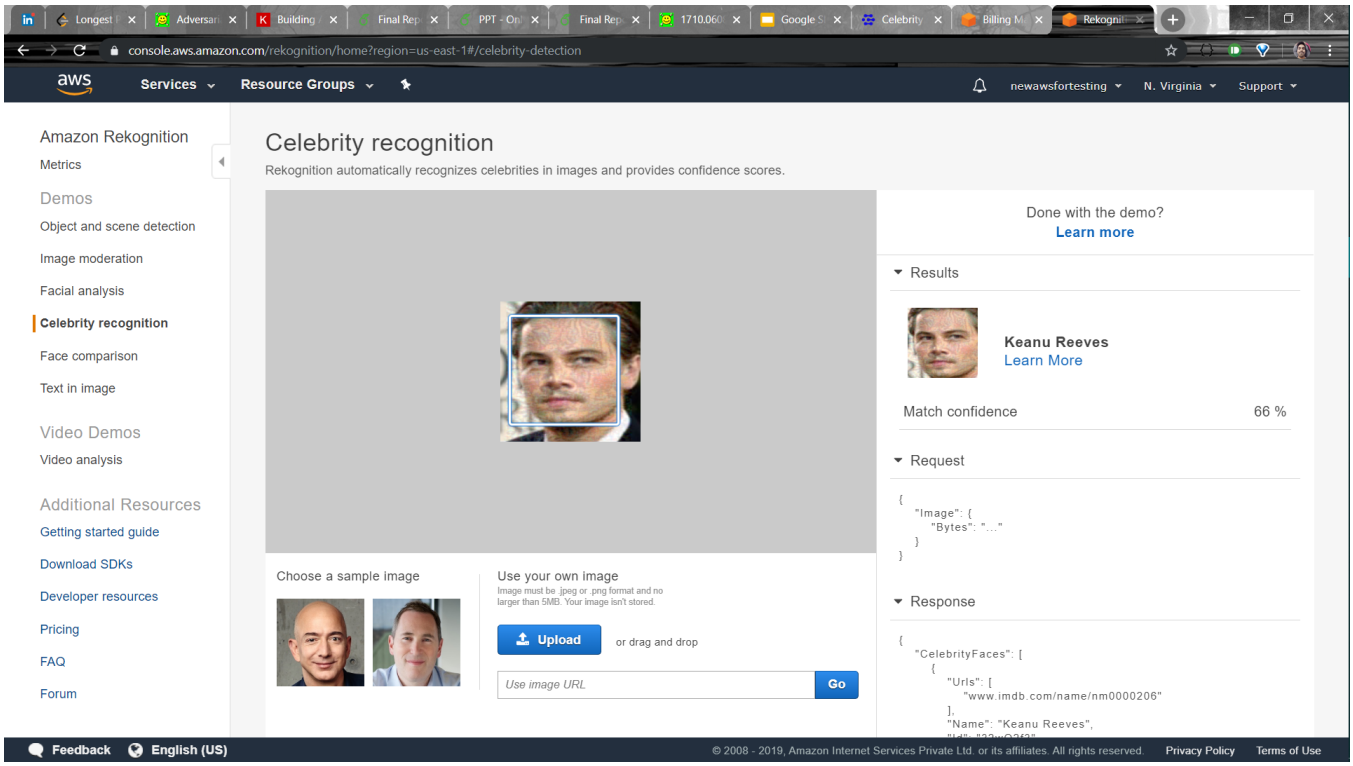


Figure 3.5: After running MI-FGSM algorithms on the same image, it categorizes the celebrity as Matt Damon.

CHAPTER 4

CONCLUSION

One step based method such as FGSM algorithm assume a linear decision boundary in the input space, which is not suitable for practical attack scenarios. The effectiveness of FGSM can be increased to some extent by applying FGSM in iteration. It just works like neural network optimization, and because of the fact that the greedy nature of optimization algorithms overfit the model and tends to fall into poor local maxima. Adversarial examples generated by optimization algorithms are not likely to transfer across models. For simple black-box scenario, in cases when gradients are restricted to outsider, substitution based attack might work. But if we are completely unaware of the model's parameter, input, weights etc, we need to skip the poor optima. So by associating a velocity parameter, adversarial examples can attack black-box models [6]. Robust models can also be fooled using this technique.

Hence there is a need to develop suitable defense methods against these adversarial attacks. There is an arms race going on between adversarial attacks and defenses. A lot of defense methods were proposed recently but none of the defense methods can guarantee against all types of adversarial attacks. That's why there are really a lot of defenses against different kinds of attacks. One of the most basic and effective defense methods is Adversarial Training , which is extensive training the model for lot of adversarial examples. But it consumes a lot of computation resources and still there may be some variations against which the model might not be trained for. Gradient Masking acts as a proper defense for white-box models. Even for attacks where gradient is hidden or there are models which does not use gradient, it works really well. But if adversary figures out the gradient change, gradient masking no longer works.Then there is another defense based on Bayesian uncertainty estimates which differentiates between clean and adversarial examples . Randomization of gradients of neural networks [21], Defensive distillation, introducing randomness, data augmentation are another techniques of defenses for different attack settings.

REFERENCES

[1] Z. Yin, S. Chawla, and W. Liu, *Adversarial Attack, Defense, and Applications with Deep Learning Frameworks*, vol. 1. 2019.

[2] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, "Query-efficient hard-label black-box attack:an optimization-based approach,"

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2014.

[4] F. Liao, M. Liang, Y. Dong, and X. H. Tianyu Pang, "Defense against adversarial attacks usinghigh-level representation guided denoiser," *Arxiv*, vol. 1712.02976v2, 2018.

[5] A. Kurakin, I. J. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, J. Wang, Z. Zhang, Z. Ren, A. L. Yuille, S. Huang, Y. Zhao, Y. Zhao, Z. Han, J. Long, Y. Berdibekov, T. Akiba, S. Tokui, and M. Abe, "Adversarial attacks and defences competition," *ArXiv*, vol. abs/1804.00097, 2018.

[6] N. Papernot, P. D. McDaniel, and I. J. Goodfellow, "Transferability in machine learning: from phenomena to blackbox attacks using adversarial samples," *ArXiv*, vol. abs/1605.07277, pp. 1–13, 2016.

[7] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision:a survey," 2018.

[8] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," *2016 IEEE European Symposium on Security and Privacy (EuroSP)*, pp. 372–387, 2015.

[9] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 2805–2824, 2017.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial networks," *ArXiv*, vol. abs/1406.2661, 2014.

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *ArXiv*, vol. abs/1706.06083, 2017.

[12] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, 2015.

[13] Bektas, Sebahattin, Sisman, and Yasemin, "The comparison of l1 and l2-norm minimization methods," *International Journal of the Physical Sciences*, vol. vol.5, 2017.

[14] A. Mustafa, S. K. Ling Shao, M. Hayat, R. Goecke, and J. Shen, "Adversarial defense by restricting the hidden spaceof deep neural networks," *ArXiv*, vol. 1904.00887v4, 2018.

[15] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *ArXiv*, vol. abs/1810.00069, 2018.

[16] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, "Ensemble adversarial training: Attacks and defenses," *ArXiv*, vol. abs/1705.07204, 2017.

[17] Y. LeCun and C. Cortes, "The mnist database of handwritten digits," 2005.

[18] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *ArXiv*, vol. abs/1607.02533, 2016.

[19] D. Goodman, T. Wei, and B. X-Lab, "Cloud-based image classification service is not robust to simpletransformations:a forgotten battlefield," *International Journal of the Physical Sciences*, vol. 1906.07997v1, 2017.

[20] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," pp. 9185–9193, 2017.

[21] J. Atif, C. Gouy-Pailler, A. A. Laurent Meunier, H. Kashima, and F. Yger, "Theoretical evidence for adversarial robustness through randomization," *ArXiv*, vol. 1902.01148v2, 2019.