KLE Society's
KLE Technological University

A Mini Project Report

On

Defense against Adversarial Perturbations based Attacks for Deep Neural Networks

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering

In

Computer Science and Engineering

Submitted By

Varsha Chalageri	01FE19BCS033
Ankita Mane	01FE19BCS052
Bhavana Kumbar	01FE19BCS244
Neha Kardant	01FE20BCS422

Under the guidance of
Mr. Shashidhara Vyakaranal.

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

HUBLI–580 031 (India).

Academic year 2021-22

KLE Society's
KLE Technological University

2021 - 2022

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Mini Project entitled "Defense against Adversarial Perturbations based Attacks for Deep Neural Networks" is a bonafide work carried out by the student team Ms. Varsha Chalageri - 01FE19BCS033, Ms. Ankita Mane - 01FE19BCS052, Ms. Bhavana Kumbar - 01FE19BCS244, Ms. Neha Kardant - 01FE20BCS422, in partial fulfillment of completion of Fifth semester B. E. in Computer Science and Engineering during the year 2021 – 2021. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said programme.

Guide **SoCSE Head**

Mr. Shashidhara Vyakaranal **Dr. Meena S.M**

External Viva:

Name of the Examiners Signature with date

1.

2.

ABSTRACT

Deep neural networks have demonstrated remarkable success in solving various complex tasks that were difficult to solve in the past using the conventional machine learning methods. The need for deep learning algorithms and its applications is increasing in daily life. It has become an inevitable part for most of the applications in the present day scenarios. However, recent studies have revealed that the deep neural networks are unfortified to the present adversarial attacks. The addition of imperceptible perturbations to the inputs cause the neural networks to fail and predict incorrect outputs. In practice, adversarial attacks pose a major challenge towards the success of deep learning by affecting the performance of the deep learning models. Thus, it becomes very essential to provide robustness to the deep learning models. In this project, we attempt to implement and provide an overall discussion about some of the adversarial attacks and certain countermeasures against them.

Keywords: Deep neural networks, Adversarial examples, Adversarial attacks, Adversarial defense

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of some individuals whose professional guidance and encouragement helped us in the successful completion of this report work.

We take this opportunity to thank Dr. Ashok Shettar (Vice-Chancellor, KLE Technological University, Hubli) and Dr. Prakash Tewari (Dean of Academic Affairs, KLE Technological University, Hubli).

We also take this opportunity to thank Mr. Shashidhara Vyakaranal (Assistant Professor, School of Computer Science and Engineering) who is our guide for having provided us with an academic environment that nurtured our practical skills contributing to the success of our project.

We sincerely thank Mr. Mahesh Patil and Mr. K.M.M Rajashekharaiah Mini Project Coordinators for their support, inspiration and wholehearted cooperation during the course of completion.

We sincerely thank Dr. Meena S. M. HoS, School of Computer Science and Engineering for her support, inspiration and wholehearted cooperation during the course of completion.

Our gratitude will not be complete without thanking the Almighty God, our beloved parents, our seniors and our friends who have been a constant source of blessings and aspirations.

Varsha Chalageri - 01FE19BCS033
Ankita Mane - 01FE19BCS052
Bhavana Kumbar - 01FE19BCS244
Neha Kardant - 01FE20BCS422

Chapter No. **TABLE OF CONTENTS** **Page No.**

1. INTRODUCTION **1-7**

1.1 Motivation **2**

1.2 Literature Survey **2**

1.3 Problem Definition **6**

1.4 Applications of Proposed System **6**

1.5 Objectives **7**

1.6 Scope and constraints **7**

2. REQUIREMENT ANALYSIS **8-9**

2.1 Functional Requirements **8**

2.2 Non-Functional Requirements **8**

2.2.1 Performance requirements **9**

2.3. Software and Hardware requirement specifications **9**

3. SYSTEM DESIGN **10-14**

3.1 Architecture of the system **10**

3.1.1 Attack Phase **10**

3.1.2 Defense Phase **11**

3.1.3 High Level architecture Design **12**

3.2 Detailed Design **13**

4. DATASET **15-18**

4.1 Description **15**

4.2 Data Preprocessing Techniques **18**

5. IMPLEMENTATION **19-26**

5.1 Proposed Methodology **19**

5.1.1 Adversarial Training **19**

5.1.2 Defensive Distillation **20**

5.1.3 FGSM Attack **22**

5.2 Evaluation Metrics **25**

6. RESULTS AND DISCUSSIONS **27-28**

7. CONCLUSION **29**

8. BIBLIOGRAPHY **30**

Chapter 1

INTRODUCTION

Deep neural networks are massive neural networks whose design or architecture is structured as a series of layers of neurons, each one of them representing an individual computing or logistic unit. They are generally connected by links with some different biases and weights along their way and transmit the result of the activation function on its input to the neurons of the immediate next layer. Deep neural networks usually mimic the biological neural networks of the human brains to gain understanding and build data from examples. Thus, they have the strength to touch upon sophisticated tasks that cannot be easily modelled as linear or nonlinear issues. With the evolution of deep neural network models, Deep learning has achieved a massive progress within the fields of classification of images, speech recognition, translation of languages, reconstruction of brain circuits etc. The use of deep learning applications can be mostly seen in security and safety crucial environments like self-driving cars, detection of malwares, drones, robotics etc. However, recent works show that deep neural networks are extremely at risk of tiny perturbations to the input image.

Usually, the addition of visually subliminal perturbations to the original input images may end up in the failures of image detection, object detection and semantic segmentation. These perturbed images are referred to as adversarial examples and such adversarial examples pose a great security risk to the deployment of commercial machine learning systems. Therefore, making deep neural networks more robust to the adversarial examples is very crucial and nevertheless a difficult task.

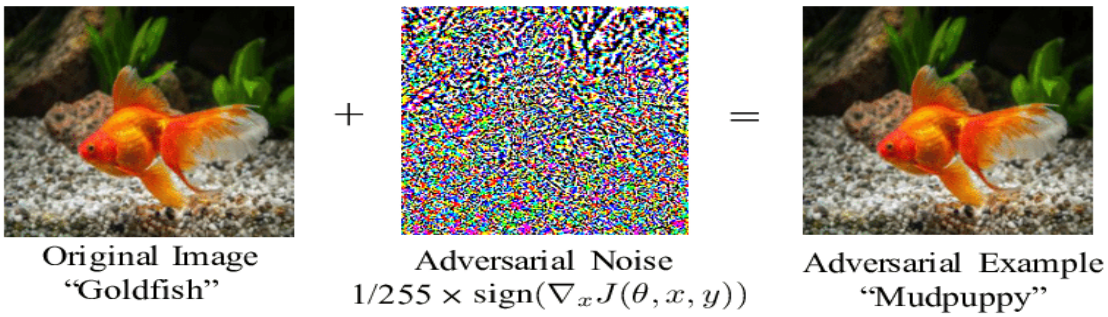

Figure 1: Example of an Adversarial attack

To mitigate or filter out the adversarial effects on the deep learning models, we propose appealing defense strategies against the adversarial attacks. Adversarial training appears to be more effective for providing robustness to the models by reducing the adversaries against some of the common adversarial attacks. By considering certain attack and defense strategies, a comparative study has been proposed to understand the nature and behaviour of the adversarial examples.

1.1 Motivation

The importance of deep learning and its applications are increasing day by day to a large extent. The recent advances in the field of machine learning and deep neural networks have enabled the researchers to resolve multiple complex and practical issues, especially considering the image or text classification tasks. However, recent studies revealed that deep neural networks are frail towards the adversarial examples and can easily get confused or affected by them. In most of the cases, these modifications can be so unnoticatable that a human observer too cannot recognise the modifications, yet the classifier misclassified the output.

Most of the adversarial attacks usually aim to deteriorate the performance of classifiers by fooling the deep learning algorithm. The image perturbations or adversaries can also fool multiple network classifiers which can be a serious threat affecting most of the models. Usually, the attacked models give more confidence on the incorrect prediction. The ardent implications of these results created an interest in the adversarial attacks and defense against them for the deep learning based applications.

1.2 Literature Survey

The common attack scenarios and the classification of attacks based on different criteria are described here and also summarizes the recent advancements in different types of the adversarial attacks and their countermeasures [1].

Common Attack Scenarios can be classified as:

• By the kind of outcome the adversaries desire:

- Non-Targeted Attacks
- Targeted Attacks

• By the amount of information or knowledge that the adversary has concerning the machine:
- White box
- Black box with probing
- Black box without probing

• By the manner in which the adversary will cater data into the model:
- Digital attack
- Physical attack

In Non Targeted attacks, the aim of the adversary is to classify and predict the false label while in case of the Targeted Attacks, the aim of the adversary is to change the prediction of the classifier to a specified target class.

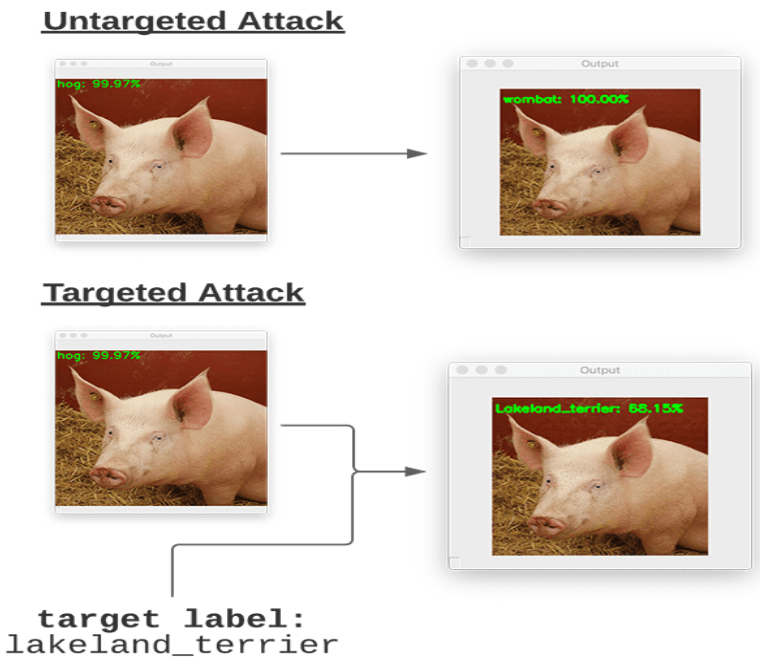

Figure 2: Targeted and Untargeted Attack example.

A survey on adversarial attacks and their defense mechanisms has been explained in this paper. The adversarial examples can be generated with the motivation of two goals specifically Attack and Defense while the adversarial goals which impact the output of a classifier can be confidence reduction of the target model and misclassification of output. The defenses used against these adversarial attacks are being developed either by training modification or by modifying the networks or by using external models. The strategies present under these categories can be divided as only detection or complete defense [2].

Deep neural networks nowadays have become more popular and have achieved success in solving many of the machine learning tasks with ease. They have been deployed commercially in different types of recognition problems such as in the domain of classification of images, graphs, text and speech recognition with astonishing success. The different strategies or countermeasures against Adversarial examples are Gradient masking/ Obfuscation and Robust optimization [3].

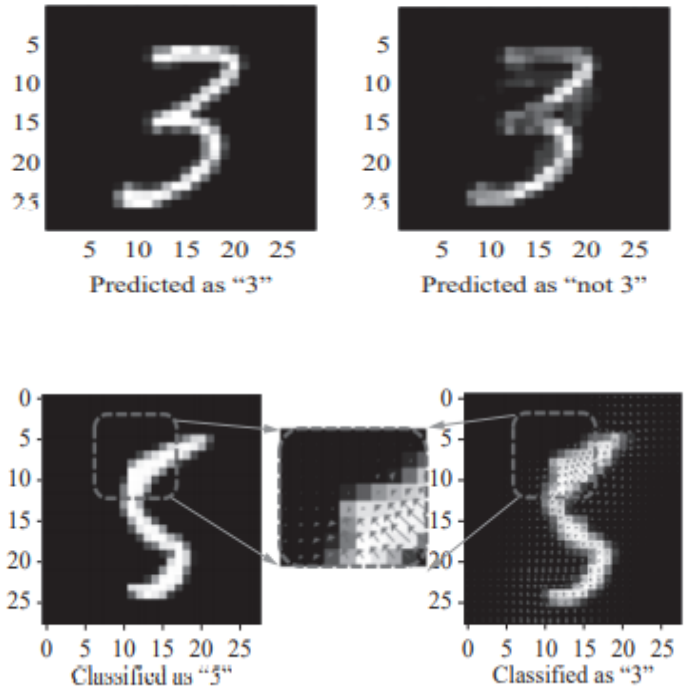

Figure 3: Gradient Masking Example

High-level representation of a guided denoiser as a defense strategy is generally used for the image classification tasks. High level guided denoiser overcomes the problem of standard denoiser mainly, the amplification of noise by making use of a loss function which is defined as the difference between the outputs of the target model, usually activated by the denoised images and clean images. The target model obtained as a result after the denoising process is now more robust to either black box or white box adversarial attacks [4].

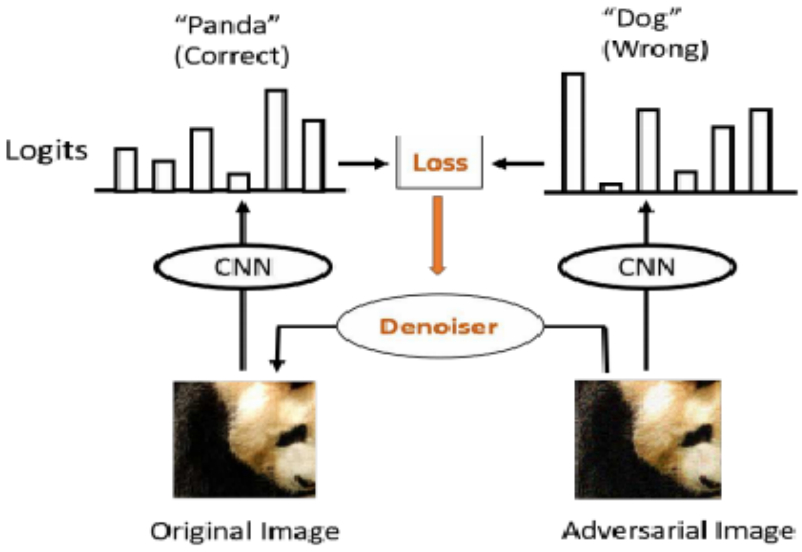

Figure 4: High level representation of a guided denoiser

A complete defense framework has been described in detail to secure the deep neural networks against the adversarial attacks. The detection of adversarial examples can be accomplished by using two detectors that are complementary in nature. They are also adaptive to the features of adversarial perturbations. The adversarial examples are filtered and cleaned out in this complete process. Unnoticeable perturbations are filtered out by the minor alteration detectors while the noticeable perturbations are filtered out by the statistical detectors [5].

The proposed framework comprises mainly of 3 modules:

- Detection of Adversarial examples
- Cleaning of Adversarial perturbations
- Adversarially targeted network which is completely trained

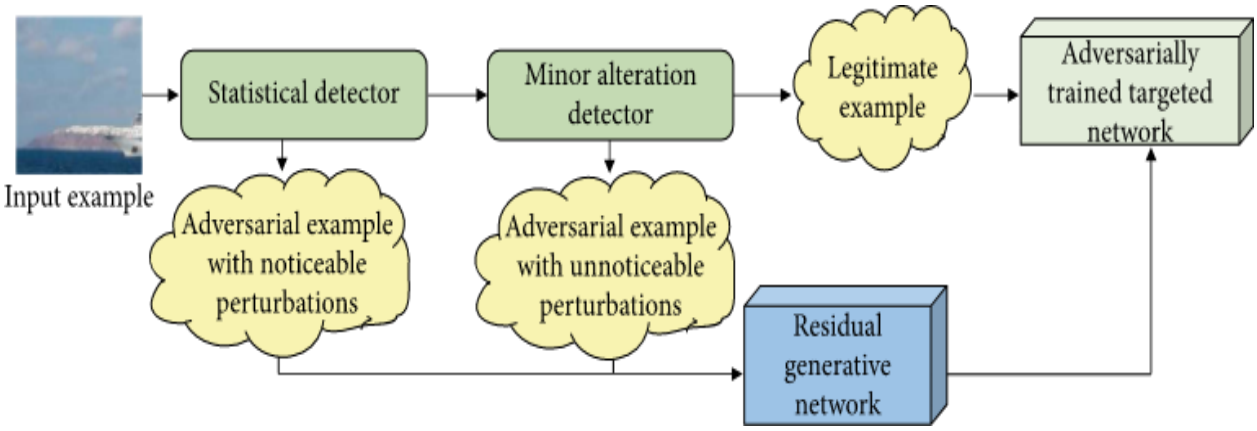

Figure 5: Overview of complete defense framework using complementary detectors

1.3 Problem Statement

Defense Against Adversarial Perturbations based Attacks using deep neural networks.

1.4 Applications

Deep learning strategies are applied in certain security and safety critical functions. The real life applications of adversarial attacks can be very dangerous. For example, one can modify a traffic sign to be misclassified or misinterpreted by an autonomous vehicle which can lead to a fatal accident. Hence, it becomes very essential to make the classifiers more robust to such kinds of adversarial attacks to avoid misinterpretation or misclassification. Therefore, the safety and security functions regarding the deep neural networks have become a major matter of concern. In such real life scenarios, the defense against adversaries plays a pivotal role in providing robustness to the classifiers.

1.5 Objectives

- To secure the deep neural network model.
- To preserve the accuracy and performance of the model.
- To enhance the DNN model's accuracy and performance of the model.
- To reduce the effect of adversarial noise or perturbations.
- To filter out or remove the adversarial examples.

1.6 Scope and Constraints

When we consider the adversarial patterns, usually the adversaries come up with their own adversarial goals and capabilities. As the deep learning applications are at the risk of adversarial attacks, the security of these applications is generally measured concerning the adversarial goals, mainly misclassification which leads the model to misclassify the output. Hence, there is a scope to design robust learning strategies that are irrepressible towards adversarial examples.

While devising the relisilent techniques, there are some challenges because currently some of the defense techniques are not adaptive to different types of adversarial attacks. One type of defense method can block one type of attack but it can leave vulnerability open to another kind of attack. For example, if the defense is successful against white box attacks, it may leave vulnerability open to black box attacks or vice-versa. It may decrease the prediction accuracy of the real time model built and may degrade its performance. Therefore, keeping in mind the constraints posed, the defense strategies need to be implemented which becomes a challenging task.

Chapter 2

REQUIREMENT ANALYSIS

Requirement Analysis is also referred to as Requirement Engineering. It is the process of defining or describing all expectations of the users for a particular application that is to be built or needs to be modified. In software engineering, it is occasionally noted loosely through names which include requirements collecting or requirements capturing. It includes all of the obligations which can be carried out to discover the needs of various stakeholders.

2.1 Functional Requirements

A functional requirement elucidates a feature of a system as a specification among outputs and inputs. It implies what a software system needs to do and the way it needs to function, basically they're product features that concentrate on user needs.

The functional requirements are as follows -

- The system should be able to accept the input image.
- The system should be able to classify the image.
- The system should be able to detect the adversarial attack.
- The system should be able to defend the adversarial attack.

2.2 Non Functional Requirements

A non-functional requirement describes the standards that may be used to decide upon the working of a system, in place of particular behaviors. It specifies the best characteristic of a system and describes the quality attribute of a system. It describes the system primarily based on certain factors which are vital to the success of the system.

The non-functional requirements are as follows -

- The system should be robust and reliable.

- The system should be able to classify the output with greater than 75% accuracy.

Performance Requirements - The system should accept legitimate input and produce expected results by minimizing the effect of the adversaries by 70%.

2.3 Hardware Requirements

- Computer
- HardDisk

2.4 Software Requirements

- Python3
- Pytorch
- Tensorflow
- Keras
- Numpy
- Pandas
- Scikit-Learn
- MatplotLib

Chapter 3

SYSTEM DESIGN

System design process provides a complete detailed information about the system and its components to enable the process of implementation and make it easy to understand parallely with the architecture of the system.

3.1 Architecture Design

The architecture design is divided into two phases:

- Attack phase
- Defense phase

3.1.1 Attack phase

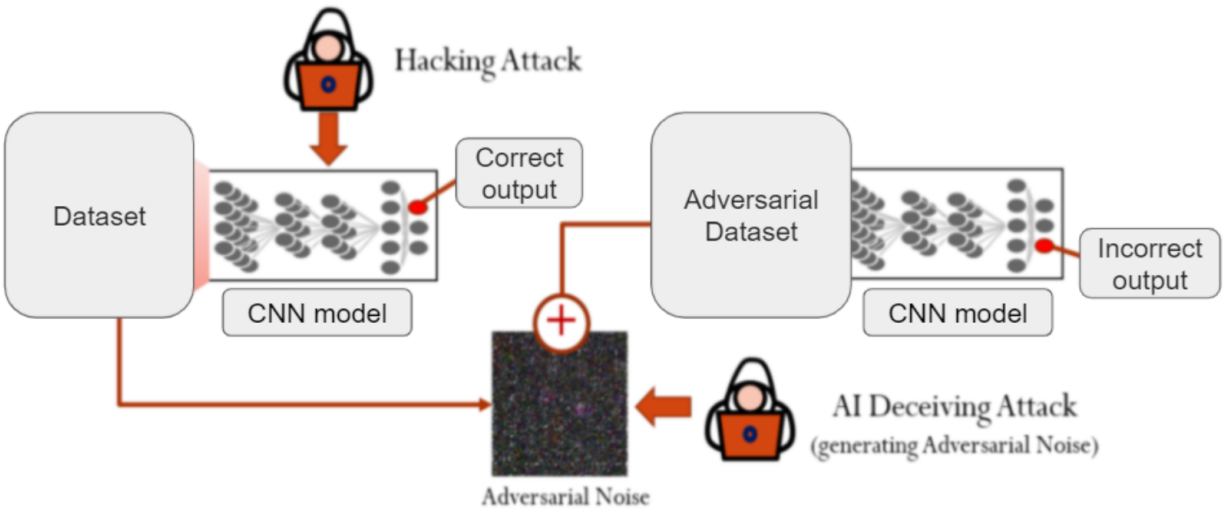

Figure 6: It provides a basic understanding of the architecture of adversarial attack

The CNN model built is trained with clean samples of images. Initially, the model detects the images correctly with a decent amount of accuracy. When adversarial noise is added to the original image dataset, the model predicts incorrectly. Due to the adversarial attack, the CNN model starts predicting the original input image as an image of a different class other than its original class. Thus the model misclassified the output giving rise to the adversarial images.

3.1.2 Defense Phase

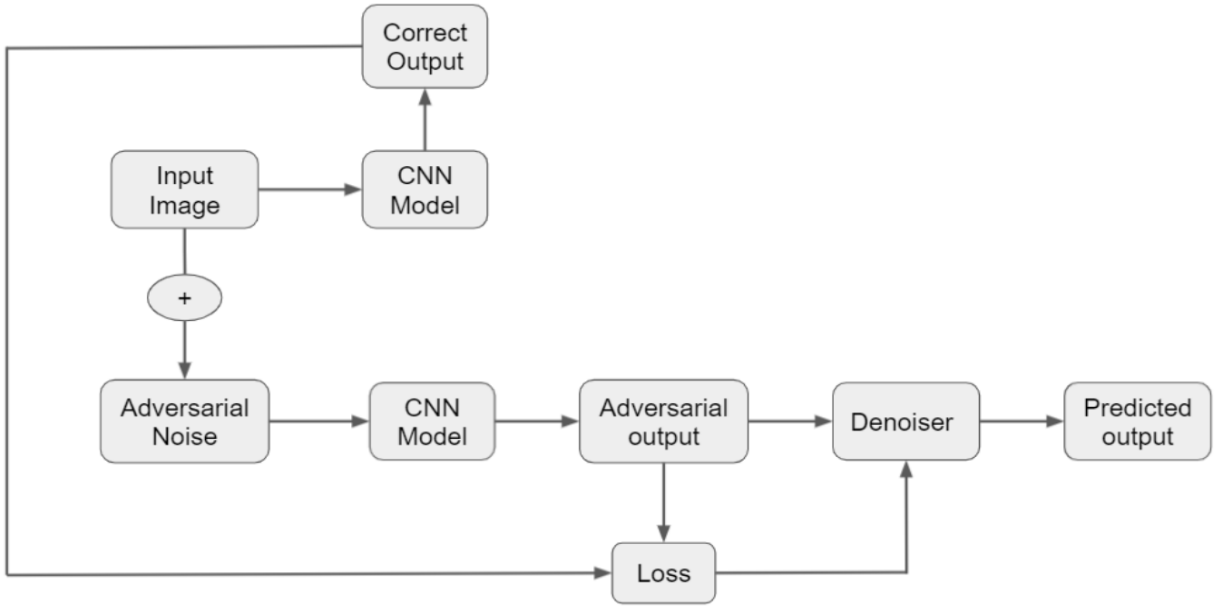

Figure 7: It provides a basic understanding of the architecture of adversarial defense

The adversarial examples are generated when the noise is added to the input image. So, a general idea here is denoising the adversarial examples to get back the proper prediction of the original image. The adversaries here can be removed with the help of a denoiser. Here, the loss function can be defined as the difference between the correct output and the adversarial output. Based upon the difference in the loss function, the denoiser predicts the output generally called as the predicted output. If the correct output and the predicted output are nearly equal, then it can be said that the defense against the adversaries is successful. This provides us a general idea about the defense phase.

3.1.3 High Level Architecture Design

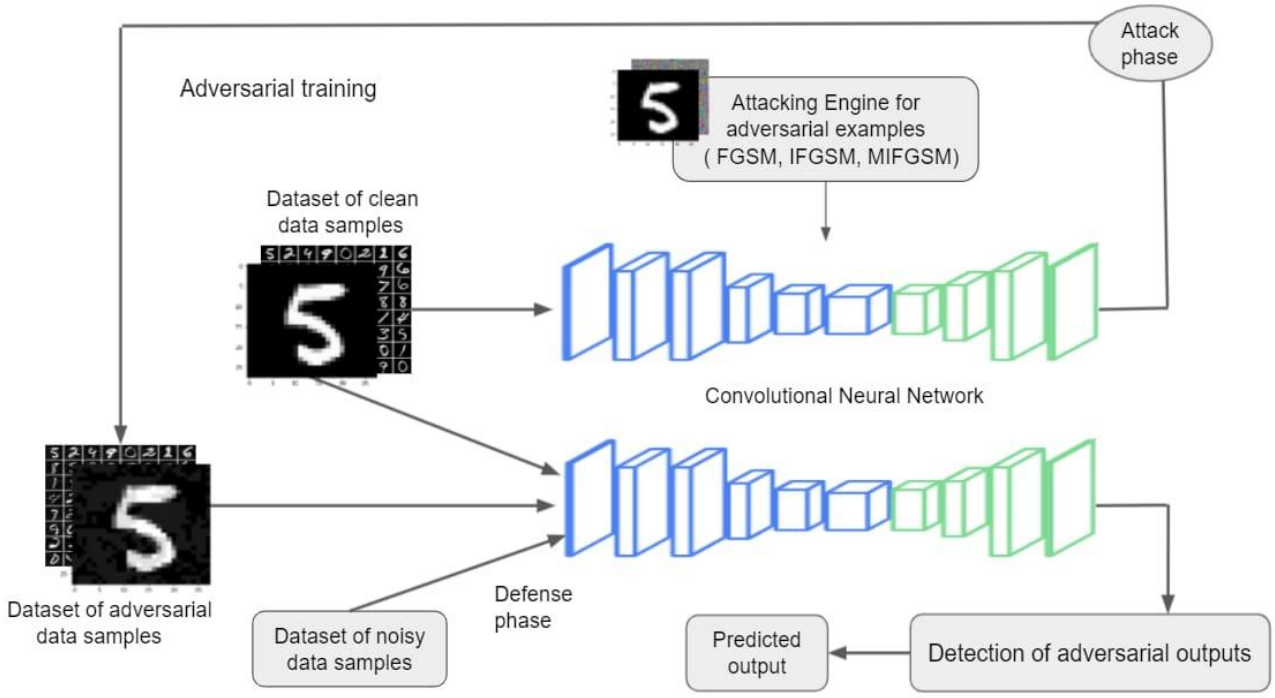

Figure 8: High Level Architecture design of adversarial attack and defense

A Dataset of clean samples which comprises clear images with no added noise or perturbation is fed to the built CNN Model for it to get trained. The model is trained with these clean samples and the correct output is obtained as a result.

During the attack phase, perturbations are added to the clean data samples to generate adversarial data samples. Different types of attack mechanisms are tried on the built CNN Model with help of an attack engine.

The attacks namely are :

- FGSM Attack : Fast Gradient Sign Method does the calculation of the loss function gradients with respect to the original input images and further new images are created which will maximize the loss using the sign of the gradients.

- IFGSM Attack : An Iterative FGSM Attack is a straightforward method to extend the fast method FGSM in an iterative manner.
- MIFGSM Attack: Momentum Iterative Fast Gradient Sign Method combines the FGSM attack and IFGSM attack.

The adversarial images generated are given to the CNN model which was previously trained with regular images.The model then learns from these adversarial images and trains itself to distinguish between clear images and adversarial images. The model detects the adversarial outputs and without getting fooled, the model identifies and predicts the correct output. It is possible through adversarial training.

3.2 Detailed Design

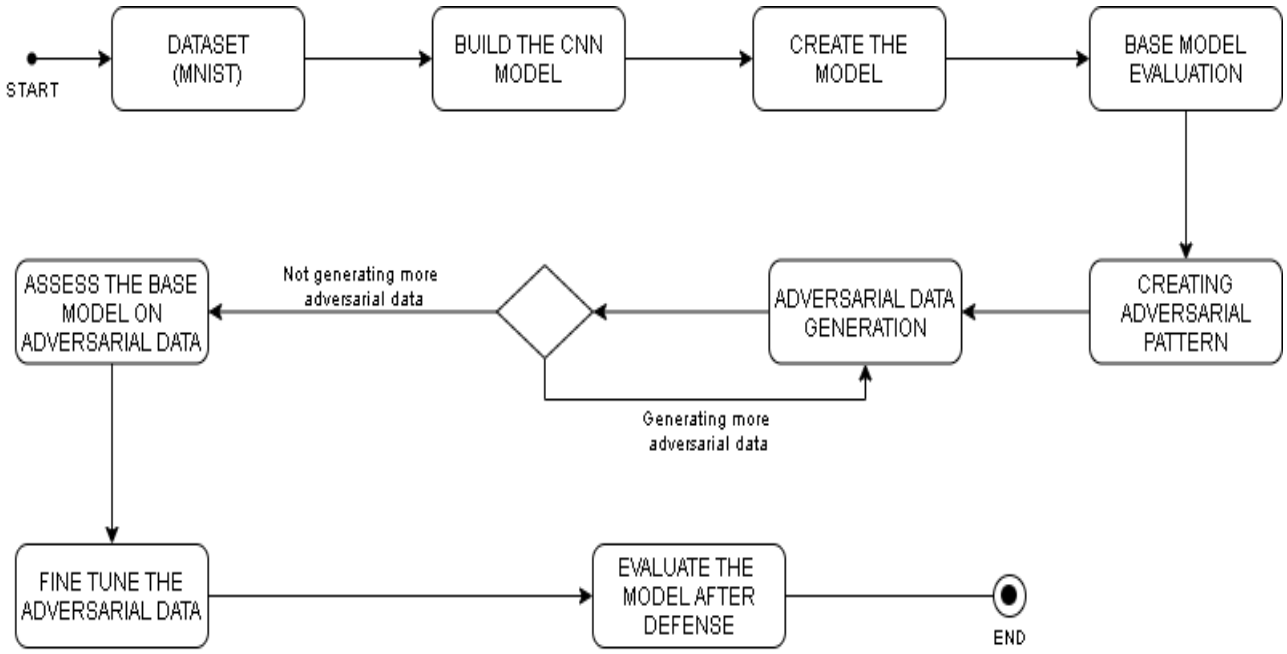

Figure 9: Activity diagram of adversarial attack and defense

Using the dataset, preferably MNIST, a basic CNN model is built and trained. The accuracy of the base model is then evaluated. During the attack, adversarial patterns are generated and fed into the model due to which it results in generation of adversarial dataset and the model's accuracy drips down and the model now gives false outputs. As a part of the defense method, more adversarial images are generated. After enough adversarial images are generated ,the model then gets trained with the adversarial images and the adversarial training here as a defense mechanism.

The CNN Model can now differentiate between regular images and adversarial images. After defense,the model is again checked with the model accuracy. Model accuracy is retrieved back as it was before.

Chapter 4

DATASET

4.1 Description

The CNN model was trained using different datasets on which types of attacks and their respective defense mechanisms were carried out and a comparative study was obtained for each of the datasets used.

The datasets used are -

1) **MNIST**

   It is a dataset of handwritten digits which are between 0 and 9 (10 classes) and it consists of small square 60000 28x28 pixel grayscale images.

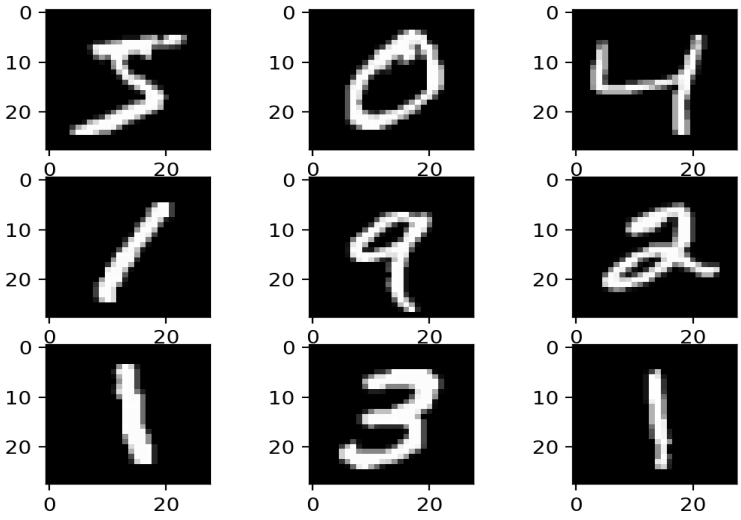

Figure 10: MNIST Dataset

2) CIFAR10

It is a dataset of 32x32 colour images and 60000 in total. It consists of 10 classes with around 6000 images per class. Out of 60000, 50000 are taken as training images and 10000 are taken as testing images.

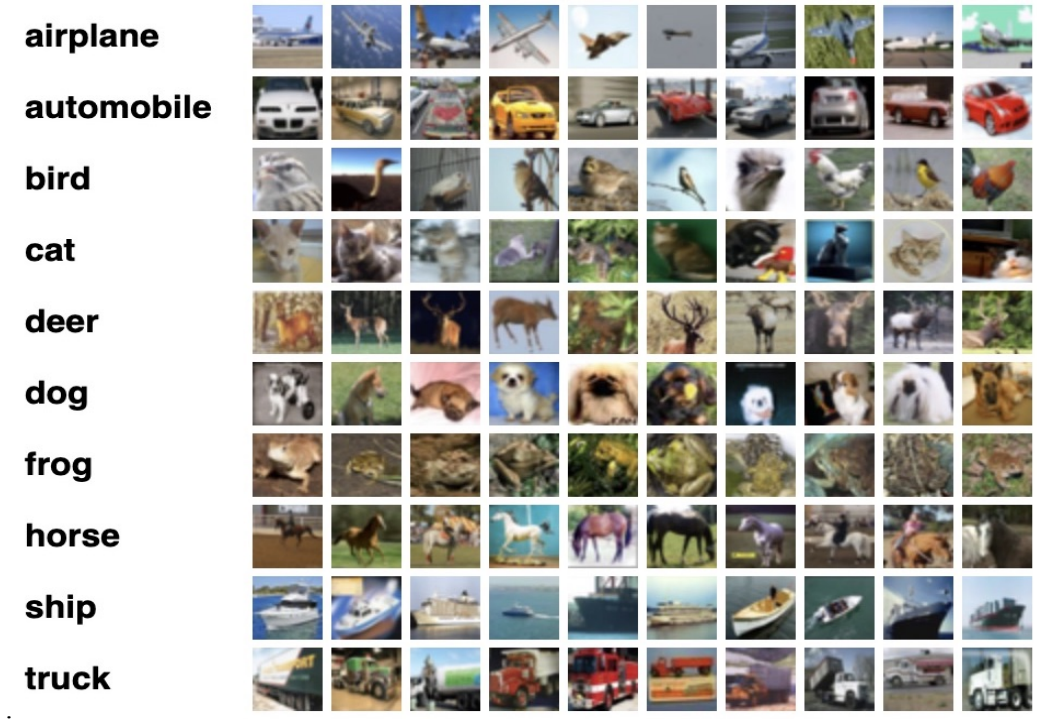

Figure 11: CIFAR10 Dataset

Dataset source for MNIST and CIFAR10

- Tensorflow library
- Torch library

3) CUSTOM DATASET

It is a KLETECH Dataset collected by the students during the course project.

It consists of 3 classes with around 200 images in each class contributing to about 600 images in total.

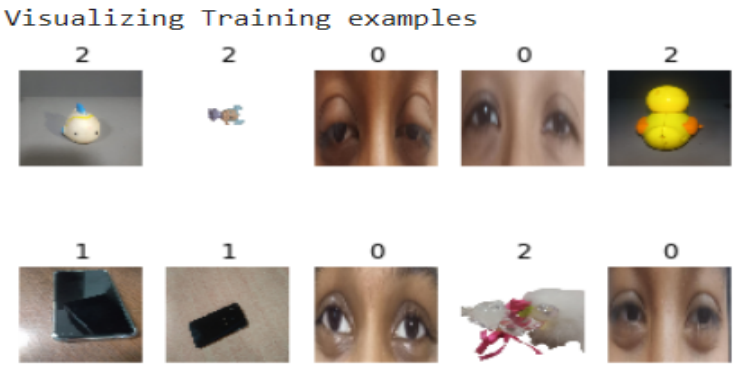

Figure 12: Training examples of custom dataset

Class 0 - Eyelids
Class 1 - Phone
Class 2 - Toys

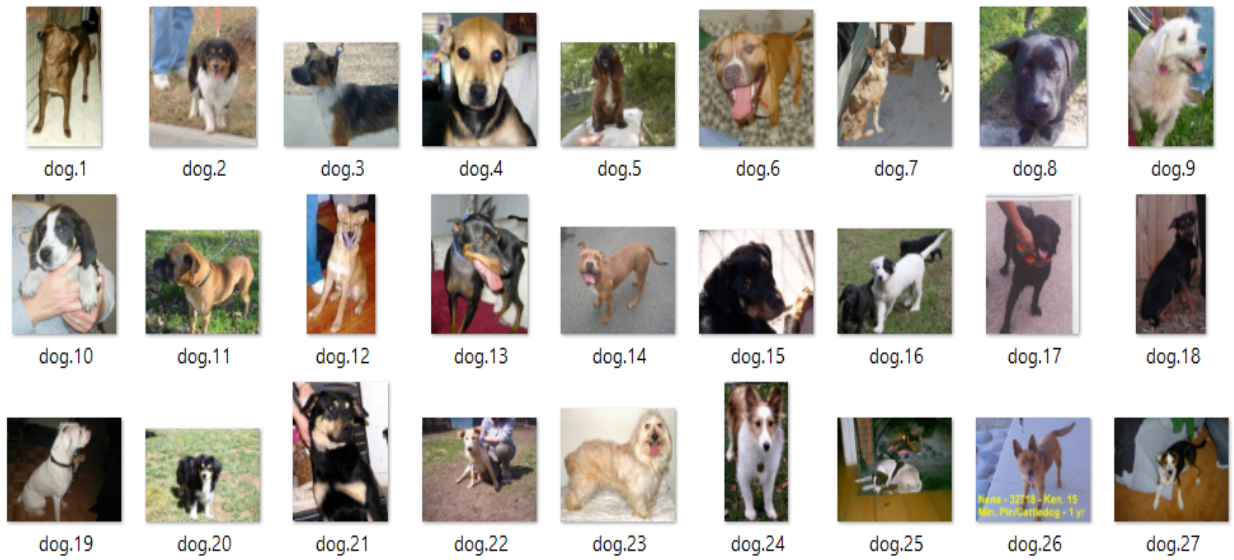

Figure 13: Dog Dataset

4.2 Data preprocessing techniques

In a generalised manner, some amount of preprocessing was done on each of the datasets. It includes reshaping, rescaling, normalising and resizing the images.

The input images are to be loaded as numpy arrays and are reshaped to account for the number of channels. They are then normalised to the interval [0,1] and the ground truth labels are one hot encoded to make it compatible for carrying out further operations. The class vectors are converted to binary class matrices and are then operated.

Chapter 5

IMPLEMENTATION

5.1 Proposed Methodology

There are various different types of attack and defense mechanisms. The methodology is to implement various kinds of attacks and perform a comparison study to explore the best defense techniques suitable for each of the proposed attacks. Statistically, to have a broader idea about their behaviour, the same attacks and defense techniques have been implemented on different datasets using the CNN model.

For the proposed method, CNN is most suited for the classification task that is to be performed for adversarial attack and defense. A simple sequential model has been used with accuracy as the main evaluation measure.

The implementation is performed on the following 3 types of attacks

- FGSM
- IFGSM
- MIFGSM

The defense strategies applied to the attacks are -

- Adversarial Training
- Defensive Distillation

5.1.1 Adversarial Training

The main idea of the adversarial training is to improve the robustness of the model by generating a lot of adversarial examples and injecting them into the training set. The model built is trained on these examples so that it can learn from the adversarial data and make the predictions appropriately. Hence, it is generally called a brute force approach. The augmentation here can be performed by training the model with the original input data and the generated adversarial data

ensuring that the training is performed properly. However, adversarial training is not suitable for certain black box attacks and two step attacks. Nevertheless, it performs well single step attacks as well as on white box attacks.

5.1.2 Defensive Distillation

The defensive distillation is generally a strategy which will add flexibility to the process of classification algorithm so that the model is less vulnerable to exploitation caused due to the adversaries. During the training process of defensive distillation, one model is trained to predict the output probabilities of another model which was trained earlier on the baseline standard model to highlight and compare the accuracy.

5.1.3 FGSM Attack And Adversarial Training

The implementation is done using the keras and tensorflow framework on MNIST dataset and on CIFAR10 dataset as well. During pre-processing, we normalise the data in the interval of [0,1] and reshape it. Also one hot encode the labels. A simple sequential model has been used which fits the data well and provides the baseline accuracy around 98%.

Adversarial attacks are performed by creating adversarial patterns and the base accuracy on the adversarial images is evaluated. By comparing the accuracy before the attack and after the attack, the accuracy has been reduced indicating that the attack has been performed.

Defense against such attacks is done using an adversarial example generator. A bunch of adversarial images are generated on which the model is trained so that the model can learn from adversarial data. To increase the accuracy after defense, we train the model by generating more and more images.

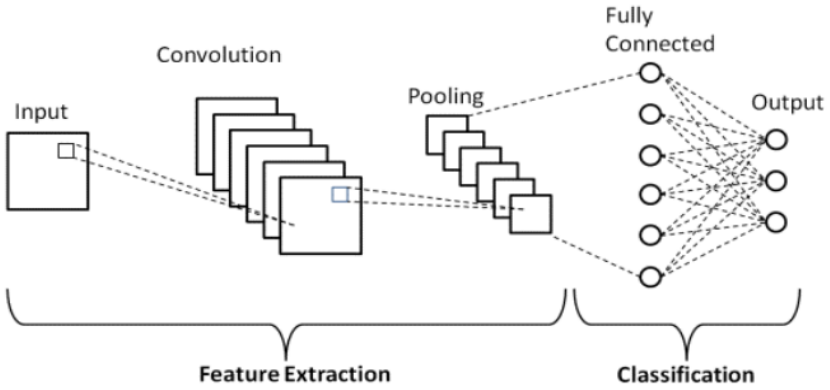

Figure 14: Basic CNN Architecture

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 10, 10, 32)	320
conv2d_1 (Conv2D)	(None, 4, 4, 64)	18496
conv2d_2 (Conv2D)	(None, 2, 2, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 1, 1, 64)	0
dropout (Dropout)	(None, 1, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 32)	1056
dropout_2 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 10)	330

Total params: 59,210
Trainable params: 59,210
Non-trainable params: 0

Figure 15: Summary of Sequential Model

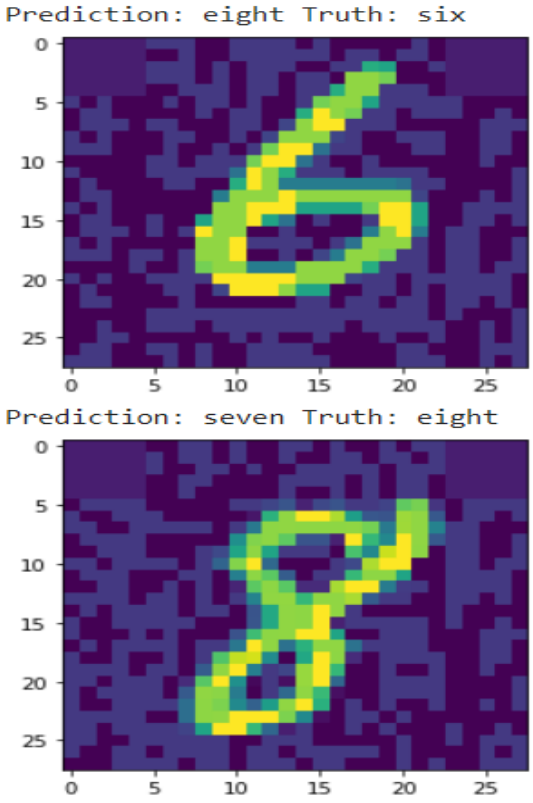

Figure 16: FGSM attack implementation using tensorflow and keras using MNIST dataset
(Images have been misclassified)

The above technique has been implemented on a custom KLETECH dataset as well where in for instance, a sample image is taken for visualisation. The model predicts the original image correctly as a labrador retriever with high confidence but when an adversarial attack is performed on it, it predicts the label class incorrectly as saluki and the accuracy for prediction falls down to 13%. It indicates that the attack has been performed as the accuracy has been drastically reduced when compared to the baseline accuracy. The defense method used for this attack is again the adversarial training and after the defense strategy is applied, the accuracy again is increased and the model becomes more robust to these kinds of attacks.

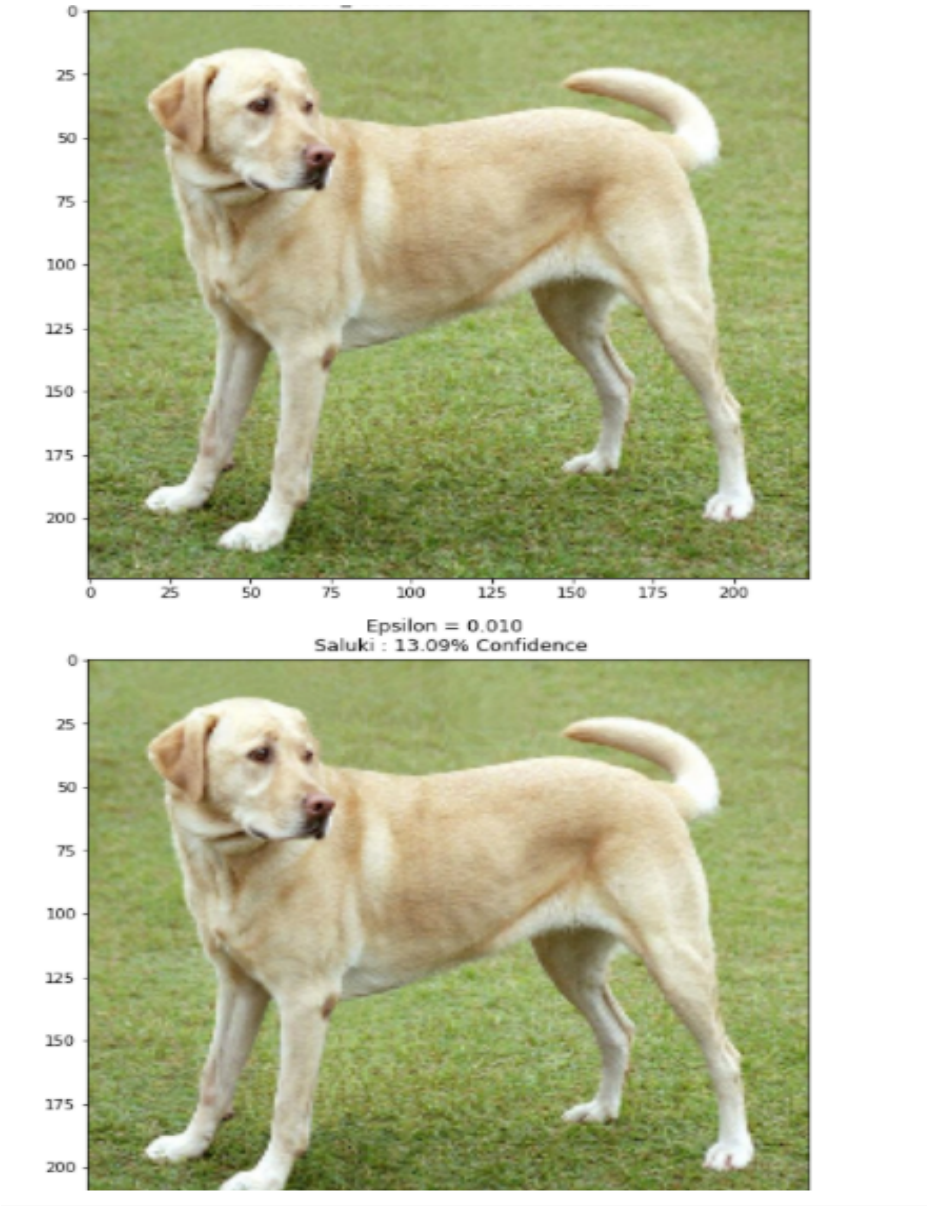

Figure 17: FGSM attack implementation on custom dataset

FGSM, IFGSM, MIFGSM attacks were applied on the MNIST dataset and were implemented using PyTorch. Defensive distillation is used here as a defense strategy for these attack mechanisms.

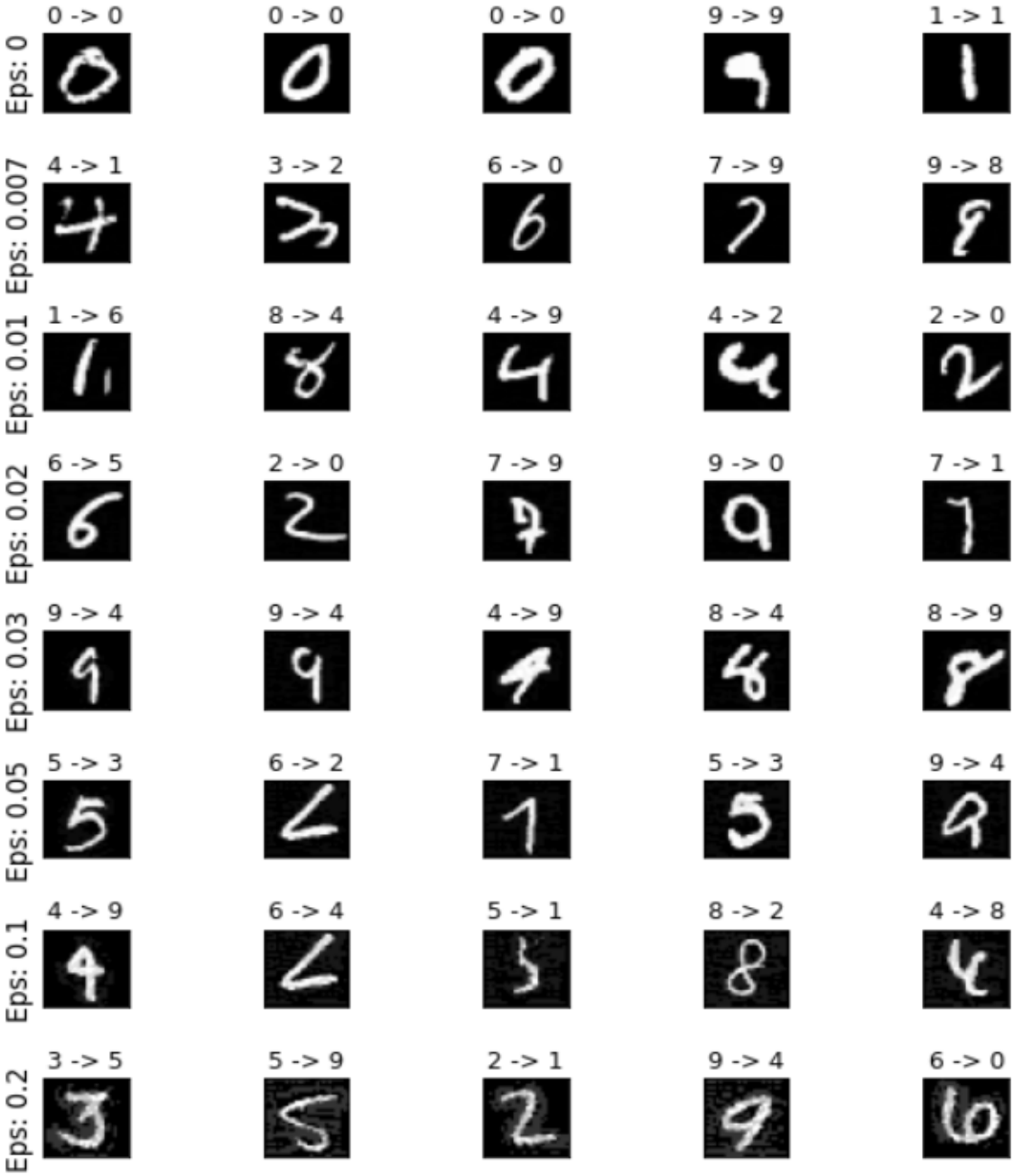

Figure 18: Sample adversarial examples in the epsilon range of 0 to 0.3

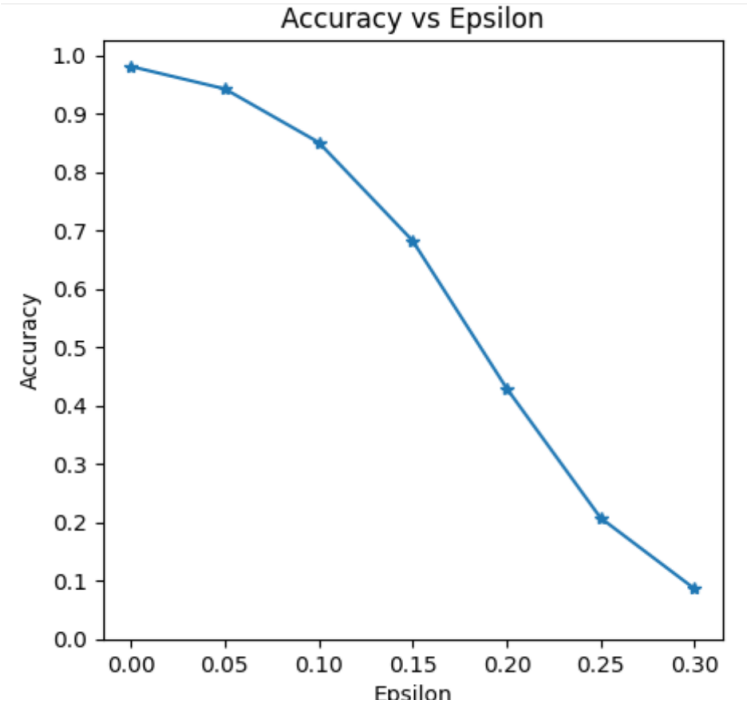

Figure 19: Accuracy vs epsilon plot

The test accuracy decreases as epsilon increases. It occurs because larger epsilons means maximising the loss by taking a larger step in that direction.But as the epsilon increases, we observe that the perturbations or noise becomes more perceptible. Therefore, one can notice a tradeoff between decrease in accuracy and the perceptibility.

5.2 Evaluation Metrics

Evaluation Measures play a crucial role in determining the performance of the model. Accuracy is the most important evaluation measure and is commonly used and a confusion matrix is often used to explain the performance of a classification type model.

		Predicted	
		Negative (**N**) **-**	Positive (**P**) **+**
Actual	Negative **-**	True Negatives (**TN**)	False Positives (**FP**) **Type I error**
	Positive **+**	False Negatives (**FN**) **Type II error**	True Positives (**TP**)

Figure 20: Confusion matrix

.

True positive and true negatives are the observations that are correctly predicted so our aim is to minimize false positives and false negatives.

- **Accuracy:** It is given by the ratio of correctly predicted observation to the total observations. It is a great measure when we have symmetric datasets where values of false positive and false negatives are almost same.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Chapter 6

RESULTS AND DISCUSSIONS

To understand the behaviour of the models and to differentiate them, a table has been formulated to summarise the findings obtained. It gives an overall idea about the performance of different attack and defense methods used for implementation. The behaviour of the model on different datasets is also noted down.

Table 1: **Comparison results of different attack and defense mechanisms**

TYPE OF ATTACK	TYPE OF DEFENSE MECHANISM	MNIST DATASET			CIFAR-10 DATASET		
		Before Attack accuracy (in %)	After Attack accuracy (in %)	After Defense accuracy (in %)	Before Attack accuracy (in %)	After Attack accuracy (in %)	After Defense accuracy (in %)
FGSM	Adversarial Training	98.21	20.40	79.93	85.84	31.84	68.16
FGSM	Defensive Distillation	97.08	28.84	72.24	84.12	34.27	65.73
IFGSM		96.92	30.54	66.38	82.34	38.77	61.23
MIFGSM		97.05	30.10	66.35	84.76	38.58	61.42

When we consider Adversarial Training, the type of attack used is FGSM. For the MNIST dataset, the before attack accuracy is 98.21% and the after attack accuracy is 20.40%. As there is significant decrease in accuracy, the goal of misclassification for FGSM attack has been successful. After the attack, when adversarial training is performed, the accuracy has been increased to around 79.93% indicating that the defense has been successful.

The next defense mechanism is defensive distillation and particularly three types of attacks are performed. Considering the MNIST dataset, for FGSM attack, the test accuracy reduces from 97.8% to 24.84% with epsilon in the range from 0 to 0.3 whereas for IFGSM attack, considering the number of iterations as 10, the test accuracy reduces from 96.92% to 30.24%. For MIFGSM, the decay factor is taken as 1.0 with 10 iterations and the test accuracy reduces from 97.05% to 30.10%. Overall, these attacks on the proposed system performed well.

During defensive distillation, the temperature is taken as 100 and the number of filters have been reduced to half in each layer to minimise the number of parameters. By observing the accuracy after the defense, there is significant increase in the accuracy compared to the accuracy after the attack, indicating that the defense has been successful.

Table 2: Comparison of MNIST with KLETECH dataset for FGSM Attack and Adversarial Training method

TYPE OF ATTACK	TYPE OF DEFENSE MECHANISM	MNIST DATASET			KLETECH DATASET		
		Before Attack accuracy (in %)	After Attack accuracy (in %)	After Defense accuracy (in %)	Before Attack accuracy (in %)	After Attack accuracy (in %)	After Defense Accuracy (in %)
FGSM	Adversarial Training	98.21	20.40	79.93	84.12	34.27	68.73

Similarly, the above methods are carried out on the CIFAR10 dataset and the observations are noted while the FGSM attack with Adversarial Training method is carried out on the KLETECH custom dataset as well to get a broad understanding of the behaviour of the attacks and defense mechanism.

Chapter 7

CONCLUSION

A comparative study on the types of attacks and defense has been done based on the implementation and the accuracy has been used as a measure for the comparison. Based upon the observations, we see that the MNIST Dataset is more finely tuned as the accuracy obtained is more when compared to cifar10 and custom dataset.

However, all the types of attacks on the proposed system could reduce more than 70% of the accuracy after the attack, indicating that the attack mechanisms pose a threat to deep learning systems. While defending these attacks seems to be a challenging task. For those attacks, with a goal of misclassification mainly FGSM attack, adversarial training and defensive distillation are used as defense strategies.When Adversarial training is used as a defense strategy with FGSM Attack, it performs better than it does with that of Defensive distillation. Hence, it makes the system more robust and resilient to adversarial attacks with a goal of misclassification. Yet there is scope for figuring out the more robust techniques for other adversarial goals and capabilities such that those techniques are resilient to all kinds of adversaries.

BIBLIOGRAPHY

[1]. Chakraborty, Anirban, et al. "Adversarial attacks and defences: A survey." *arXiv preprint arXiv:1810.00069* (2018).

[2]. Akhtar, Naveed, and Ajmal Mian. "Threat of adversarial attacks on deep learning in computer vision: A survey." *Ieee Access* 6 (2018): 14410-14430.

[3]. Xu, Han, et al. "Adversarial attacks and defenses in images, graphs and text: A review." *International Journal of Automation and Computing* 17.2 (2020): 151-178.

[4]. Liao, Fangzhou, et al. "Defense against adversarial attacks using high-level representation guided denoiser." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[5]. Sun, Guangling, et al. "Complete Defense Framework to Protect Deep Neural Networks against Adversarial Examples." *Mathematical Problems in Engineering* 2020 (2020).

[6]. Prakash, Aaditya, et al. "Deflecting adversarial attacks with pixel deflection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[7]. Zantedeschi, Valentina, Maria-Irina Nicolae, and Ambrish Rawat. "Efficient defenses against adversarial attacks." *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 2017.

[8]. Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks." *arXiv preprint arXiv:1706.06083* (2017).

[9]. Zhu, Dingyuan, et al. "Robust graph convolutional networks against adversarial attacks." *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019.

[10]. Kos, Jernej, and Dawn Song. "Delving into adversarial attacks on deep policies." *arXiv preprint arXiv:1705.06452* (2017).