

**DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION**

**BY**

**NG SHI QI**

**A REPORT**

**SUBMITTED TO**

**Universiti Tunku Abdul Rahman**

**in partial fulfillment of the requirements**

**for the degree of**

**BACHELOR OF COMPUTER SCIENCE (HONOURS)**

**Faculty of Information and Communication Technology**

**(Kampar Campus)**

**JUNE 2024**

## REPORT STATUS DECLARATION FORM

**Title:** DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION

**Academic Session:** JUN 2024

I \_\_\_\_\_ NG SHI QI \_\_\_\_\_  
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in  
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



\_\_\_\_\_  
(Author's signature)



\_\_\_\_\_  
(Supervisor's signature)

**Address:**

72, LRG BKT MINYAK INDAH,  
TMN BKT MINYAK INDAH,  
14000 BKT MERTAJAM.

\_\_\_\_\_  
Ng Hui Fuang

Supervisor's name

**Date:** \_\_\_\_ 19/09/24 \_\_\_\_

**Date:** \_\_\_\_ 20/9/2024 \_\_\_\_

<b>Universiti Tunku Abdul Rahman</b>			
Form Title : <b>Sample of Submission Sheet for FYP/Dissertation/Thesis</b>			
Form Number: <b>FM-IAD-004</b>	Rev No.: <b>0</b>	Effective Date: <b>21 JUNE 2011</b>	Page No.: <b>1 of 1</b>

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: \_\_\_\_\_19/09/24\_\_\_\_\_

**SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS**

It is hereby certified that    NG SHI QI    (ID No:    21ACB04550   ) has completed this final year project entitled “**DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION**” under the supervision of    DR NG HUI FUANG    (Supervisor) from the Department of    COMPUTER SCIENCE   , Faculty of    INFORMATION AND COMMUNICATION TECHNOLOGY   .

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,



\_\_\_\_\_  
(NG SHI QI)

\*Delete whichever not applicable

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : \_\_\_\_\_  \_\_\_\_\_

Name : \_\_\_\_\_ NG SHI QI \_\_\_\_\_

Date : \_\_\_\_\_ 19/09/24 \_\_\_\_\_

## **ACKNOWLEDGEMENTS**

I extend my heartfelt gratitude to my supervisor, Dr. Ng Hui Fuang, for granting me the valuable opportunity to participate in this research project on adversarial defence in image recognition. His unwavering support throughout the research process has been invaluable.

I am also deeply grateful to my friends for their patience, unwavering support, and love, especially during challenging times. Lastly, I would like to thank my parents and family for their constant love, support, and encouragement throughout this journey.

## ABSTRACT

Deep neural networks were found to be extremely useful and perform exceedingly well in machine learning tasks such as computer vision, speech recognition, natural language processing and in various domains such as healthcare system and autonomous car system. The high accuracy exhibited by the deep learning models in machine learning tasks have attracted people into using them in various real-world scenarios including those safety-oriented ones. However, it would seem that the high accuracy that was exhibited by those machine learning models does not necessary means that they are reliable, or robust enough to be employed in our daily lives directly. It was found recently that these high-performance models can be fooled by adversarial examples, which are perturbed inputs that are almost indiscernible from normal inputs to human eyes but can create a huge disruption in the behavior of machine learning models. If this inherent nature of machine learning models was exploited by adversaries, dire consequences will occur. Hence, defensive methods should be deployed to prevent the machine learning models to be robust against these adversarial examples. In the reviewed papers, we found that researchers have tried incorporating randomness into the models by various methods such as RSE and Adv-BNN, but only Adv-BNN has combined adversarial training with BNN which infuse randomness into their defensive strategy, and other methods rarely investigated the effect of combining adversarial training and randomness incorporation into one defensive method, as well as investigating the effect of combining adversarial purification and adversarial training. In this thesis, we propose a defensive method combining a preprocessing pipeline that introduces noises, adversarial purification and adversarial training, and we have investigated various methods of incorporating noises the effect of doing so in an adversarial defense system.

# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>REPORT STATUS DECLARATION FORM</b>	<b>ii</b>
<b>FYP THESIS SUBMISSION FORM</b>	<b>iii</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vi</b>
<b>TABLE OF CONTENTS</b>	<b>vii</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF TABLES</b>	<b>xi</b>
<b>LIST OF SYMBOLS</b>	<b>xii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Background Information	1
1.2 Problem Statement	2
1.3 Motivation	3
1.4 Project Scope	4
1.5 Project Objective	5
1.6 Contributions	5
1.7 Report Organization	5
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>6</b>
2.1 Adversarial Attack	6
2.2 Adversarial Defense	8
2.2.1 Proactive Defense Algorithm	8
2.2.2 Passive Defense Algorithm	10
2.3 Additional Notes on Adversarial Defense	12
2.4 Limitation of Previous Studies	12

<b>CHAPTER 3 METHOD</b>	<b>14</b>
3.1 Noise Incorporation Preprocessing Pipeline	14
3.1.1 Gaussian Noise	14
3.1.2 Salt and Pepper Noise	14
3.1.3 Uniform Noise	15
3.1.4 Gaussian Blur	15
3.2 Adversarial Purification by Diffusion Model	16
3.3 Proposed Methodology	17
<b>CHAPTER 4 EXPERIMENT</b>	<b>20</b>
4.1 Hardware Setup	20
4.2 Environmental Setup	21
4.3 Model Architecture and Dataset	21
4.4 Evaluation Benchmarks and Metrics	22
4.5 Result Reproduction Attempts	23
4.6 Noise Incorporation Pipeline	23
4.7 Testing Pipeline with Adversarial Attack	29
4.8 Additional Remarks	31
4.9 Implementation Issues and Challenges	31
<b>CHAPTER 5 CONCLUSION</b>	<b>44</b>
<b>REFERENCES</b>	
<b>WEEKLY LOG</b>	
<b>POSTER</b>	
<b>PLAGIARISM CHECK RESULT</b>	
<b>FYP2 CHECKLIST</b>	



## LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 1.1	Figure 1.1 An example of adversarial attack, where an image of panda was misclassified by GoogLeNet after the injection of perturbation using FGSM	2
Figure 1.2	A physical perturbation applied to a stop sign causing the LISA-CNN classifier to misclassify the stop sign as a speed limit of 45km/h	3
Figure 2.1	Illustration of adversarial attack	6
Figure 3.1	The forward and reverse process of diffusion model	16
Figure 3.2	Equation for forward diffusion process	16
Figure 3.3	Equation for reverse diffusion process	16
Figure 3.4	Forward diffusion process	17
Figure 3.5	Reverse diffusion process	17
Figure 3.6	Proposed Methodology	17
Figure 3.7	Proposed Preprocessing Pipeline	18
Figure 4.1	Effect of Adding Gaussian Noise	24
Figure 4.2	Experiment of Gaussian Noise	25
Figure 4.3	Effect of Adding Salt and Pepper Noise	25
Figure 4.4	Experiment of Salt and Pepper Noise	26
Figure 4.5	Effect of Adding Uniform Noise	26
Figure 4.6	Experiment of Uniform Noise	27
Figure 4.7	Effect of Gaussian Blur	27
Figure 4.8	Experiment of Gaussian Blur	28
Figure 4.9	Applying the final pipeline	29
Figure 4.10	Comparison among clean image, adversarial image and transformed adversarial image and their classification results	29

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 4.1	Specifications of NVIDIA GeForce GTX 1080	20
Table 4.2	Specifications of NVIDIA GeForce RTX4070	20
Table 4.3	Specifications of Kaggle Cloud NVIDIA Tesla P100	20
Table 4.4	Specifications of the laptop	21
Table 4.5	Results on applying two different pipelines after PGD-10 attack	30
Table 4.6	Results on applying two different pipelines and sent for adversarial purification after PGD-10 attack.	31

## LIST OF ABBREVIATIONS

<i>AI</i>	Artificial Intelligence
<i>LISA-CNN</i>	LISA-Convolutional Neural Network
<i>FGSM</i>	Fast Gradient Sign Method
<i>BIM</i>	Basic Iterative Attack
<i>PGD</i>	Projected Gradient Descent
<i>ZOO</i>	Zeroth Order Optimization Attack
<i>NES</i>	Natural Evolutionary Strategies
<i>FreeAT</i>	Adversarial Training for Free
<i>YOPO</i>	You Only Propagate Once
<i>DNN</i>	Deep Neural Network
<i>JPEG</i>	Joint Photographic Experts Group
<i>RSE</i>	Random Self-Ensemble
<i>BNN</i>	Bayesian Neural Network
<i>CNN</i>	Convolutional Neural Network
<i>GAN</i>	Generative Adversarial Network
<i>GPU</i>	Graphics Processing Unit
<i>EDM</i>	Elucidating Diffusion Model
<i>WRN</i>	WideResNet
<i>DDPM</i>	Denoising Diffusion Probabilistic Model
<i>SGM</i>	Score-Based Generative Model
<i>Score SDE</i>	Stochastic Differential Equation

# Chapter 1

## Introduction

In this chapter, we present the background and motivation of our research, our contributions to the field, and the outline of the thesis.

### 1.1 Background Information

Ever since the first deep convolutional network was implemented by Alex Krizhevsky et al [1], which significantly reduces the error rate of image classification, allowing the authors to outperform other competitors and let them won the ImageNet challenge in 2012, many deep learning model based on their work has been proposed in the following years which further improved the classification accuracy. Deep neural networks were found to be extremely useful and perform exceedingly well in machine learning tasks such as computer vision, speech recognition, natural language processing and in various domains such as healthcare system and autonomous car system. The high accuracy exhibited by the deep learning models in machine learning tasks have attracted people into using them in various real-world scenarios including those safety-oriented ones.

However, it is unfortunate that a high accuracy of these models in performing their tasks does not guarantee a high robustness in them, as it was found recently that these high-performance models can be fooled by adversarial examples, which are perturbed inputs that are almost indiscernible from normal inputs to human eyes but can create a huge disruption in the behavior of machine learning models. These adversarial examples are created by inserting a small amount of specially designed perturbations into normal inputs, and they can cause a significant drop of performance in the machine learning models. As the inserted perturbation is small, the adversarial examples resemble valid inputs to a human but are capable of causing a mistake in the model's predictions. The creation of an adversarial example is known as an adversarial attack, and an illustration of this process was shown in Figure 1.1, where the image on the left is the normal input of a panda, and the image on the right was an adversarial example, created by inserting a small perturbation into the normal input.

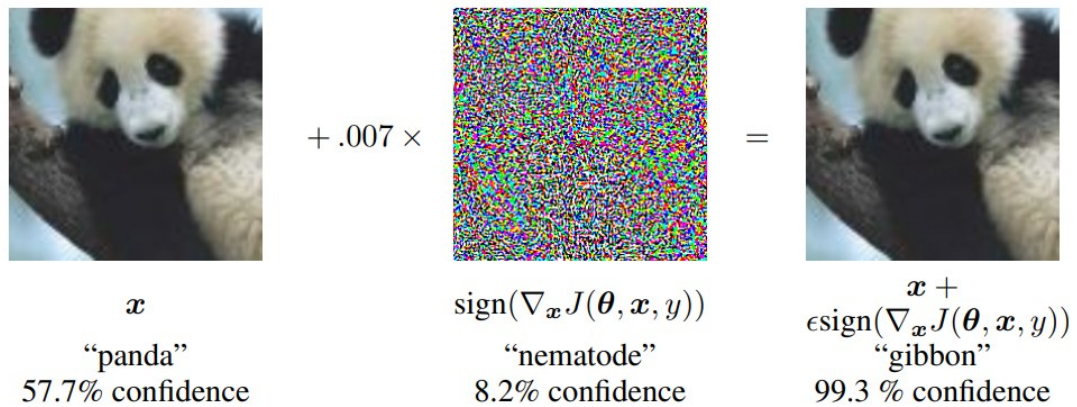


Figure 1.1 An example of adversarial attack, where an image of panda was misclassified by GoogLeNet after the injection of perturbation using FGSM [2]

While the adversarial example given above was in the field of image recognition, adversarial attacks are also found in other machine learning applications such as natural language processing field [3]. Adversarial attacks also come in many forms, including (1) poisoning attack, (2) evasion attacks and (3) model extraction. The example given in Figure 1.1 is under the category of evasion attack. Poisoning attack occurs during the training stage of a machine learning life cycle, where the attacker contaminates the data by injecting malicious samples and causing the model to underperform as a result. Evasion attacks are performed during the deployment stage, where the attackers feed manipulated data to the pre-trained model during deployment, causing misclassification. Last but not least is the model extraction where the attackers try to probe a black box model so that they can reconstruct the model for their own usage.

In this thesis we will focus only on evasion attacks in the field of image recognition. Related adversarial attack and defense strategies will be covered in Chapter 2.

## 1.2 Problem Statement

It is said that the development of an AI model is under carefully controlled conditions, whereas the conditions of the deployment of model in the real world are rarely perfect and risks are abundant. A metaphor is used, describing the development as a greenhouse and the deployment as a jungle [4]. This goes to show that we have to take some approaches to make the AI model robust to adversity, and the way to do that is by performing adversarial defenses.

The existence of adversarial attack has posed a tremendous threat to the robustness and trustworthiness of machine learning models. It is of significant importance to ensure that our deep learning models are robust enough to defense against adversarial attacks for us to employ these models in real-life scenarios like public security, communication, and financial transaction.

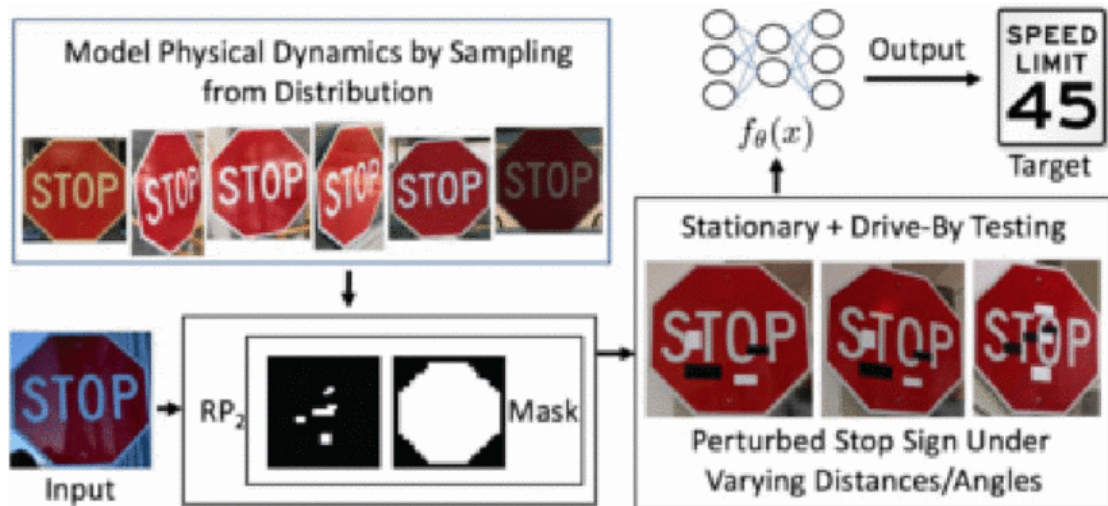


Figure 1.2 A physical perturbation applied to a stop sign causing the LISA-CNN classifier to misclassify the stop sign as a speed limit of 45km/h [5]

Figure 1.2 shows a physical-world attack on traffic sign classification task where a stop sign is being misclassified as a speed limit of 45km/h. Imagine this adversarial attack being employed in a real-world autonomous car system, where the autonomous car mistook a stop sign for a speed limit sign and made a wrong decision during its driving course. Disastrous consequences would have incurred and lead to the loses of properties and human lives. Hence, the robustness of machine learning models is vital if we are going to employ these models in real life settings so that safety and security can be guaranteed, and adversaries cannot easily hack into the model and cause harm to the society. Dire implications can incur if the machine learning models are hacked by adversaries since these models are widely adopted in nearly all aspects of our lives now. To sum up, the research on adversarial machine learning are of interest not only to the researchers but also to the general public.

### 1.3 Motivation

## CHAPTER 1

The aim of the thesis is to research and make improvement on the current defense methods against adversarial attacks in the field of image recognition so that the adversarial robustness of machine learning models can be enhanced. As mentioned in the problem statement, the discovery that the addition of a small amount of perturbation, imperceptible to human, to the input image can lead to the misclassification of a classifier has come to be a shock to the society. The fragility of machine learning classifier to such perturbation can lead to countless loss if they are exploited by the adversaries. Hence, it is vital for us to come out with solutions that can improve the robustness of machine learning systems that are capable of not only solving the problem but also in an efficient and computationally inexpensive way.

### **1.4 Project Scope**

At the end of the project, we will come out with an adversarial defense method that can enhance the adversarial robustness of image classification machine learning models. The method will incorporate noises in the input images as well as combine both methods of adversarial purification and adversarial training, and both will harness the use of a diffusion model in different way.

In this project, we will:

#### **1. Propose and Research Methods and Effects of Incorporating Noise**

Incorporating noises play a vital role in the adversarial defense method that we propose. We propose to add noises to the input images before going into the adversarial purification process, as incorporating noises in the input will incorporate randomness and it is a common approach used by other researchers to counter adversarial attack. We would like to find out if incorporating noise in our proposed method will help in improving adversarial robustness, and also explore and find out different methods of noise incorporation. The process of adversarial purification using a diffusion model also involves the addition of noise in a series of steps.

## **2. Research the Effect of Combining Adversarial Purification and Adversarial Training**

We aim to determine whether the integration of adversarial purification and adversarial training can enhance adversarial robustness, and whether the incurred costs are justified by the improvement in robustness. We will assess the individual performance of these methods based on the computational cost and time required for model training, both independently and when combined.

### **1.5 Project Objective**

The main research objectives are to (1) reduce the computational resources needed in making a model robust to adversarial attack, (2) propose an adversarial defense method that is easy to implement, and (3) propose an adversarial defense method that is general enough to apply to any machine learning models.

### **1.6 Contributions**

We investigated various ways of incorporating noise in the adversarial defense pipeline and the effect of incorporating those noise. The research is significant as it provides insights of noise incorporation helps in defending machine learning models against adversarial attacks, which can negatively impact the trustworthiness of machine learning models in performing tasks especially safety-critical tasks such as medical diagnostic tasks and autonomous driving tasks.

### **1.7 Report Organization**

This report is organised into 6 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 Method, Chapter 4 Experiment and Chapter 5 Conclusion. Chapter 2 reviews some of the existing adversarial attacks and defenses, and the limitations of current research. Chapter 3 will discuss our method; Chapter 4 shows the experimentation done to investigate how to improve the adversarial robustness of a deep learning system. Chapter 5 concludes our research



## Chapter 2

### Literature Review

Section 2.1 summarizes the existing adversarial attack methods. Following is Section 2.2 which is related to existing defense methods. Section 2.3 discusses additional notes on defense methods regarding proven defense and certification.

#### 2.1 Adversarial Attack

In this section, we briefly summarise the existing adversarial attack methods. Adversarial attack is the process of generating an adversarial example given a natural sample and a victim model. To put it in a more formal way, given a natural input  $\mathbf{x}_0$ , an adversarial attack aims to find a small perturbation  $\delta$  such that the adversarial example  $\mathbf{x}^* = \mathbf{x}_0 + \delta$  looks no different from  $\mathbf{x}_0$  to human but will be misclassified by the victim model [6]. This process is illustrated in Figure 2.1 below.

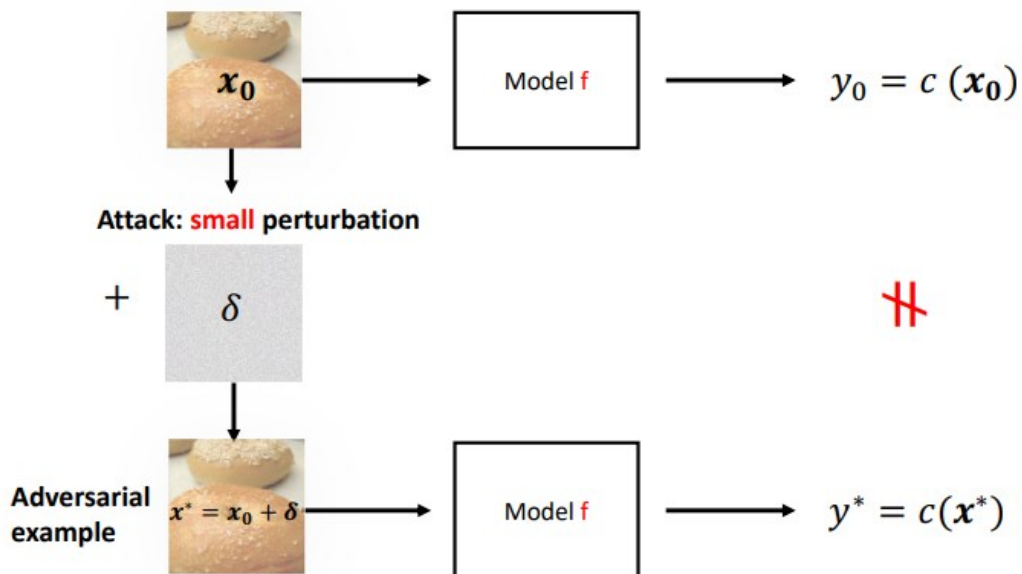


Figure 2.1 Illustration of adversarial attack [6]

Adversarial attacks have two kinds of attack goals in general, which are targeted and untargeted attacks. An untargeted attack is considered successful if the input is predicted with any labels other than the original correct label, whereas a targeted attack is only successful when the adversarial example is classified as the target class.

## CHAPTER 2

Generally speaking, there are two types of adversarial attack settings, which are (1) black box settings and (2) white box settings. In black box attack settings, the attacker does not have access to the model parameters. Black box attack is further divided into soft-label and hard-label black box setting. When the predicted scores are available, it is a soft-label black box attack, and when only the predicted labels are revealed, it is a hard-label black box attack. Whereas in white box attack settings, attackers have access to the model, including the model's parameters and its architecture. In addition, there is also the gray-box attack where only partial information of the model is known to the attackers. Most of the time, however, attacks are under the black box attack settings as the access to the target model is usually limited.

Next, according to the amount of information required to perform the adversarial attack, there are three categories of adversarial attack, namely (1) gradient-based, (2) score-based, and (3) decision-based [6].

Gradient-based attacks are under white-box attack settings where it is assumed that the attackers have access to the model parameters. Gradient-based attacks generate adversarial examples through gradient back-propagation, some examples are Fast Gradient Sign Method (FGSM) [2], Basic Iterative Attack (BIM), Projected Gradient Descent (PGD).

Score-based attacks are performed when output scores of the victim classifier are available but not the detailed information like model parameters. In such situations attackers may try to estimate the gradient with the score information and generate adversarial examples based on the estimated gradient. Adversarial attacks that fall under this category include Zeroth Order Optimization Attack (ZOO) [7] and NES attack.

Decision-based attacks are employed when only the predicted labels are returned by the victim models, which is under the category of hard-label black box settings. One of the attack methods proposed is the transfer-based attack where instead of attacking the target model directly, they attack a substitute model that is trained purposely to mimic the target model [8]. The adversarial examples generated based on the substitute model is then transferred to attack the target model. Another attack method is the random-walk based attack where the attackers start with a sample from the targeted class and seek to minimize the perturbation while staying adversarial [9].

## 2.2 Adversarial Defense

We will discuss some of the existing adversarial defenses in this section. According to [10], adversarial defenses can be divided into two categories, which are (1) proactive defense algorithms and (2) passive defense algorithms. While proactive defense works by improving the robustness of the model itself, passive defense works by mitigating or eliminating adversarial perturbations. While adversarial defenses may lead to an increase in model robustness against adversarial attacks, it should be noted that there exists a recognized trade-off between accuracy and robustness in machine learning models [11]. That is, models with higher accuracy tend to be more vulnerable to adversarial attacks.

### 2.2.1 Proactive defense algorithm

Proactive defense schemes strengthen the model in order to achieve better robustness. It often requires retraining of the model or additional optimization on the model parameters. The most representative proactive defense is the adversarial training, which was the most successful empirical defense known [12]. Adversarial training involves generating adversarial samples with adversarial attacks and train the model using them as input data. It can be seen as a special type of data augmentation, where the training data is augmented with adversarial examples. [10] describes the basic adversarial training steps as follow, where steps 2 and 3 can be repeated for multiple iterations.

- (1) Given a training set  $\mathbf{x}$ , and a model  $f$  to be trained. We use  $\mathbf{x}$  to train the model  $f$ .
- (2) For each benign image  $x^n$ , generate its corresponding adversarial example  $\widetilde{x}^n$  by an attack algorithm, hence creating a new training set  $\mathbf{x}'$ .
- (3) Use both  $\mathbf{x}$  and  $\mathbf{x}'$  to update the model  $f$ .

Adversarial training is first proposed in [2] where the authors attacked the model with FGSM attack and adding the adversarial samples back to the training dataset to enhance the robustness of the model. However, adversarial examples created by FGSM through

only one step iteration is not strong enough and the model trained using FGSM adversarial attack is still vulnerable to iterative attack.

Adversarial training can also be expressed as a min-max optimization problem with the following objective function:

$$\min_{\theta} \max_{\delta \in S} L(\theta, x + \delta, y)$$

Here,  $x$  is the input data,  $y$  is the label, and  $L(\theta, x + \delta, y)$  is the adversarial classification loss. The outer minimization is over the network parameters whereas the maximization is over the adversarial perturbations. The min-max formulation is introduced by [13], which employed PGD to generate adversarial examples. In addition, it is noteworthy that PGD adversarial training only trains on adversarial examples. PGD adversarial training can survive a series of attack methods and it remains the baseline method for many state-of-the-art defense methods [10].

While adversarial training could make the model more robust to adversarial attacks, heavy computation load is also involved and it could lead to high cost, precluding its use on large datasets. Hence, researchers have proposed other methods that aim to make the process of adversarial training less computational expensive, such as Adversarial Training for Free (FreeAT) and You Only Propagate Once (YOPO).

Besides adversarial training, another proactive defense strategy is the neural network optimization which aims to improve the robustness of the model by improving the parameters of the network. This can be achieved by regularization, where a regularization term is added to model during training. [14] has shown that training a differentiable model with gradient regularization could achieve robustness comparable to adversarial training, while at the same time making the adversarial perturbation more interpretable by human subjects. However, gradient regularization method may take longer time to converge. As a second-order method, it may not work under all operations in all auto-differentiation frameworks [14].

Other than applying regularization to a model, neural network optimization can also be achieved by adjusting the structure of the network, for example the authors in [15] proposed to insert a mask layer in a DNN model before the linear layer that is handling classification, where the mask will filter out extra unnecessary features from the ones that are extracted by the classifier. The reasoning behind is that those unnecessary

features can be exploited by adversaries and therefore should be removed to improve the model robustness. The merits of this method is that it is easy to implement and computationally efficient.

### 2.2.2 Passive defense algorithm

Unlike proactive defense algorithms discussed above, passive defense algorithm does not strengthen the model itself, instead it aims to reduce the implications caused by the adversarial perturbation [10]. The advantage of passive defense algorithm is that robustness of the model can be enhanced without making changes to the model. In other words, the process of model training and adversarial defense is decoupled, and typically no retraining of model is required. However, the enhancement in model robustness might not be as well as proactive defense algorithm such as adversarial training.

According to [10], DNN is usually robust to randomization added to the image. However, that is not the case for adversarial examples especially those generated by gradient information. Hence, randomization is one of the techniques that could potentially mitigate adversarial attacks. [16] proposed to perform random resizing and random zero padding to the input images at inference time before feeding them to the classifier. The authors demonstrated the effectiveness of their approach in large-scale dataset as well as against iterative attacks.

Moreover, [17] proposed to perform image transformation techniques including bit-depth reduction, JPEG compression, total variance minimization and image quilting before feeding the image to a classifier. In addition, the authors in [17] suggested that an input-transformation defense should be difficult to differentiate and randomized enough to be considered strong.

Authors in [18] proposed a defense algorithm named Random Self-Ensemble (RSE) by combining two important concepts: randomness and ensemble. The randomness is incorporated into the model by adding random noise layers to the neural network before each convolution layer. On the other hand, the authors showed that their algorithm is equivalent to an ensemble of an infinite number of noisy models without any additional memory overhead, as compared to an ensemble of finite  $k$  models that will increase the model size by  $k$  folds.

## CHAPTER 2

Although RSE could improve robustness in models, the accuracy sacrificed is also non-negligible. The authors in [19] proposed another adversarial defense framework named Adv-BNN which employs a min-max formulation that combines adversarial training with Bayesian Neural Network (BNN). The authors assume that the weights in the network are stochastic and train the model with the usual techniques in BNN, and conduct the experiment with the hopes that the randomness in the weights in BNN could provide better robustness in the model. Adv-BNN has produced significant improvement over RSE and adversarial training. An observation made by the authors is that although BNN has no defense functionality itself, combining BNN with adversarial training will improve the robustness against adversarial attacks significantly.

Adversarial purification represents an alternative method for defending against adversarial attacks. This approach employs a generative model to cleanse adversarially perturbed inputs before they are passed to a classifier. One of the pioneering studies in this domain is by the authors referenced in [20], who proposed the utilization of a Generative Adversarial Network (GAN) to mitigate the impact of adversarial perturbations on the model's predictions. However, to achieve satisfactory results, it is imperative that the model is adequately trained and fine-tuned.

In recent years, diffusion models have gained prominence as another category of generative models in the realm of image synthesis tasks. A seminal work that integrates diffusion models into adversarial purification is DiffPure, as proposed by the authors cited in [21]. DiffPure employs a systematic approach involving the addition of noise and subsequent denoising through forward and reverse processes to cleanse the adversarial perturbations from the input data.

Despite the potential advantages of adversarial purification methods, they present certain challenges. Notably, they are computationally intensive, particularly in terms of the memory resources required for deployment, which limits their suitability for downstream applications. Furthermore, their high computational demands preclude their use with RobustBench, a prominent benchmark in the field of adversarial defense research. RobustBench relies on AutoAttack, a powerful ensemble of attack that is also computationally demanding. Nonetheless, adversarial purification methods show considerable promise since they do not necessitate retraining of the classifier. This

characteristic significantly enhances their generalizability compared to adversarial training, which may be less effective when encountering unknown threats.

### 2.3 Additional Notes on Adversarial Defense

Most of the defensive approaches are empirical defenses, which means that there is no theoretical guarantee that after applying the defensive algorithm the models will be truly robust against adversarial perturbations. In other words, these defenses may be broken by a new attack in the future, as in many cases in the past where a defense method was broken by other newly proposed attack, soon after the proposal of the defense method. On the other hand, there was certified defenses that guarantees a classifier's prediction to be constant within some set around the input  $x$ , often an  $l_2$  or  $l_\infty$  ball [12]. In some cases, certification mechanisms are proposed by researchers to certify the robustness of models. However, these provable defenses are usually conducted in small scale network or small dataset [10].

One theoretically proven defense algorithm was proposed in [12], which is the randomized smoothing method. The authors proved a tight robustness guarantee in  $l_2$  norm for smoothing in Gaussian noise, and their method permits the use of arbitrary large neural networks unlike other certified defenses. The authors conclude with the remark that smoothing with Gaussian noise is a promising direction for future researches into classification that is robust to adversarial attacks.

In [22], the authors proposed CNN-Cert, a certification framework for Convolutional Neural Network (CNN). According to the authors, CNN-Cert can handle various CNN architectures such as convolutional layers and max-pooling layers, and it was computationally efficient compared to other certification algorithms such as Fast-Lin and CROWN.

### 2.4 Limitation of Previous Studies

As mentioned above, adversarial training is often computationally expensive, but in overall it produces the most adversarially robust model. On the other hand, while other defensive methods such as input pre-processing method and randomization are

## CHAPTER 2

computationally inexpensive, their defensive capabilities are usually limited, where problems such as adding more randomization factor or noise to the input or model will deteriorate the performance of the model. Adversarial purification, while promising in their robustness performance and generalisability, is more computationally costly as well. In the reviewed papers, we found that researchers have tried incorporating randomness into the models by various methods such as RSE and Adv-BNN, but only Adv-BNN has combined adversarial training with BNN which infuse randomness into their defensive strategy, and other methods rarely investigated the effect of combining adversarial training and randomness incorporation into one defensive method. Additionally, the effect of combining adversarial purification and adversarial training together is not yet explored.



# Chapter 3

## Method

### 3.1 Noise Incorporation Preprocessing Pipeline

In this section, we will review several types of noise commonly encountered in computer vision applications, including Gaussian Noise, Salt and Pepper Noise, and Uniform Noise. Additionally, we will also examine Gaussian Blur, which, although not typically referred to as 'noise,' is included in this discussion for its relevance.

#### 3.1.1 Gaussian Noise

Gaussian Noise is one of the most common types of noise in computer vision, where the distribution of the noise follows a Gaussian (normal) distribution. In adversarial robustness sense, Gaussian noise can be useful to test a model's resistance to small, random perturbations, helping researchers to evaluate how robust their models are to minor, local modifications. Gaussian noise adds random variations in pixel intensity, distorting the original image, and is typically characterized by its zero mean and varying standard deviation  $\sigma$ , which determines the level of noise in the image.

The equation of Gaussian noise is as follows:

$$I_{noisy}(x, y) = I_{original}(x, y) + N(0, \sigma^2)$$

Where:

$I_{noisy}(x, y)$  is the noisy image at pixel coordinate  $(x, y)$

$I_{original}(x, y)$  is the original image at pixel coordinate  $(x, y)$

$N(0, \sigma^2)$  is the Gaussian distribution with mean 0 and variance  $\sigma^2$

#### 3.1.2 Salt and Pepper Noise

Salt and pepper noise is a type of impulsive noise commonly encountered in image processing. It is characterized by the presence of randomly occurring white and black pixels in an image, resembling the appearance of salt (white) and pepper (black) scattered across the image. By applying salt and pepper noise in an image, a small

percentage of the total pixels in the image are randomly set into the maximum intensity (salt) and minimum intensity (pepper).

Let  $P_{pepper}$  represents the probability that an affected pixel takes on the value of 0 (black, pepper) and  $P_{salt}$  represents the probability that an affected pixel takes on the value of 255 (white, salt). Then the probability that a pixel  $x'(i, j)$  will be affected by salt and pepper noise is given by:

$$x'(i, j) = \begin{cases} 0 & \text{with probability } P_{pepper} \\ 255 & \text{with probability } P_{salt} \\ x(i, j) & \text{otherwise} \end{cases}$$

### 3.1.3 Uniform Noise

Uniform noise refers to a type of random noise where the noise values are uniformly distributed over a specified range. In other words, each possible value within a certain range has an equal probability of being selected. Mathematically, the probability density function (PDF) for a uniform noise between the values  $a$  and  $b$  is given as:

$$P(x) = \frac{1}{b-a}, a \leq x \leq b$$

Where  $a$  and  $b$  are constants representing the lower and upper bound of the uniform noise distribution.

### 3.1.4 Gaussian Blur

Instead of adding noise to image, Gaussian blur is a technique commonly used to reduce noise. It is based on the Gaussian function, which creates a smooth and gradual transition from the edges of an object, making the image appear blurrier. The Gaussian function, which defines the distribution of the blur across the image, is given as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where  $x$  and  $y$  are the pixel distances from the center of the blur kernel, and  $\sigma$  is the standard deviation that controls the spread of the blur. When applying Gaussian blur, a kernel is used, and the strength of the blur is controlled by both the kernel size and the

standard deviation. A larger kernel and standard deviation will result in a stronger, more spread-out blur.

### 3.2 Adversarial Purification by Diffusion Model

Diffusion models are specialized generative models primarily used for image generation tasks. There are many variations of diffusion models, including Denoising Diffusion Probabilistic Model (DDPM), Score-Based Generative Model (SGM) and Stochastic Differential Equation (Score SDE). However, the idea behind how these models work is generally the same. These models typically involve two main procedures: the forward process, where a sample is progressively noised, and the reverse process, in which the sample is ultimately denoised to produce the final image. In this context, the resulting image will be a clean image suitable for input into the classifier.

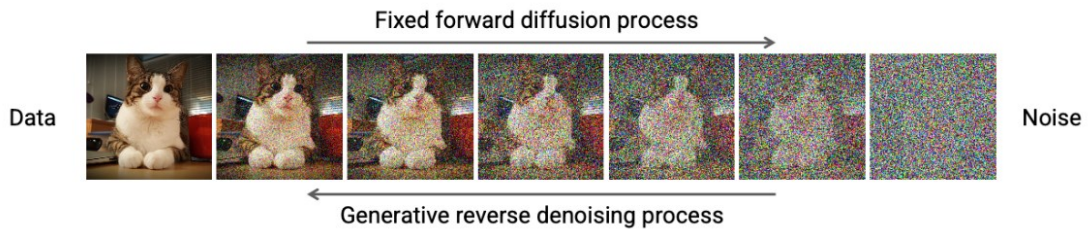


Figure 3.1 The forward and reverse process of diffusion model [23]

In the forward process, Gaussian noise is gradually added to the input image  $x_0$  through a series of  $T$  steps. Given a data point  $x_0$  sampled from the real data distribution  $q(x)$ , that is,  $x_0 \sim q(x_0)$ , the forward process can be described as a Markov chain of  $T$  steps, where each step of the noise addition is only dependent on the previous step, producing a sequence of noisy examples  $x_1, \dots, x_T$ . In each step, a new example  $x_t$  with distribution of  $q(x_t | x_{t-1})$  can be produced by adding Gaussian noise with variance  $\beta_t$  to  $x_{t-1}$ . The step sizes can be constant or being controlled by a variance schedule  $\{\beta_t \in (0,1)\}_{t=1}^T$  where the schedule can be linear, quadratic, cosine and so on.

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad q(\mathbf{x}_{1:T} | \mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1})$$

Figure 3.2 Equation for forward diffusion process

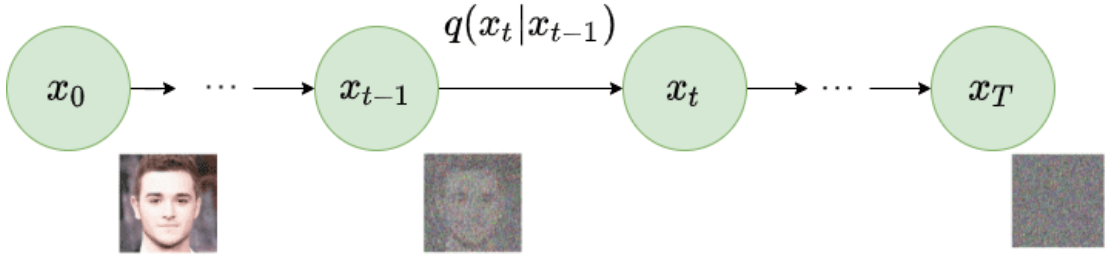


Figure 3.3 Forward diffusion process [24]

For the reverse process, the aim is to reverse the above process and recover the true input from the noised input  $x_t \sim N(0,1)$  by learning the reverse distribution  $q(x_{t-1} | x_t)$ . However, this cannot be done easily as the entire dataset is required to do so. Hence, a parameterized model  $p_\theta$  is used to approximate  $q(x_{t-1} | x_t)$  by parameterising the mean and variance. The model mentioned here could be a neural network, being used to learn the parameters to predict the mean and variance for each time step, where  $\mu_t(x_t, t)$  is the mean and  $\Sigma_\theta(x_t, t)$  is the covariance matrix in the equation shown in Figure 3.2.

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) \quad p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Figure 3.4 Equation for reverse diffusion process

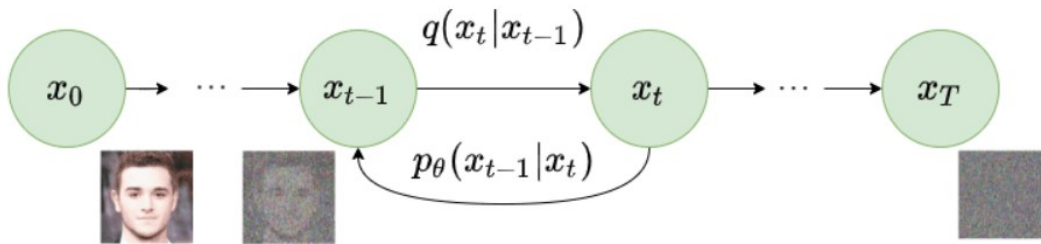


Figure 3.5 Reverse diffusion process [24]

### 3.3 Proposed Methodology

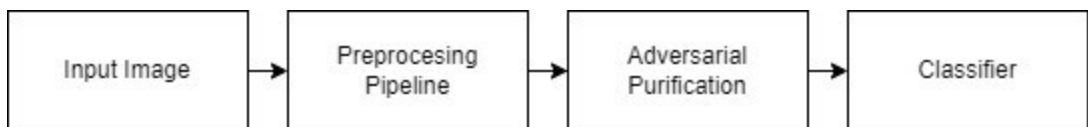


Figure 3.6 Proposed Methodology

Based on the limitations outlined in Chapter 2, we propose a defensive strategy that combines adversarial purification using a diffusion model with adversarial training. Specifically, we suggest that each clean and adversarial sample undergo a series of preprocessing steps involving the addition of Gaussian noise to the image, and make the input go through an adversarial purification process before it is input into the classifier, as shown in Figure 3.6. The incorporation of Gaussian noise is a commonly employed method by other researchers to introduce randomness and mitigate adversarial attacks. We consider adversarial perturbation as a form of special noise, in which we will introduce Gaussian noise to ‘disrupt’ the perturbation in some way. Besides Gaussian noise addition, we also investigated the effect of introducing some other types of noises including Salt and Pepper noise and Uniform noise. At the end of the preprocessing pipeline, we included the use of Gaussian Blur to mitigate the effect of the extra noises added. This design consideration is due to excessive noises can affect the classification accuracy of the classifier. Although we will feed the resulting noisy images into a diffusion model for adversarial purification later, which may recover the clean image from the noises added, it is a reasonable consideration to include Gaussian Blur, a type of image smoothing technique, to reduce the noise before adversarial purification. Figure 3.7 shows the preprocessing pipeline proposed in our system.

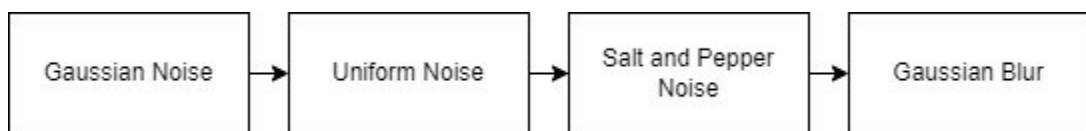


Figure 3.7 Preprocessing Pipeline Proposed

Subsequent to the introduction of Gaussian noise into the input, the sample will undergo the adversarial purification process. During this phase, the input will be passed through a pre-trained diffusion model. In our system, the pretrained diffusion model is proposed by authors in [25]. In fact, this diffusion model in [25] is designed specifically to purify adversarial images, named DiffPure, a type of Variance-Preserving SDE. The concept is to add Gaussian noise to the input image until a diffusion timestep of  $t^*$ , then recover the clean image from the diffused image at diffusion timestep  $t^*$ . The larger the diffusion timestep  $t^*$ , the greater the ability to destroy the adversarial perturbation in

## CHAPTER 3

the image, but the original semantic information in the image will be lost as well. Hence, experiments are done by the authors to find the optimal  $t^*$  for the adversarial purification task, which may differ from one dataset to another. It is notable that in our system, we proposed to apply a preprocessing pipeline that introduces extra noises into the input images, hence it would be worth to experiment and find out if the value of  $t^*$  will change due to the preprocessing pipeline. However, due to the time constraints, this will be left as a future work to be done.

Last but not least, the image is fed into the classifier, after going through the preprocessing pipeline and adversarial purification process. Any classifier can be used here, but our system uses the WideResNet-28-10 (WRN-28-10). We propose using the adversarially pre-trained model in [24], and finetuning the model parameters using the input that is preprocessed and adversarially purified, effectively combining the randomness introduced by the preprocessing pipeline, adversarial training and purification approach. However, as in the case before, this was not implemented due to time constraints and will be left as a future work. Instead, we tried to simply replace the classifier to an adversarially trained classifier without finetuning and show that combining adversarial purification and adversarial training in this way did improve the adversarial robustness of the model in the next chapter.

# Chapter 4

## Experiment

### 4.1 Hardware Setup

Some of the experiments are done on a machine equipped with Graphics Processing Unit (GPU) which is NVIDIA GeForce GTX 1080 and also NVIDIA GeForce RTX4070. Most of the experiments, however, is conducted on a cloud GPU provided by Kaggle, which is NVIDIA Tesla P100. A laptop is used to access the machine remotely and the Kaggle cloud GPU.

Table 4.1 Specifications of NVIDIA GeForce GTX 1080

Description	Specifications
CPU	Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
Operating System	Ubuntu
GPU	NVIDIA GP104 GeForce GTX 1080
Memory	16GB RAM
Storage	1TB HDD

Table 4.2 Specifications of NVIDIA GeForce RTX4070

Description	Specifications
CPU	Intel Core i7-13700 Processor @ 5.20GHz
Operating System	Ubuntu
GPU	NVIDIA GeForce RTX4070 12GB GDDR6X Graphics
Memory	32GB DDR4 DIMM RAM
Storage	512GB PCIE NVMe SSD + 1TB HDD

Table 4.3 Specifications of Kaggle Cloud NVIDIA Tesla P100

Description	Specifications
CPU	Intel Xeon
Operating System	Ubuntu
GPU	NVIDIA Tesla P100 16GB HBM2
Memory	13GB available for usage

Storage	61GB available for usage
---------	--------------------------

Table 4.4 Specifications of the laptop

Description	Specifications
Model	ASUS VivoBook X513EA_V5050EA
Processor	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
Operating System	Windows 10
Graphic	Intel® Iris® Xe Graphics
Memory	16GB RAM
Storage	256GB

## 4.2 Environment Setup

On the machines, the environment has to be set up before any experiments can be done. This includes installing libraries such as Git, Torch, Torchvision, IPython, TorchAttack and pulling GitHub repositories to get access to the code published by other authors. Our experiment uses PyTorch to build deep learning models and relevant operations such as transformation pipeline. Jupyter Notebook is used for interactively writing the code. In addition, pretrained diffusion models and classifiers are downloaded from respective sources as well.

## 4.3 Model architecture and Dataset

The main model architecture used in this project will be WRN-28-10, a popular architecture that is widely adopted and has achieved state-of-the-art performance on various image classification benchmarks, including CIFAR-10 dataset. WRN-28-10 is chosen because it involves less model parameters than its variant, WRN-70-16, which is also commonly used in the adversarial robustness research, hence will take less time to train the model. WRN-28-10 refers to the Wide Residual Network with a depth of 28 and a widening factor of 10, and it is a variant of the Residual Network (ResNet) architecture.

The WRN-28-10 has a total depth of 28 layers, which includes both convolutional and residual layers. The depth of the network plays a crucial role in capturing complex features from the input images. The widening factor of 10 indicates that the number of



channels in the convolutional layers is 10 times greater than the original ResNet architecture. This wider architecture allows the network to learn more diverse and richer features, which can improve the classification performance. In addition, the basic building block of WRN is similar to that of ResNet, consisting of two convolutional layers and a skip connection (or shortcut connection). The main difference lies in the increased number of channels due to the widening factor. The network ends with a global average pooling layer followed by a fully connected layer with softmax activation to produce the final class probabilities.

The project will be using CIFAR-10, which is a popular benchmark dataset in the field of computer vision and machine learning. CIFAR-10 consists of 60,000 32x32 colour images in 10 different classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. CIFAR-10 is chosen as the dataset because the number of classes involved is less than other datasets and has a very small image resolution of 32x32, hence it will require less computational resources to conduct the research. However, it should be noted that the small image resolution may affect the performance of the overall network, as less visual semantics can be captured within such small image resolution, and the already little semantic information may be destroyed by the addition of noises in attempt to disrupt the adversarial perturbation.

### **4.4 Evaluation metrics and benchmarks**

This research will employ both clean and robust accuracy metrics to assess the model's robustness. To benchmark our work against existing research and evaluate the performance of our model, we will utilize RobustBench [26]. For the adversarial attack conducted during evaluation, we propose to employ AutoAttack. AutoAttack is an attack method designed by authors in [27] to provide reliable evaluations of adversarial defense mechanisms, thereby mitigating the risk of generating a false sense of robustness resulting from improperly tuned attack hyperparameters and gradient masking issues. However, as an ensemble of several strong adaptive attacks, AutoAttack consumes high computational resources and takes a very long time to evaluate the model. Hence, for our experiment purposes, we will use only a subset of 512 test images instead of the whole test set for evaluation, as suggested in [25], as doing so will incur only very little discrepancy in the robust accuracy reported. In addition, during results and experiments involving adversarial purification with

diffusion model and noise incorporation, we uses PGD-10 with  $\text{eps} = 8/255$  and  $\text{step\_size} = 0.003$ . Every experiment is done using the Linf norm except during result reproduction that also involves L2 norm.

### **4.5 Result Reproduction Attempts**

In this section we describe the attempts to reproduce the results in [25] for adversarial purification and [24] for adversarial training. This is mainly to get an idea of how feasible it is for us to conduct our research based on available resources, and to avoid reporting inaccurate results because the number of images used during our evaluation is very small compared to the normal evaluation procedures.

Our attempt to reproduce results by running the standard AutoAttack with  $\text{eps} = 8/255$  for Linf norm and  $\text{eps} = 128/255$  on L2 norm on the adversarial training work by authors in [28], using a test subset of 512 images on both norms shows very similar results as reported in the papers, which is evaluated on the whole test set. We also tried their method to adversarially train the model from scratch, and upon realizing that we can never train a model as good as theirs due to resource limitations, mainly insufficient GPU memory, we decided to just use the model checkpoint provided by the authors for our own experiments.

The same resource limitation is even severe for the attempt to reproduce the results in [25] as diffusion models are particularly known for their high requirements for computational resources not only during training, but also during inferences. As a result, we can only reproduce the result by applying the randomized AutoAttack with EOT of 5 on a test subset of 32 images. However, we found that although we used a very small test subset of only 32 images, the resulting accuracy is still very similar to what was reported in the papers, showing that our evaluation method remains reliable even with the resource limitations.

### **4.6 Noise Incorporation Pipeline**

In this section we describe our experimentations with adding various types of noises in our transformation pipeline, with different parameters such as noise levels. Specifically, we experimented with Gaussian noise, Salt and Pepper noise, Uniform noise and lastly, applying Gaussian blur to smooth out the noises.

## CHAPTER 4

After applying the noises with different noise levels, we feed them to the diffusion model for purification to determine the best possible transformation pipeline that can disrupt the adversarial noises, but not so much as to degrade the classification results. The intuition is that we should choose an appropriate transformation pipeline that introduces an optimal amount of noise (randomness) into the input, but not too significant until the diffusion model cannot ‘clean’ out the added noises.

Below shows the resulting image of adding noise, and the comparison among original image and noisy image both before and after the purification process.

### 1. Gaussian Noise

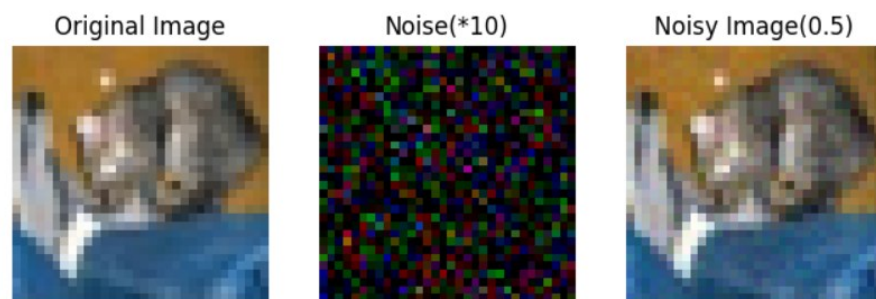


Figure 4.1: Effect of adding Gaussian noise (std = 0.02). The magnitude of the noise shown in the middle is magnified by x10 for easy visualization.

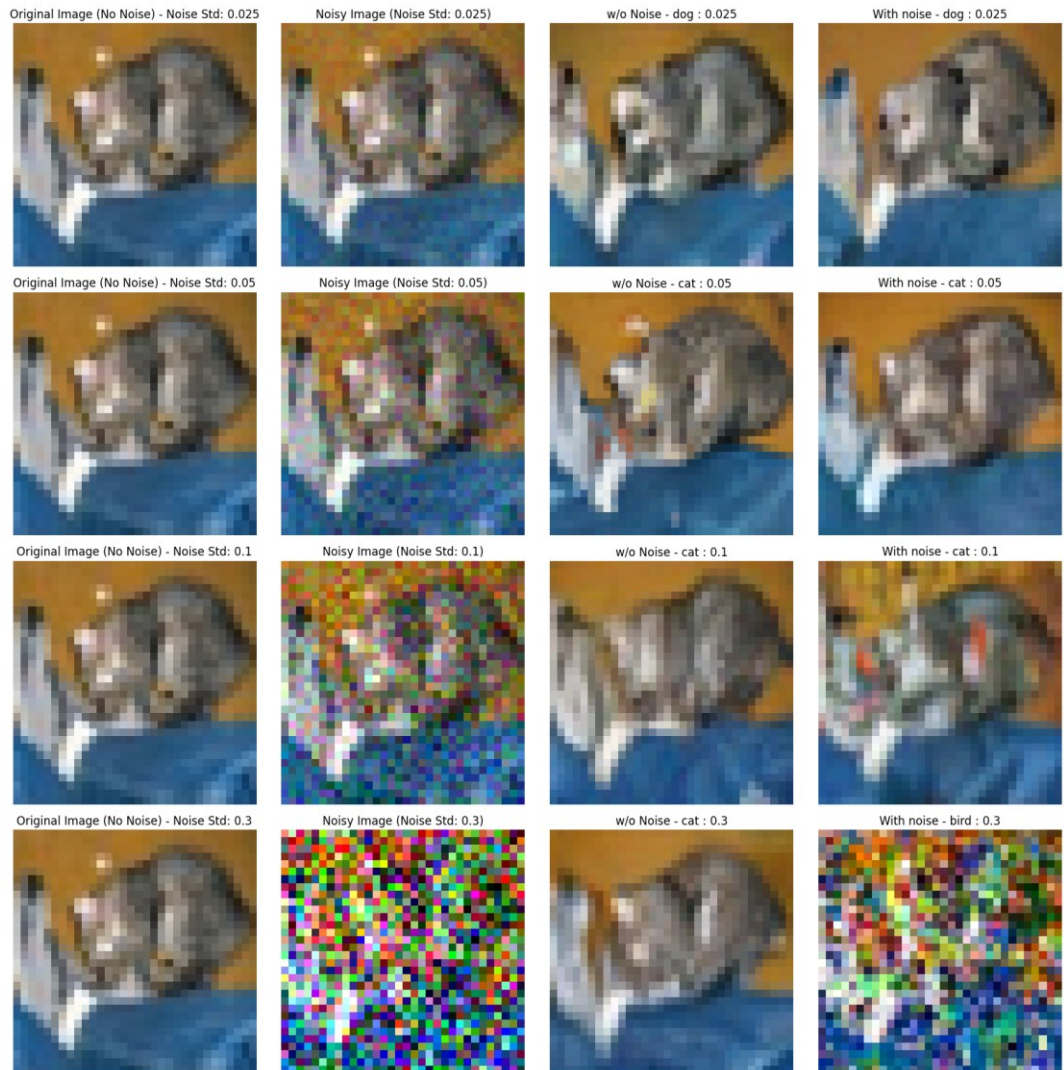


Figure 4.2: Experiment of Gaussian noises with  $\text{std} = [0.025, 0.05, 0.1, 0.3]$  from row 1 to row 4. The first two columns compare images before purification, the last two columns compare images after purification.

## 2. Salt and Pepper Noise

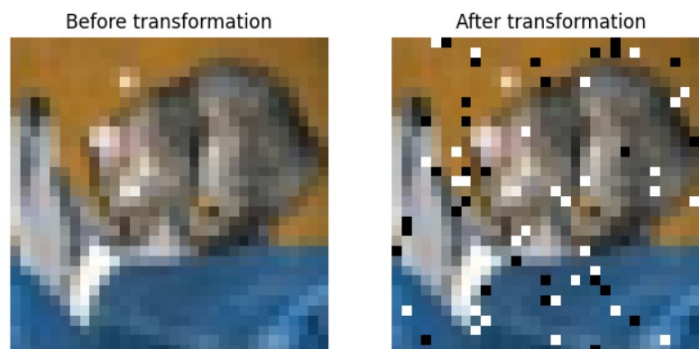


Figure 4.3: Effect of adding Salt and Pepper noise ( $\text{prob} = 0.01$ ).



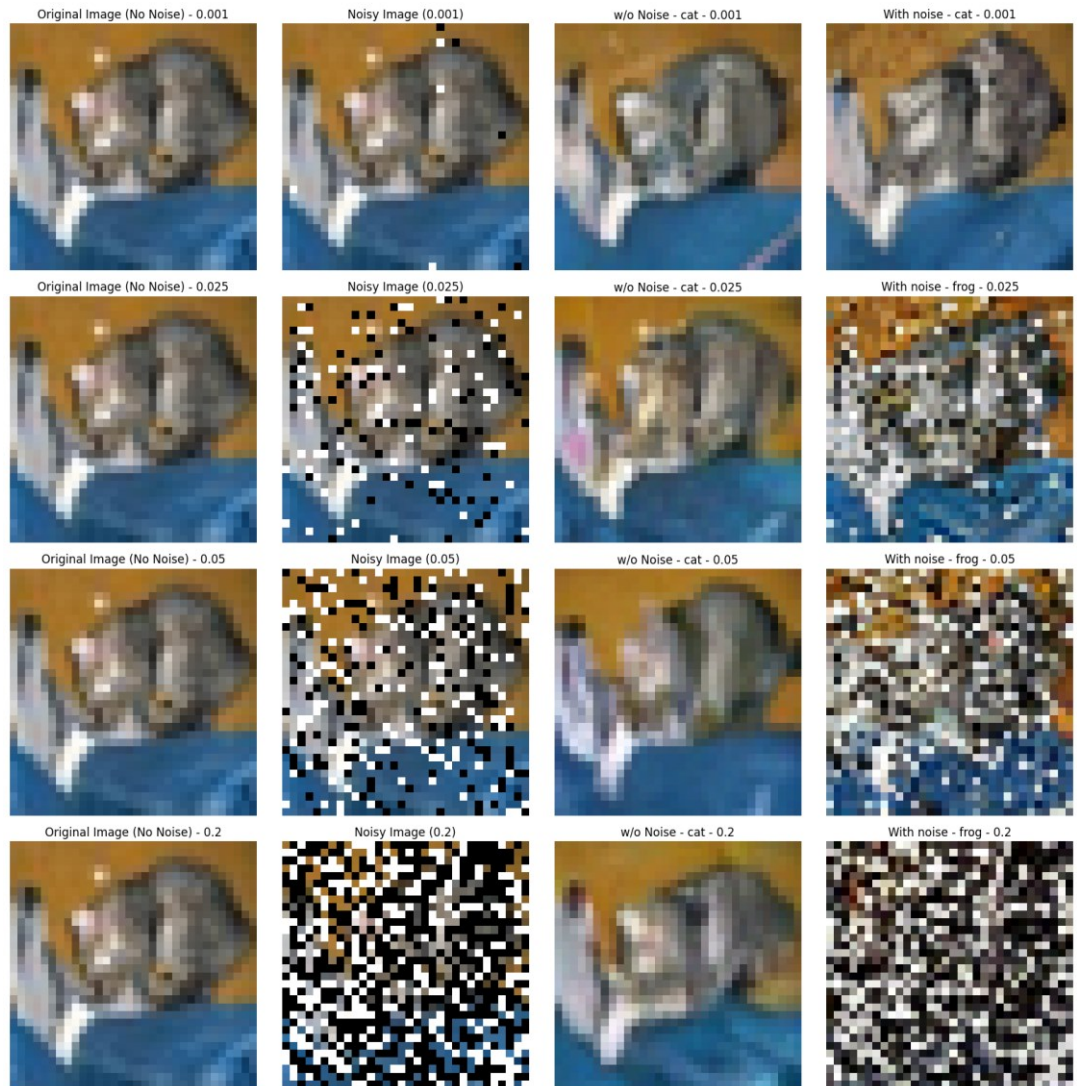


Figure 4.4: Experiment of Salt and Pepper noises with prob = [0.001, 0.025, 0.05, 0.2] from row 1 to row 4. The first two columns compare images before purification, the last two columns compare images after purification.

### 3. Uniform Noise

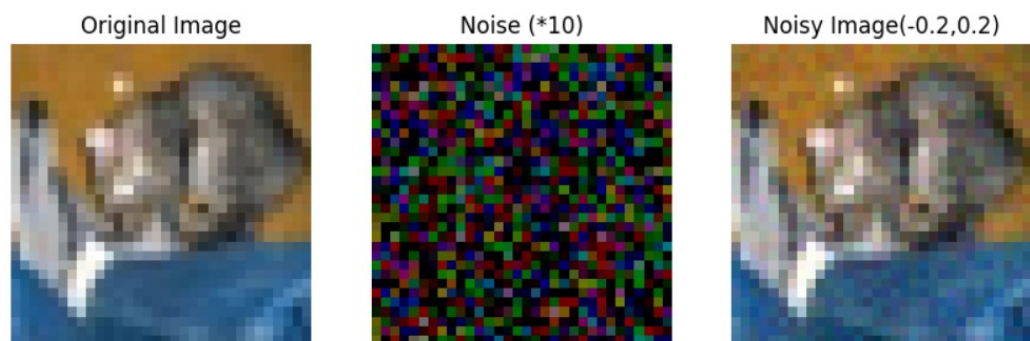


Figure 4.5: Effect of adding Uniform noise ( $a = -0.2, b = 0.2$ ). The magnitude of the noise shown in the middle is magnified by x10 for easy visualization.

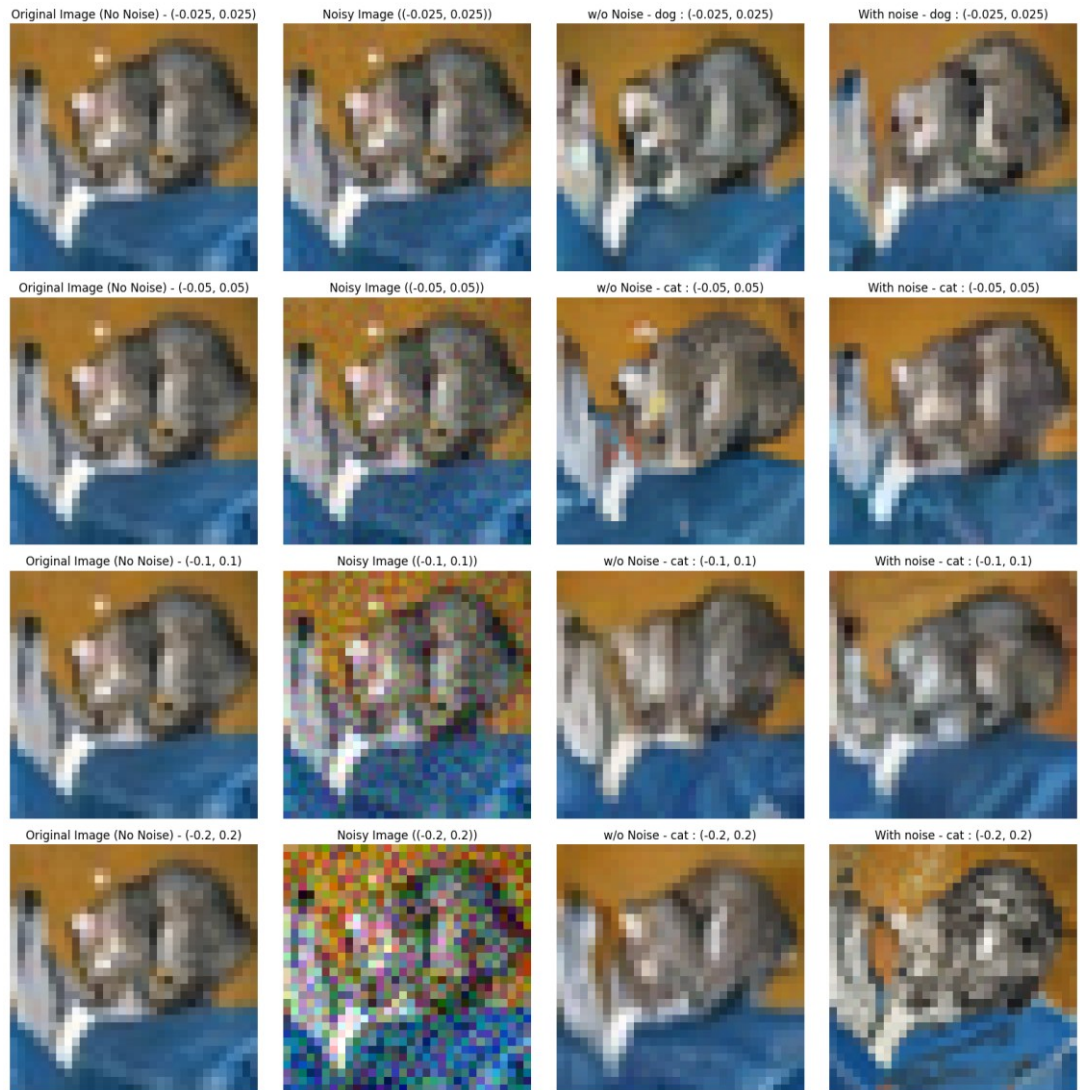


Figure 4.6: Experiment of Uniform noises with noise = [0.025, 0.05, 0.1, 0.2] from row 1 to row 4. The first two columns compare images before purification, the last two columns compare images after purification.

#### 4. Gaussian Blur

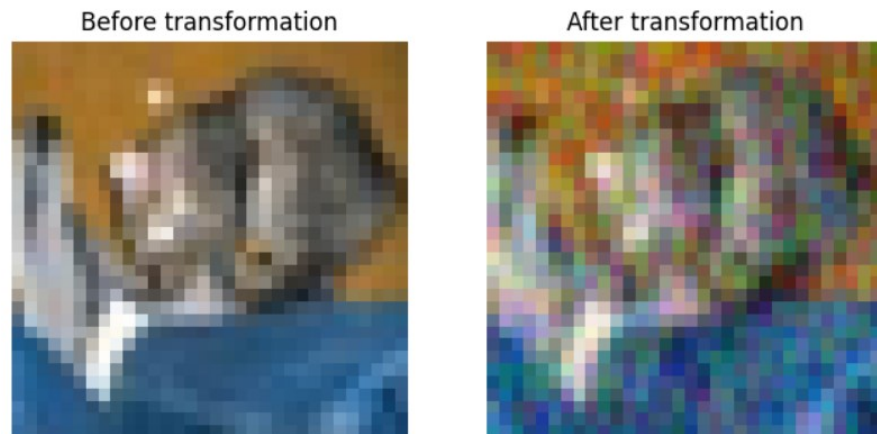




Figure 4.7: Effect of Gaussian Blur (kernel=61, sigma=(0.5,0.6)) on an image applied with a high level of noise. The noise is smoothed out quite significantly.

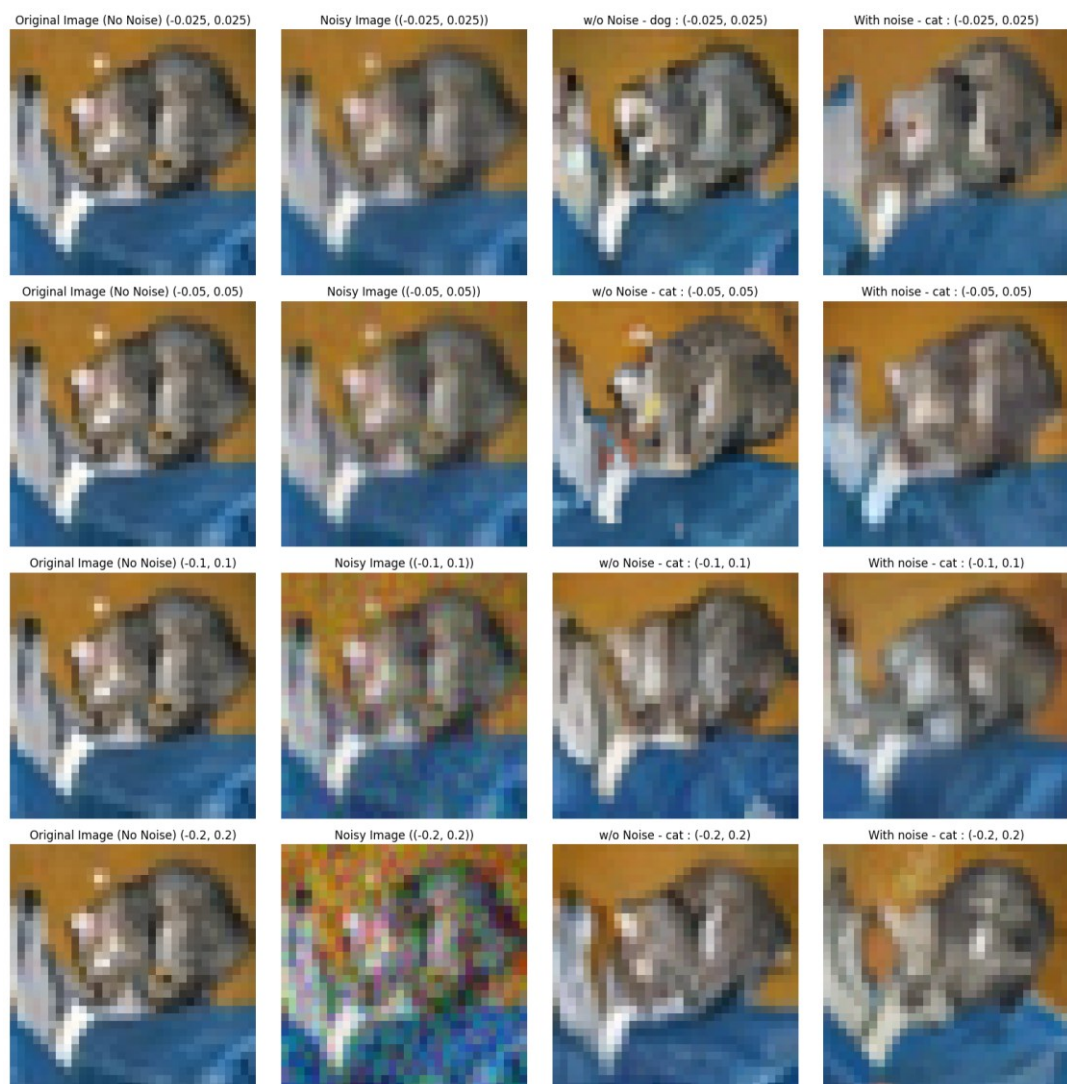


Figure 4.8: Experiment of applying Gaussian Blur (kernel=61, sigma=(0.5,0.6)) on noisy images from Figure 4.6. The first two columns compare images before purification, the last two columns compare images after purification.

After experimenting with the noises, we came out with a final preprocessing pipeline applying Gaussian noise, Uniform Noise, Salt and Pepper noise and Gaussian blur in order. Figure 4.9 shows the result of such transformations in the first row, and the resulting image after purification in the second row. We will continue our experiment with this final pipeline and make slight adjustments to the noise level in later experiments.

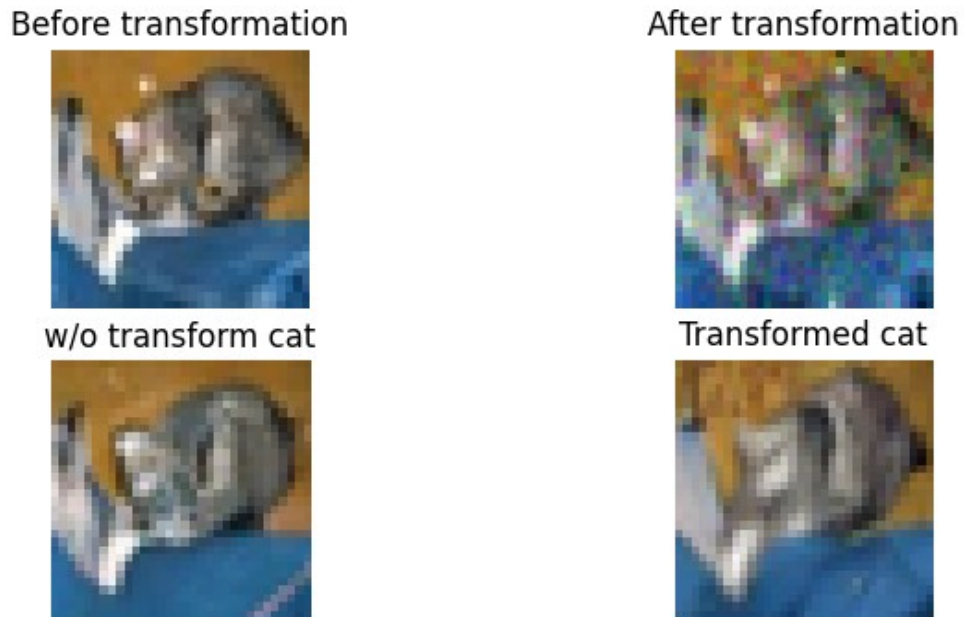


Figure 4.9: Applying the final pipeline.

#### 4.7 Testing Pipeline with Adversarial Attack

After proposing the preprocessing pipeline, we will now test the final pipeline with PGD-10 and show the classification result for the clean image, adversarial image and the transformed adversarial image with our proposed pipeline.



Figure 4.10: Comparison among clean image, adversarial image and transformed adversarial image and their classification results.

Figure 4.10 shows that by applying our preprocessing pipeline, it is possible to revert the classification result back to the correct one even after being adversarially attacked. Although every run of the code can produce different result, and it is not guaranteed that after transformation we will get back the correct result, we see that it is possible to



do so by incorporating the preprocessing pipeline, showing that noise incorporation has the ability to disrupt the adversarial noise.

Next, we evaluate the effect of the preprocessing pipeline by applying PGD-10 on a normal WRN-28-10 classifier with slightly different parameters on the preprocessing pipeline (different noise level) to observe the effect on the accuracy. In Table 4.1, Pipeline 1 consists of Gaussian noise (std=0.025), Uniform noise (a=-0.15, b=0.15), Salt and Pepper noise (0.001) and Gaussian blur (kernel=61, sigma=(0.5,0.6). Meanwhile, Pipeline 2 is similar to Pipeline 1, except with Gaussian noise(std=0.02) and Uniform noise (a=-0.2, b=0.2).

Pipeline	Clean Acc.	Robust Acc.	Robust Acc. (after applying pipeline)
1	94.78	0.11	35.02
2	94.78	0.10	34.75

Table 4.5: Results on applying two different pipelines after PGD-10 attack.

Results show that slight adjustment on the noise level does not seem to affect the clean and robust accuracy. Also, we see that incorporating the preprocessing pipeline indeed has its effects on disrupting the adversarial noise as the difference between applying the pipeline or not is significant (~35%). However, since we apply the pipeline in a whole, we cannot determine which noise is more effective on the disruption. More detailed tests can be conducted in the future to research the effect of incorporating these noises in depth.

#### 4.7 Testing Pipeline with Adversarial Attack and Purification

Then, we test out the effect of the preprocessing pipeline by feeding the output to the diffusion model for adversarial purification, as suggested by our proposed solution. PGD-10 is used as the method for evaluation, evaluated on a test subset of 512 images, with batch size of 64. Evaluation is ran twice on normal WRN-28-10, and once on an adversarially trained WRN-28-10. Pipeline 1 is the same as before, while Pipeline 2 now applies Gaussian noise(std=0.01) and Uniform noise (a=-0.1, b=0.1).

Pipeline	Clean Acc.	Robust Acc.	Robust Acc. (after applying pipeline)
1	86.91	83.98	79.30
2	90.62	81.25	76.76
2 (adv. trained)	87.3	85.35	80.08

Table 4.6: Results on applying two different pipelines and sent for adversarial purification after PGD-10 attack.

Results show that without the preprocessing pipeline, the robust accuracy after purification is better (by ~4 to 5%). However, we do not yet come to a conclusion that the pipeline is not useful in improving adversarial robustness, as our solution proposes that we feed the preprocessed, purified images to the classifier for further adversarial training by finetuning the model parameters of the adversarially trained classifier. From the results, we do see that combining adversarial purification and adversarial training (by simply replacing the classifier) has a positive effect on the robust accuracy (~4%).

#### 4.8 Additional Remarks

Previously, we have evaluated our proposed solution by incorporating our pipeline **after** the adversarial images have been created using PGD-10, using a test subset of 512 images. We have also evaluated using AutoAttack (Rand, EOT=5) with a test subset of 32 images by incorporating the pipeline **before** the adversarial attack is conducted. Since the pipeline was introduced even before the attack, it is considered only pure noise and is not a part of the defense strategy, but we observe that the noise indeed will disrupt the adversarial attack in some way. Hence, to some extent, it shows that adversarial robustness is actually the model ability to generalize to input, but not overfitting to local pixel features.

#### 4.9 Implementation Issues and Challenges

Training a deep neural network is a time and resource-intensive process, especially for large model sizes, large input number of inputs and batch size. While we have proposed our solution, we have not yet done full experiment on its effectiveness due to time and resource constraint. Incorporating diffusion model in our solution is the biggest challenge as its usage simply requires much better resource than what is available.

# Chapter 5

## Conclusion

In recent years, deep neural networks have excelled in machine learning tasks, demonstrating high accuracy in areas such as computer vision, speech recognition, and natural language processing. This performance has led to their adoption in critical real-world applications, including safety-oriented domains like healthcare and autonomous vehicle systems. However, the impressive accuracy of these models does not guarantee their reliability or robustness for direct deployment in daily life. Recent research has revealed that these high-performance models can be susceptible to adversarial examples—subtly perturbed inputs that are nearly indistinguishable from normal inputs to the human eye but can cause significant disruptions in machine learning model behaviour. Exploiting this vulnerability could have dire consequences, making it essential to develop defensive methods to enhance the robustness of machine learning models against adversarial attacks. The primary aim of this thesis is to research and improve upon current defense methods against adversarial attacks in the field of image recognition, thereby enhancing the adversarial robustness of machine learning models.

Our thesis proposes a novel defense method that combines a preprocessing pipeline, adversarial purification and adversarial training. In addition, we investigated various methods of incorporating noises the effect of doing so in an adversarial defense system. For future research, we recommend exploring the effect of noise incorporation in depth, finetuning the diffusion timestep and other hyperparameters in the diffusion model, and also investigate the effect of finetuning the adversarially pretrained classifier using the preprocessed, purified output in our proposed solution.

## REFERENCES

## REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. doi:10.1145/3065386
- [2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
- [3] S. Qiu, Q. Liu, S. Zhou, and W. Huang, "Adversarial attack and defense technologies in Natural Language Processing: A Survey," *Neurocomputing*, vol. 492, pp. 278–307, 2022. doi:10.1016/j.neucom.2022.04.020
- [4] P.-Y. Chen, "Securing AI systems with adversarial robustness," IBM Research Blog, <https://research.ibm.com/blog/securing-ai-workflows-with-adversarial-robustness> (accessed Sep. 11, 2023).
- [5] K. Eykholt *et al.*, "Robust physical-world attacks on Deep Learning Visual Classification," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. doi:10.1109/cvpr.2018.00175
- [6] Y. Li, M. Cheng, C.-J. Hsieh, and T. C. Lee, "A review of adversarial attack and defense for classification methods," *The American Statistician*, vol. 76, no. 4, pp. 329–345, 2022. doi:10.1080/00031305.2021.2006781
- [7] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models," *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017. doi:10.1145/3128572.3140448
- [8] N. Papernot *et al.*, "Practical black-box attacks against machine learning," *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017. doi:10.1145/3052973.3053009
- [9] W. Brendel, J. Rauber, and M. Bethge, "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models," *International Conference on Learning Representations*, Feb. 2018.
- [10] J. Wang *et al.*, "Adversarial attacks and defenses in deep learning for image recognition: A survey," *Neurocomputing*, vol. 514, pp. 162–181, 2022. doi:10.1016/j.neucom.2022.09.004
- [11] Dimitris Tsipras, Shibani Santurkar, L. Engstrom, A. D. Turner, and A. Madry, "Robustness May Be at Odds with Accuracy," May 2018.

## REFERENCES

- [12] J. M. Cohen, E. Rosenfeld, and J. Zico Kolter, “Certified Adversarial Robustness via Randomized Smoothing,” *arXiv (Cornell University)*, Feb. 2019.
- [13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.
- [14] A. Ross and F. Doshi-Velez, “Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.  
doi:10.1609/aaai.v32i1.11504
- [15] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi, “DeepCloak: Masking Deep Neural Network Models for Robustness Against Adversarial Samples,” *arXiv (Cornell University)*, Feb. 2017.
- [16] C. Xie, Y. Wang, Z. Zhang, Z. Ren, and A. L. Yuille, “Mitigating Adversarial Effects Through Randomization.,” *International Conference on Learning Representations*, Feb. 2018.
- [17] C. Guo, M. Rana, M. Cisse, and van, “Countering Adversarial Images using Input Transformations,” *arXiv (Cornell University)*, Oct. 2017.
- [18] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh, “Towards Robust Neural Networks via Random Self-ensemble,” Dec. 2017, doi: <https://doi.org/10.48550/arxiv.1712.00673>.
- [19] X. Liu, L. Yao, C. Wu, and C.-J. Hsieh, “Adv-BNN: Improved Adversarial Defense through Robust Bayesian Neural Network,” *arXiv (Cornell University)*, Oct. 2018, doi: <https://doi.org/10.48550/arxiv.1810.01279>.
- [20] P. Samangouei, M. Kabkab, and R. Chellappa, “Defense-gan: Protecting classifiers against adversarial attacks using generative models,” *arXiv.org*, <https://arxiv.org/abs/1805.06605>.
- [21] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, “Diffusion Models for Adversarial Purification,” *arXiv.org*, May 16, 2022.  
<https://arxiv.org/abs/2205.07460>.
- [22] A. Boopathy, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel, “CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3240–3247, Jul. 2019, doi: <https://doi.org/10.1609/aaai.v33i01.33013240>
- [23] “Denoising Diffusion-based Generative Modeling: Foundations and Applications,” *Denoising Diffusion-based Generative Modeling: Foundations and Applications*. <https://cvpr2022-tutorial-diffusion-models.github.io/>
- [24] S. K. Adaloglou Nikolas, “How diffusion models work: the math from scratch,” *AI Summer*, Sep. 29, 2022. <https://theaisummer.com/diffusion-models/>

## REFERENCES

- [25] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, “Diffusion Models for Adversarial Purification,” *arXiv.org*, May 16, 2022. <https://arxiv.org/abs/2205.07460>
- [26] F. Croce *et al.*, “RobustBench: a standardized adversarial robustness benchmark,” *arXiv.org*, Oct. 19, 2020. <https://arxiv.org/abs/2010.09670>
- [27] F. Croce and M. Hein, “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,” *arXiv.org*, Aug. 04, 2020. <https://arxiv.org/abs/2003.01690>.
- [28] Z. Wang, T. Pang, C. Du, M. Lin, W. Liu, and S. Yan, “Better Diffusion Models Further Improve Adversarial Training,” *arXiv.org*, Jun. 01, 2023. <https://arxiv.org/abs/2302.04638>.

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 1
Student Name & ID: NG SHI QI 2104550	
Supervisor: DR NG HUI FUANG	
Project Title: DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

**Reviewed deep learning related knowledge.**

## 2. WORK TO BE DONE

**Continue to review related knowledge, and review related papers.**

## 3. PROBLEMS ENCOUNTERED

N/A

## 4. SELF EVALUATION OF THE PROGRESS

**No issues in the progress.**



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 3</b>
<b>Student Name &amp; ID: NG SHI QI 2104550</b>	
<b>Supervisor: DR NG HUI FUANG</b>	
<b>Project Title: DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

**Reviewed deep learning related knowledge and related papers.**

## 2. WORK TO BE DONE

**Reimplement results by other authors.**

## 3. PROBLEMS ENCOUNTERED

**The work done by other authors may have outdated libraries involved, and debugging is required to run the code properly.**

## 4. SELF EVALUATION OF THE PROGRESS

**Progress may be delayed due to unexpected bugs.**



\_\_\_\_\_  
Supervisor's signature



\_\_\_\_\_  
Student's signature



# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 5</b>
<b>Student Name &amp; ID: NG SHI QI 2104550</b>	
<b>Supervisor: DR NG HUI FUANG</b>	
<b>Project Title: DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

**Tried to reimplement some research outcome by other authors, specifically, adversarial training.**

## 2. WORK TO BE DONE

**Review the code by other authors, try to understand what the code is doing.**

## 3. PROBLEMS ENCOUNTERED

**Lack in experiences in training a deep learning network caused a lack of understanding to interpret how well the network is training.**

## 4. SELF EVALUATION OF THE PROGRESS

**Progress may be delayed due to lack of understanding in how training a deep learning network works.**



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 7
Student Name & ID: NG SHI QI 2104550	
Supervisor: DR NG HUI FUANG	
Project Title: DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

**Reviewed the adversarial training code implemented by other authors, tweaked some learning hyperparameters and learnt some more advanced techniques in training a network.**

## 2. WORK TO BE DONE

**Redirect attention and try to implement other related works.**

## 3. PROBLEMS ENCOUNTERED

**The time required to reimplement an adversarial training network is too long, and due to lack of good computing resource especially GPU memory, we found that it is impossible to reproduce the results of the author.**

## 4. SELF EVALUATION OF THE PROGRESS

**A change of attention to other research problem should be done as there is not much to do in adversarial training part. Need to look at other research area for possible contribution.**



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 9</b>
<b>Student Name &amp; ID: NG SHI QI 2104550</b>	
<b>Supervisor: DR NG HUI FUANG</b>	
<b>Project Title: DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

**Not much progress due to midterms.**

## 2. WORK TO BE DONE

**Try to reimplement adversarial purification works.**

## 3. PROBLEMS ENCOUNTERED

**Schedule is tight.**

## 4. SELF EVALUATION OF THE PROGRESS

**Progress is delayed.**



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 11</b>
<b>Student Name &amp; ID: NG SHI QI 2104550</b>	
<b>Supervisor: DR NG HUI FUANG</b>	
<b>Project Title: DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

**Reviewed diffusion models related knowledge and some deep learning general knowledge to understand what the paper is saying.**

## 2. WORK TO BE DONE

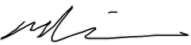
**Reimplement adversarial purification work and study how to implement it in my own research.**


## 3. PROBLEMS ENCOUNTERED

**Tight schedule and loss of access to the machine due to hardware issues.**

## 4. SELF EVALUATION OF THE PROGRESS

**Progress is delayed.**

  
\_\_\_\_\_  
Supervisor's signature

  
\_\_\_\_\_  
Student's signature

# POSTER



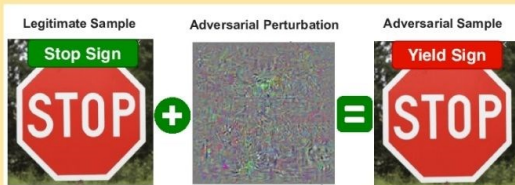
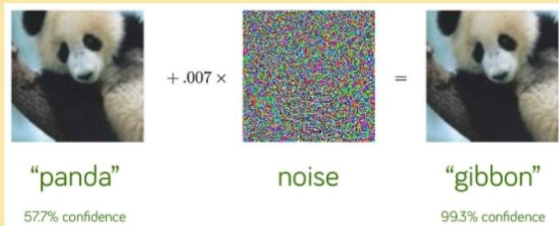
DONE BY NG SHI QI  
SUPERVISED BY DR NG HUI FUANG

FACULTY OF INFORMATION COMMUNICATION AND TECHNOLOGY (FICT)

## DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION

### WHAT IS ADVERSARIAL ATTACK?

The act of *inserting a small amount* of specially designed *perturbations* into normal inputs, and they can *cause a significant drop of performance* in the machine learning (ML) models.



### IMPLICATIONS

*Poses threats to the robustness* of ML models. Dire consequences can occur if ML models employed in real life settings (e.g. autonomous car) are being adversarially attacked.

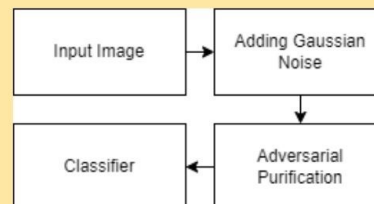
### OBJECTIVE

Propose adversarial defense methods to *improve ML robustness* that is *less computationally costly, easy to implement* and *more generalisable*.

### CONCLUSION

ML models, while having high accuracy, may exhibit poor robustness, making them vulnerable to adversarial attacks. Actions must be taken to enhance their robustness so that they can be applied in real life safely.

### PROPOSED METHOD



Three major components:

- *Gaussian noise addition* to incorporate randomness and 'disrupt' perturbation.
- *Adversarial purification* to remove adversarial perturbation.
- *Adversarial training* to make the classifier more robust to adversarially perturbed examples.

## PLAGIARISM CHECK RESULT

### FYP Check

#### ORIGINALITY REPORT

<b>14%</b>	<b>8%</b>	<b>11%</b>	<b>4%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
<b>PRIMARY SOURCES</b>			
<b>1</b>	Jia Wang, Chengyu Wang, Qiuzhen Lin, Chengwen Luo, Chao Wu, Jianqiang Li. "Adversarial attacks and defenses in deep learning for image recognition: A survey", <i>Neurocomputing</i> , 2022 Publication	<b>3%</b>	
<b>2</b>	<a href="http://www.arxiv-vanity.com">www.arxiv-vanity.com</a> Internet Source	<b>2%</b>	
<b>3</b>	Submitted to Universiti Tunku Abdul Rahman Student Paper	<b>1%</b>	
<b>4</b>	Yao Li, Minhao Cheng, Cho-Jui Hsieh, Thomas C. M. Lee. "A Review of Adversarial Attack and Defense for Classification Methods", <i>The American Statistician</i> , 2022 Publication	<b>1%</b>	
<b>5</b>	<a href="http://www.researchgate.net">www.researchgate.net</a> Internet Source	<b>&lt;1%</b>	
<b>6</b>	Submitted to University of Strathclyde Student Paper	<b>&lt;1%</b>	
<b>7</b>	Submitted to University of Sydney		

<b>Universiti Tunku Abdul Rahman</b>			
<b>Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)</b>			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1




**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

<b>Full Name(s) of Candidate(s)</b>	NG SHI QI
<b>ID Number(s)</b>	21ACB04550
<b>Programme / Course</b>	BACHELOR OF COMPUTER SCIENCE
<b>Title of Final Year Project</b>	DEFENCE AGAINST ADVERSARIAL ATTACK IN IMAGE RECOGNITION

<b>Similarity</b>	<b>Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)</b>
<b>Overall similarity index: <u>14</u> %</b>  <b>Similarity by source</b> Internet Sources: <u>8</u> % Publications: <u>11</u> % Student Papers: <u>4</u> %	
<b>Number of individual sources listed of more than 3% similarity: <u>0</u></b>	
<b>Parameters of originality required and limits approved by UTAR are as Follows:</b> (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

  
 \_\_\_\_\_  
 Signature of Supervisor

\_\_\_\_\_  
 Signature of Co-Supervisor

Name: Ng Hui Fuang

Name: \_\_\_\_\_

Date: 20/9/2024

Date: \_\_\_\_\_



**UNIVERSITI TUNKU ABDUL RAHMAN**

**FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY  
(KAMPAR CAMPUS)**

**CHECKLIST FOR FYP2 THESIS SUBMISSION**

Student Id	21ACB4550
Student Name	NG SHI QI
Supervisor Name	DR NG HUI FUANG

TICK (√)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

\*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

\_\_\_\_\_  
(Signature of Student)

Date: 20/09/2024