HANDLING ADVERSARIES IN IMAGE RECOGNITION DEEP NEURAL NETWORKS

Pivithuru Thejan Amarasinghe

(209307X)

Degree of Master of Science

Department of Computer Science and Engineering

Faculty of Engineering

University of Moratuwa
Sri Lanka

March 2022

HANDLING ADVERSARIES IN IMAGE RECOGNITION DEEP NEURAL NETWORKS

Pivithuru Thejan Amarasinghe

(209307X)

Thesis/Dissertation submitted in partial fulfilment of the requirements for the degree Master of Science in Data Science

Department of Computer Science and Engineering

Faculty of Engineering

University of Moratuwa
Sri Lanka

March 2022

Declaration

I declare that the thesis as my own work and it doesn't contain any sort of external material without an acknowledgement. Also, according to my knowledge I haven't used any previously published materials in order to prepare the thesis. All the work carried out to prepare the thesis are individual work and do not copy any existing work without an acknowledgement.

I like to grant permission to the University of Moratuwa to reuse or share my work in any sort of format with external parties. Also, I like to keep my right to use my work in future.

Signature: Date:

I supervised the research conducted by the above candidate.

Signature of the supervisor: Date:

Acknowledgments

I wish to sincerely thank my supervisor, Dr. Charith Chitraranjan for providing me with the necessary guidance throughout the project. He provided me with ideas from the beginning of the project and helped me with finding the research idea. Also, special gratitude should be given to our MSc coordinator for encouraging me to full fill my research under dedicated timelines. Also, I would like to extend my gratitude to all the academic staff of the University of Moratuwa for the great contribution they made for me during study. Also, I am grateful to my family members and all the friends who helped me throughout the project.

Abstract

Deep neural networks play a vital role in image recognition. There are so many mission-critical applications that use deep neural networks for image recognition. With the popularization of deep neural networks, attackers have identified their downsides of them when it comes to image recognition. Some ways can create images that can fool even deep neural networks. These images are commonly known as adversarial images. So attackers use these adversarial images to fool image recognition neural networks to develop a negative picture about using neural networks for image recognition. And even sometimes, attackers use these loopholes to conduct criminal activities as well. Keeping all these aspects in mind the idea of the research is to develop a viable solution that can tackle the main two attack techniques. The research will focus on developing adversarial images using main attacking techniques and developing a defense mechanism for those attacks. The defense technique used in the research is a combination of two techniques called adversarial training and defense distillation. As the outcome of the project accuracy of the proposed solution is measured against a typical deep neural network-based image recognition system using data samples containing adversarial images.

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

Deep neural networks can classify images with near-human-level performance. As a result, image classifying deep neural networks are used in critical applications like autonomous cars, face recognition systems, handwriting recognition systems, etc. But as we all know there should be a difference between human vision and computer vision. Recent studies show that adversarial images can cause an image classifying deep neural network to misbehave from their expected behavior. So, introducing defense systems for adversarial attacks is an essential requirement to ensure trust against deep neural networks that use for image classification.

Adversarial attacks conducted against image classifying neural networks can be divided into two major types,

- Type I Error Based Attacks
- Type II Error Based Attacks

1.1 Type I Error Based Attacks

Type I Error-based attack is where the original image is changed in a way that is obvious to the naked human eye but if we feed this image to an image classifying neural network it will still classify this image as belonging to the original class. This means the image classifying deep neural network cannot identify the change which is obvious to the naked human eye. For example, this type of attack can be conducted on a face recognition system where modified images can be used to fool a face recognition system and reduce the trust of the users toward the system.

1.2 Type II Error Based Attacks

Type II Error based attack is where the original image is changed in a way that is not significant to the naked human eye but if we feed this image to an image classifying neural network it will classify that image as some other thing. For example, this type of attack can be conducted on autonomous vehicles where road signs are changed in a way that can mislead an autonomous vehicle and lead the vehicle to an accident.

1.3 Defense Methods

Two main approaches against adversarial attacks are,

- Proactive defense
- Reactive defense

These two approaches have advantages as well as disadvantages, based on the problem we can use either one of the approaches or a combination of these two approaches.

1.3.1 Proactive Defense

Proactive defense makes image classifying neural networks robust to adversarial examples. Network distillation is a prominent technique used in proactive defense where the model is trained twice. First, the model is trained with ground truth labels and then trained from the predictions of the previous machine learning model to increase the robustness of the current machine learning model. Retraining the model with noise samples is another proactive defense technique against adversarial attacks where we can use existing algorithms to generate adversarial samples. Apart from these two main techniques, there are improvements that can be done to neural network architectures to increase robustness against adversarial attacks.

1.3.2 Reactive Defense

Reactive defense techniques can be applied to already existing image classifying neural networks. Adversarial detection is one of the techniques where we can use a simple neural network as a binary classifier to identify adversarial inputs before feeding them to an image classifying neural network. Input reconstruction is another technique where adversarial samples are reconstructed into clean inputs that cannot affect the prediction of image classifying neural networks. Finally, network verification is a promising reactive defense technique for unseen attacks where we check inputs that violate the characteristics of image classifying neural networks that we identified during our training phase.

1.4 Contribution of Research

This research tries to suggest a generic approach that can be used against type I and type II attacks. Existing solutions focus on one type of attack. Therefore, there is an opportunity to provide a solution that can defend against both types of attacks. The proposed defense technique would be a proactive defense mechanism against adversarial attacks.

2 RESEARCH PROBLEM

There are two main categories of adversarial attacks against image classifying neural networks which are type I error-based adversarial attacks and type II error-based adversarial attacks. Methods like fast gradient sign, evolution algorithm, and iterative methods are used to generate adversarial samples to conduct attacks. These attacks have given negative impacts on critical systems like autonomous cars, face recognition systems, etc. Therefore, there is a requirement for defense mechanisms that can countermeasure these attacks. Proactive defense mechanisms, as well as reactive defense mechanisms, are currently available against these attacks. But what is missing in existing solutions is a combined approach that can be used as a defensive mechanism for both adversarial attack types.

3 RESEARCH OBJECTIVES

- Conduct a literature review about existing solutions to mitigate type I and type II error-based attacks on image classifying neural networks.
- Design a defense mechanism that can mitigate both types of attacks.
- Based on the literature, develop adversarial samples that can attack image classifying neural networks.
- Test the proposed defense mechanism on developed adversarial samples.

4 LITERATURE REVIEW

Using deep neural networks to solve complex problems has become a widespread practice these days. Out of these, using deep neural networks for image classifying has obtained remarkable results. Solutions like AlexNet [1] built on top of the ImageNet dataset and LeNet model [2] built on top of the MNIST data set are prominent image classifying neural networks that have shown promising results. Though these brilliant solutions gave excellent results with time researchers have found loopholes in these solutions.

Deep Neural Networks are Easily Fooled - High Confidence Predictions for Unrecognizable Images [3] is a research conducted in 2015 by a group of researchers where they have shown loopholes in image classifying neural networks used in business-critical applications. To test whether Deep Neural Networks will provide false positive predictions for noise images, researchers have used a DNN trained to near reasonable performance. As data sources, they have used MNIST and ImageNet datasets for this research. For the ImageNet dataset, they have selected the well-known "AlexNet" architecture [1] and for the MNIST dataset, they have selected the Caffe-provided LeNet model [2]. To generate novel images to test DNNs researchers have used evolutionary algorithms with direct and indirect encoding and gradient ascent.

When looking into results obtained by the researchers within 50 iterations, each iteration of direct encoding continuously produced unidentifiable images for each digit type identified by MNIST DNNs with more than 99.99% confidence. Even after 20,000 iterations, according to the researchers the direct encoding failed to create high-confidence images for many categories of ImageNet. However, according to the researchers, direct encoding did manage to produce images for 45 classes that are classified with more than 99% confidence to be natural images. Though there are more strokes and other regularities, indirect encoding still led to MNIST DNNs labeling unidentifiable images as digits with 99.99% confidence within a few generations. In five independent iterations, indirect encoding produced many images of ImageNet with DNN confidence scores ≥ of 99.99%, but that is unrecognizable.

To eliminate this behavior from DNNs, researchers have tried to train networks to identify fooling images. They have trained DNN 1 on a dataset, then evolved images that produce a high confidence score for DNN 1 for the n classes in the dataset, then they have taken those images and added them to the dataset in a new class n + 1. Then they have trained DNN 2 on this enlarged "+1" dataset. When considering the results of training networks to recognize fooling images, the evolution had still produced many noise images of MNIST for DNN 2 with confidence scores of 99.99%. The median confidence score had significantly decreased from 88.1% for DNN 1 to 11.7%

for DNN 2 of ImageNet. According to the researchers ImageNet DNNs were better equipped against adversarial images than MNIST DNNs.

The Limitations of Deep Learning in Adversarial Settings [4] is a research where researchers have conducted type II error-based attacks on image classifying neural networks using the MNIST dataset. Compared to other approaches researchers' have computed a direct mapping from the input image to the output image to develop an explicit adversarial image. Their approach changes a portion of input features leading to reduced perturbation of the source inputs. More formally what they try to achieve is to solve the following optimization problem where X means feature vector and Y* means desired output.

$$arg_{min}\|\delta x\| \; s.t. F(X + \delta x) = Y^*$$

Here X^* is the adversarial sample and Y^* is the desired adversarial output. Solving the above equation is not trivial because properties of image classifying neural networks make that equation non-linear and non-convex. Therefore, researchers have crafted adversarial samples by creating a mapping from input perturbations to output variations. This approach is different from other approaches because it used output variations to find corresponding input perturbations. According to the researchers how changes made to inputs affect a DNN's output stems from the evaluation of the forward derivative. This forward derivative is used to construct adversarial saliency maps indicating input features to include in perturbation δx to produce adversarial samples inducing a certain behavior from the image classifying neural network. Researchers further mention that forward derivative approaches are much more powerful than gradient descent techniques. Forward derivative approaches can be used for both supervised and unsupervised architectures and allow adversaries to generate information for a wider range of adversarial samples. Researchers have validated their adversarial sample generation technique against LeNet architecture [5] with the MNIST dataset. Researchers have shown that any input can be perturbed to be misclassified as any target class with a 97.10% success rate while perturbing on average 4.02% of the input features per sample.

In the research Distillation as a Defence to Adversarial Perturbations against Deep Neural Networks [6] researchers have introduced defensive distillation to reduce the effectiveness of adversarial samples on image classifying neural networks. They have analytically investigated the generalizability and robustness properties granted using defensive distillation when training image classifying neural networks. Based on results their defensive distillation can reduce the successfulness of adversarial image creation from 95% to less than 0.5% on an image classifying neural network that they used during the study. Distillation, the key concept that is used in this research was

formally introduced in [7]. Initially, distillation was introduced to train a DNN using knowledge transferred from a different DNN thinking about reducing computational costs. But researchers have used distillation in this research as a knowledge transfer mechanism to increase the resilience of image classifying neural networks against adversarial samples.

Explaining and Harnessing Adversarial Examples [8] is a research conducted by a group of researchers in Google where they claim adversarial attacks against image classifying neural networks possible due to the linear nature of the image classifying neural networks. They provide adversarial training as a solution to overcome adversarial attacks against image classifying neural networks. Their claim of the linear behavior of image classifying neural networks in high dimensional spaces is sufficient to generate adversarial images gives a fast method to generate adversarial samples for adversarial training. Using the MNIST dataset they have proved generic regularization strategies like dropout, pretraining, and model averaging do not provide significant resistance against adversarial attacks on image classifying neural networks.

According to the researchers precision of an individual input feature is limited. Since the precision of the features is limited classifiers will not respond differently to an input $x' = x + \eta$ if η is less than the precision of the feature. Researchers expect the machine learning model to assign to the same class to x and x' as long as $\|\eta\| < \epsilon$. When considering the dot product between the weight vector and the adversarial sample

$$W^T x' = W^T x + W^T \eta$$

The adversarial change causes the activation to increase by $W^T \eta$. Researchers have tried to increase this adversarial perturbation subject to the max norm constraint on η by assigning $\eta = sign(w)$. According to the researchers, a simple linear machine learning model can have adversarial images if its input has sufficient dimensions. But previous justifications from other research invoked hypothesized properties of neural networks, such as their highly non-linear nature. But this research's hypothesis based on linearity is simpler and can also explain why SoftMax regression is vulnerable to adversarial images.

When considering how adversarial attacks can be conducted in the physical world, Adversarial Examples in The Physical World [9] research has moved a step forward compared to other researches where those researches have tried to directly feed adversarial to the image classifying neural network. This research has focused on how cameras and other sensors can generate adversarial samples that can fool an image classifying neural network. Researchers explain this behavior by providing adversarial

images obtained from a mobile-phone camera to an ImageNet dataset machine learning model and measuring the prediction accuracy.

Famous research DeepFool: a simple and accurate method to fool deep neural networks [10] introduced DeepFool algorithm that can be used to easily generate perturbations that can be used to fool an image classifying neural network. Results indicate that the DeepFool algorithm outperforms other adversarial generating algorithms because it tries to define a small perturbation that can change the prediction of a given image classifying neural network. Also, as a part of the research researchers have proposed a simple and accurate method to compute the robustness of image recognition neural networks.

4.1 Adversarial Sample Generation

These adversarial sample generations are commonly mentioned in literature as adversarial attacks. Though these attacks can be divided into Type I error and Type II error-based attacks, there are several subcategories of adversarial attacks and some of them are,

- Poisoning Attacks

Poisoning attack algorithms allow attackers to insert few fake samples into the training dataset of the image classifying neural network. Due to these fake samples' failures happen to image classifying neural networks in the form of low accuracy and misclassifications in the training set. Poisoning attacks are commonly used in scenarios where attackers can access training datasets.

- Evasion Attacks

In evasion attacks, attackers do not have access to the training data set or to the parameter set of the image classifying neural network. Attackers create adversarial samples that cannot be identified by image classifying neural networks. A sample scenario would be pasting small pieces of tape on the stop road signs to misguide an autonomous vehicle.

- Targeted Attacks

In targeted attacks attacker tries to alter the input feature set of a sample to induce a specific prediction from the image classifying neural network. A sample scenario would be an attacker trying to show himself as a highly credible client to a credit evaluation model of a financial organization.

- Non-Targeted Attacks

Non-targeted attacks only try to generate incorrect predictions from the image classifying neural network. They do not try to generate a specific prediction for a given sample compared to a targeted attack.

- White-Box Attacks

During a white-box attack, the attacker is aware of the image classifying neural network including architecture, parameters, gradients, data, etc. Therefore, the attacker can carefully craft adversarial samples to miss classify the image classifying neural network. But in real-world scenarios conducting this type of attack is highly impractical.

- Black-Box Attacks

During a black-box attack, the attacker does not have any internal information related to the image classifying neural network. Attacker feeds inputs to the image classifying neural network and based on predictions given attacker tries to identify weaknesses in the image classifying neural network.

- Semi-white (Gray) Box Attacks

Semi-white box attacks are a more practical way of generating adversarial samples. In this approach, the attacker develops a generative model for producing adversarial images in a white-box environment. Then the attacker can use this new model to create the adversarial sample in a black-box environment.

4.1.1 White-box Attacks

During a white-box attack when the classifier is C and the image that we need to generate an adversarial image is (x, y), the attacker's objective is to generate a sample x' that is like the original image x but that can misguide the machine learning classifier C. This can be formally defined as,

find x' fulfilling $\|x' - x\| \leq \in$

In a way $C(x') = t \neq y$

- Deep Fool [10]

In this approach, the attack algorithm tries to find a path so the data point can go pass the decision boundary of the image classifying neural network as shown in figure 1. Hyper-plane F3 can be defined as $F3 = \{z: F(x)_4 - F(x)_3 = 0\}$. In every attacking step Deep Fool algorithm linearizes the decision hyperplane using Taylor expansion

$$F3' = \{x: f(x) \approx f(x_0) + (\nabla_x f(x_0) \times (x - x_0)) = 0\}$$

Here $f(x) = F(x)_4 - F(x)_3$

Using F3' Deep Fool algorithm calculates an orthogonal vector w that can be used to permute x_0 to be classified into class 3 by the image classifying neural network.

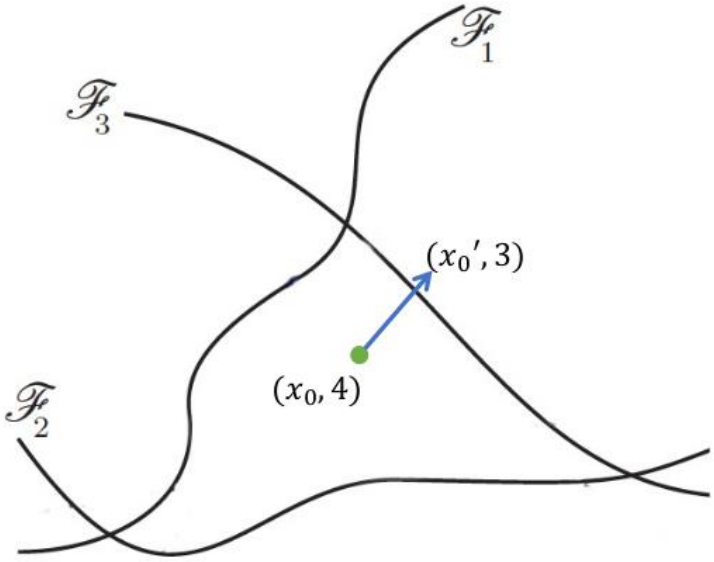


Figure 1: Hyper-planes

Hyper-plane F1 separates the data points in class 1 and class 4. Similarly, hyper-plane F2 separates class 2 data points from class 4 and hyper-plane F3 separates class 3 data points from class 4. The sample x_0 crosses the F3 hyperplane therefore it's classified as class 3 by the image classifying neural network. Image is taken from [11]

- Fast Gradient Sign Method [12]

This is a one-step algorithm to generate adversarial samples. This algorithm is faster compared to iterative adversarial sample generating algorithms because the algorithm involves only one backpropagation step. Therefore, this algorithm is used in scenarios where there is a demand to create many adversarial samples.

$x' = x+ \in sign(\nabla_x L(\theta, x, y))$ for a non-targeted attack

$x' = x- \in sign(\nabla_x L(\theta, x, y))$for a targeted attack

When further investigating these equations, for a targeted attack above equation is treated as a one-step gradient decent problem that needs to be solved.

$$minimize\ L(\theta, x', t)$$

subject to $\|x' - x\|_\infty \leq \in$ and $x' \in [0,1]^m$

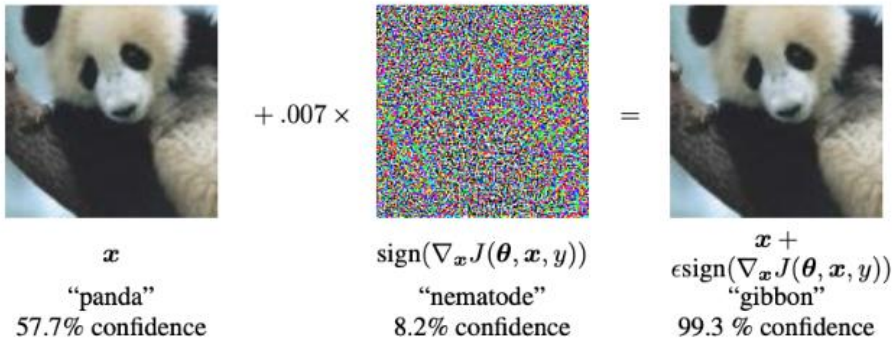


Figure 2: Adversarial image from the fast gradient
By adding a perturbation that cannot be identified by a naked human eye panda class is miss-classified to gibbon class. Image is taken from [12].

- Biggio's Attack [13]

This algorithm optimizes the discriminant function to mislead the image classifying neural network.

$$g(x) = < w, x > + b$$

If $g(x)$ is positive x is classified as belonging to class 'c' or else x is classified as not belonging to class 'c'. The algorithm creates a new sample x' to minimize the discriminant value $g(x')$ while managing $|x - x'|$ small. If $g(x')$ is negative x' is no longer belonging to class 'c'. But there is no significant difference between x and x' to the naked eye.

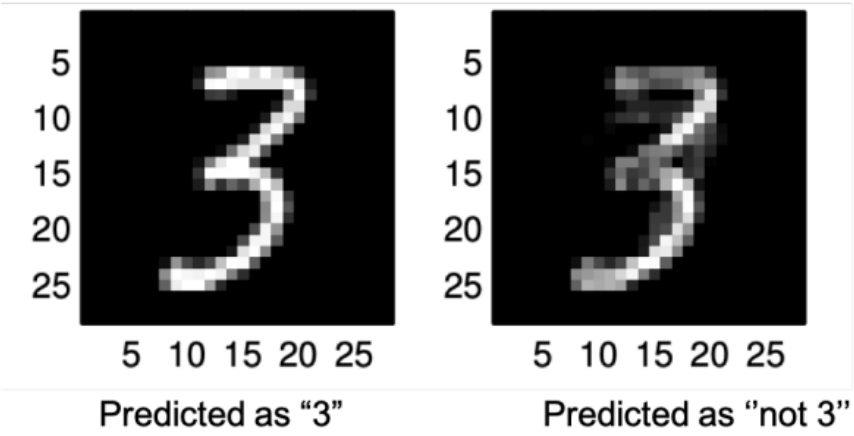


Figure 3: Adversarial image from Biggio’s attack
Biggio’s attack algorithm creates a permuted image that causes the image classifying neural network to miss-classify. Image is taken from [13]

- Szegedy’s L-BFGS attack [14]

This is the very first algorithm that introduced adversarial samples that can miss-classify an image classifying neural network. The algorithm tries to identify minimum distorted sample x’ with object to,

minimize $|x - x'|$

subject to $C(x') = t$

This problem can be solved by introducing a loss function.

minimize $c|x - x'| + L(\theta, x', t)$

In this loss function, the first term depicts the closeness between x and x’. The second term allows the algorithm to find a x’ that has a minor loss to t such that an image classifying neural network will predict x’ belongs to t. By changing c the algorithm allows finding a x’ that is quite like x that can fool image classifying neural networks.

- Jacobian-based Saliency Map Attack [15]

This is a greedy attack algorithm that continuously manipulates the most influential pixel to the output of an image classifying neural network. The algorithm relies on obtaining the Jacobian matrix of the score function F.

$$J_{F(x)} = {\partial F(x)}/{\partial x} = \left\{ {\partial F_j(x)}/{\partial x_i} \right\}_{i \times j}$$

The above function will model F(x)'s change for a change in input x. In a targeted attack scenario x' is crafted such that it is classified by the model as t which is the targeted label. The algorithm repeatedly changes pixel x_i, which will affect $F_t(x)$. Due to this $\Sigma_{i \neq j} F_j(x)$ will change which causes the model to give the largest score to label t.

- Carlini & Wagner's Attack [16]

This algorithm also tries to identify minimum distorted perturbation. The algorithm is quite similar to Szegedy's L-BFGS attack [14]. But this algorithm uses margin loss $f(x, t)$ rather cross-entropy loss $L(x, t)$. The advantage of using margin loss is that when $C(x') = t$, the margin loss value $f(x', t) = 0$, the algorithm will directly minimize the distance from x' to x.

- Spatially Transformed Attacks [17]

Though other algorithms try to individually permute pixels of an image, this algorithm does small spatial transformations to the image. Here we translate, rotate, and distort local image features marginally. The change is minor enough that cannot be distinguished by the naked human eye, but that change is powerful enough to fool an image classifying neural network.

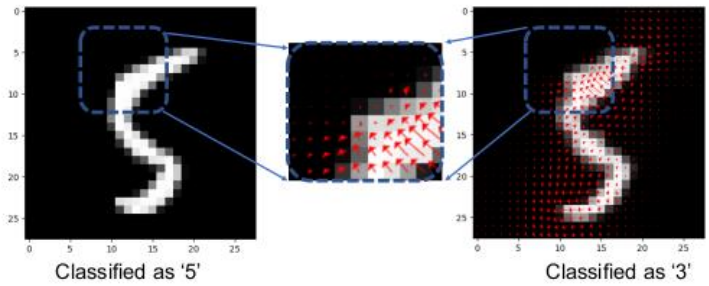


Figure 4: Adversarial image from the spatial transformation
The top part of the image is slightly permuted to be thicker. Therefore, the new image is classified as 3 instead of 5 by the image classifying neural network. Image is taken from [17].

- Unrestricted Adversarial Samples

While other approaches consider adding minor perturbations into images, this method creates unrestricted adversarial images. These images do not look like the original image, but these are still justifiable to the naked human eye. Most of the existing defense mechanisms fail to identify these as adversarial images because those defense systems only consider minimal perturbations to the original image.

- Universal Attacks

Rather than considering creating an adversarial sample for a specific sample, there are algorithms to mislead image classifying neural network's decision on all testing samples. Here algorithm tries to find a perturbation δ such that an image classifying neural network misleads all the samples. This reveals the inherent weakness of image classifying neural networks to all the samples.

- L_p Attacks

Most studies use l_2 or l_∞ normalizations to generate adversarial samples. But there are other types of l_p normalizations that can be used to generate adversarial samples. One pixel attack is one such attack where it constrains l_0 norm of the perturbation $|x_0 - x|$ to reduce the number of pixels that allow changing. This attack is a clear example of the poor robustness of image classifying neural networks. The elastic-net attack is also another such attack where it constraints the perturbations l_1 and l_2 norm together. According to the researcher's elastic-net attack can mislead an image classifying neural networks which have a strong defense against l_2 or l_∞ attacks.

- Ground Truth Attack

Theoretically, this can be described as a way to find slightly changed adversarial images. This is the very first work to measure the exact robustness of image classifying neural networks. But this approach is slow and not scalable for large datasets. This method encodes the model parameters F and data (x, y) as the subjects of a linear programming system, and then solves to check whether there is an eligible image x_0 in x's neighborhood that can fool the model.

4.1.2 Black-box Attacks

- Substitute Model

Research [18] was the first to introduce a successful method that can generate adversarial samples that can fool an image classifying neural network in a black-box setting. This algorithm exploits the transferability property of adversarial samples. This means if x' can attack image classifying neural network F, then x' can attack F' as well that has a matching structure to F. Main steps of this algorithm are as follows,

  - Make a substitute training set
  - Feed the generated training dataset X into the target classifier F to label Y. Now train a model F' using (X, Y).
  - Augment (X, Y) and retrain model F' to increase its accuracy.
  - Use white-box attack methods and create adversarial samples for F'. These samples will be adversarial to F as well due to transferability property.

- Zeroth Order Optimization Based Black-Box Attack

This method differs from the previously mentioned substitute method because this approach assumes that the attacker has access to the output score from the target model's output [19]. In this approach, there is no requirement to create a substitute training set or a substitute machine learning model. This approach tries to scrape gradient information for target image x by looking into F(x).[19] provides a specific algorithm for it.

Above mentioned black-box attacks need lots of interactions with the machine learning model. This might not be the case in practical scenarios. There are efficient ways of creating adversarial samples for black-box attacks with limited input queries. Estimating gradient information from outputs is an efficient way to create adversarial samples.

4.1.3 Grey-box Attacks

A grey-box attack framework was introduced by [20]. As the first step, the algorithm trains a Generative Adversarial Network targeting the model of interest. Then adversarial samples can be crafted using the generative model. As an advantage generative model-based adversarial sample generation gives quicker and natural-looking adversarial samples.

4.1.4 Poisoning Attacks

Poisoning attacks are applied on graph neural networks due to their transductive learning behavior. A poisoning attack uses the knowledge about the image classifying neural networks to generate adversarial samples. These adversarial samples are used in the training dataset to conduct the poisoning attack.

- Koh's Model [21]

This model tries to interpret image classifying neural networks on how the model's prediction would affect an input change. This model can easily quantify the change in the final loss without retraining the machine learning model when a single training image is modified. Using this attacker can identify the most effective adversarial samples to conduct the poisoning attack.

- Poison Frogs [22]

This method allows to insertion of an adversarial image with the true label to the training set, to cause the trained machine learning model to wrongly classify a target test image. This algorithm tries to solve x'.

$$x' = arg_x min \|Z(x) - Z(x_t)\| + \beta \|x - x_b\|$$

Here x_t and y_t are true sample points. Also, x_b and y_b are base points. After inserting x' into the training dataset model will predict x' belongs to y_b. If we use this model to predict x_t model will predict it to the same class that belongs to x' because of the minor distance between x' and x_t.

4.1.5 Physical Attacks

Compared to previously mentioned digital attacks, there are physical attacks as well. In scenarios where systems use cameras to input images, there is a chance of physically creating adversarial samples and inputting them. Examples like stickers attached to road signs can mislead an autonomous vehicle and cause severe damage. These kinds of physical attacks can affect practical applications like face recognition, autonomous vehicles, etc.

Research [9] tries to generate physical adversarial samples. They have tried whether FGSM and BIM can be used to create physical adversarial images by checking their robustness under natural transformation. Here robust means the created samples remain adversarial after the transformation process. They have printed adversarial images created from FGSM and BIM and used cell phones to take pictures of printed

adversarial samples. Based on the lighting and shooting angle these pictures can or cannot be to an image classifying neural network. Based on the research large portion of the pictures were still good enough to fool an image classifying neural network. Especially adversarial images created using FGSM have shown more success rate of being adversarial to the classifier after printing them and taking photos.

Research [23] creates physical adversarial samples by changing road signs that can mislead a road sign recognition system. The researchers have put stickers on a "stop" sign and tested their approach's effectiveness. As the first step, they have conducted an l1-norm based attack on digital images to identify the areas for perturbation. Then they have conducted an l2 norm-based attack on areas that were found during the previous step to identify colors. Finally, they have printed stickers based on the colors they have found and put those in the identified spots.

Figure 5: Physical attacks
The attacker puts some stickers on a road sign to mislead an autonomous vehicle. Image is taken from [23]

Also, 3D printed objects can be used as physical adversarial attacks to mislead image classifying neural networks. In these kinds of attacks, special attention needs to be put to the texture of the object as well because it has a significant impact on the adversarial effectiveness of the printed object.

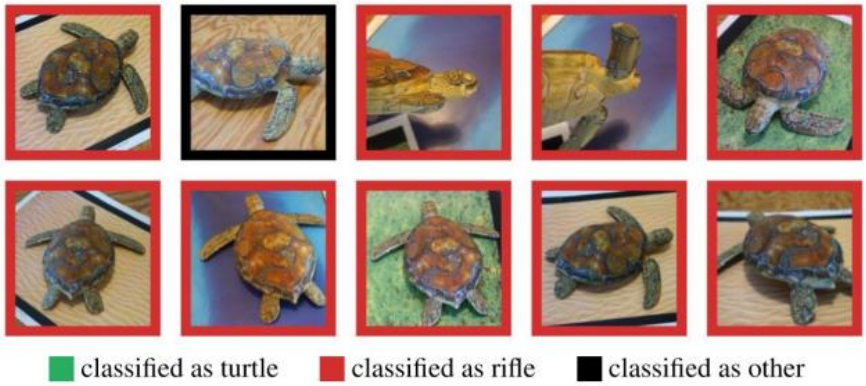


Figure 6: 3D printed physical attacks
Image classifying neural network correctly classifies the original image but fails to classify the adversarial sample. Image is taken from [24].

4.2 Countermeasures

There are reactive defense techniques as well as proactive defense techniques. These two main techniques can be further categorized into subcategories. Based on literature three main subcategories can be used as reactive or proactive defense techniques.

- Gradient Masking
- Robust Optimization
- Adversarial Examples Detection

4.2.1 Gradient Masking

Since most of the attack algorithms consider gradient information to generate adversarial samples, this approach will hide gradient information from attack algorithms and confuse them.

- Defensive Distillation

As mentioned, previously distillation was first introduced in [7] to reduce the size of deep neural networks. But [6] formally introduced defense distillation as a technique to countermeasure adversarial samples. It has defined a four-step process as the solution.

- Develop a model F using the given training set (X, Y) by setting the temperature of the SoftMax function to T.
- Calculate the scores given by $F(X)$.

- Develop another model F' using SoftMax function at temperature T using a dataset with soft labels $(X, F(X))$.
- Use F' as the model to do predictions on test data.

This method is highly effective because this technique makes the SoftMax value of the target class close to 1 and the SoftMax value of other classes close to 0.

- Shattered Gradients [25]

This technique tries to protect the model from adversarial samples by processing input data. This technique adds a non-smooth non-differentiable pre-processor g (). Then the model will be trained on g(x). Since the model f(g(x)) cannot be differentiable from x, the new model has protection against adversarial samples. This technique will not allow the attack algorithm to find $\partial F(x)/\partial x$.

- Stochastic Gradients

This technique tries to randomize image classifying neural networks to confuse adversarial samples. This technique trains several models on the training dataset and randomly picks a model to do a prediction on a test sample. Due to this attack algorithm has no idea about the predicting model. Therefore, the success rates of adversarial samples get low.

- Exploding & Vanishing Gradients

This technique adds a generative model before the image classifying neural network. Therefore, the final model is an extremely deep neural network. Due to the deepness of this model partial derivatives of each layer causes ∂L(x) / ∂x to be extremely low. This causes the attack algorithm to accurately create adversarial samples. Based on literature gradient masking cannot eliminate adversarial samples 100%. But it's a successful method against adversarial samples.

4.2.2 Robust Optimization

By changing the manner of training robust optimization method tries to improve the robustness of image classifying neural networks against adversarial samples. Based on literature robust optimization techniques are mainly based on two major areas.

Learn model parameters θ* to minimize average adversarial loss.

$$\Theta^* = arg\ min_{\theta \epsilon \Theta} E_{x \sim D} max_{\|x' - x\| \leq \epsilon} L(\theta, x', y)$$

Learn model parameters θ* to maximize the average minimal perturbation distance.

$$\Theta^* = arg\ max_{\theta \epsilon \Theta} E_{x \sim D} \min C(x') \neq y \ \|x' - x\|$$

Robust optimization methods get the prior knowledge about possible attack types that can be applicable for an image classifying neural network. Based on prior knowledge robust optimization techniques create image classifying neural networks more robust against the specific attack.

- Regularization Methods

In the early stages regularization was an effective way to countermeasure adversarial samples. Training image classifying neural networks with regularization for some scenarios heuristically helped to withstand adversarial samples.
Researchers [26] suggest adding regularizations to parameters during model training will create more stable image classifying neural networks against adversarial samples. Research suggests constraining the Lipschitz constant L_k between, so that outcome of each layer will not have a huge impact from small input changes.

$$\forall x, \delta, \|h_k(x; W_k) - h_k(x + \delta; W_k)\| \leq L_k \|\delta\|$$

Research [27] introduces a deep contractive network mechanism to regularize training of image classifying neural networks. This technique proposes adding a penalty on partial derivatives at each layer into backward propagation therefore a minor change in the input image will not cause a substantial change in the output of each layer.

- Adversarial Retraining

Research [8] first suggested adding adversarial samples with true labels while training image classifying neural networks. By doing this image classifying neural networks will know the correct label y' of adversarial sample x'. Therefore, image classifying neural networks will classify x' properly when the attacker feeds it. To generate adversarial samples for training research [8] has used a non-targeted FGSM.

$$x' = x + \in sign\big(\nabla_x L(\theta, x, y)\big)$$

This approach of training has improved the robustness of image classifying neural networks. Research [9] has provided a more improved version of research [8] where they have introduced scalability to the approach. They have suggested batch normalization to improve adversarial training for large datasets. This approach of adversarial training has a good resistance against FGSM based adversarial attacks. But

for iterative attacks, this approach is still vulnerable. To overcome this weakness research [28] suggests using projected gradient descent based adversarial samples for adversarial neural network training instead of step-by-step approaches like FGSM. This approach will try to identify the most misleading sample x_{adv} and train image classifying neural networks using it.

$$x_{adv} = arg\ max_{x' \in B(x)} L(x', F)$$

This approach provides good resistance against small datasets. But against large datasets, this approach faces performance issues though it provides resistance against adversarial samples.

Research work mentioned in [29] introduces an ensemble approach against adversarial attacks. This approach can protect an image classifying neural network from one-step attacks. Here classifier's training dataset is augmented from adversarial samples generated by other pre-trained classifiers. If we need to improve the robustness of the classifier F, then we need pre-trained classifiers F1, F2, and F3. For sample x we need to create adversarial samples from F1, F2, and F3 using a single-step attack approach. Due to the transferability property these adversarial samples can mislead classifier F. Therefore, training F with these adversarial samples will give more resistance against single-step attacks. This approach of ensemble technique is more efficient compared to FGSM or PGD based one-step attacks. Also, this technique provides good resistance against black-box attacks.

Research [30] introduces an accelerated adversarial training technique by repurposing backward pass calculations. The algorithm derives

The gradient of the loss to input image:

$$\partial L(x + \delta\theta) / \partial x$$

The gradient of the loss to machine learning model parameters:

$$\partial L(x + \delta\theta) / \partial \theta$$

in one back propagation iteration. Therefore, this approach provides a highly accelerated adversarial technique.

The reluplex algorithm introduced by research [31] verifies the robustness of image classifying neural networks. If the model F is given the Reluplex algorithm will

identify the value of minimal perturbation distance $r(x; F)$. Minimal perturbation distance will say that the image classifying neural network is safe against any attack that has a perturbation less than $r(x; F)$. After applying the Reluplex algorithm to the whole dataset we can identify the percentage samples that are safe against attacks less than $r(x; F)$.

Moreover, there are certificates $C(x, F)$ that can verify the robustness of image classifying neural networks. These certificates are trainable. So, training image classifying neural networks to verify these certificates will provide more robustness. There are a few methods to design these kinds of certificates. Some of them are,

- The lower bound of minimal perturbation

Research [32] introduces a lower boundary $C(x, F)$ for the minimal perturbation distance of F on x based on the Cross-Lipschitz Theorem. This $C(x, F)$ mainly relies on F and x. So, it is easy to calculate $C(x, F)$ for a hidden layer.

- Upper bound of adversarial loss

Research [33] introduces an upper boundary $U(x, F)$ that is greater than adversarial loss $L_{adv}(x)$.

$$L_{adv}(x) = max_{x'}\{max_{i \neq y} Z_i(x') - Z_y(x')\}$$

$U(x, F)$ acts if $U(x, F) < 0$, then adversarial loss $L(x, F) < 0$. Therefore, the machine learning model always gives the highest score for the original label.

4.2.3 Adversarial Examples Detection

Adversarial sample detection is another method to guard image classifying neural networks against adversarial attacks. These try to identify input is adversarial or not, instead of directly feeding inputs into image classifying neural network. Research [34] introduces three categories of threats that adversarial detection techniques need to handle.

- Zero-knowledge adversary – where the attacker can only get to image classifying neural network's parameters, not to the detection model.
- Perfect knowledge adversary – where attacker knows about the model and detection scheme with machine learning model's parameters.
- Limited knowledge adversary - where the attacker knows about the model and detection scheme. Although no idea about the detection scheme's parameters.

Research [35] introduces auxiliary machine learning models that aim to identify adversarial examples. Here image classifying neural network is trained with K + 1 labels with an extra label for adversarial samples. Research [36] introduces heuristically identifying adversarial samples from others. Adversarial images are found to put a reasonable weight on the later principal components where the natural images have a reasonable weight on early principal components. Therefore, researchers identify adversarial images based on PCA. There are researches focused on checking the accuracy of the image's prediction. This assumes that natural images have consistent predictions compared to adversarial samples. Research [37] changes the input sample itself. For every input image, research minimizes the color depth of the image. This color change will not change the prediction for a natural image. But for an adversarial image color change will cause a prediction change.

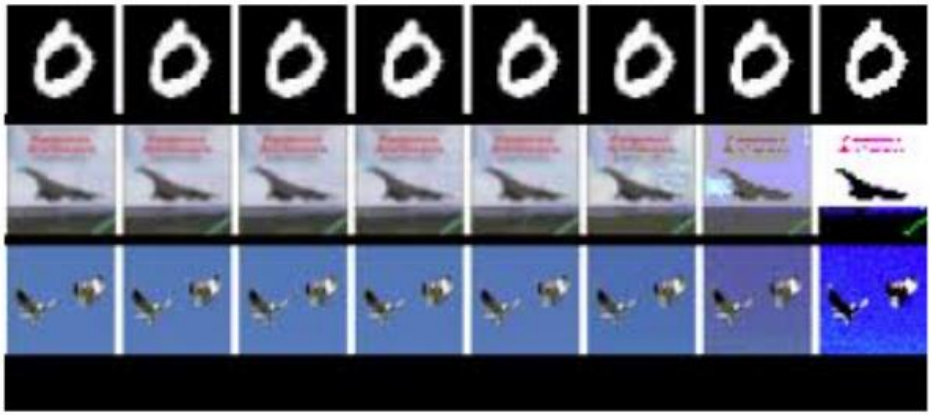


Figure 7: Color depth variation

From left to right, the color depth is minimized from 8-bit to 1-bit. Image is taken from [37]

Image classifying neural networks vulnerable to adversarial attacks because they do not work properly in the low probability space of data. This generalization issue comes due to the complex nature of image classifying neural networks. However linear machine learning models are also susceptible to adversarial attacks. In most cases, adversarial samples are close to the decision boundary of the trained model. Many studies suggest it's impossible to create an optimally robust classifier. Researchers' main reason for it is that distribution of each class is not well concentrated therefore there are enough opportunities to create adversarial samples.

Transferability is a key concept that allows the generation of adversarial samples. Transferability means adversarial samples generated to target an image classifying neural network can be used against a similar kind of image classifying neural network.

This transferability is more applicable to single-step attacks (FGSM) than iterative methods (BIM). Nevertheless, the transferability property can be effectively used in the black box environment to generate adversarial images.

When considering the impact of adversarial attacks on computer vision, there are impacts toward face recognition, object detection, and semantic segmentation. Research [38] attacks face recognition systems at digital and physical levels. Digital level attacks were based on the L-BFGS method. Physical level attacks were conducted asking participants to wear 3D printed glass frames while researchers optimize the color of glass frames at the digital level.

Figure 8: Physical attack from 3D printed glass frames
The subject on left wears a pair of glasses and the subject is identified as movie star Milla Jovovich by the face recognition system. Image is taken from [38].

The goal of object detection and semantic segmentation is to learn a model that can associate a series of labels with a given image. Research [39] generates a perturbation that can work as an adversarial and cause the classifier to give the wrong prediction on all the output labels for a given input image.

Image classifying neural networks have obtained fabulous success in various domains of machine learning. But existing adversarial generation techniques have created critical concerns about image classifying neural networks. To overcome there are so many researches that suggest approaches to create protection methods against adversarial samples.

5 METHODOLOGY

The research is trying to develop a solution that can mitigate type I and type II-based attacks. As the first step, the research needs to develop adversarial samples for type I and type II. Due to computation limitations, the research has to go with MNIST dataset than the ImageNet dataset. To create adversarial samples, the research uses existing methods in the literature. As defense mechanisms for adversarial images, the research tries a combined approach of distillation and adversarial retraining. These two techniques will be tried against both types of adversarial samples. As result evaluation criteria the research is going to split the MNIST dataset into a training set and test set and check the success of the novel approach.

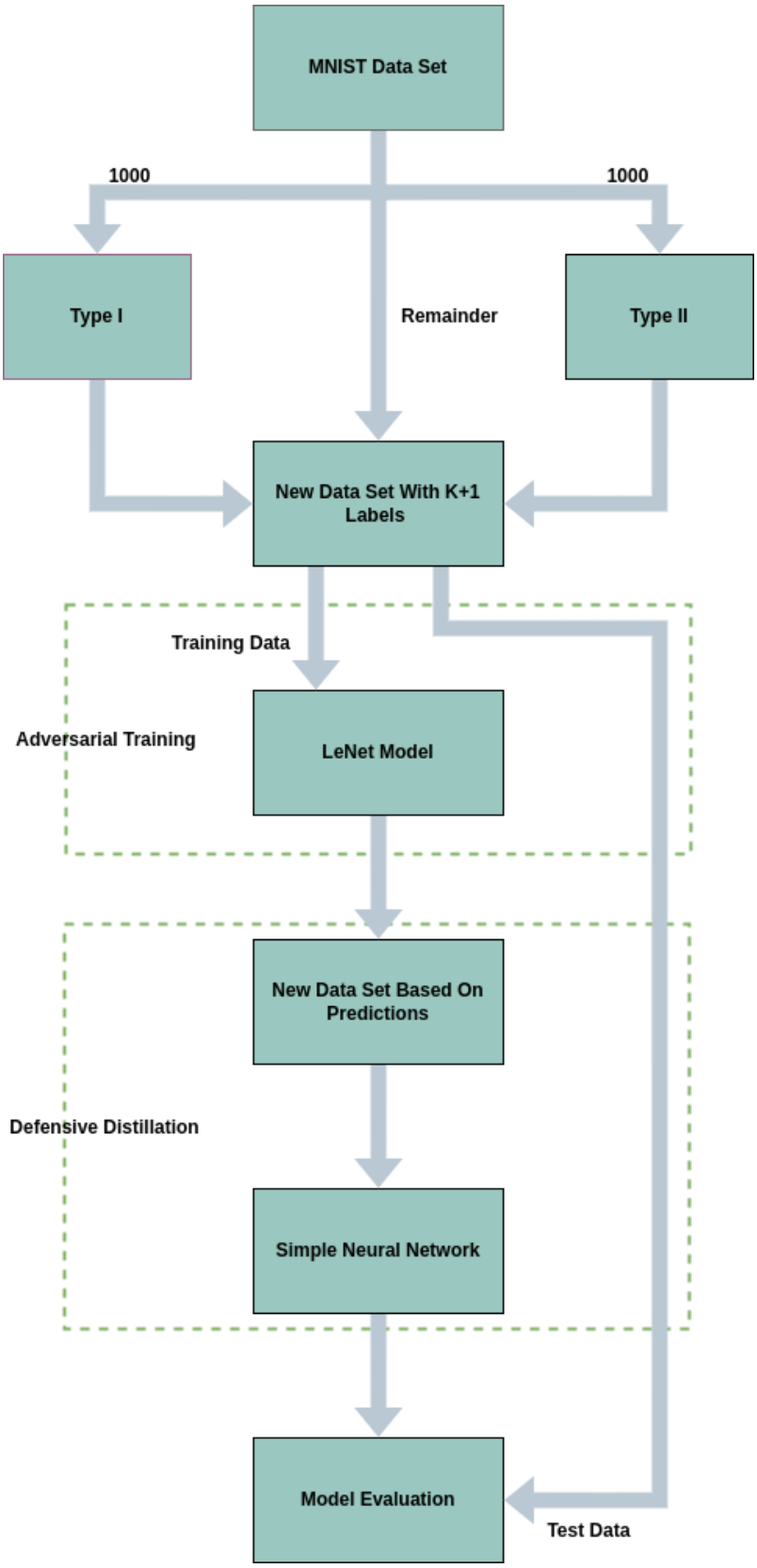


Figure 9: High-level methodology

The research methodology consists of three main components. They are,

- Adversarial image generation
- Adversarial training
- Defence distillation

As the first step, the research creates adversarial images using the MNIST dataset. The research selects roughly 1% of the MNIST data to create type I error-based adversarial images. Then the research selects roughly 1% of the MNIST data to create type II error-based adversarial images. Using these new adversarial images, the research creates a new training dataset. That is the first step of the methodology.

In the new dataset, the research has an additional class label to represent images created using type I error. These images do not belong to any category in the MNIST dataset. So, the research added a new class label to represent them. After creating the new dataset, the research uses some parts of the dataset for model training and the remainder for evaluating the models. Both training and testing datasets have adversarial samples related to two types of errors so that the research can evaluate the accuracy of the solution.

As depicted in the diagram, model training has two models. The first model is related to adversarial training. There the research lets the machine learning model to train with the adversarial samples so that it can withstand adversarial images. Then in the second model, the research uses defense distillation which is another technique to counter adversarial images. As the dataset for the second model, the research uses the predictions given by the first model.

Finally, for the evaluation, the research uses the test dataset. Test dataset first runs through the first model which has the adversarial training capability. Then the prediction given from the first model runs through the second model which has the defense distillation capability. The second model's prediction is the final prediction and it's used to evaluate the accuracy of the whole solution.

5.1 Dataset

As the preliminary dataset, the research is using MNIST data. In the research methodology, the research uses the MNIST dataset to create a modified dataset to evaluate our solution. MNIST is a 28 x 28-pixel grayscale images of handwritten single digits between 0 and 9. The dataset consists of total 70,000 images. Out of those 50000 images are used as training dataset and another 10000 images are used as validation dataset and the remaining 10000 images are used as test dataset. All the

images are labeled and altogether there are ten classes. When it comes to MNIST data the research cannot straight away use it for machine learning purposes. The research had to perform a pre-processing step to use the MNIST dataset for machine learning purposes. Since the research used Keras to develop neural networks first the research converted the MNIST dataset into 4-dimensional NumPy arrays. In the original form, the MNIST dataset has (70000, 28, 28) dimensions. So as reshaping the research converted this to (70000, 28, 28, 1). Then as the other pre-processing step, the research normalized the MNIST dataset. So, the research divided the RGB code of each pixel value by 255 because the maximum RGB code value is 255. After performing the pre-processing steps, the research divided the MNIST dataset into the train, validation, and test datasets.

5.1.1 Type I

The idea is to modify the original image in a manner that can be seen as a total noise image. But still to the image classifying neural network it is still the original image. So, the image classifying neural network still classifies it as an image belonging to the class of the original image.

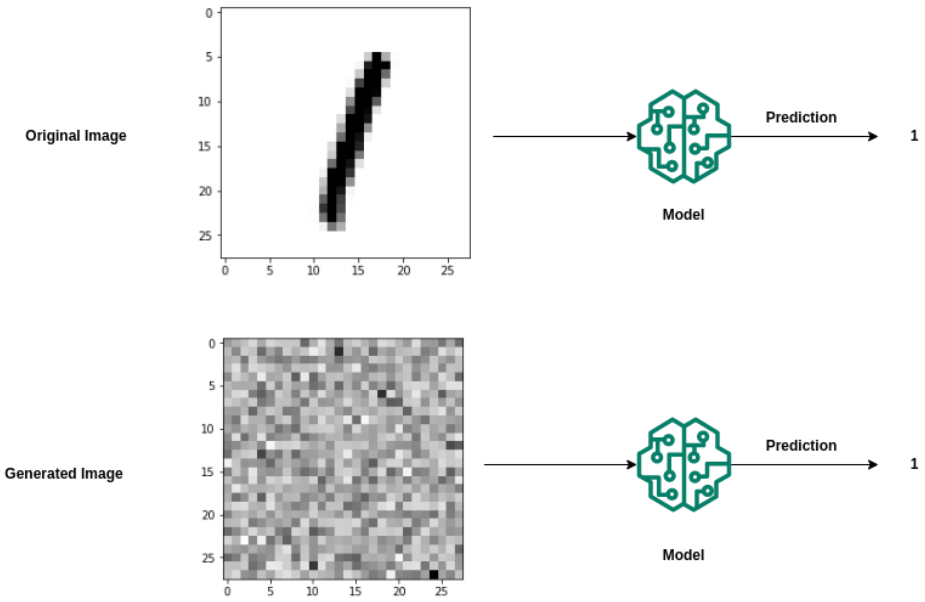


Figure 10: Image Classifying Neural Network's Behaviour for A Type I Image

The generated image can be found by solving a simple optimization problem. As the cost function, the research can use the below function.

$$C = \frac{1}{2} \| y_{goal} - F(\overline{x}) \|_2^2$$

Here y_{goal} means the label of the original image. Then $\overline{x}$ means the adversarial image. As the first step of the image generating process, the research adds some random noise to the original image such that it can be observable to the naked human eye. But the problem here is that the image classifying neural network will classify this as belonging to some other class. But what the research wants is to classify this to the original image class by the image classifying neural network. In order to achieve that the research needs to find $\overline{x}$.

This is similar to what we do while training neural networks. There we use backpropagation to find the biases and the weights of the neural network. But for this case, the research needs to find derivatives of the cost function with respect to input using backpropagation. Then the research has to gradually update the modified image using the derivatives of the cost function. The equation for that process is the below one.

$$\overline{x} = \overline{x} - \theta \times \nabla c$$

5.1.2 Type II

The idea is to modify the original image in a manner that cannot be identified by the naked human eye. But the image classifying neural network will classify this modified image to a different class from the original class due to this change.

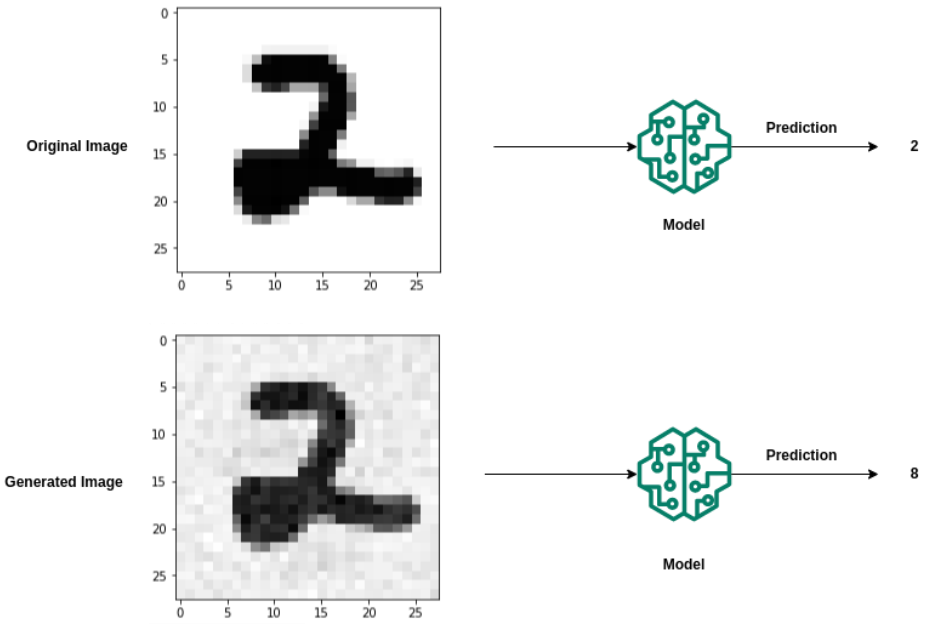


Figure 11: Image Classifying Neural Network's Behaviour for A Type II Image

In order to find the adversarial image, the research needs to minimize the following cost function.

$$C = \frac{1}{2} \| y_{goal} - F(\overline{x}) \|_2^2 + \lambda \| \overline{x} - \dot{x} \|_2^2$$

Here y_{goal} means the desired image class that the research needs to classify by the image classifying neural network. $F(\overline{x})$ means the classification given by the image classifying neural network if we feed the original image. Finally $\overline{x}$ means the original image and the $\dot{x}$ represents the adversarial image. As the first step, the research adds some random noise to the original image and creates a new image. For a Type II attack, the research needs to reduce the difference between the original image and the image generated using random noise. Also, the image classifying neural network needs to classify the new image into a different class compared to the original image's class. In order to do that the research needs to find the $\overline{x}$.

A similar process can be used here as well. The research just wants to find the derivatives of our cost function and use gradient descent to update the new image created by adding some noise to our original image. One difference here compared to Type I is that the research needs to consider the difference between the original image and the modified image. To capture all those requirements the new equation to create Type II error-based images is the below one.

$$\overline{x} = \overline{x} - \theta \times \left(\nabla c + \lambda (\overline{x} - \dot{x}) \right)$$

5.2 New Dataset

The new data set consists of Type I and Type II errors. From the original MNIST data set we used 1000 images and created Type I images out of them. These images are noise images for the naked human eye but the image classifying neural network still classifies them as belonging to their original image class. So, the image introduced a new label "10" to these Type I images. Also, the research keeps track of their original image classes because the research needs them in the evaluation phase. After all, the research needs to check the accuracy improvement of our solution.

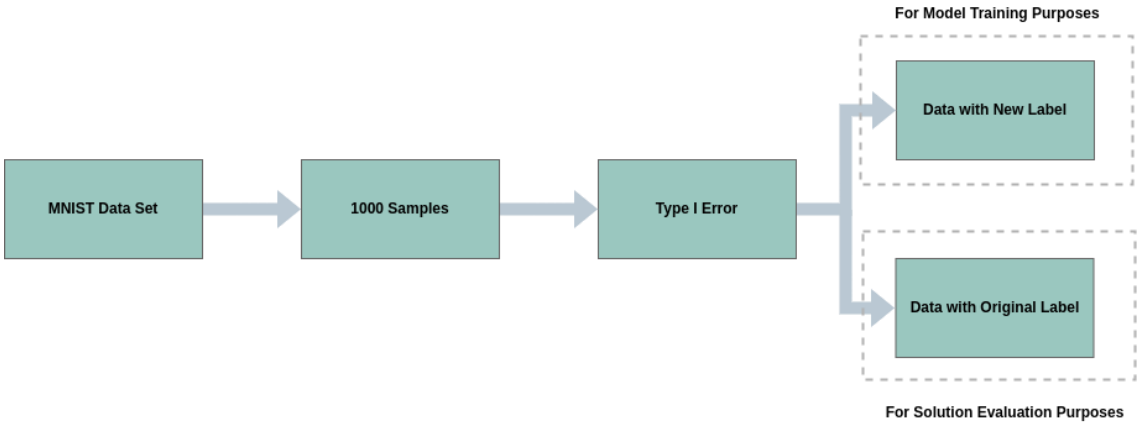


Figure 12: New Data Sets for Type I Error

In order to create Type II error-based images, the research selected another 1000 images from the MNIST data set and modified them in such a way that they cannot be identified by the naked human eye using the technique mentioned in the above sections. But image classifying neural network identifies this small change and labels that modified image into a different class label. But the research gave the original class labels of those images and created the data set. That data set can be used for model training as well as solution evaluation purposes.

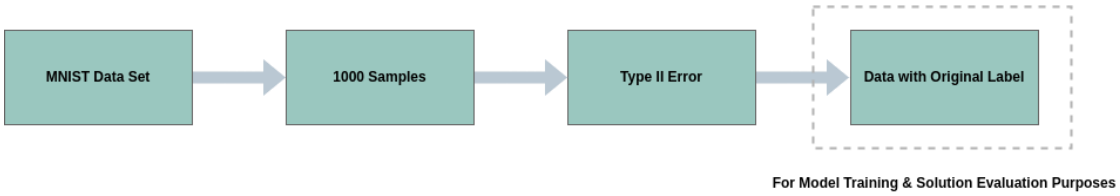


Figure 13: New Data Sets for Type II Error

Once the research has created the Type I and Type II error-based data, the research added them to the original MNIST data set by replacing their original data points. Then the research got a new data set with 11 labels. Ten labels to represent 0 to 9 digits and 10 to represent Type I error-based images. New data set is used for model training and model accuracy calculation purposes.

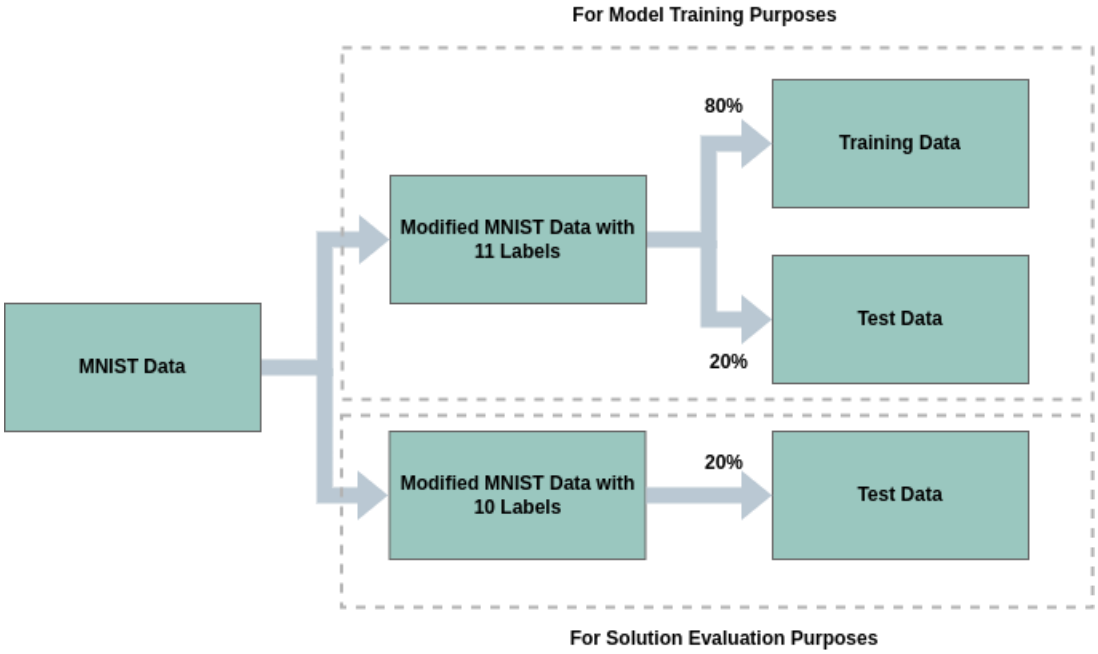


Figure 14: New Data Set with Type I & Type II Errors

5.3 Adversarial Training

During the adversarial training phase, the research trains a LeNet model [2] using the new data set which has 11 labels. This will allow the image classifying neural network to train with adversarial samples so that knowledge can be used to properly perform predictions for a test data set which has adversarial images. So, the research developed a LeNet model [2] using Keras. The research followed the same architecture introduced for the LeNet model [2] and did some minor modifications in order to support our new data set with adversarial images.

The LeNet model [2] consists of seven layers. These layers consist of three convolutional layers, 2 subsampling layers, and two dense layers.

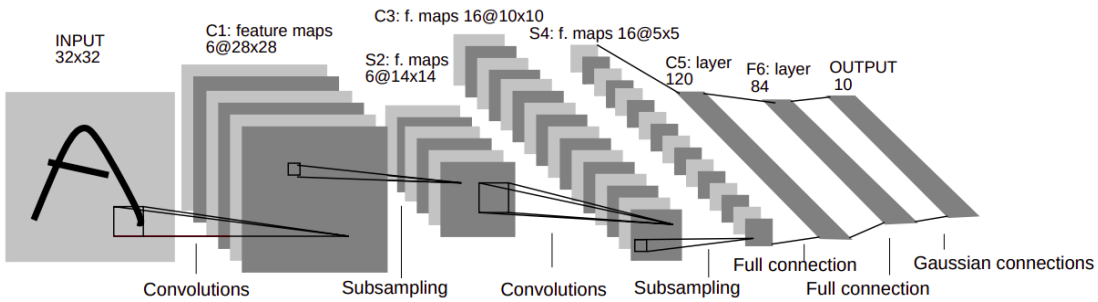


Figure 15: The LeNet Architecture [2]

The first layer is the input layer, and it is not considered as a layer because nothing is learnt from that layer. In the LeNet model [2] input is designed to take 32 x 32 inputs. Since the MNIST data set has 28 x 28 inputs the research needs to add padding. The first convolutional layer produces six feature maps as outputs. To create them the first convolutional layer uses a 5 x 5 kernel. These six feature maps will have 28 x 28 dimensions. As the next layer, we have the first subsampling layer. Its haves the feature map received from the previous layer. So, it creates six feature maps with 14 x 14 dimensions. As the third layer, the research has again a convolutional layer. This layer will create sixteen 10 x 10 feature maps. As the fourth layer again, the research has a sub-sampling layer that will create sixteen 5 x 5 feature maps. As the fifth layer again, the research has a convolutional layer where it converts 2D feature maps into 1D feature maps so the research can feed them to the dense layers. For the first dense layer, the research has 84 neurons according to the original LeNet model [2] architecture. The final dense layer which is the output layer has ten neurons according to the LeNet model [2] architecture. But for the adversarial training, the research needs eleven neurons because the research has eleven class labels in the new data set. So that is a minor modification done by the research for the LeNet model [2].

Layer	**No of Filters**	**Filter Size**	**Size of the Feature Map**	**Activation Function**
Input			32 x 32 x 1	
Conv 1	6	5 x 5	28 x 28 x 6	tanh
Sub Sampling 1		2 x 2	14 x 14 x 6	
Conv 2	16	5 x 5	10 x 10 x 16	tanh
Sub Sampling 2			5 x 5 x 16	
Conv 3	120	5 x 5	120	tanh
Dense 1			84	tanh
Dense 2			11	SoftMax

Table 1: Features of each layer of the upgraded LeNet model

Actually, the research needs another LeNet model [2] with ten neurons in the output layer to compare the accuracy of our overall solution. This model will be trained using the original MNIST data set and to test its accuracy the research will provide the MNIST data set with adversarial images but with ten image classes.

Layer	No of Filters	Filter Size	Size of the Feature Map	Activation Function
Input			32 x 32 x 1	
Conv 1	6	5 x 5	28 x 28 x 6	tanh
Sub Sampling 1		2 x 2	14 x 14 x 6	
Conv 2	16	5 x 5	10 x 10 x 16	tanh
Sub Sampling 2			5 x 5 x 16	
Conv 3	120	5 x 5	120	tanh
Dense 1			84	tanh
Dense 2			10	SoftMax

Table 2: Features of each layer of the base LeNet model

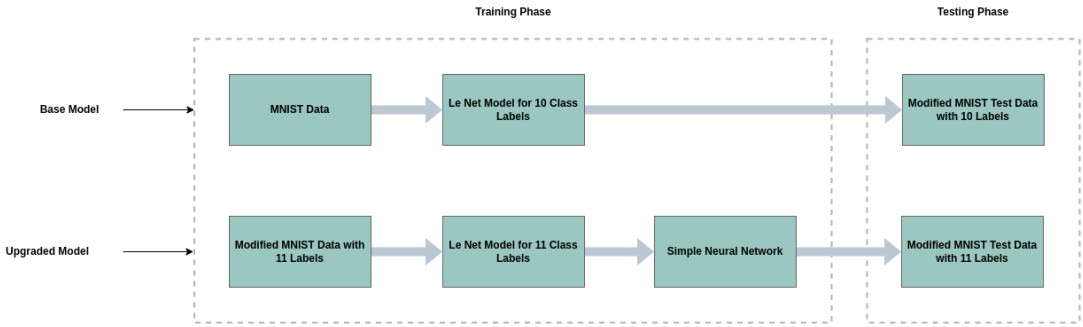


Figure 16: Adversarial Training

5.3.1 Defense Distillation

During the defense distillation phase, the research trains a simple neural network using the predictions given in the previous model used for adversarial training.

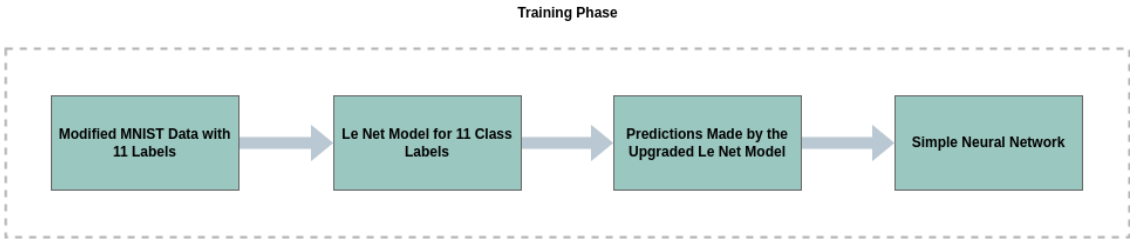


Figure 17: Defense Distillation

The simple neural network used for the defense distillation consist of a single convolutional layer, a single sub-sampling layer, and two dense layers. The final dense layer is the output layer and it consists of eleven neutrons to represent eleven class labels contained in the data set. The convolutional layer and the first dense layer use

'tanh' as the activation function. For the final layer, we use 'SoftMax' as the activation function since this is a classification problem.

Layer	No of Filters	Filter Size	Size of the Feature Map	Activation Function
Input			32 x 32 x 1	
Conv 1	6	5 x 5	28 x 28 x 6	tanh
Sub Sampling 1		2 x 2	14 x 14 x 6	
Dense 1			84	tanh
Dense 2			11	SoftMax

Table 3:Features of each layer of the simple neural network

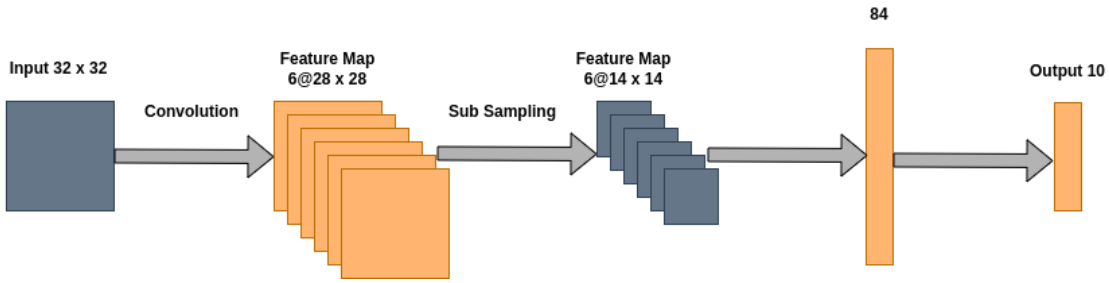


Figure 18: Architecture of the simple neural network

5.3.2 Evaluation

For evaluation purposes, the research uses the base to identify the improvements made by the proposed solution. The research is going to identify the accuracy of the base model using the test data set which consists of adversarial images. But this test data set has only ten class labels. In order to identify the accuracy of the proposed solution, the research is using the test data set with adversarial images which have eleven class labels.

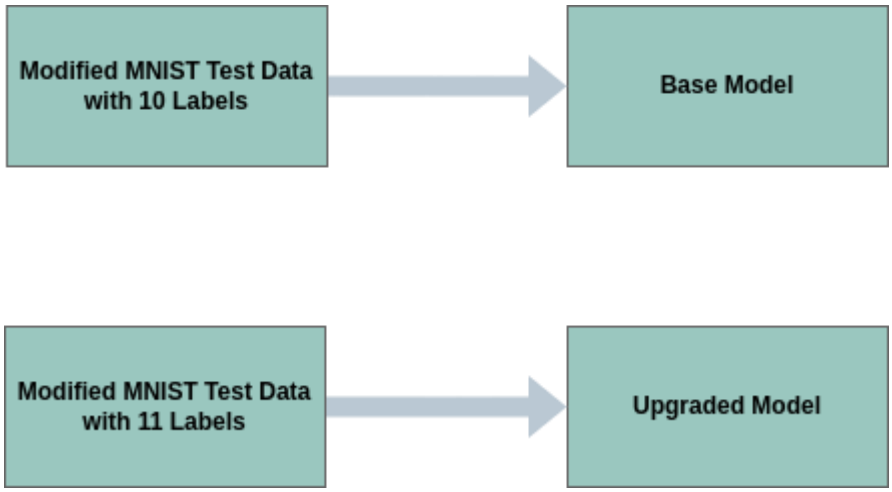


Figure 19: Model Testing

For both data sets the research is using hold-out-cross-validation with the stratified sampling. So, both training data and testing data will have the same class distribution. This applies to both data sets.

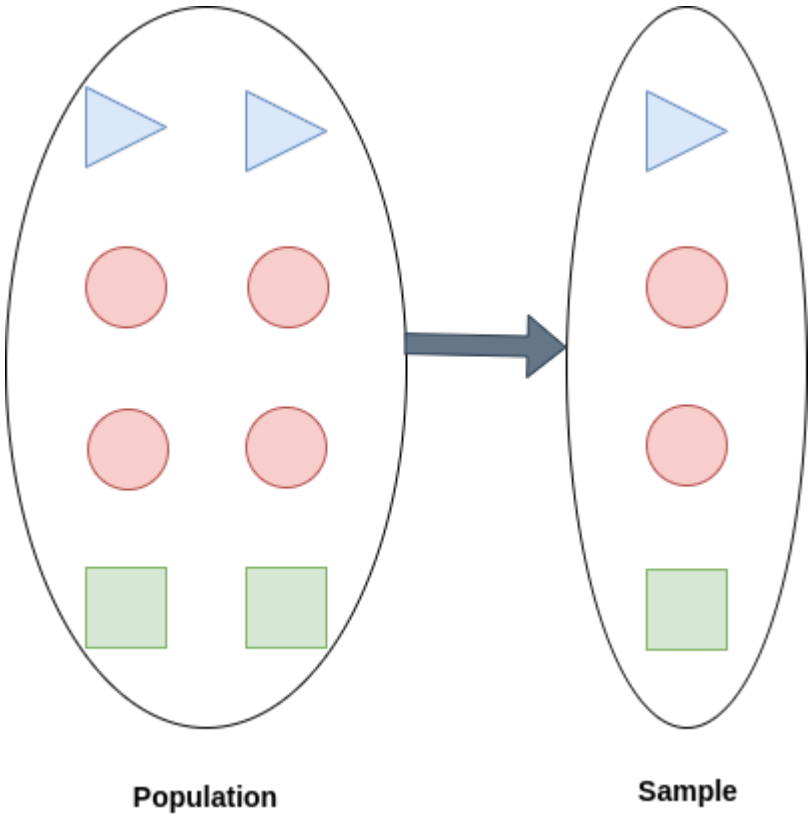


Figure 20: Stratified Sampling

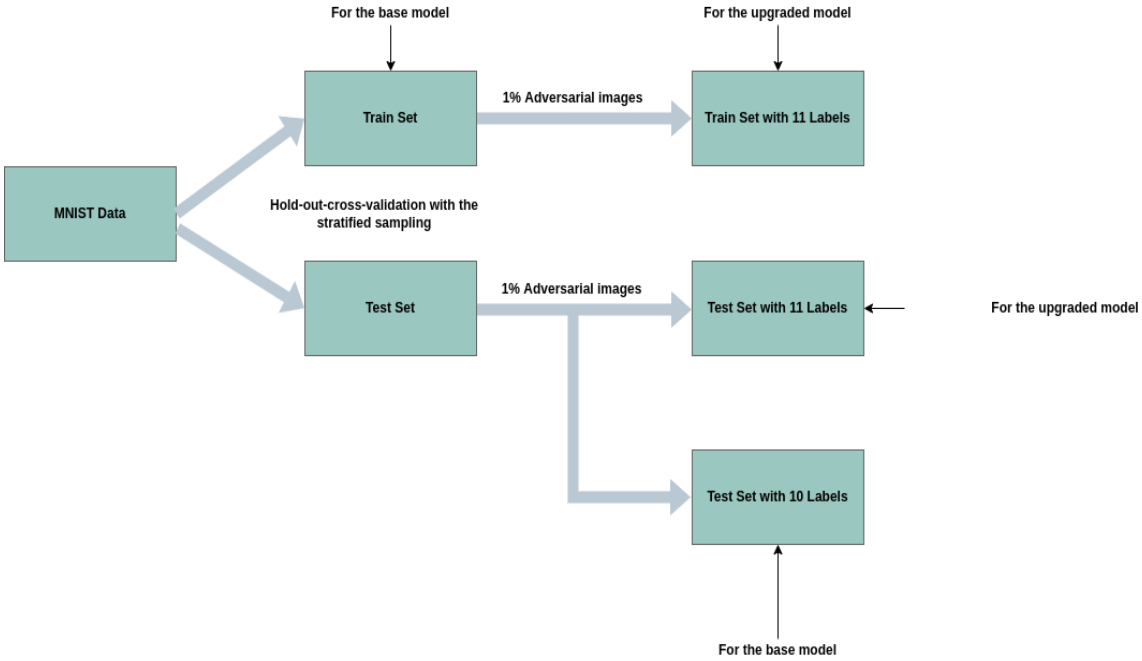
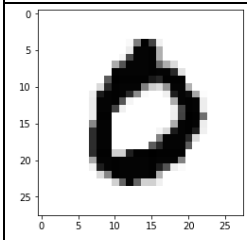
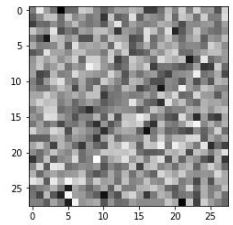
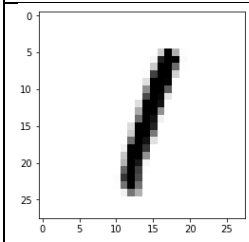
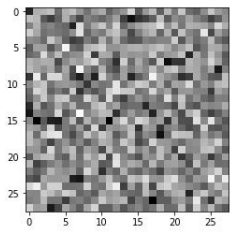
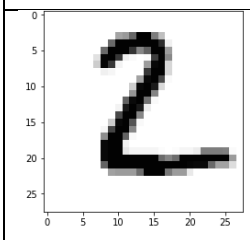
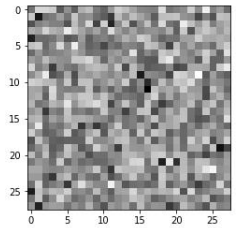
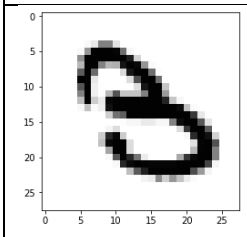
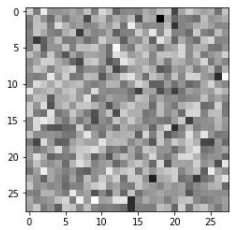
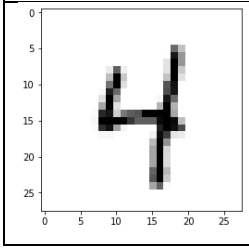
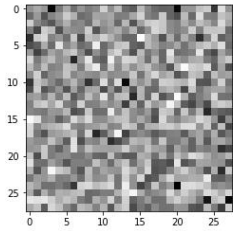


Figure 21: Training & Testing Data Sets

6 Results and Evaluation

6.1 Adversarial Images

Adversarial images related to Type I & Type II were developed. The below diagram shows Type I adversarial images generated for each number. The research's adversarial image generation techniques have been able to generate noise images to fool the image classifying neural network such that it still classifies the noise image to the original class. Ideally, these should be identified as noise images.

Original Image	Original Class	Adversarial Image	Predicted Class
	0		0
	1		1
	2		2
	3		3
	4		4

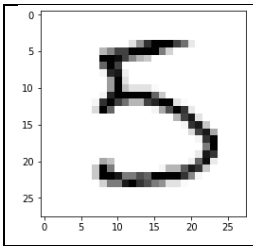
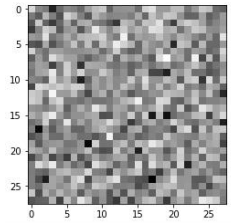
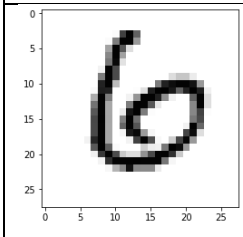
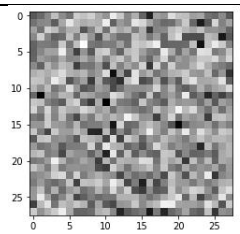
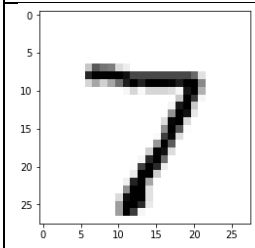
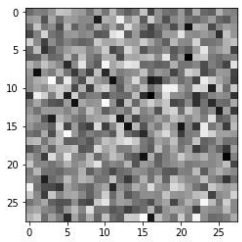
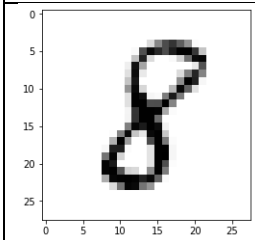
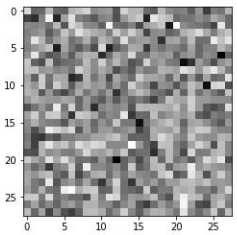
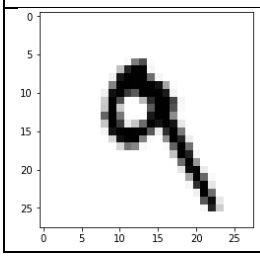
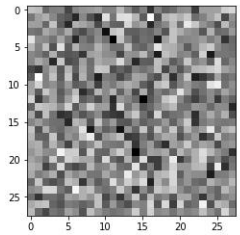
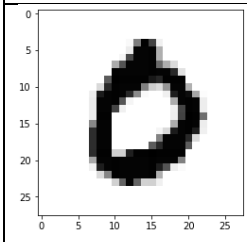
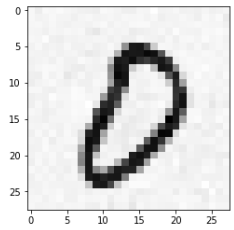
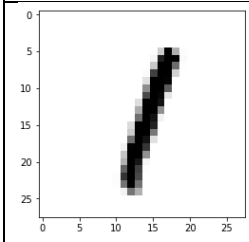
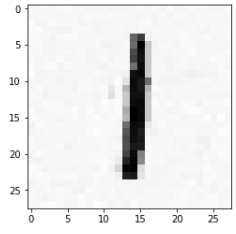
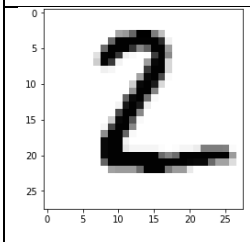
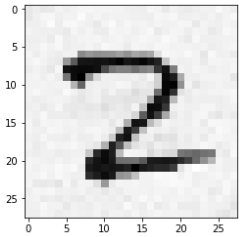
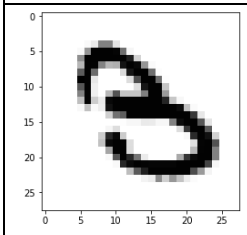
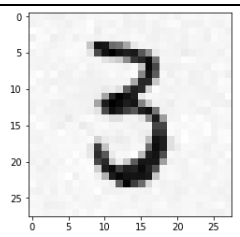
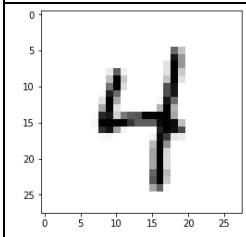
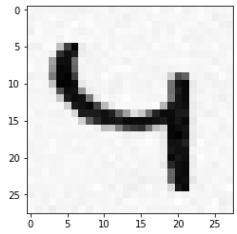
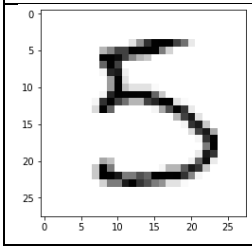
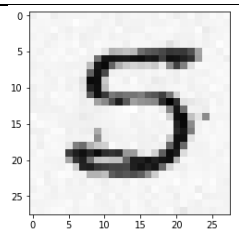
	5		5
	6		6
	7		7
	8		8
	9		9

Table 4: Type I Error

The below diagram shows the Type II adversarial images generated for each digit. These images are similar to the original image. But the image classifying neural network still can't identify it. So, it classifies these adversarial images to a different class compared to their original class. Ideally, the image classifying neural network should be able to omit the noise added to the original image.

Original Image	Original Class	Adversarial Image	Predicted Class
	0		2
	1		8
	2		3
	3		8
	4		7
	5		0

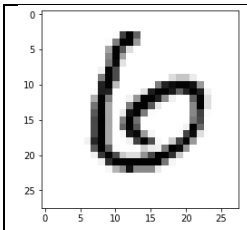
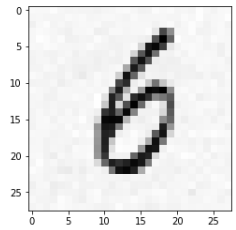
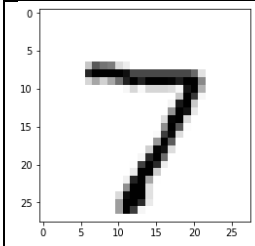
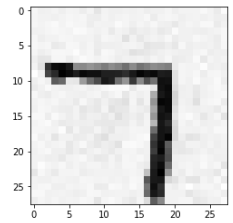
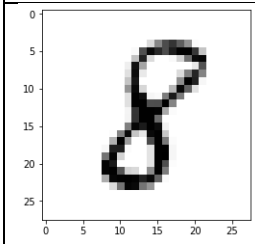
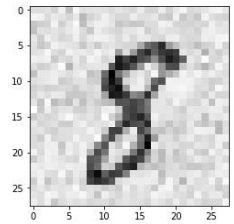
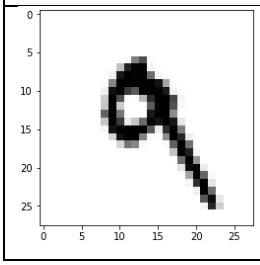
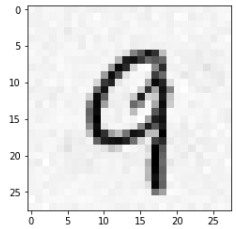
	6		3
	7		3
	8		1
	9		5

Table 5: Type II Error

6.2 Model Training

For model training, first the research created a training data set and a test data set. This was done using stratified sampling. The research used stratified sampling for a few reasons. One reason is to make sure the research has the same class distribution in the test data set and in the training data set. The other reason is to make sure the research creates adversarial samples for each image class-based according to the original distribution.

Data set	Number of data points
Train data set	49000
Test data set	21000
Type I error in train data set	490
Type I error in test data set	210
Type II error in train data set	490
Type II error in test data set	210

Table 6: Number of data points in data sets

The research had to create data sets with different class counts for training and testing purposes. For the upgraded model the research uses train and test data sets with 11 class labels so that the research can represent noise images. But for the base model, the research uses 10 class labels for the training data set. For testing purposes for the base model, the research uses 10 class labels and 11 class labels. In order to test the base model with 11 class labels, the research added 11 neurons for the output layer but trained with data having 10 class labels.

Data set	Number of classes
Train data set for the base model	10
Train data set with adversaries for the upgraded model	11
Test data set with adversaries for the base model	10
Test data set with adversaries for the upgraded model	11

Table 7: Number of classes for data sets

6.3 Accuracy

Accuracy was measured using test and train data sets for the base model and the upgraded model. The research has tried different scenarios with testing and training data to identify the effectiveness of the upgraded model. The research measures the accuracy of the upgraded model with 11 class label testing data with adversarial images. Also, the research measured the accuracy of the base model with 11 and 10 class label testing data with adversarial images.

Model	Precision	Recall	F1 Score
Training accuracy base model	98.96%	98.96%	98.96%
Testing accuracy base model with adversaries	88.29%	89.09%	88.69%
Training accuracy upgraded model	99.22%	99.24%	99.23%
Testing accuracy LeNet (trained with adversaries) model with adversaries	94.6%	94.6%	94.6%
Testing accuracy upgraded model with adversaries	97.98%	97.99%	97.98%

Table 8: Model accuracies with 2% adversarial data

As we can see there is a roughly ten percent accuracy difference between the base model and the upgraded model if the research converts 2% of the data in two adversarial images. To clearly identify this accuracy difference, the research tried the adversarial image percentage of the data set.

Model	Precision	Recall	F1-Score
Training accuracy base model	98.70%	98.69%	98.69%
Testing accuracy base model with adversaries	87.54%	88.85%	88.19%
Training accuracy upgraded model	99.39%	99.40%	99.40%
Testing accuracy LeNet (trained with adversaries) model with adversaries	94.71%	94.73%	94.7%
Testing accuracy upgraded model with adversaries	98.05%	98.06%	98.05%

Table 9: Model accuracies with 4% adversarial data

As we can see there is a roughly 10% accuracy difference between the base model and upgraded model if the research converts 4% data into adversarial images.

Model	Precision	Recall	F1-Score
Training accuracy base model	99.20%	99.19%	99.20%
Testing accuracy base model with adversaries	87.31%	89.50%	88.39%
Training accuracy upgraded model	99.43%	99.42%	99.42%
Testing accuracy LeNet (trained with adversaries) model with adversaries	94.77%	94.79%	94.76%
Testing accuracy upgraded model with adversaries	97.98%	97.97%	97.98%

Figure 22: Model accuracies with 6% adversarial data

Model	Precision	Recall	F1-Score
Training accuracy base model	88%	87%	88%
Testing accuracy base model with adversaries	78%	80%	79%
Training accuracy upgraded model	91%	91%	91%
Testing accuracy LeNet (trained with adversaries) model with adversaries	86%	86%	86%
Testing accuracy upgraded model with adversaries	88%	87%	88%

Table 10: Model accuracies with 6% adversarial data for MNIST fashion data

As we can see there is a roughly 10% accuracy difference between the upgraded model and the base model when the research converts 6% of the data into adversaries. With different adversarial percentages, a common observation we see is that the accuracy of the upgraded model does not change drastically. The upgraded model has a constant accuracy compared to the base model. Constant accuracy for different adversarial image percentages means that the defense mechanism has good resistance to adversarial images.

The research observed similar accuracy reduction for MNIST fashion data set as well. For the MNIST fashion data set the upgraded model gives 88% accuracy against adversarial images. But the Le Net accuracy has a 10% reduction against adversarial images even in the MNIST fashion data set as well. Modern defense techniques like stochastic gradient decent masking techniques [40] provide promising results against adversarial images. Since those techniques are tested

against coloured images, the research must fulfil computation limitations to compare results.

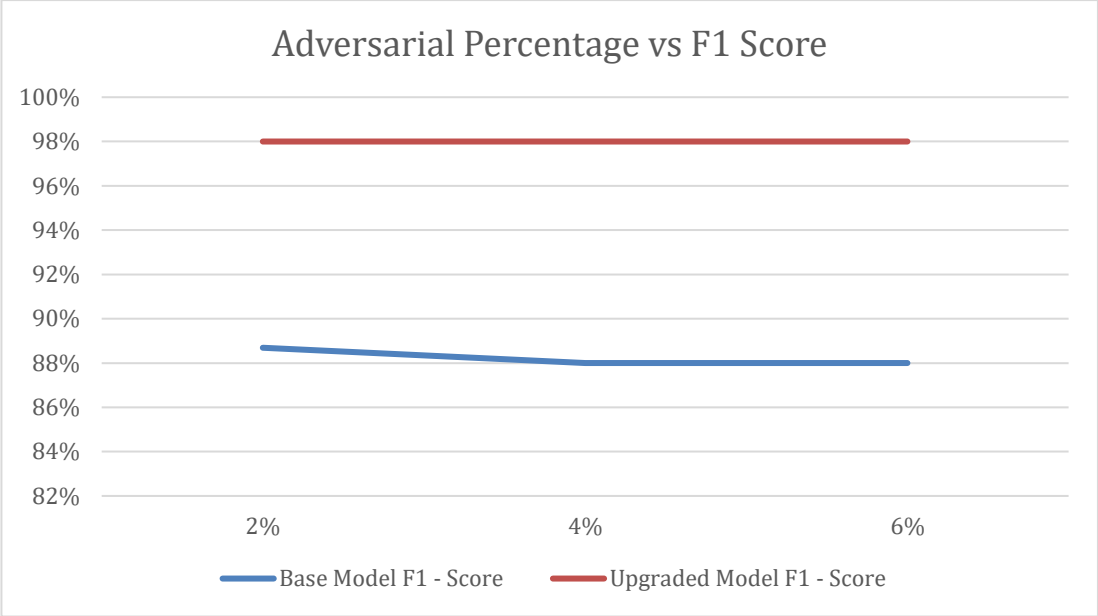
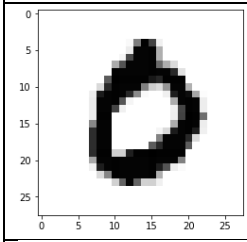
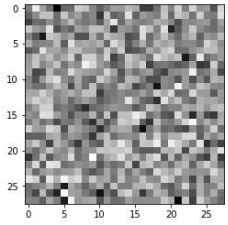
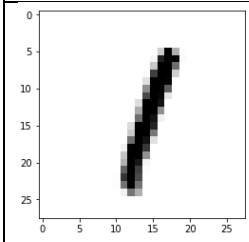
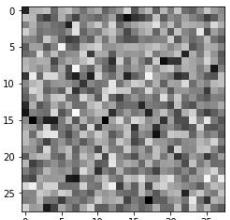
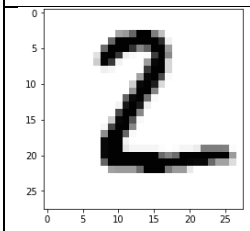
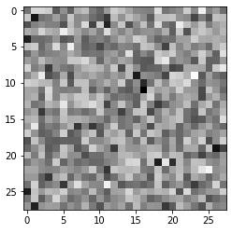
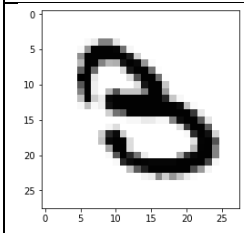
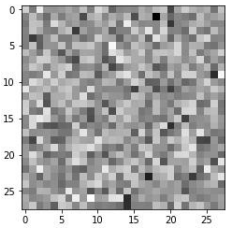
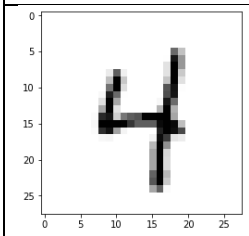
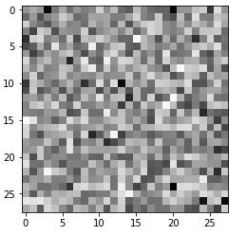
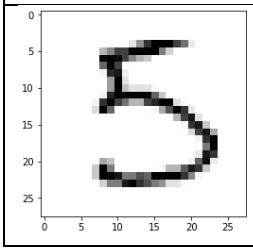
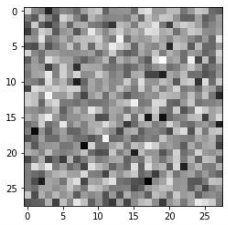
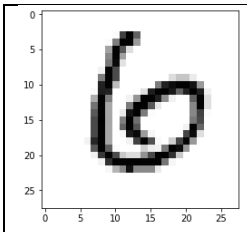
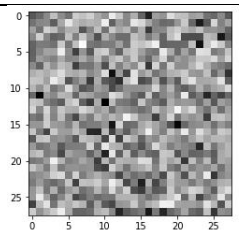
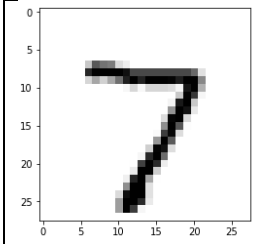
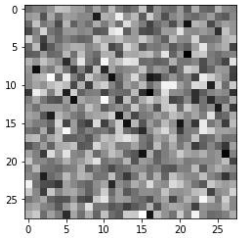
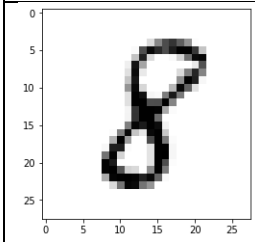
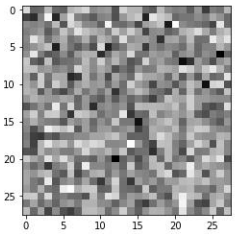
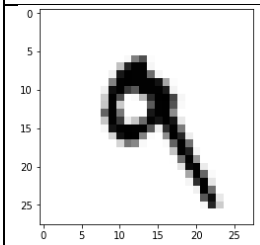
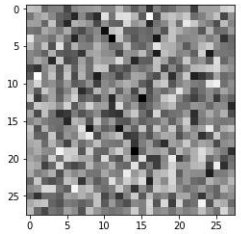
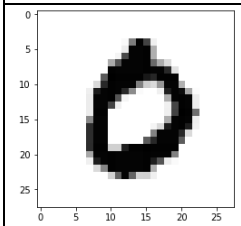
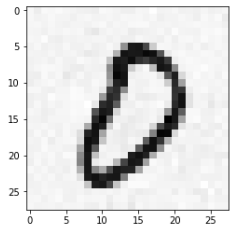
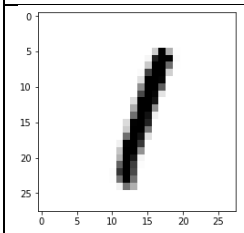
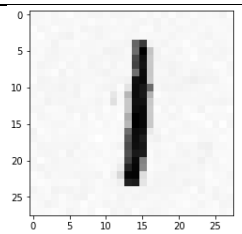
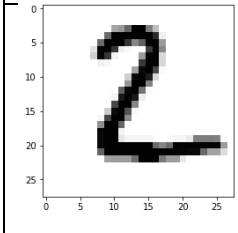
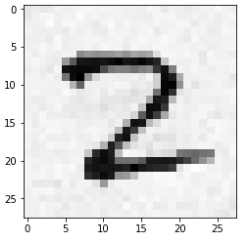


Figure 23 : F1-Score in the base model compared to the upgraded model

Apart from accuracy comparison, the research tried some sample adversarial images with the upgraded model to view the prediction results.

Original Image	Original Class	Adversarial Image	Base Model Predicted Class	Upgraded Model Predicted Class
	0		0	10
	1		1	10
	2		2	10
	3		3	10
	4		4	10
	5		5	10

	6		6	10
	7		7	10
	8		8	10
	9		9	10
	0		2	0
	1		8	1
	2		3	2

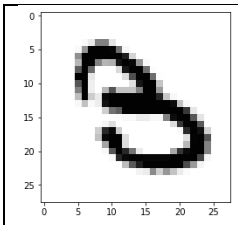
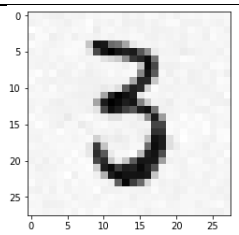
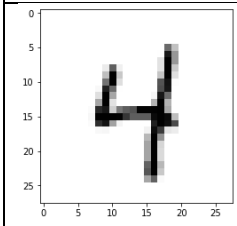
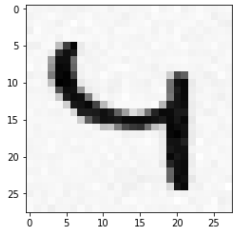
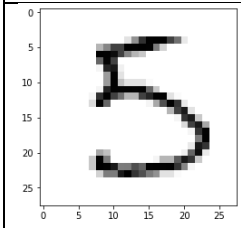
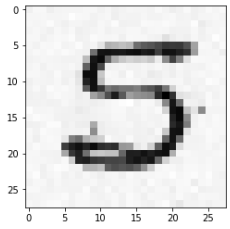
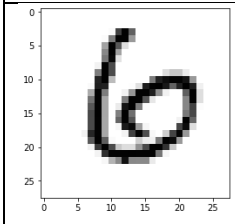
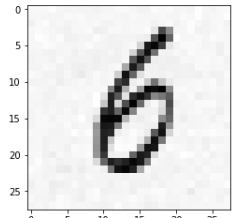
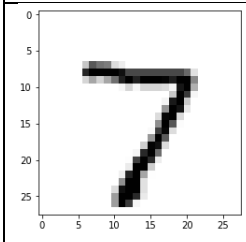
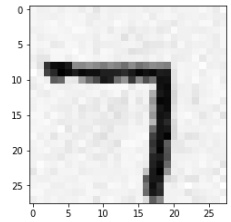
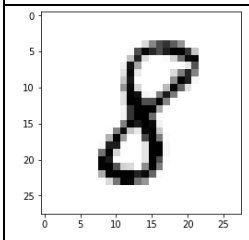
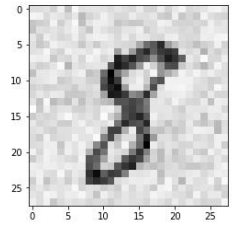
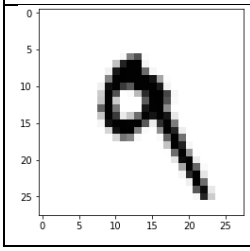
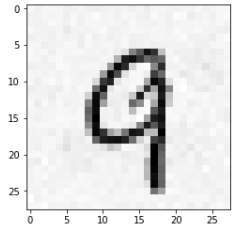
	3		8	3
	4		7	4
	5		0	5
	6		3	6
	7		3	7
	8		1	8
	9		5	9

Table 11: Special Scenarios with the Upgraded Model

By checking out the behavior of the upgraded model for adversarial images, the upgraded model can clearly identify the differences and mitigate them and provide the desired output. This can be further confirmed by the accuracies we got for the test data set for the upgraded model. Most of the time it’s more than 97% and accuracy does not change significantly when the research increases the adversarial image percentage. But this is not the case for the base model which is vulnerable to adversarial images.

7 Conclusion

The research was focused on creating defense mechanisms for Type I and Type II error-based adversarial attacks against image classifying neural networks. As a part of the research, the research generated adversarial images for both types of errors which can fool an image classifying neural network. To counter these attacks, the research tried a combined approach of adversarial training and defense distillation. According to the results test approach provide good resistance against adversarial images. When the research increases the adversarial sample count in the data set the tested approach gave good resistance and gave consistent accuracy of more than 97%. But this is not the case for normal image classifying neural networks. In this occasion, the research used Le Net [2] as the base model and its accuracy drastically reduces with the increase of adversarial samples. Therefore, the research can conclude that the proposed methodology provides good resistance, and this approach can be used as a defense option against adversarial attacks.

8 REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[3] Nguyen, Anh & Yosinski, Jason & Clune, Jeff. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 427-436. 10.1109/CVPR.2015.7298640.

[4] Papernot, Nicolas & McDaniel, Patrick & Jha, Somesh & Fredrikson, Matt & Celik, Z. Berkay & Swami, Ananthram. (2016). The Limitations of Deep Learning in Adversarial Settings. 372-387. 10.1109/EuroSP.2016.36.

[5] Y. LeCun, L. Bottou, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[6] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," 2016 EEE Symposium on Security and Privacy (SP), San Jose, CA, 2016, pp. 582-597, doi: 10.1109/SP.2016.41.

[7] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in Deep Learning and Representation Learning Workshop at NIPS 2014. arXiv preprint arXiv:1503.02531, 2014.

[8] Goodfellow, Ian & Shlens, Jonathon & Szegedy, Christian. (2014). Explaining and Harnessing Adversarial Examples. arXiv 1412.6572.

[9] Kurakin, Alexey & Goodfellow, Ian & Bengio, Samy. (2016). Adversarial examples in the physical world.

[10] S. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2574-2582, doi: 10.1109/CVPR.2016.282.

[11] Moosavi-Dezfooli, Seyed-Mohsen & Fawzi, Alhussein & Frossard, Pascal. (2016). DeepFool: a simple and accurate method to fool deep neural networks. CVPR.

[12] Gu, Shixiang & Rigazio, Luca. (2014). Towards Deep Neural Network Architectures Robust to Adversarial Examples.

[13] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In Joint European conference on machine learning and knowledge discovery in databases, pp. 387–402. Springer, 2013.

[14] Szegedy, Christian & Zaremba, Wojciech & Sutskever, Ilya & Bruna, Joan & Erhan, Dumitru & Goodfellow, Ian & Fergus, Rob. (2013). Intriguing properties of neural networks.

[15] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 372 387. IEEE, 2016a.

[16] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 39-57, doi: 10.1109/SP.2017.49.

[17] Xiao, C., Zhu, J.-Y., Li, B., He, W., Liu, M., and Song, D. Spatially transformed adversarial examples. arXiv preprint arXiv:1801.02612, 2018b.

[18] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security, pp. 506–519. ACM, 2017.

[19] Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397, 2017.

[20] Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 15–26. ACM, 2017.

[21] Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., and Song, D. Generating adversarial examples with adversarial networks. ArXiv preprint arXiv:1801.02610, 2018a.

[22] Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 1885–1894. JMLR. org, 2017.

[23] Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean label poisoning attacks on neural networks. In Advances in Neural Information Processing Systems, pp. 6103–6113, 2018a.

[24] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning models. arXiv preprint arXiv:1707.08945, 2017.

[25] Buckman, J., Roy, A., Raffel, C., and Goodfellow, I. Thermometer encoding: One hot way to resist adversarial examples. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=S18Su--CW.

[26] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[27] Gu, S. and Rigazio, L. Towards deep neural network architectures robust to adversarial examples. arXiv preprint arXiv:1412.5068, 2014.

[28] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

[29] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. ArXiv preprint arXiv:1705.07204, 2017.

[30] Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! ArXiv preprint arXiv:1904.12843, 2019.

[31] Carlini, N., Katz, G., Barrett, C., and Dill, D. L. Provabl minimally distorted adversarial examples. arXiv preprint arXiv:1709.10207, 2017.

[32] Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In Advances in Neural Information Processing Systems, pp. 2266–2276, 2017.

[33] Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. arXiv preprint arXiv:1801.09344, 2018a.

[34] Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 3–14. ACM, 2017a.

[35] Grosse, K., Manoharan, P., Papernot, N., Backes, M., and Mc- Daniel, P. On the (statistical) detection of adversarial examples. arXiv preprint arXiv:1702.06280, 2017

[36] Hendrycks, D. and Gimpel, K. Early methods for detecting adversarial images. arXiv preprint arXiv:1608.00530, 2016.

[37] Xu, W., Evans, D., and Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155, 2017.

[38] Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1528–1540. ACM, 2016.

[39] Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., and Yuille, A. Adversarial examples for semantic segmentation and object detection. In Proceedings of the IEEE International Conference on Computer Vision, pp. 1369–1378, 2017b.

[40] Xie, Meiyan & Xue, Yunzhe & Roshan, Usman. (2019). Stochastic Coordinate Descent for 01 Loss and Its Sensitivity to Adversarial Attacks. 299-304. 10.1109/ICMLA.2019.00056.