Generative Model Based Adversarial Defenses for Deepfake Detectors

School of Computer Science & Applied Mathematics
University of the Witwatersrand

Kavilan Nair
1076342

Supervised by Professor Richard Klein

August 21, 2023

A Research Report submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Science by Coursework and Research Report in Computer Science

Abstract

Deepfake videos present a serious threat to society as they can be used to spread misinformation through social media. Convolutional Neural Networks (CNNs) have been effective in detecting deepfake videos, but they are vulnerable to adversarial attacks that can compromise their accuracy. This vulnerability can be exploited by deepfake creators to evade detection. In this study, we evaluate the effectiveness of two generative adversarial defense mechanisms, APE-GAN and MagNet, in the context of deepfake detection. We use the FaceForensics++ dataset and a CNN victim model based on the XceptionNet architecture, which we attack using the iterative fast gradient sign method at two different levels of ϵ, $\epsilon = 0.0001$ and $\epsilon = 0.01$. We find that both APE-GAN and MagNet can purify the adversarial images and restore the performance of the victim model to within 10% of the model's accuracy on benign fake inputs. However, these methods were less effective at restoring accuracy for adversarial real examples and were not able to significantly restore accuracy when the adversarial attack was aggressive ($\epsilon = 0.01$). We recommend that an adversarial defense method be used in conjunction with a deepfake detector to improve the accuracy of predictions. APE-GAN and MagNet are effective methods in the deepfake context, but their effectiveness is limited when the adversarial attack is aggressive.

Declaration

I, Kavilan Nair, hereby declare the contents of this Research Report to be my own work. This Research Report is submitted for the Master of Science by Coursework and Research Report in Computer Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

Acknowledgements

The Research Report presented here is a culmination of challenging yet rewarding work over the period of one and a half years. I would first like to thank my supervisor, Dr Richard Klein, for all the support and guidance throughout the duration of this research. I thoroughly enjoyed working under your supervision and tapping into your vast and in-depth knowledge throughout the countless research group meetings.

I would also like to thank my former employer, DataProphet for enabling me to pursue my Masters. They were extremely supportive and understanding throughout the whole process. To my DataProphet mentor, Holly Arrowood, thank you for all the advice, guidance and counsel throughout the whole process.

Lastly to my family and partner, thank you for always believing in me and providing me with a supportive environment that has enabled me to pursue what I want out of life. I would not have been able to do this without you all.

Contents

Preface
Abstract i
Declaration ii
Acknowledgements iii
Table of Contents iv
List of Figures vi
List of Tables viii

1 Introduction **1**
1.1 Motivation 1
1.2 Problem Statement 2
1.3 Research Questions 2
1.4 Outline 3

2 Background **4**
2.1 Computer vision models 4
2.1.1 Convolutional Neural Networks 4
2.2 Adversarial examples 5
2.2.1 Adversarial Attacks 5
2.2.2 Adversarial Defenses 7
2.3 Generative Models 8
2.3.1 Autoencoder 8
2.3.2 Generative Adversarial Networks 9

3 Related Work **11**
3.1 Deepfake Generation and Dataset 11
3.1.1 DeepFakes 11
3.1.2 Face2Face 11
3.1.3 FaceSwap 12
3.1.4 NeuralTextures 12
3.2 Deepfake Detection 13
3.3 Adversarial Deepfakes 13
3.4 Adversarial Defenses 14
3.4.1 APE-GAN 14
3.4.2 MagNet 16

4 Adversarial Defense Workflow **18**
4.1 Dataset . . . 18
4.2 Victim Model . . . 18
4.3 Crafting Adversarial Dataset . . . 19
4.4 Training Defense Model . . . 19
4.5 Evaluation of Defense Model . . . 20

5 Datasets and Victim Models **21**
5.1 MNIST . . . 21
5.1.1 Dataset . . . 21
5.1.2 Victim Model . . . 22
5.1.3 Adversarial Dataset . . . 22
5.2 CIFAR-10 . . . 23
5.2.1 Dataset . . . 23
5.2.2 Victim Model . . . 23
5.2.3 Adversarial Dataset . . . 24
5.3 FaceForensics++ . . . 25
5.3.1 Dataset . . . 25
5.3.2 Victim Model . . . 25
5.3.3 Adversarial Dataset . . . 29

6 Results and Analysis **34**
6.1 MNIST . . . 34
6.1.1 APE-GAN . . . 34
6.1.2 MagNet . . . 38
6.1.3 Analysis . . . 41
6.2 CIFAR-10 . . . 41
6.2.1 APE-GAN . . . 41
6.2.2 MagNet . . . 44
6.2.3 Analysis . . . 47
6.3 FaceForensics++ . . . 47
6.3.1 APE-GAN . . . 48
6.3.2 MagNet . . . 52
6.3.3 Analysis . . . 56
6.4 Future recommendations . . . 58

7 Conclusion **60**

References **65**

List of Figures

2.1 Autoencoder high-level model architecture . . . 9
2.2 GAN high-level model architecture . . . 9
2.3 DCGAN discriminator high-level architecture . . . 10
2.4 DCGAN generator high-level architecture . . . 10

3.1 NeuralTextures deepfake generation method [Thies *et al.* 2019] . . . 12
3.2 APE-GAN training architecture . . . 15
3.3 Pipeline combining APE-GAN and target model . . . 16
3.4 MagNet reformer training architecture . . . 17

5.1 MNIST examples for each class 0-9 . . . 21
5.2 MNIST FGSM sample images $\epsilon = 0.1$ to $\epsilon = 0.5$ . . . 22
5.3 CIFAR-10 examples for each class . . . 23
5.4 CIFAR-10 FGSM sample images $\epsilon = 0.01$ to $\epsilon = 0.1$ . . . 24
5.5 Source and target video snapshots from real test set . . . 25
5.6 FaceForensics++ fake video snapshots from test set showing the DeepFakes, Face2Face, FaceSwap and NeuralTextures method . . . 26
5.7 Deepfake classifier detection pipeline . . . 27
5.8 Real frame with face detection model applied . . . 27
5.9 XceptionNet CNN architecture [Chollet 2017] . . . 28
5.10 Model prediction pipeline example . . . 29
5.11 Real video prediction and Adversarial real video prediction . . . 31
5.12 DeepFakes video prediction and adversarial DeepFakes video prediction . 31
5.13 Face2Face video prediction and adversarial Face2Face video prediction . 31
5.14 FaceSwap video prediction and adversarial FaceSwap video prediction . 32
5.15 Comparing adversarial attacks on real images . . . 32
5.16 Comparing adversarial attacks on fake images . . . 33
5.17 NeuralTextures video prediction and adversarial NeuralTextures video prediction . . . 33

6.1 MNIST Benign, Adversarial (FGSM $\epsilon = 0.1$), APE-GAN Purified Adversarial 35
6.2 MNIST Benign, Adversarial (FGSM $\epsilon = 0.2$), APE-GAN Purified Adversarial 35
6.3 MNIST Benign, Adversarial (FGSM $\epsilon = 0.4$), APE-GAN Purified Adversarial 36
6.4 MNIST Benign, Adversarial (FGSM $\epsilon = 0.5$), APE-GAN Purified Adversarial 36
6.5 Benign APE-GAN MNIST reconstructed images . . . 37
6.6 MNIST Benign, Adversarial (FGSM $\epsilon = 0.1$), MagNet Purified Adversarial 39
6.7 MNIST Benign, Adversarial (FGSM $\epsilon = 0.2$), MagNet Purified Adversarial 39

6.8 MNIST Benign, Adversarial (FGSM $\epsilon = 0.4$), MagNet Purified Adversarial 40
6.9 MNIST Benign, Adversarial (FGSM $\epsilon = 0.5$), MagNet Purified Adversarial 40
6.10 Benign MagNet MNIST reconstructed images 41
6.11 CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.01$), APE-GAN Purified Adversarial 42
6.12 CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.05$), APE-GAN Purified Adversarial 42
6.13 CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.1$), APE-GAN Purified Adversarial 43
6.14 Benign APE-GAN reconstructed CIFAR-10 images 44
6.15 CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.01$), MagNet Purified Adversarial 45
6.16 CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.05$), MagNet Purified Adversarial 46
6.17 CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.1$), MagNet Purified Adversarial 46
6.18 Benign MagNet CIFAR-10 reconstructed images 47
6.19 Benign, Adversarial (I-FGSM-0.0001), APE-GAN Purified Adversarial, APE-GAN Reconstructed Benign example frame comparison 50
6.20 Benign, Adversarial (I-FGSM-0.01), APE-GAN Purified Adversarial, APE-GAN Reconstructed Benign example frame comparison 52
6.21 Benign, Adversarial (I-FGSM-0.0001), MagNet Purified Adversarial, MagNet Reconstructed Benign example frame comparison 55
6.22 Benign, Adversarial (I-FGSM-0.01), MagNet Purified Adversarial, MagNet Reconstructed Benign example frame comparison 56

List of Tables

3.1 Adversarial Deepfake results summary 14

5.1 Basic CNN MNIST accuracies for benign and adversarial (FGSM) 23
5.2 Basic CNN CIFAR-10 accuracies for benign and adversarial (FGSM) . . . 24
5.3 XceptionNet deepfake detector accuracies on benign FaceForensics++ dataset 29
5.4 XceptionNet adversarial attack results for varying levels of ϵ 30
5.5 XceptionNet Deepfake detector evaluation metrics 30

6.1 MNIST APE-GAN accuracy results for the various levels of ϵ 37
6.2 MagNet MNIST architecture 38
6.3 MNIST MagNet accuracy results for the various levels of ϵ 38
6.4 CIFAR-10 APE-GAN performance 43
6.5 MagNet CIFAR-10 autoencoder architecture 44
6.6 CIFAR MagNet Performance 45
6.7 XceptionNet evaluation metrics on benign dataset 47
6.8 APE-GAN FaceForensics++ architecture 48
6.9 APE-GAN I-FGSM-0.0001 evaluation metrics 49
6.10 APE-GAN I-FGSM-0.0001 accuracies per FaceForensics++ subset 49
6.11 APE-GAN I-FGSM-0.01 evaluation metrics 51
6.12 APE-GAN I-FGSM-0.01 accuracies per FaceForensics++ subset 51
6.13 MagNet FaceForensics++ autoencoder architecture 53
6.14 MagNet I-FGSM-0.0001 evaluation metrics 53
6.15 MagNet I-FGSM-0.0001 accuracies per FaceForensics++ subset 53
6.16 MagNet I-FGSM-0.01 evaluation metrics 55
6.17 MagNet I-FGSM-0.01 accuracies per FaceForensics++ subset 55
6.18 Defense model F1-score summary 57
6.19 F1-score summary when applying the models trained with $\epsilon = 0.01$ on the $\epsilon = 0.0001$ test set 58
6.20 F1-score summary when applying the models trained with $\epsilon = 0.0001$ on the $\epsilon = 0.01$ test set 58

Chapter 1

Introduction

1.1 Motivation

The rapid development in image generation techniques over the last decade through both deep learning and traditional image processing techniques has enabled some remarkable results. These techniques that have been developed have many applications, from enriching training data for machine learning projects, up-scaling image resolution, art creation and character model generation. While this technology is extremely valuable in a variety of different applications, it can also be used in malicious ways.

Deepfakes are generated images or videos that contain a person who has been manipulated. Deepfakes have many uses, they can be used in the film and TV industry to assist in CGI techniques such as de-ageing an actor or video dubbing, to improve lip movement to match the dubbed translated audio more accurately. Deepfakes are not inherently malicious but they can be used in malicious ways. The technique can be used to create false videos with forged audio of important people such as political leaders. These videos can spread through social media extremely quickly without any validation and result in the spread of misinformation which can be extremely dangerous.

Deepfakes pose a threat to society and therefore methods to detect and flag manipulated media is critical to prevent the spread of misinformation. Deep learning computer vision techniques that utilise convolutional neural networks have proved effective to a certain extent in identifying deepfake images and videos.

While the identification of deepfake images and videos has had success through utilising deep learning models trained on labelled examples, there are ways that these models can be attacked which reduces their effectiveness. Adversarial attacks can alter images in ways that are imperceptible to the human eye, but cause a machine learning model to predict incorrectly. Given that deepfakes can be used with malicious intent, we must assume that methods to get around the detection will be implemented.

It is therefore vital to identify when an image contains an adversarial perturbation and to determine if the underlying image content is a deepfake. Techniques for defending against adversarial attacks currently exist and continue to be researched.

This research investigates different adversarial defense mechanisms in the context of deepfake detection. There has been substantial research within the field of adversarial attacks and defense but very little in the context of deepfake detection. Recent adversarial defense mechanisms use generative models and are effective against many adversarial attacks when tested on simple datasets such as MNIST, Fashion-MNIST and CIFAR-10. Deepfake images are of a higher resolution and contain more complex imagery than those contained in these datasets. Two different defense models will be implemented and compared to determine how effective the techniques are in the deepfake setting.

1.2 Problem Statement

Deepfake detectors are vulnerable to adversarial attacks in both white-box and black-box settings. It is hypothesised that APE-GAN and MagNet models can be integrated with deepfake detectors to improve their robustness to adversarial attacks.

This research investigates the efficacy of the APE-GAN and MagNet adversarial defense mechanisms in the context of deepfake detection. The FaceForensics++ deepfake dataset is used and a CNN classifier based on the XceptionNet architecture is used as a victim model.

1.3 Research Questions

1. **How well does the deepfake detector perform on the FaceForensics++ dataset?**
   It is important to first establish a baseline for the deepfake detector. The detector should be evaluated on the FaceForensics++ dataset and the key evaluation metrics analysed.

2. **How well does the victim CNN model perform on the adversarial dataset?**
   The victim CNN model should be evaluated on the adversarial dataset using the iterative fast-gradient sign method white-box adversarial attack to establish a baseline for comparison.

3. **Are the implemented APE-GAN and MagNet effective defenses on simple datasets?**
   The defenses should first be implemented and evaluated on simple datasets such as MNIST and CIFAR-10. This will validate that the implemented defenses are effective in this setting before applying them to the deepfake context.

4. **Are APE-GAN and MagNet effective defenses against adversarial examples in a deepfake setting?**
   The APE-GAN and MagNet defenses should be trained and tested on the adversarial FaceForensics++ dataset. This will indicate if the models can be used in a more complex setting such as deepfake detection.

5. **Which defense performs better between APE-GAN and MagNet?**
   The two approaches will be evaluated on the adversarial FaceForensics++ dataset and the evaluation metrics of the defenses compared.

1.4 Outline

The structure of this document is as follows: Chapter 2 covers the background by providing some insight into CNNs, autoencoders, GANs, adversarial attacks and defenses. Chapter 3 contains the related work and includes deepfake creation and detection techniques. It also details APE-GAN and MagNet, generative model based adversarial defenses. Chapter 4 covers the general adversarial defense workflow and formally defines the problem and how its evaluated. Chapter 5 covers all datasets used in the research, the victim models and the subsequent adversarial datasets that were crafted. Chapter 6 contains the results and analysis of the APE-GAN and MagNet methods applied to the MNIST, CIFAR-10 and FaceForensics++ adversarial datasets. Lastly, Chapter 7 provides the conclusion to the research.

Chapter 2

Background

This chapter covers fundamental technical concepts that are required background knowledge for the research conducted in this report. It covers the basics of CNNs, which form the basis of modern computer vision and the techniques used to detect deepfake imagery. adversarial examples and generative models which form the basis of the adversarial defense methods investigated. The background into adversarial attacks highlights the risk of such an attack in the machine learning context and offers insight the attack algorithms. Furthermore, adversarial defenses are presented to understand how an adversarial attack may be mitigated. Lastly, the autoencoder and GAN networks are explained as they form the foundation of the adversarial defenses investigated.

2.1 Computer vision models

2.1.1 Convolutional Neural Networks

Convolutional neural networks (CNN) are a type of artificial neural network that was first proposed by Lecun *et al.* [1998] to classify handwritten characters. Traditional fully connected artificial neural networks cannot extract spatial information from the vector of input pixel values provided. The high dimensionality of the image also results in requiring extremely large networks with many parameters. CNNs make use of convolutional layers that can extract local features from an image by convolving a set filter with the image. CNNs also contain max pooling layers which sub-sample the image and reduce the dimensionality of the image.

Krizhevsky *et al.* [2012] successfully implemented a deep convolutional neural network that drastically outperformed other state of art methods when classifying images contained in the ImageNet dataset. The dropout technique was leveraged in this implementation to avoid overfitting. This was a large breakthrough in the field of computer vision using convolutional neural networks and ever since, CNNs have been extremely successful when applied to image classification-related tasks and have formed the foundation of modern computer vision.

2.2 Adversarial examples

Deep neural networks have been used extensively in learning tasks and have shown exceptional performance on certain tasks such as image recognition. A counter-intuitive property of a neural network is that they are sensitive to small perturbations and can cause the network to misclassify the input data. These non-random changes that can be optimised to maximise the prediction error are known as adversarial examples [Szegedy *et al.* 2014]. A benign example is the opposite of this and does not contain any modification.

Deep neural networks, therefore, contain blind spots and can be exploited. This is a concerning characteristic of a neural network as a malicious actor can create adversarial examples to intentionally fool them. Self-driving cars often use neural network based object detectors and a malicious actor can place a carefully crafted sticker on the sign that causes the network to misclassify it as something else [Sitawarin *et al.* 2018]. This type of attack is known as an evasion attack where the input is modified such that the model misclassifies the input [Laskov and Lippmann 2010]. The other type of attack is known as a poisoning attack which corrupts the training data to achieve the desired output.

Adversarial attacks are malicious modifications of the input data that cause the model to predict incorrectly. An evasion attack can be targeted or non-targeted [Ozdag 2018]. A targeted attack is when an adversary modifies the input data with a perturbation that results in the model classifying the input data as a specific class other than the true label. A non-targeted attack results in the model classifying the input as a class that is not the true class.

Adversarial attacks can also be classified as either white-box or black-box attacks [Ozdag 2018]. A white-box attack is when the adversary has full access to the target model and the training data. In contrast, a black-box attack is when the adversary has limited knowledge of the target model and can only query the model by providing input and observing the output.

Adversarial attacks are evaluated according to two main criteria. The first criterion is the number of queries required to be made to the target model in order to create an adversarial example. The larger the number of queries, the longer it will take to form an adversarial example. The second criterion is known as the perturbation norm, which measures the error between the benign input and the adversarial input. The l_2 and l_∞ norms are the two most common measures used for quantifying the difference between the benign input and the adversarial input [Wu *et al.* 2021].

2.2.1 Adversarial Attacks

There are many different types of adversarial attacks and new ones are always being researched. This section covers a few to demonstrate the details of how an attack works.

Goodfellow *et al.* [2015] hypothesise that neural networks are still mostly linear and are vulnerable to linear adversarial perturbations. Even though non-linear functions such as sigmoid are used as activation functions for the neurons, the models mostly operate in the linear range of the function by design so that they are easy to optimise. The "fast gradient sign method" (FGSM) is a proposed method for crafting adversarial examples [Goodfellow *et al.* 2015]. The method is calculated by backpropagating gradients from the output layer back to the input image and nudging the pixel values in the direction that maximises the loss.

The fast gradient sign method to create an adversarial example is shown below in Equation (2.1) where adv_x is the adversarial image, x is the original image, ϵ is a configurable multiplicative factor parameter that represents the small error introduced to add or subtract from each pixel. $\nabla_x J(\theta, x, y_{true})$ is the gradient of the loss function where θ represents the model parameters, x the input image and y_{true} the label of the input image. The $sign$ method is used to determine if the value should be added or subtracted. If the sign of the gradient is positive, an increase in pixel intensity will increase the loss and if the sign is negative, a decrease in pixel intensity will result in an increase in the loss. This is in essence a form of gradient ascent, the opposite to gradient descent which is used to reduce loss and train neural networks. If the pixels are nudged by a large enough amount, the classifier will classify the modified image incorrectly without the pixel changes being noticeable to the human eye.

$$adv_x = x + \epsilon * sign(\nabla_x J(\theta, x, y_{true})). \tag{2.1}$$

The "basic iterative method" or "iterative fast gradient sign method" (I-FGSM) is an extension of the fast gradient sign method proposed by Kurakin *et al.* [2017] which iteratively generates adversarial examples according to a set step size. The method is outlined below in Equation (2.2). The step size is specified by α and N is the iteration. $Clip_{X,\epsilon}$ is a function that performs pixel clipping for the image so that the modified image will be near to the original image. The method is run iteratively until it fools the victim model or until it has reached a maximum number of iterations.

$$\mathbf{X}_0^{adv} = \mathbf{X}, \mathbf{X}_{N+1}^{adv} = Clip_{X,\epsilon}\{\mathbf{X}_N^{adv} + \alpha sign(\nabla_X J(\mathbf{X}_N^{adv}, y_{true}))\}. \tag{2.2}$$

In contrast to the white-box attacks above, where we have access to the model weights and gradients, black-box attacks have limited access to the target model and can only query the model by providing inputs and recording the outputs. There are two main types of black-box attacks. The first type of black-box attack is based on query feedback, where the attacker generates a perturbed input and queries the model continuously until the perturbed input is able to fool the model. The second type of black-box attack leverages the transferability property of adversarial examples [Papernot *et al.* 2016]. The transferability property means that a substitute model can be trained to perform the same task as the victim model and used to craft adversarial examples that are also effective against the victim model. This can be done with little information needed about the victim model.

A gray-box attack is another type of adversarial attack which only requires partial knowledge of the system compared to white-box attacks which require full knowledge of the system [Xiao *et al.* 2018]. Gray-box attacks are a more realistic scenario than white-box attacks, but research is mainly conducted on white-box attacks because they represent the worst case scenario.

Adversarial examples are crafted to be as close as possible to the original image as the changes should be undetectable to human observers. This can be quantified using the l_2 and l_∞ norms shown in Equations (2.3) and (2.4) where x is the benign image and y is the adversarial image. The l_2 norm measures the Euclidean distance between all the pixels in the image and sums these distances. The l_∞, which is also known as the max norm, measures the maximum pixel differences between the two images and returns the maximum difference. Adversarial attacks aim to modify the image as little as possible, meaning that these two measurements should be as close to zero as possible. If this value increases too much, the adversarial perturbations can become perceivable to the viewer.

$$l_2 = ||x - y||_2 = \sqrt{\sum_{k=1}^{n} |x_k - y_k|^2}. \tag{2.3}$$

$$l_\infty = ||x - y||_\infty = \max_i |x_i - y_i|. \tag{2.4}$$

2.2.2 Adversarial Defenses

Defense strategies have also been researched and developed to protect machine learning models against adversarial attacks. These methods do sometimes have some unintended side effects such as performance overhead. There currently does not exist a robust defense strategy for all types of adversarial attacks. New and improved attacks are continuously being developed, which also make it hard to defend against.

Adversarial training is a defense strategy that can be implemented in one of two ways. The first method involves crafting adversarial examples using different known methods and adding these examples to the training set used for the model [Szegedy *et al.* 2014]. The second method involves using a modified objective function based on the attack method [Goodfellow *et al.* 2015]. An example of this would be to use the objective function that is derived from the fast gradient sign method shown in Equation (2.5). The value of α was set to 0.5 by Goodfellow *et al.* [2015] and this was deemed to work well enough to not warrant any further optimisation. The new objective function is now comprised of the original loss function with a weighting of α and the additional component, which is the loss function with respect to the adversarial example.

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x + \epsilon * sign(\nabla_x J(\theta, x, y_{true})). \tag{2.5}$$

Gradient hiding is another defense technique that has been designed to restrict an adversary from gradient attacks. It achieves this by ensembling the neural network

model with other machine learning models that are not differentiable, such as decision trees and random forest models. Gradient hiding is only effective against white-box attacks that attempt to backpropagate gradients through the network. A substitute model can however be trained and subsequent adversarial examples generated from the substitute model to defeat the gradient hiding defense [Papernot *et al.* 2017].

The transferability property of neural networks makes them vulnerable to black-box adversarial attacks. Blocking this property is therefore vital to defend against these attacks. Hosseini *et al.* [2017] proposed a method to block the transferability property. This can be achieved by adding a NULL label for adversarial examples in the dataset. This means that if an adversarial perturbation is contained in an image, the network will classify this as an adversarial image and reject it. The limitation of this method is that the model only predicts inputs that are deemed non-adversarial. The drawback with this approach is that the model will indicate when an input is adversarial but can not remove the adversarial noise and classify it. Human intervention is required to manually perform the classification on the input.

2.3 Generative Models

Generative models learn the joint probability distribution $P(X, Y)$ compared to discriminative models which learn the conditional probability $P(Y|X)$. Generative models are able to generate new inputs by sampling from a statistical data manifold of the true dataset. When conditioned on an adversarial image, they may be able to reconstruct the benign input image as the data manifold on which they were trained does not include adversarial noise. This section covers autoencoders and Generative Adversarial Networks which are presented as defense techniques in Chapter 3.

2.3.1 Autoencoder

The autoencoder is a generative model which comprises a neural network that learns a lower-dimensional representation of the input data [Hinton and Salakhutdinov 2006]. The network contains two components, an encoder and a decoder shown in Figure 2.1. The encoder network reduces the high-dimensional input data to a latent space, z, which is a lower-dimensional space. The latent space is also known as the information bottle-neck layer and the encoder must learn to summarise enough information into a low dimensional representation to enable the decoder to build the reconstruction of the input, $\hat{x}$. The decoder network reconstructs the input data from the latent space representation. Reconstruction loss is a metric that can be defined as the difference between the input, x and the reconstructed output $\hat{x}$. The neural network is trained by minimising the reconstruction loss. A common loss function used is the mean squared error, although other perceptual losses are also sometimes used to emphasise different visual properties. The more accurately the model reconstructs the data, the better the latent space representation.

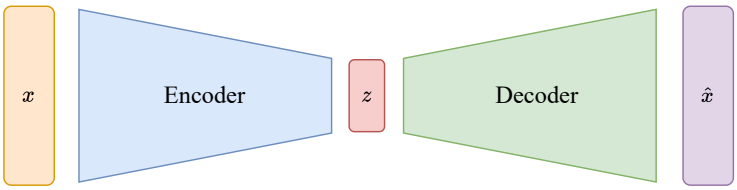


Figure 2.1: Autoencoder high-level model architecture

2.3.2 Generative Adversarial Networks

A Generative Adversarial Network (GAN) is a type of generative model that contains two networks, a generator (G), and a discriminator (D) [Goodfellow *et al.* 2014]. The structure of a GAN is shown in Figure 2.2. The objective of the generator is to produce fake data that can fool the discriminator. The discriminator classifies the data as real or fake. The generator model samples z, which contains noise and produces fake data. The generator and discriminator networks are often implemented as neural networks and both are trained through backpropagation.

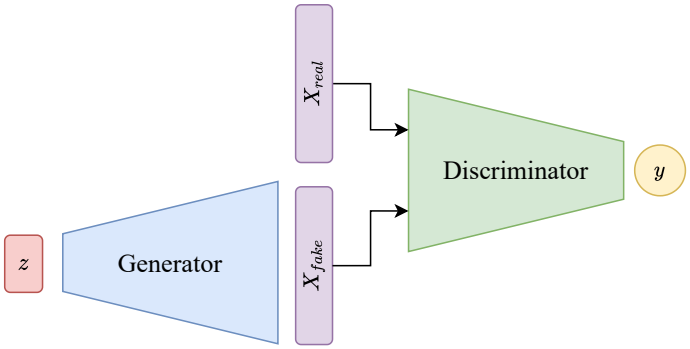


Figure 2.2: GAN high-level model architecture

The generator and discriminator networks are trained simultaneously and compete against each other in an adversarial two-player minimax game represented by the value function shown in Equation (2.6). $G(z)$ represents the generator's output when provided with noise z, $D(G(z))$ represents the discriminator's prediction that the generated data is real. $D(x)$ is the discriminator's prediction that the real data is real. $\mathbb{E}_{x\sim p_{data}(x)}$ is the expected value for all the real data instances and $\mathbb{E}_{z\sim p_z(z)}$ is the expected value of all the inputs generated from noise z.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]. \tag{2.6}$$

Radford *et al.* [2015] introduced the Deep Convolutional Generative Adversarial Network (DCGAN) which leverages convolutional layers from the CNN model architecture. Figures 2.3 and 2.4 show example architecture diagrams for a discriminator and generator respectively. This technique has been shown to learn underlying representations and successfully produce detailed images. The original GAN approach was susceptible to an unstable training process that made it difficult for the generator model to converge. The DCGAN paper provided a few suggestions to aid in the training of stable GANs. Some of the suggested techniques include the following:

- Use batch normalisation for the generator and discriminator models
- Generator should use the rectified linear unit (ReLU) activation function for all layers except the last output layer, where it should use the hyperbolic tangent function (Tanh)
- All discriminator layers should use the LeakyReLU activation function.

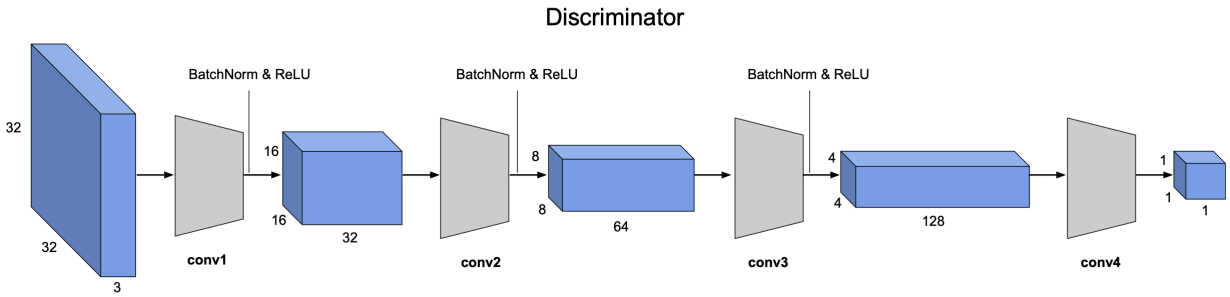


Figure 2.3: DCGAN discriminator high-level architecture

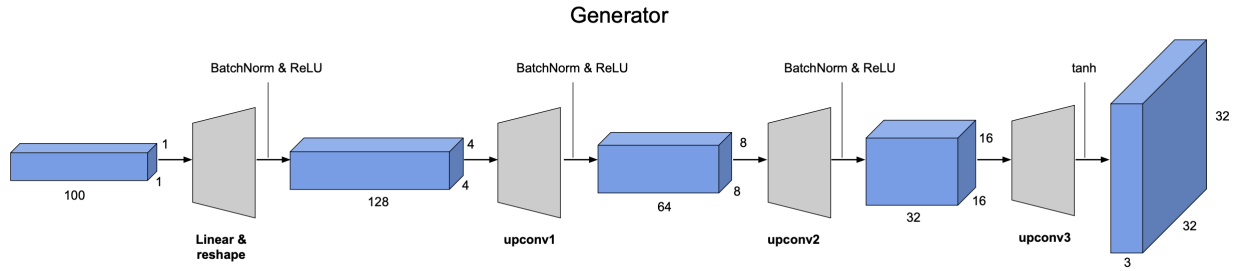


Figure 2.4: DCGAN generator high-level architecture

Arjovsky *et al.* [2017] introduced the Wasserstein GAN (WGAN) which helps prevent the vanishing gradient problem present in the traditional GAN setup, which leads to instable training meaning that the model does not converge when training. The technique uses weight clipping to ensure a smooth gradient. The WGAN uses the Wasserstein distance which is also known as the earth movers distance to measure the difference between the real and generated images data distribution.

Chapter 3

Related Work

This chapter covers related work such as the previous research conducted into deepfake datasets and the detection of deepfakes using convolutional neural networks. Additionally, adversarial defense techniques using generative models such as GANs and autoencoders are also covered.

3.1 Deepfake Generation and Dataset

FaceForensics++ is a benchmark dataset proposed by Rössler *et al.* [2018] and has been used extensively by different researchers developing deepfake detection models. The FaceForensics++ dataset consists 1000 videos which contain of over 1.8 million individual manipulated images that have been generated by four different methods: DeepFakes, Face2Face, FaceSwap and NeuralTextures. The dataset contains images with varying image resolutions and compression. Face manipulation techniques can either use reenactment or replacement techniques. The reenactment techniques transfer the expressions from a source actor to a target actor whilst maintaining the identity of the target actor. Replacement techniques remove the original face and replace it with a generated manipulated face.

3.1.1 DeepFakes

The DeepFake's generation method uses an autoencoder based approach which consists of two autoencoders with a shared encoder. The image output from the autoencoder is merged and blended with the target image using Poisson image editing [Pérez *et al.* 2003].

3.1.2 Face2Face

Face2Face is a face-swapping technique that uses reenactment techniques designed by Thies *et al.* [2016]. The facial expressions of a source actor are transferred to a target actor. The method makes use of traditional image processing techniques and does not use any deep learning networks.

3.1.3 FaceSwap

This technique extracts the facial region from one image and places it onto another image. It achieves this by using multiple different machine learning models. The first model uses a CNN to perform face detection on an image. The second model performs face alignment so that all faces have a consistent size and pose using a ResNet based model [He *et al.* 2015]. Multiple faces can be present in an image and a face recognition model is used to identify which is the face of interest and discard the other faces. A face segmentation network is then used to segment out the face of the target actor. Lastly, autoencoder models are used to perform the actual face swap which involves reconstructing the face of the target actor with the source actor. It performs this on a frame-by-frame basis to craft the manipulated video.

3.1.4 NeuralTextures

NeuralTextures proposed by Thies *et al.* [2019] introduce Deferred Neural Rendering. Their approach combined traditional image processing pipelines with learnable components. The process is shown in Figure 3.1. The process requires a source image and target image with both source and target actors present in the frame. The first step is to obtain the background from the target actor frame by removing the target actors face. The second step involves extracting the source actors expression and combining it with the identity of the target actor. The NeuralRenderer then combines the background image with the NeuralTexture to produce a realistic output The approach is successful in transferring textures from one image to another. The model used to achieve these results is derived from the U-Net architecture and combines the encoder and decoder networks with skip connections [Ronneberger *et al.* 2015].

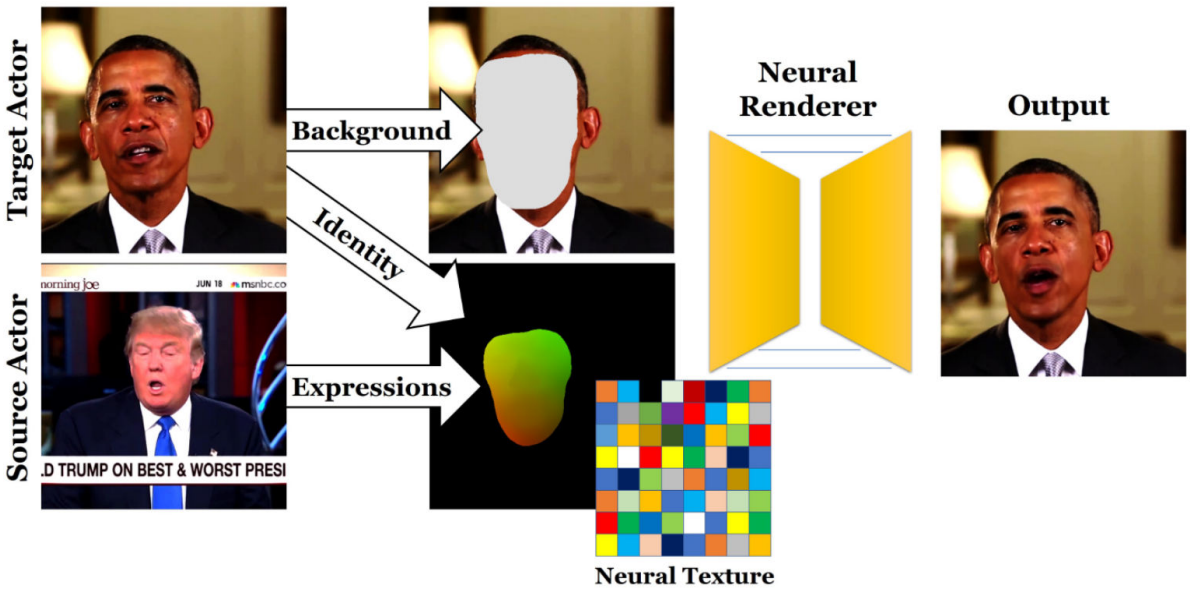


Figure 3.1: NeuralTextures deepfake generation method [Thies *et al.* 2019]

3.2 Deepfake Detection

Multiple techniques have been applied to the problem of detecting deepfakes. In addition to the deepfake dataset, the FaceForensics++ team also developed a deepfake classifier. The classifier was a frame-by-frame CNN model which performed binary classification on the extracted faces of each image. The base model used was XceptionNet with initial weights from a model trained on ImageNet [Chollet 2017]. Transfer learning was utilized by replacing the final layer with a fully connected layer and further training the model on the deepfake dataset. This model performed with an overall accuracy of 99.26% on the uncompressed dataset. The performance degraded to an accuracy of 81% when evaluated on the dataset with the most aggressive compression applied.

MesoNet is a CNN based deepfake detection network introduced by Afchar *et al.* [2018] that contains a few layers to allow the model to focus on the mesoscopic properties of the image. The model was also trained using the training subset of the FaceForencics++ dataset and evaluated on its test set. It performed with an accuracy of 95.23% on the uncompressed FaceForensics++ dataset and an accuracy of 70.47% on the compressed dataset.

CNNs only use the spatial data contained in an image. Deepfakes are most often crafted to create videos and contain artefacts between frames that can be noticeable. CNNs cannot capture the temporal information present in video data. 3D convolutional neural networks (3D-CNN) are an extension of the traditional CNNs that can capture spatiotemporal features by performing both 3D convolution operations and 3D pooling [Tran *et al.* 2015]. Wang and Dantcheva [2020] trained multiple 3D CNN models on the FaceForensics++ dataset and compared their performance with the traditional CNN models. The best performing model was the Two-Stream Inflated 3D ConvNet (I3D) [Carreira and Zisserman 2018]. It performed with an overall accuracy of 87.43% on the compressed dataset which is higher than the XceptionNet model which only achieved an accuracy of 81%.

3.3 Adversarial Deepfakes

Deepfake technology can be used by malicious attackers to spread misinformation. Deepfake models are currently being used to flag manipulated data across different social media platforms. A malicious attacker is therefore likely to try to bypass these detector models by perturbing the manipulated images to form an adversarial example. As described earlier, these perturbations can be extremely small and can be imperceptible. Neekhara *et al.* [2020] demonstrated that deepfake detectors are vulnerable to adversarial examples in both a white-box and black-box setting.

Both frame-by-frame CNN models and a 3D-CNN model were attacked and the effectiveness of the attack was measured. The frame-by-frame CNN models tested were based on the XceptionNet and MesoNet architectures. The XceptionNet model was able to perform with an accuracy of 99.26% on the full dataset and MesoNet performed with an accuracy of 95.23%. The iterative fast gradient sign method was used to construct

adversarial examples in a white-box setting. Furthermore, robust white-box attacks that are also effective when the image is compressed were also crafted. The black-box attacks were crafted using a method based off the Natural Evolutionary Strategies Method (NES) and also contain a robust method that is effective when compressed [Wierstra *et al.* 2011]. The l_∞ norm was used to measure the difference between the adversarial image and the benign image.

Table 3.1 shows a summary of the results presented by Neekhara *et al.* [2020]. XceptionNet and MesoNet are attacked using both white-box and black-box methods. The l_∞ norm and the overall binary accuracy of the classifier on the compressed adversarial examples (Acc-C%) were recorded. The white-box attack does reduce the performance of the models significantly even when the videos are compressed. The robust white-box attack is even more effective and reduces the accuracy of the models to 1.77% and 0.5% for the XceptionNet and MesoNet models respectively. The black-box attacks are overall less effective than the white-box attacks but are still effective in fooling the detectors. The robust black-box attack reduces the accuracy of the models to 16.06% and 21.67%.

Table 3.1: Adversarial Deepfake results summary

	White-box		***Robust White-box***		***Black-box***		***Robust Black-Box***	
Metric	l_∞	*Acc-C%*	l_∞	*Acc-C%*	l_∞	*Acc-C%*	l_∞	*Acc-C%*
XceptionNet	0.004	41.54	0.013	1.77	0.045	36.31	0.047	16.06
MesoNet	0.007	7.28	0.025	0.50	0.063	21.82	0.053	21.67

This paper demonstrated that the state-of-the-art deepfake detectors can be easily defeated by leveraging adversarial attacks in both a white-box and black-box setting. It is therefore vital to develop defences against these attacks to create robust deepfake detectors.

3.4 Adversarial Defenses

3.4.1 APE-GAN

Adversarial Perturbation Elimination with GAN (APE-GAN) is an adversarial defense technique that leverages the generator component in the GAN architecture to purify adversarial examples [Jin *et al.* 2019]. The method has been tested against white-box attacks on MNIST, CIFAR-10 and the ImageNet (Top 1) dataset and has been shown to be successful.

A GAN is trained using adversarial examples and benign examples. Figure 3.2 illustrates how the training of the GAN should be setup so that the GAN can learn how to eliminate the adversarial perturbations. X^{adv} is input into the generator which then produces $G(X^{adv})$ which is typically the X_{fake} variable in a normal GAN setup. $G(X^{adv})$ and X are then fed into the discriminator component which then predicts which is real and which is fake. Both the Discriminator and the Generator are trained as you would a typical GAN.

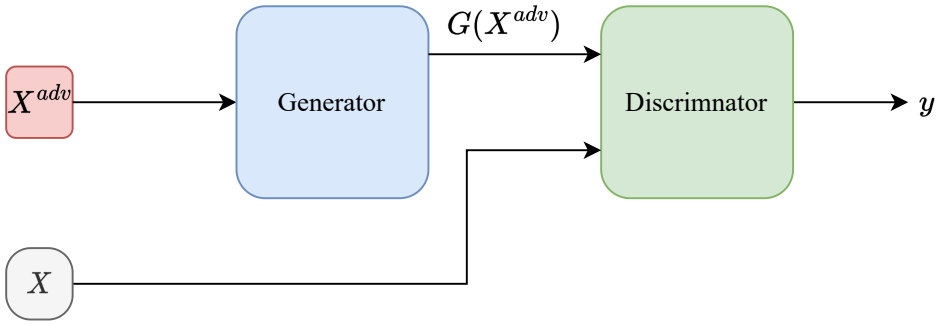


Figure 3.2: APE-GAN training architecture

The APE-GAN architecture aims to train a generator function, G, that can estimate the original clean counterpart, X, for a given adversarial input image, X^{adv}. This is achieved by optimising a generator network with weights and biases denoted as θ_G, using an adversarial perturbation elimination loss function, l_{ape}. The generator network G, parametrized by θ_G, is trained using the following objective in equation (3.1) where X^{adv_k} represents adversarial images to corresponding original clean images X_k for k = 1, ..., N.

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{k=1}^{N} l_{ape}(G_{\theta_G}(X^{\text{adv}_k}), X_k). \tag{3.1}$$

To complement the generator G, a discriminator network, D, is also trained to solve the adversarial zero-sum problem. The goal of the discriminator is to distinguish between reconstructed images $G(X^{adv})$ and original clean images. The discriminator D is trained to maximise the following objective shown in equation (3.2).

$$\max_{\theta_D} \min_{\theta_G} \quad \mathbb{E}_{X \sim p_{\text{data}}(X)}[\log D_{\theta_D}(X)] - \mathbb{E}_{X^{\text{adv}} \sim p_G(X^{\text{adv}})}[\log D_{\theta_D}(G_{\theta_G}(X^{\text{adv}}))]. \tag{3.2}$$

The loss function for the discriminator, l_d, is shown below in equation (3.3):

$$l_d = -\sum_{n=1}^{N} \left[\log D_{\theta_D}(X) + \log D_{\theta_D}(G_{\theta_G}(X^{\text{adv}}))\right]. \tag{3.3}$$

Equation (3.4) shows the generator loss function, l_{ape}, consists of two components: pixel-wise Mean Square Error (MSE) loss and adversarial loss.

$$l_{ape} = \xi_1 l_{mse} + \xi_2 l_{adv}. \tag{3.4}$$

Equation (3.5) describes the pixel-wise MSE loss l_{mse}, defined as the mean squared difference between the original clean image X and the reconstructed image.

$$l_{mse} = \frac{1}{W \cdot H} \sum_{x=1}^{W} \sum_{y=1}^{H} (X_{x,y} - G_{\theta_G}(X^{\text{adv}_{x,y}}))^2. \tag{3.5}$$

Equation (3.6) is the adversarial loss l_{adv}, calculated based on the probabilities of the discriminator D over all reconstructed images.

$$l_{adv} = \sum_{n=1}^{N} \left[1 - \log D_{\theta_D}(G_{\theta_G}(X^{\text{adv}_n}))\right]. \tag{3.6}$$

Overall, the APE-GAN architecture aims to train a generator that produces images without adversarial perturbations by utilising the adversarial zero-sum game between the generator and discriminator networks. The generator is optimised to minimise the weighted sum of pixel-wise MSE loss and adversarial loss, while the discriminator aims to maximise its ability to distinguish between the original clean images and the reconstructed images.

Once the GAN is trained, only the Generator network is required to purify adversarial examples and feed them into the target model. This overall inference pipeline is shown in Figure 3.3.

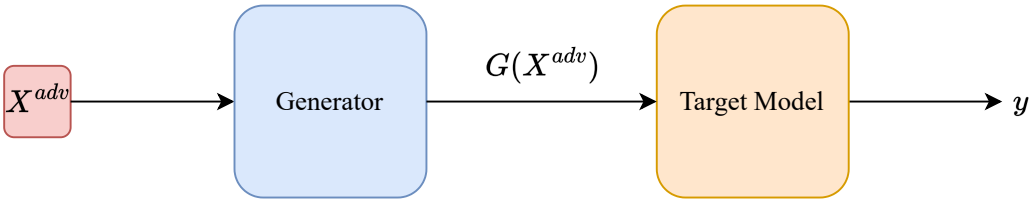


Figure 3.3: Pipeline combining APE-GAN and target model

Yang *et al.* [2021] introduces an improved implementation of APE-GAN named APE-GAN++. This method improves the performance of APE-GAN and the stability of training the GAN by utilising the WGAN-GP loss. This method was tested on multiple white-box adversarial attack methods on MNIST and CIFAR-10 datasets and performed better than the original APE-GAN method.

3.4.2 MagNet

MagNet is a defense strategy that was proposed by Meng and Chen [2017]. It is agnostic of the attack method used and does not affect the training of the target model. MagNet consists of two main components, one or more detector networks and reformer networks. The role of the detector network is to identify adversarial examples and benign examples. The reformer network modifies any adversarial input towards the manifold of benign examples. MagNet makes use of an autoencoder architecture for its detector. The reconstruction error is used to estimate the distance between an input example and the data manifold of the benign dataset. It is trained to reproduce the training dataset with minimal reconstruction error whilst still being able to perform

well on the validation dataset. If the reconstruction error is greater than a determined threshold, it is classified as an adversarial example.

The detector network does not need to be used if the reformer is trained to perform well on benign input images in addition to adversarial input. The reformer network proposed in the paper also uses an autoencoder model architecture. The reformer autoencoder calculates the reconstruction loss between the reconstructed image and the benign image as shown in the architecture diagram of the reformer network in Figure 3.4. The loss used is the mean-squared-error loss and is shown in equation (3.7), This new image is then fed into the target model. The MagNet research indicated that it performs well against black-box and gray-box attacks but can be defeated in a white-box setting.

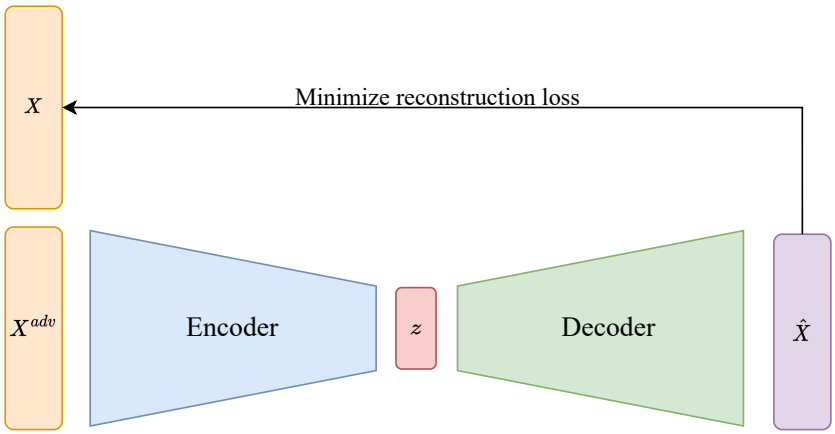


Figure 3.4: MagNet reformer training architecture

$$l_{mse} = \frac{1}{N}\sum_{i=1}^{N}\|X_i - \hat{X}_i\|^2. \tag{3.7}$$

Chapter 4

Adversarial Defense Workflow

This chapter provides a description of the adversarial defense workflow in the context of image classification. It covers obtaining the datasets, victim model, crafting the adversarial examples and how to evaluate an adversarial defense. It also details important terminology that will be used throughout the next chapters.

4.1 Dataset

The first step in the adversarial defense workflow is obtaining a dataset. The dataset serves as the foundation for training and evaluating the victim model. When selecting a dataset, it is essential to consider its size, diversity, and relevance to the problem domain. The dataset should be representative of the real-world scenarios the model will encounter, ensuring that the model learns patterns and features that generalise well.

To avoid data leakage and ensure fair evaluation, the dataset should be split into separate sets: a training set, a validation set, and a test set. The training set is used to train the victim model, while the validation set is used to tune hyperparameters and make decisions during the training process. The test set remains untouched during training and is solely used to evaluate the model's performance after training.

It is important to maintain clear separation between the training and test data to prevent the model from memorising specific examples from the test set, as this could lead to overfitting and misleading evaluation results.

4.2 Victim Model

The Convolutional Neural Network (CNN) architecture is a common and effective choice for image classification tasks. CNNs are particularly well-suited for tasks involving image data due to their ability to learn hierarchical features and patterns. The victim model is trained on the chosen dataset using the backpropagation algorithm, where it adjusts its weights and biases to minimise a predefined loss function.

The term “victim model” is used because this model is the target of adversarial attacks. Adversarial attacks aim to find inputs that can cause the victim model to misclassify them. These inputs are referred to as adversarial examples. Despite being robust on normal data, the victim model is susceptible to making incorrect predictions on these adversarial examples, which raises concerns about the model’s reliability and security in real-world applications.

The goal of the adversarial defense workflow is to improve the robustness of the victim model against such attacks without compromising its accuracy on regular data. By implementing effective adversarial defense mechanisms, we aim to enhance the model’s resilience and make it more reliable in critical applications where security and trustworthiness are paramount.

4.3 Crafting Adversarial Dataset

Adversarial examples are carefully crafted inputs that are designed to cause the victim model to make incorrect predictions. These examples are created by applying imperceptible perturbations to the original data, which can lead to significant changes in the model’s outputs. In contrast to adversarial examples, the examples found in the original dataset which do not contain any perturbations are called benign examples.

Crafting an adversarial dataset involves generating a set of adversarial examples that can be used to evaluate the robustness of the victim model and test the effectiveness of various defense mechanisms. In order to craft an adversarial example an attack method should be chosen such as FGSM, PGD, or any other specific technique based on the model and the level of adversarial robustness desired.

Using the selected attack method, adversarial examples are generated by adding small, calculated perturbations to the original benign examples. The goal is to make these adversarial examples as similar to the original examples as possible, while causing the model to misclassify them.

The magnitude of the perturbations plays a crucial role in the effectiveness of the adversarial examples. Small perturbations may lead to low success rates of attacks, while large perturbations might become noticeable to human observers, defeating the purpose of adversarial attacks.

4.4 Training Defense Model

Training a defense model involves developing a new model to improve the victim model’s robustness against adversarial attacks. The goal is to create a defense model that can purify the input such that the victim model can accurately classify both clean and adversarial examples.

The defense model is often trained on both the benign and adversarial dataset such that it can generalise and purify adversarial noise and not alter the input image if there is no adversarial noise present. By incorporating adversarial examples into the

training data, the defense model becomes better equipped to recognise and mitigate such perturbations during inference.

4.5 Evaluation of Defense Model

Evaluating the defense model is a critical step to determine its effectiveness in mitigating adversarial attacks. The evaluation process involves testing the defense model against various types of adversarial attacks and measuring its performance using relevant metrics. The defense models chosen in this report purify the input images for the downstream victim model. In order to evaluate the defense method, the performance of the downstream classifier on the purified images is determined.

Classifying deepfake images is a binary classification problem. The overall classification accuracy will be calculated using Equation (4.1), where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative. This metric does not provide complete insight into the model's performance. Additional metrics such as the precision shown in Equation (4.2), recall shown in Equation (4.3) and the F1-score in Equation (4.4) will also be calculated.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}. \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP}. \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN}. \tag{4.3}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{4.4}$$

The evaluation metrics should only be calculated on the test subset to ensure there was no data leakage. It is important to keep these splits consistent for both the victim model and the defense model. The victim model's accuracy should be determined on the benign and adversarial dataset to form a baseline for comparison. Subsequently the defense model should be used to purify both the adversarial and benign dataset to determine the performance of the model. The defense model should be able to purify the adversarial perturbations and improve the victim model's accuracy on the adversarial dataset. The defense model should also not alter benign images such that the victim model performs poorly on the this dataset and therefore it is important to determine the benign accuracy of the defense technique used. In addition to accuracy the precision and recall metrics should also be recorded to fully evaluate the defense model.

Chapter 5

Datasets and Victim Models

This chapter presents the various datasets and victim models used throughout the research. It describes the details of the three datasets, MNIST, CIFAR-10 and the FaceForensics++. The victim models chosen for each of the datasets are also provided along with the adversarial datasets that are crafted by attacking the victim models. White-box attacks are more effective and efficient than a black-box attacks and can be considered a worse-case scenario. For this reason, only a white-box attack was evaluated in this research.

5.1 MNIST

5.1.1 Dataset

The MNIST dataset is a classic machine learning dataset that has been used extensively in the context of adversarial attack and defense methods [LeCun *et al.* 2010]. It contains grayscale images of handwritten digits with a resolution of 28x28. Figure 5.1 shows an example for each hand written digit for each of the ten possible digits. The dataset contains 60 000 training examples and 10 000 test examples.

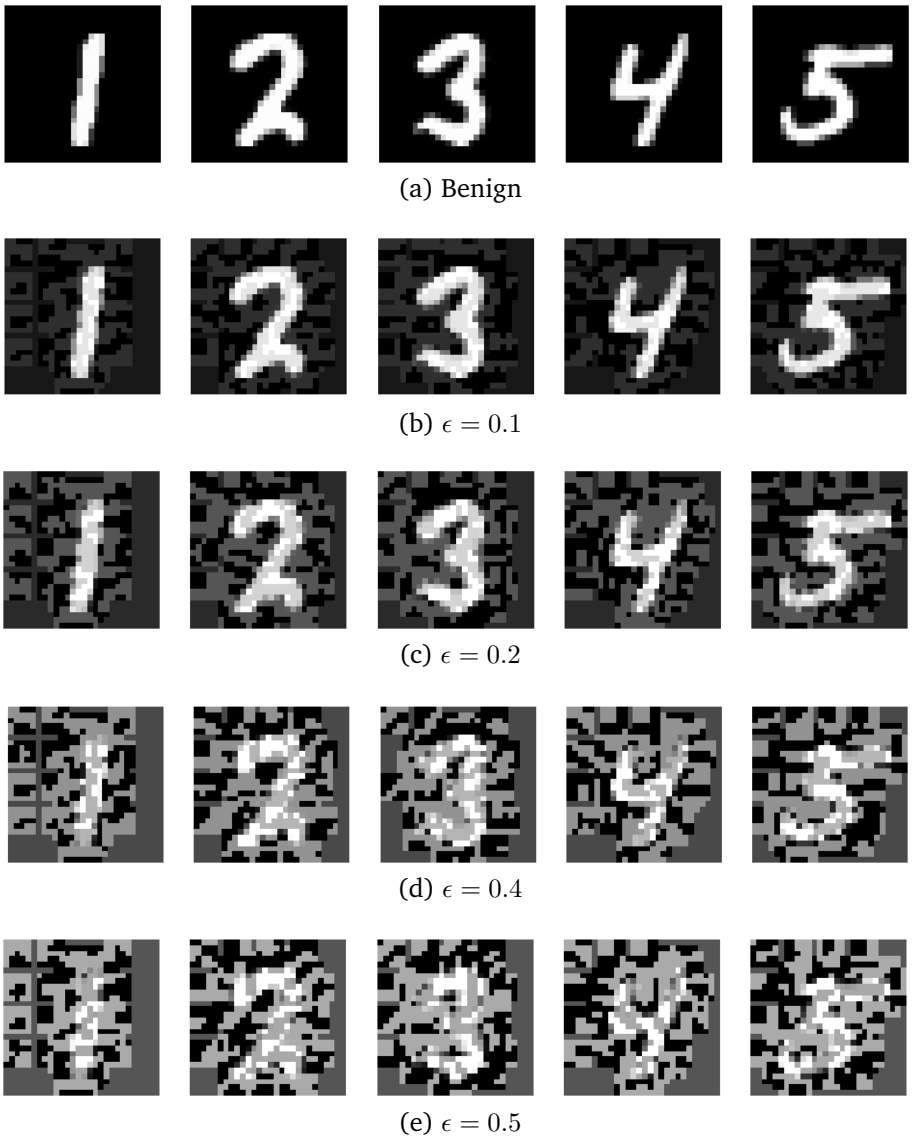
Figure 5.1: MNIST examples for each class 0-9

5.1.2 Victim Model

A simple CNN model was trained on the MNIST dataset, it contained 2 convolutional layers and 2 fully connected layers. It performed with an accuracy of 98.5% on the benign MNIST test dataset. This model is the victim model for the evaluation on the MNIST dataset.

5.1.3 Adversarial Dataset

The fast gradient sign method was used to attack the victim model using varying levels of ϵ. Figure 5.2 contains a few sample images from the benign dataset and the adversarial datasets that have been attacked with various levels of ϵ.



(a) Benign

(b) $\epsilon = 0.1$

(c) $\epsilon = 0.2$

(d) $\epsilon = 0.4$

(e) $\epsilon = 0.5$

Figure 5.2: MNIST FGSM sample images $\epsilon = 0.1$ to $\epsilon = 0.5$

Table 5.1 details the performance of the CNN classifier on the benign and adversarial dataset. The performance of the victim model begins to drop significantly when the

ϵ value is set to 0.2. The larger, the value of ϵ, the more aggressive the attack. The accuracy of the victim model drops below 5% when the ϵ value is set to 0.4 and higher, indicating that the more aggressive attacks impede the model more significantly.

Table 5.1: Basic CNN MNIST accuracies for benign and adversarial (FGSM)

ϵ	**Benign**	**Adversarial**
0.1	98.50	72.92
0.2	98.50	25.52
0.4	98.50	4.47
0.5	98.50	2.93

5.2 CIFAR-10

5.2.1 Dataset

The CIFAR-10 dataset is a another widely used benchmark dataset for image classification tasks in the field of computer vision [Krizhevsky *et al.* 2009]. The dataset consists of 60 000 colour images, each with a size of 32x32 pixels, covering 10 different classes. Figure 5.3 contains a single image example from each of the ten classes contained in the dataset.

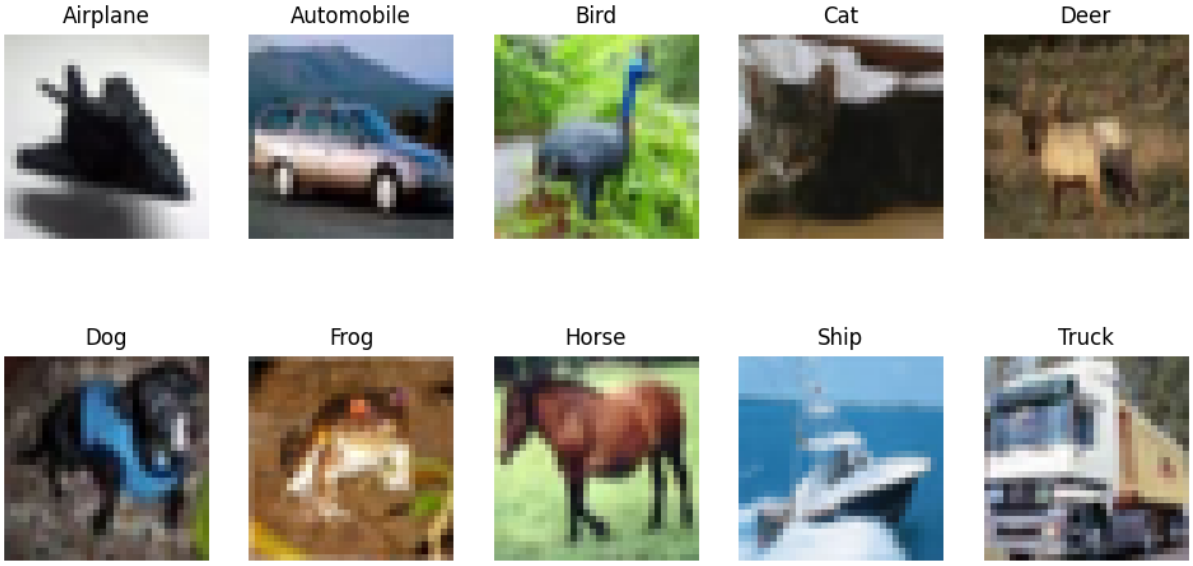


Figure 5.3: CIFAR-10 examples for each class

5.2.2 Victim Model

A CNN model that contains four convolutional layers, batch normalisation and max pooling was implemented and trained on the CIFAR-10 dataset. The model was able to achieve an accuracy of 84.20% on the CIFAR-10 test dataset.

5.2.3 Adversarial Dataset

The model was attacked with the fast gradient sign method with varying ϵ values. Table 5.2 contains the accuracy of the model at the different ϵ values. The model's performance drops down to 10% when $\epsilon = 0.01$ and close to 0% when the ϵ value is increased to 0.05 and 0.1.

Table 5.2: Basic CNN CIFAR-10 accuracies for benign and adversarial (FGSM)

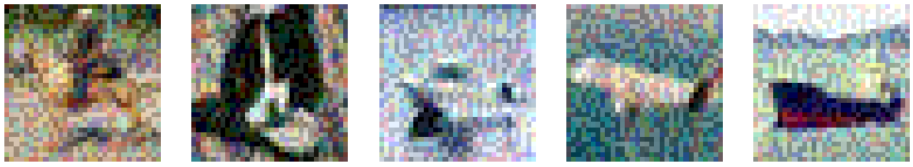
ϵ	**Benign**	**Adversarial**
0.01	84.20	9.54
0.05	84.20	0.64
0.1	84.20	2.45

Figure 5.4 contains a few sample images from the benign dataset and the adversarial datasets that have been attacked with various levels of ϵ.

(a) Benign

(b) $\epsilon = 0.01$

(c) $\epsilon = 0.05$



(d) $\epsilon = 0.1$

Figure 5.4: CIFAR-10 FGSM sample images $\epsilon = 0.01$ to $\epsilon = 0.1$

Each class contains 6,000 images, and the dataset is evenly balanced, meaning that each class has the same number of images. The images are colour images, meaning that they have three colour channels (Red, Green, and Blue), providing more detailed information compared to grayscale images. This dataset is slightly higher resolution

than the MNIST dataset but also contains the three colour channels making more complex.

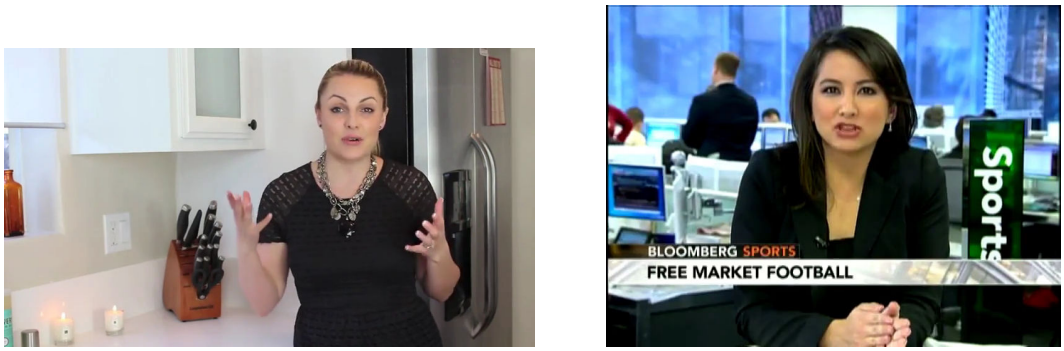
5.3 FaceForensics++

5.3.1 Dataset

The FaceForensics++ dataset contains 1000 videos obtained from YouTube which contain mainly frontal unobscured shots of the person. This was done to ensure successful deepfake generation from the techniques used. The train-test split indices for the dataset are provided by the original paper and contain 860 videos that belong to the training dataset and 140 videos that belong to the test set. It is important to run these experiments on the same indices to insure there is no data leakage between the train and test set. It also means that the experiments can be repeated and evaluated on identical datasets.

The deepfake generation methods all need to be provided with a source video and a target video. The techniques either use face replacement techniques, where the face from the source video is transferred to the target video or they use face reenactment where the expression from the actor in the source video is applied to the target video. All methods first crop the face section of the frame and apply the transformation to that block and stitch it back to the original video.

Figure 5.5 shows a single frame from the source video and the target video. Figure 5.6 contains frames from the generated deepfakes using the four different methods, namely, DeepFakes, Face2Face, FaceSwap and NeuralTextures.



(a) Source video snapshot (b) Target video snapshot

Figure 5.5: Source and target video snapshots from real test set

5.3.2 Victim Model

Deepfake detection methods for videos can use a variety of different techniques. Some methods use multiple frames of the video to detect temporal features whilst others only evaluate the spatial features on a per-frame basis. The chosen victim model is a frame-by-frame CNN based model that only evaluates the face crop from each frame of the

(a) DeepFakes (b) Face2Face

(c) FaceSwap (d) NeuralTextures

Figure 5.6: FaceForensics++ fake video snapshots from test set showing the DeepFakes, Face2Face, FaceSwap and NeuralTextures method

video. Each frame is classified as real or as fake. The detection pipeline overview is shown in Figure 5.7.

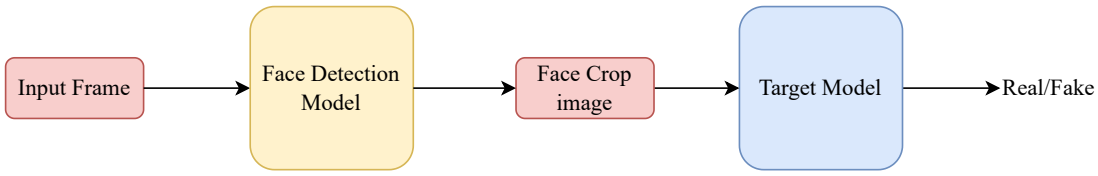
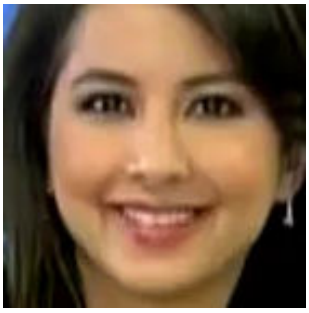


Figure 5.7: Deepfake classifier detection pipeline

The implementation provided by Neekhara *et al.* [2020] makes use of a frontal face detection model from the open-source dlib library [King 2017]. The detection model uses a Histogram of Oriented Gradients (HOG) descriptor with a Support Vector Machine (SVM) to provide the prediction. One thing to note is that this model is not invariant to face rotation and is only successful with frontal facial images. This is suitable for the FaceForensics++ dataset as the dataset has been curated to only contain videos with people looking head-on at the camera. This was also chosen as the inference time of this model is quicker than CNN based approaches. Figure 5.8 illustrates how the face detection model detects a face and the region around it successfully to enable cropping of just the person in the frame.

(a) Real frame with face crop region



(b) Cropped facial image

Figure 5.8: Real frame with face detection model applied

The cropped facial image is required to be classified as real or fake. The FaceForensics++ paper provided a CNN method that is based on the XceptionNet architecture which uses depthwise separable convolutional layers [Chollet 2017]. The implementation makes use of an XceptionNet model that has been pretrained on ImageNet. Transfer learning was then utilised to train the deepfake detector by swapping out the final layer of the pretrained network to predict real or fake. Figure 5.9 provides a detailed architecture diagram.

Pre-trained model weights for the XceptionNet based deepfake detector pipeline were obtained from Neekhara *et al.* [2020]. Figure 5.10 shows the full model pipeline working successfully in identifying a real frame from a video and correctly identifying a

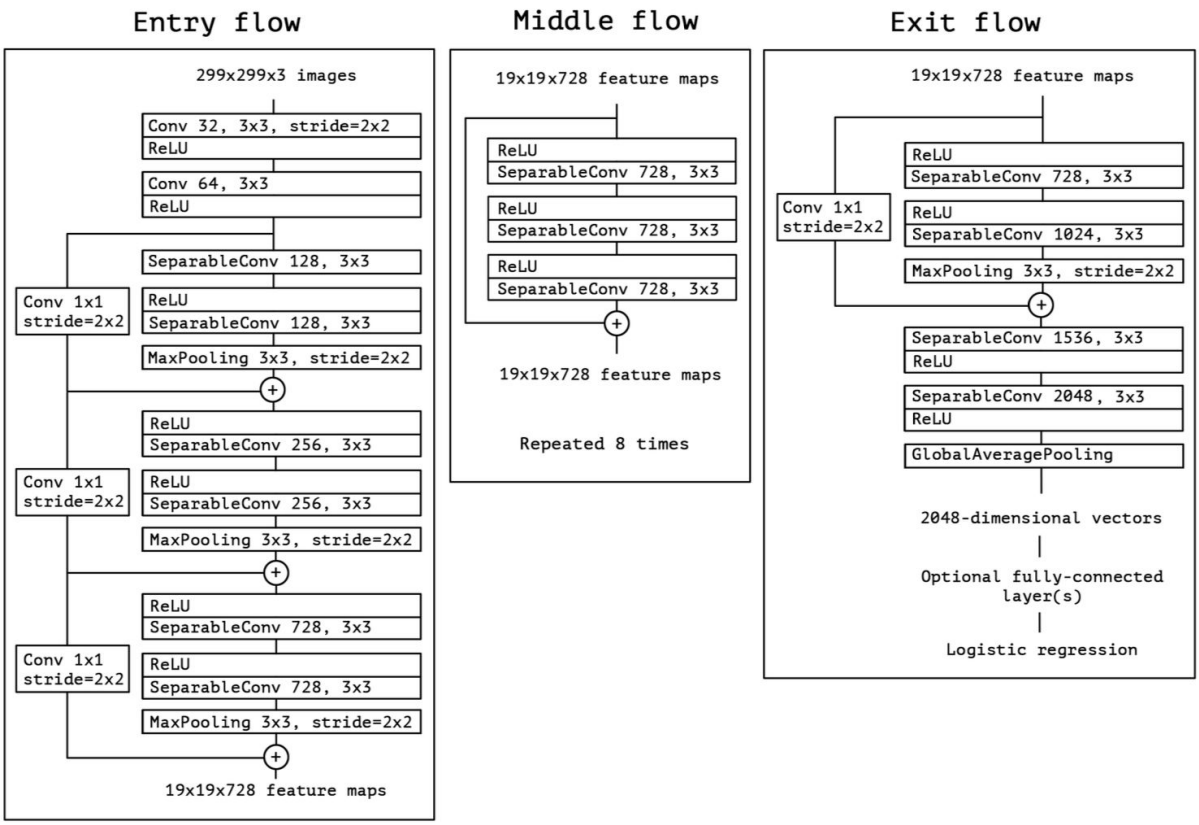
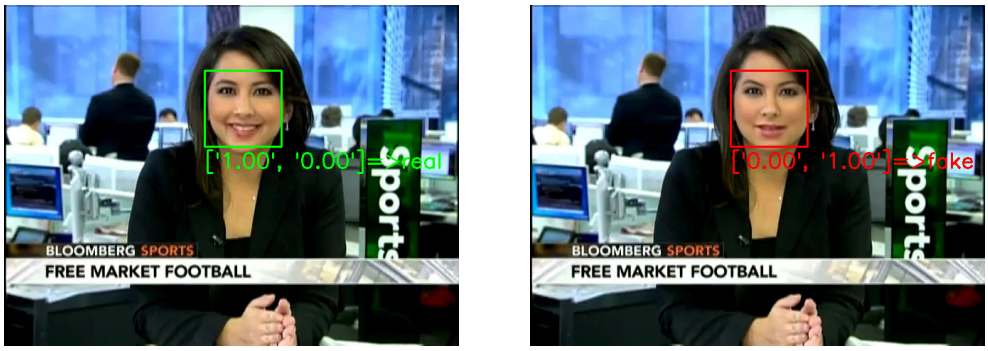


Figure 5.9: XceptionNet CNN architecture [Chollet 2017]

deepfake frame from the manipulated video. The green box indicates that the frame has been classified as a real frame and the red box indicates it has been classified as a fake frame. The array values shown in the image indicate the model's prediction probability for each of the classes, real and fake.



(a) Prediction on real frame (b) Prediction on fake frame

Figure 5.10: Model prediction pipeline example

The deepfake detection pipeline was run using the weights from the model provided by Neekhara *et al.* [2020] and the performance is presented in Table 5.3. The accuracies are calculated for all the frames of the videos in the FaceForensics++ test set. The accuracy of the model on the real videos is 83.44% and over 93% for the DeepFakes, Face2Face and FaceSwap subsets of the FaceForensics++ dataset. The model only performs with an accuracy of 31.75% on the NeuralTextures subset which is extremely low. The DeepFakes, Face2Face and FaceSwap accuracies are comparable to those obtained by Neekhara *et al.* [2020] but the NeuralTextures performance was much lower. This does mean that the model is unusable on the NeuralTextures generation method but will still be included in this research for completeness. The cause for this is unfortunately unknown.

Table 5.3: XceptionNet deepfake detector accuracies on benign FaceForensics++ dataset

	Real	**DF**	**F2F**	**FS**	**NT**
Accuracy (%)	83.44	96.94	96.23	93.79	31.75

5.3.3 Adversarial Dataset

The XceptionNet Deepfake detector was attacked using the iterative fast gradient sign method. This attack method is parameterised by ϵ, α, and the max number of iterations to perform. A larger ϵ value results in a more aggressive attack that introduces more adversarial noise. The number of maximum iterations to perform was set to 100 and a fixed step size (α) of 1/255 was used. The I-FSGM method was applied to all subsets in the FaceForensics++ dataset including the real videos from the dataset. Varying levels of ϵ were evaluated and the l_∞ norm average values were calculated and included in the table. This measures the difference between adversarial and benign images. The

higher the level of ϵ, the more aggressive the attack is and the greater the l_∞ norm. These results are contained Table 5.4.

Table 5.4: XceptionNet adversarial attack results for varying levels of ϵ

	Real	**DF**	**F2F**	**FS**	**NT**	l_∞
Benign Acc(%)	83.44	96.94	96.23	93.79	31.75	-
I-FGSM-0.1 Acc(%)	0.00	0.00	0.00	0.00	0.00	0.0119
I-FGSM-0.01 Acc(%)	0.00	0.00	0.00	0.00	0.00	0.0097
I-FGSM-0.001 Acc(%)	1.19	36.55	4.6	5.71	1.79	0.0010
I-FGSM-0.0001 Acc(%)	1.02	22.69	5.02	4.83	0.00	0.0001

The adversarial attack managed to completely fool the detector on every frame when the ϵ value was set to 0.1 and 0.01 as the accuracies for the real videos and deepfake methods all dropped to 0. The attack became slightly less successful when the ϵ value is set to 0.001, however, it was still making the detector perform with an accuracy close to zero. Only once the ϵ value was set to 0.0001 does the accuracy for some of the deepfake methods rise to approximately 5% whilst the model still performs with an accuracy of 22.69% on the DeepFakes method.

In addition to the accuracy results, Table 5.5 contains evaluation metrics which detail the precision, recall and F1-score for the victim model on the benign FaceForensics++ dataset. These results have excluded the NeuralTextures approach due to the poor performance and would skew these results unfairly. The model performed with a weighted F1-score of 0.93.

Table 5.5: XceptionNet Deepfake detector evaluation metrics

	Precision	**Recall**	**F1-score**
Real	0.87	0.83	0.85
Fake	0.94	0.96	0.95
Weighted	0.92	0.93	0.93

Figure 5.11 shows the same frame and the model prediction on the benign image and the adversarial image. These images look identical to the human eye but the model classifies them differently. The benign image is correctly identified as a real frame, whilst the model predicts the adversarial image incorrectly as a fake. It is just as problematic for a deepfake detector to detect real frames incorrectly as this could also lead to misinformation.

Figures 5.12 to 5.14 and 5.17 all show the deepfake methods and their respective classification. The benign deepfake frames get correctly identified apart from the NeuralTextures technique as the model performs poorly on this method. The adversarial frames for all methods show that through a targeted white-box attack, the classifier fails to predict correctly.

Figure 5.15 contains 5 sample real images that have been attacked using the iterative fast gradient sign method at the two levels, $\epsilon = 0.0001$ and $\epsilon = 0.01$. The deepfake

(a) Prediction of real video frame

(b) Prediction of adversarial real video frame

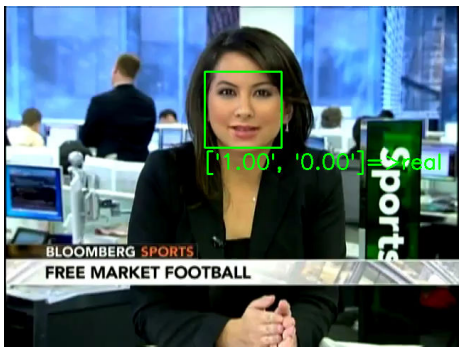
Figure 5.11: Real video prediction and Adversarial real video prediction

(a) Prediction of DeepFakes video frame

(b) Prediction of adversarial DeepFakes video frame

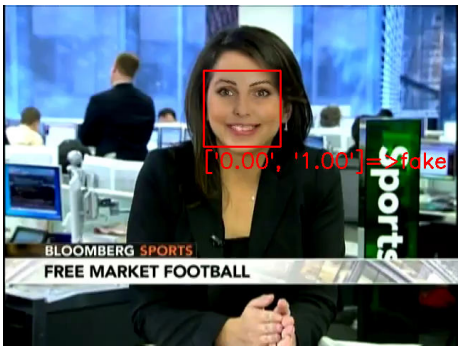
Figure 5.12: DeepFakes video prediction and adversarial DeepFakes video prediction

(a) Prediction of Face2Face video frame



(b) Prediction of adversarial Face2Face video frame

Figure 5.13: Face2Face video prediction and adversarial Face2Face video prediction



(a) Prediction of FaceSwap video frame

(b) Prediction of adversarial FaceSwap video frame

Figure 5.14: FaceSwap video prediction and adversarial FaceSwap video prediction

classifier performs with an accuracy of 1% and 0% on the respective datasets. The performance difference is slight and the adversarial noise is undetectable in both sets of images.

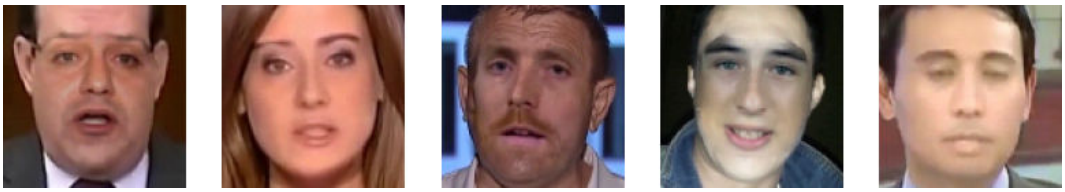
(a) Real $\epsilon = 0.0001$

(b) Real $\epsilon = 0.01$

Figure 5.15: Comparing adversarial attacks on real images

Figure 5.16 contains 5 sample fake images that have been attacked using the iterative fast gradient sign method at the two levels, $\epsilon = 0.0001$ and $\epsilon = 0.01$. The deepfake classifier performs with an accuracy of 22.69% and 0% on the respective FaceForensics++ (DeepFakes) subset. The performance difference is much larger but the difference in adversarial noise is still undetectable. This indicates that a small amount of noise can fool the underlying victim model.

(a) Fake $\epsilon = 0.0001$



(b) Fake $\epsilon = 0.01$

Figure 5.16: Comparing adversarial attacks on fake images

(a) Prediction of NeuralTextures video frame

(b) Prediction of adversarial NeuralTextures video frame

Figure 5.17: NeuralTextures video prediction and adversarial NeuralTextures video prediction

Chapter 6

Results and Analysis

This chapter evaluates the efficacy of the APE-GAN and MagNet adversarial defenses. The approach is first tested on the MNIST and CIFAR-10 datasets. It is then tested in the context of deepfake detection. Each defense method is tested at various levels of white-box attacks to determine the effectiveness. In addition to this, the defense models were used to reconstruct the benign dataset to determine if the models could be applied to both benign and adversarial input. All results provided in this chapter are calculated based on the test split from the datasets. The weights for all models are randomly initialised to ensure that the results are reproducible. To further ensure this, all experiment results are taken as an average of 10 independently trained models with the same hyper-parameters.

6.1 MNIST

6.1.1 APE-GAN

The APE-GAN model makes use of a DCGAN architecture and also uses the Wassertien loss and gradient penalty for calculating the generator loss. An APE-GAN defense model was trained on each of the the adversarial datasets at the varying levels of ϵ and the results presented in this section.

Figure 6.1 shows a few example images of how APE-GAN works at removing adversarial noise when the attack ϵ is set to 0.1. The first row in the image shows the benign MNIST images, the second row shows the handwritten numbers with the adversarial noise applied to them and lastly, the third row shows what the digits look like after the noise has been removed by the generator of the GAN. These plots have been made to get a sense of how APE-GAN is working.

Figure 6.2 shows the same plot as before but with $\epsilon = 0.2$. The attack introduces much more noticeable noise but the model is still able to purify it and reproduce the digit with most of its integrity still in place. Figure 6.3 shows some example MNIST images when an $\epsilon = 0.4$ has been applied to the dataset. The attack is a lot more aggressive and the adversarial images are almost unrecognisable to the naked eye. APE-GAN is

still able to remove most of the noise and reproduce images that look very similar to the original digit. Figure 6.4 shows what the images look like when an $\epsilon = 0.5$ is applied. The adversarial image contains an immense amount of noise. APE-GAN is still able to purify the adversarial noise and reconstruct the digits to some extent.

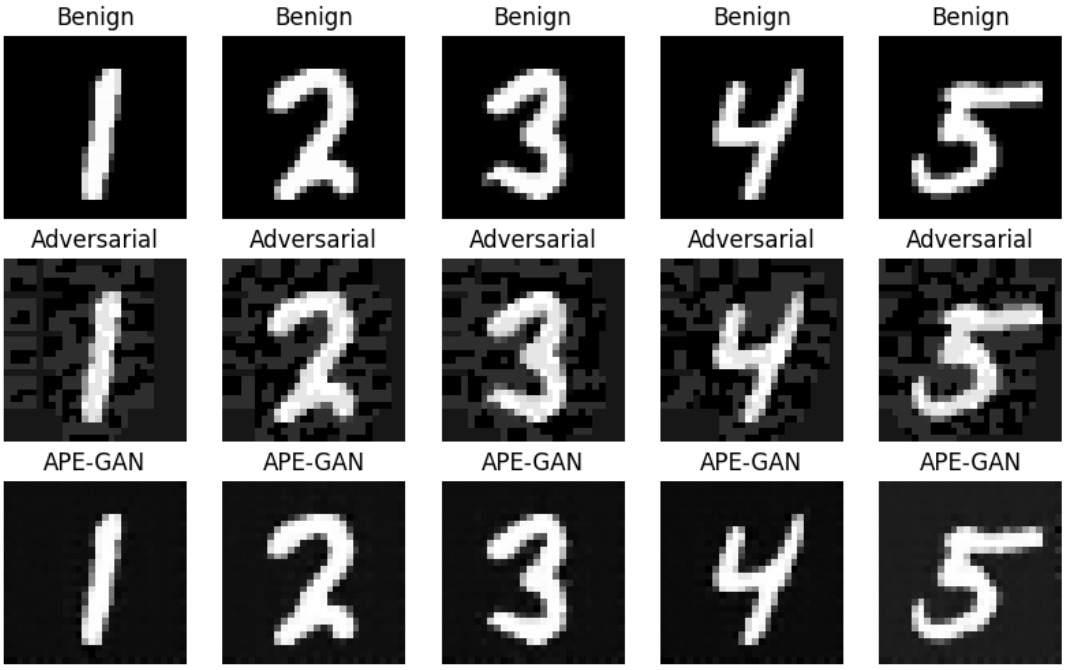


Figure 6.1: MNIST Benign, Adversarial (FGSM $\epsilon = 0.1$), APE-GAN Purified Adversarial

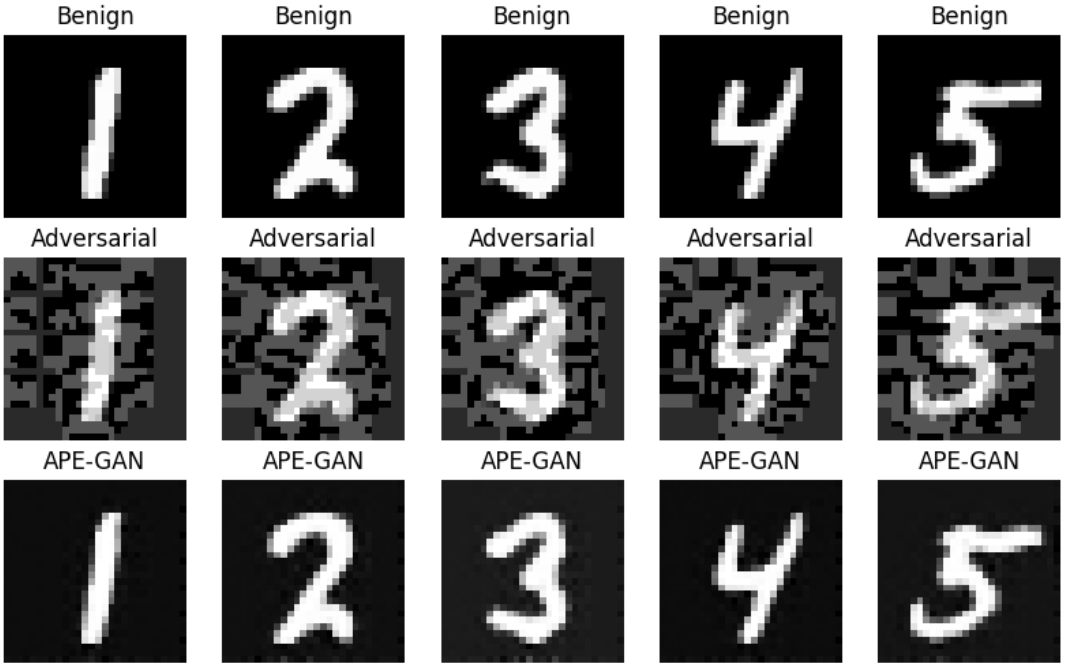


Figure 6.2: MNIST Benign, Adversarial (FGSM $\epsilon = 0.2$), APE-GAN Purified Adversarial

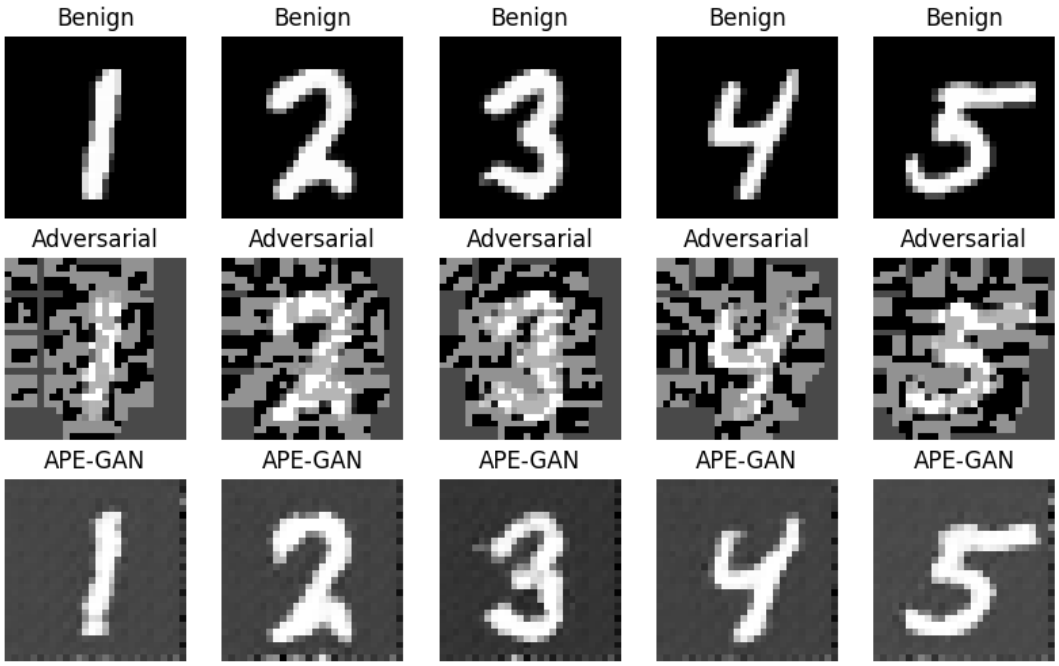


Figure 6.3: MNIST Benign, Adversarial (FGSM $\epsilon = 0.4$), APE-GAN Purified Adversarial

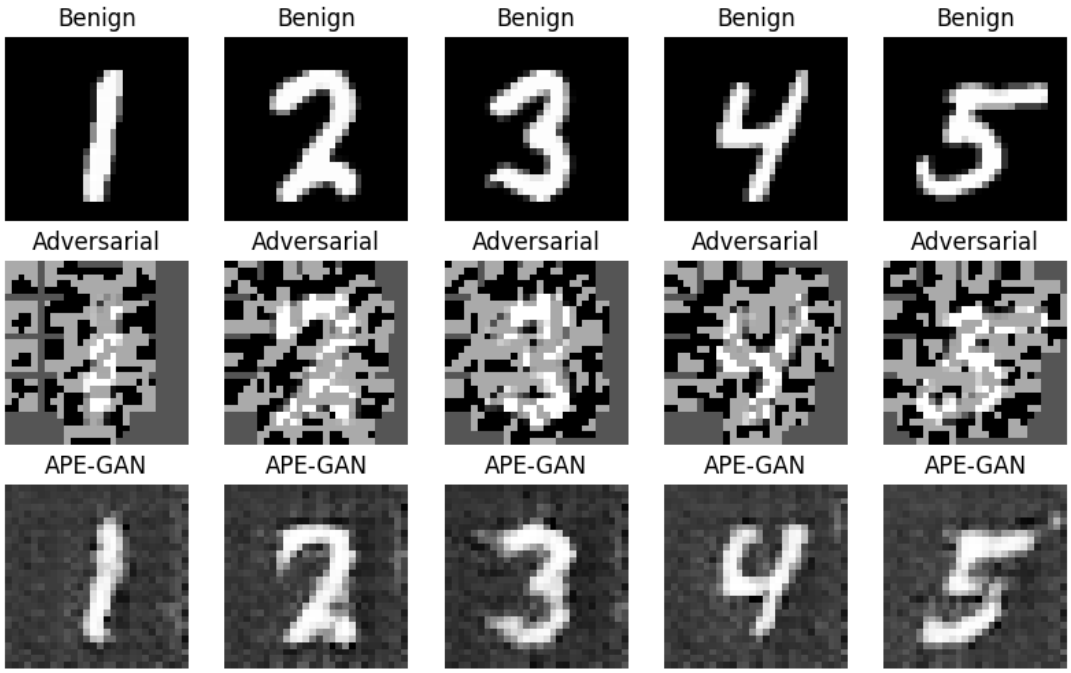


Figure 6.4: MNIST Benign, Adversarial (FGSM $\epsilon = 0.5$), APE-GAN Purified Adversarial

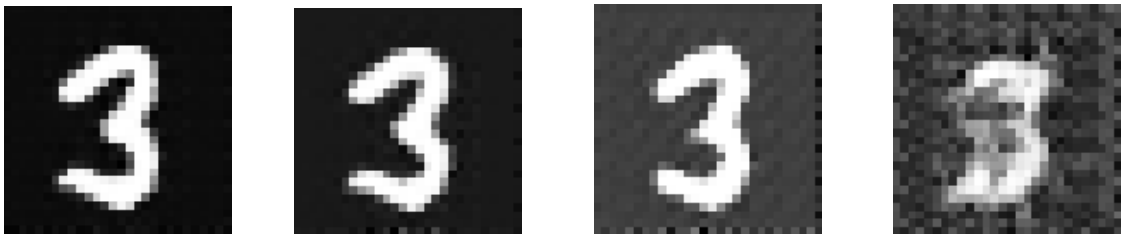
Table 6.1 contains the performance of the APE-GAN technique when applied to the adversarial examples and also when applied to the benign images. If it can reconstruct both benign and adversarial examples, APE-GAN can be integrated seamlessly with the classification pipeline. If APE-GAN only performs well on adversarial images, an additional component that determines if the input image is adversarial or not will need to be integrated.

Table 6.1: MNIST APE-GAN accuracy results for the various levels of ϵ

ϵ	**Benign**	**Adversarial**	**APE (Adversarial)**	**APE (Benign)**
0.1	98.50	72.92	97.54	98.75
0.2	98.50	25.52	97.62	98.72
0.4	98.50	4.47	94.24	97.62
0.5	98.50	2.93	94.73	54.68

APE-GAN performs with accuracies above 97.5% which is close to the original accuracy when the ϵ value is 0.1 and 0.2. The performance drops slightly when the model is applied to the adversarial attacks where the ϵ values have been set higher at 0.4 and 0.5. The underlying classifier still performs well with accuracies of 94.24% and 94.73%.

Figure 6.5 shows four benign images that have been reconstructed using the respective APE-GAN models that were trained on the varying ϵ values. The model is able to still reconstruct benign images successfully up to an ϵ value of 0.4.



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.2$ (c) $\epsilon = 0.4$ (d) $\epsilon = 0.5$

Figure 6.5: Benign APE-GAN MNIST reconstructed images

The model does however fail when ϵ is set to 0.5 and applied to the benign images. This is because the model is required to change the input images so drastically that it starts to hinder the performance on the benign images. This is shown in Figure 6.4 where the adversarial images get extremely distorted. An ϵ value of 0.5 is aggressive and is noticeable to a human.

APE-GAN successfully purifies adversarial examples and maintains the integrity of reconstructed benign images for all the attacks except for when the ϵ is set to 0.5. It is still considered successful because the most aggressive attack introduces extreme artefacts that are easily noticeable.

6.1.2 MagNet

The MagNet reformer model was implemented as in the original MagNet paper [Meng and Chen 2017]. The architecture of the autoencoder neural network is shown in Table 6.2. All the Conv2D layers except the last in the model use 3 input channels, 3 output channels and a kernel size of 3 with a stride and padding of 1. The last convolutional layer differs by using an output channel number of 1. The following hyperparameters were used as in the original paper: a learning rate of 0.001, the Adam optimiser, a batch size of 256 and the model was trained for a total of 100 epochs.

Table 6.2: MagNet MNIST architecture

MagNet Reformer Architecture	
Conv2D Sigmoid	$3 \times 3 \times 3$
Average Pooling	2×2
Conv2D Sigmoid	$3 \times 3 \times 3$
Conv2D Sigmoid	$3 \times 3 \times 3$
Upsampling	2×2
Conv2D Sigmoid	$3 \times 3 \times 3$
Conv2D Sigmoid	$3 \times 3 \times 1$

The reformer MagNet models were all trained on the relevant adversarial dataset with varying ϵ values. The benign, adversarial and purified images were plotted for visual inspection and are shown in Figures 6.6 to 6.9. MagNet is able to reform the adversarial images when ϵ is 0.1 and 0.2. The model begins to struggle to reconstruct the images with the higher ϵ values due to the large increase of perturbations contained in the image.

The performance of the defense method on the MNIST dataset is shown in Table 6.3. The MagNet reformer model performs well when ϵ is 0.1 and 0.2. The underlying classifier accuracy increases up to 95.80% and 93.93% respectively on the adversarial test dataset. The model was also applied to the benign input and the accuracy of the underlying classifier increased slightly to 98.65% and 98.58%.

Table 6.3: MNIST MagNet accuracy results for the various levels of ϵ

ϵ	**Benign**	**Adversarial**	**MagNet (Adversarial)**	**MagNet (Benign)**
0.1	98.50	72.92	95.80	98.65
0.2	98.50	25.52	93.93	98.58
0.4	98.50	4.47	82.51	98.33
0.5	98.50	2.93	47.73	96.86

MagNet still performed decently when ϵ was set to 0.4 as the accuracy of the detector went up from 4.47% to 82.51%. This is however much lower than the original model classifier. At this level of ϵ and when ϵ is 0.5 the reconstructed images from the benign images resulted in the model maintaining accuracies of 96.83% and 96.86%. MagNet does however fail at reconstructing the images well when $\epsilon = 0.5$ as the accuracy drops

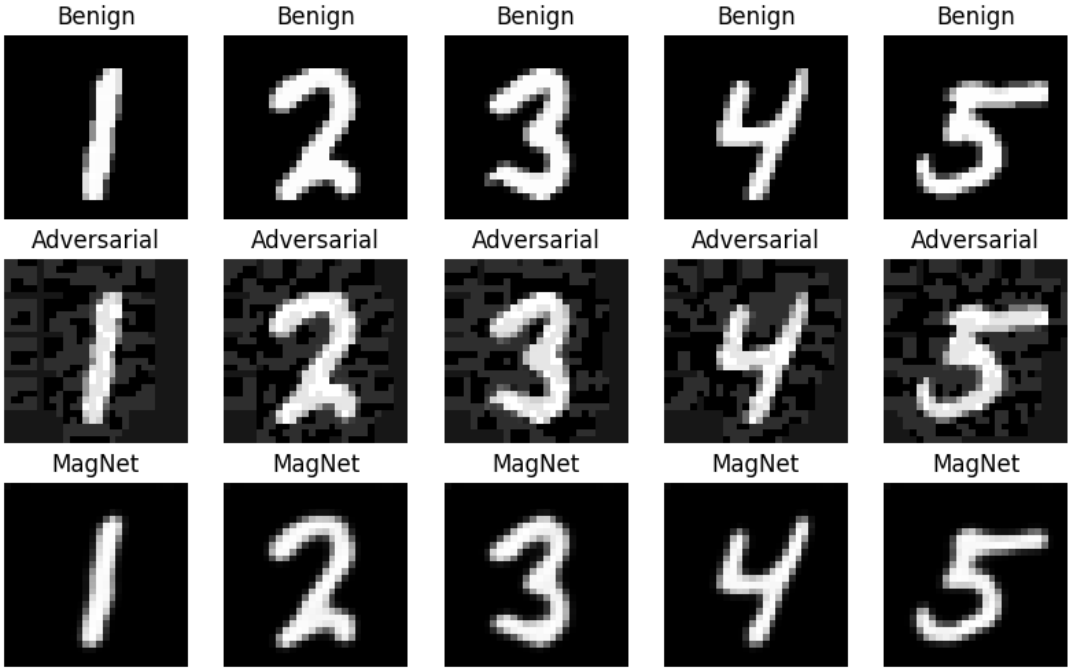


Figure 6.6: MNIST Benign, Adversarial (FGSM $\epsilon = 0.1$), MagNet Purified Adversarial

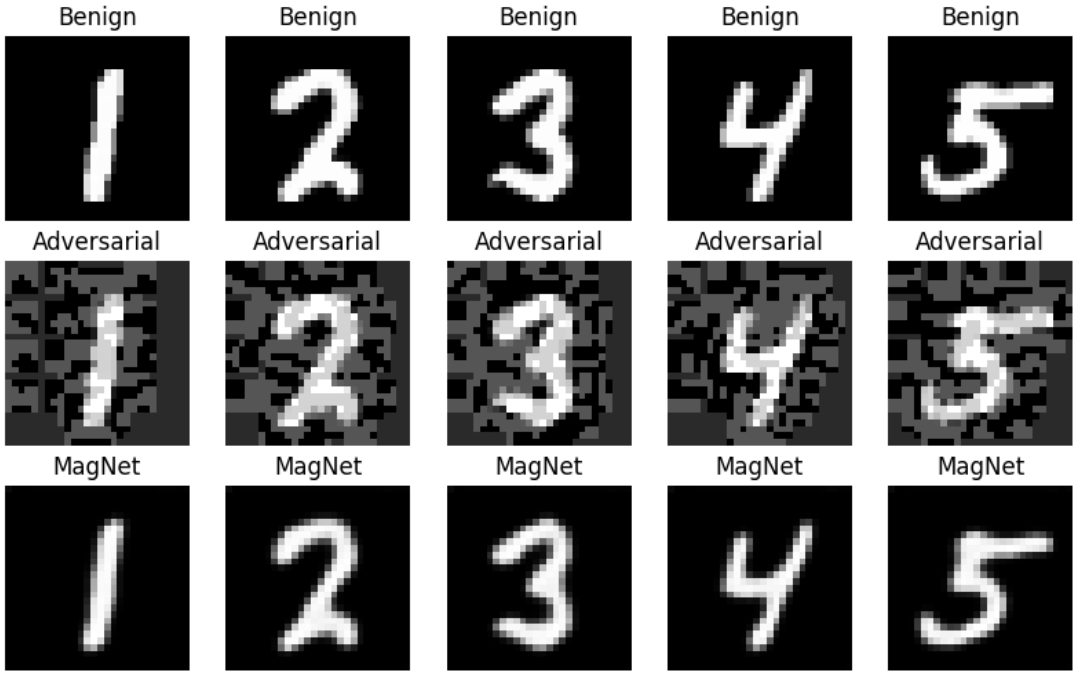


Figure 6.7: MNIST Benign, Adversarial (FGSM $\epsilon = 0.2$), MagNet Purified Adversarial

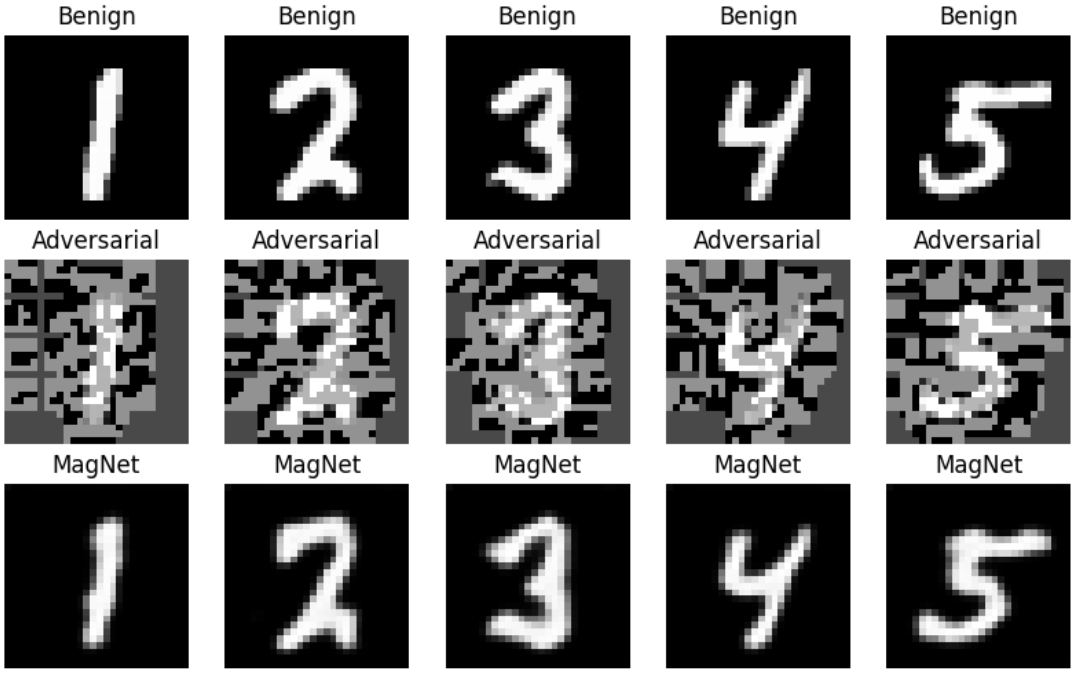


Figure 6.8: MNIST Benign, Adversarial (FGSM $\epsilon = 0.4$), MagNet Purified Adversarial

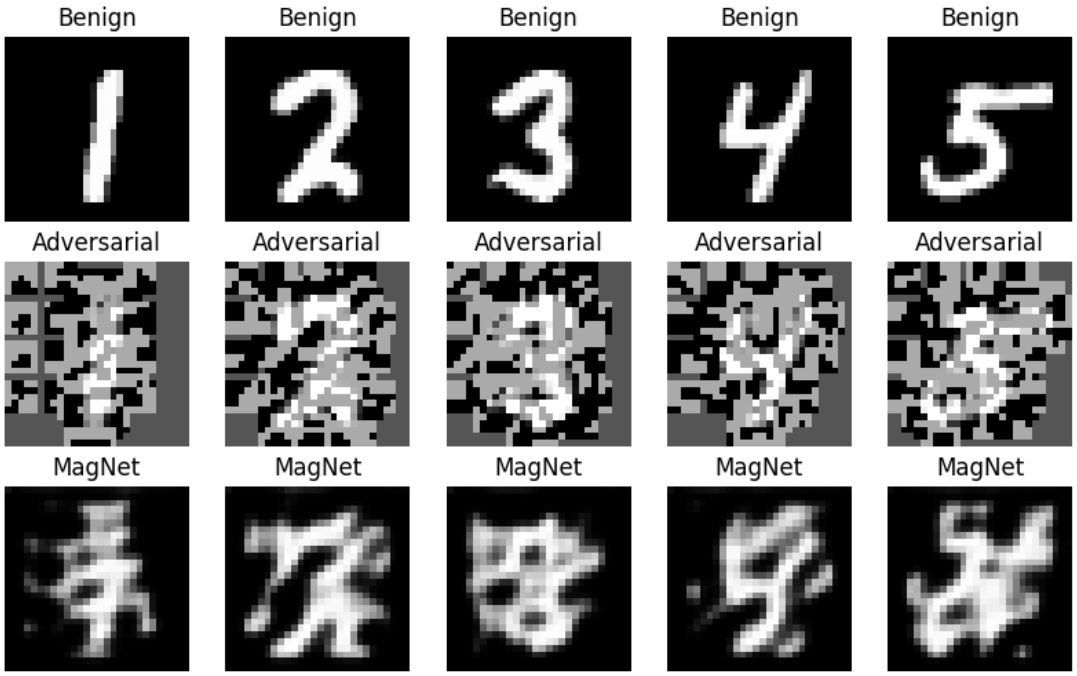
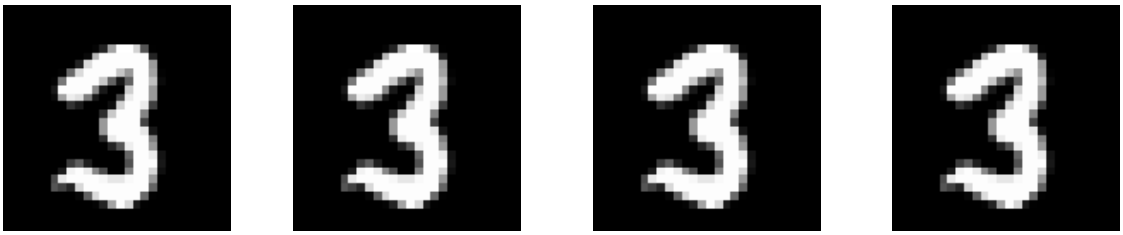


Figure 6.9: MNIST Benign, Adversarial (FGSM $\epsilon = 0.5$), MagNet Purified Adversarial

below 50%. The failure to reconstruct properly can be seen in Figure 6.9 where the purified images lack sharpness and only loosely resemble their benign counterparts.

Figure 6.10 shows the ability of the reformer MagNet model to correctly reconstruct benign images at all attack levels. This indicates that the model can be used on both adversarial and benign images.



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.2$ (c) $\epsilon = 0.4$ (d) $\epsilon = 0.5$

Figure 6.10: Benign MagNet MNIST reconstructed images

6.1.3 Analysis

The APE-GAN method performed extremely well at eliminating the adversarial noise on the MNIST dataset. It did however to distort benign images significantly when the ϵ value was set to 0.5. Adversarial attacks are most effective when they can not be noticed by a human. If the noise is aggressive, it becomes very easy to tell that the image or video has been tampered with. The noise was clearly evident when ϵ was set to 0.5 and therefore we can still conclude that APE-GAN is a valid approach to purifying adversarial images.

Similarly, MagNet was able to perform without sacrificing accuracy when the value of ϵ is low on the MNIST dataset. The performance does drastically drop off when the adversarial attack is more aggressive. APE-GAN was able to still perform with high accuracy when the attack was aggressive on the MNIST dataset.

6.2 CIFAR-10

6.2.1 APE-GAN

Figures 6.11 to 6.13 contain some example benign images from CIFAR-10, the adversarial images and the purified images by APE-GAN. The perturbations when $\epsilon = 0.01$ are barely noticeable to the human eye and APE-GAN purifies the images. The perturbations when $\epsilon = 0.05$ do become noticeable to the naked eye and APE-GAN does reconstruct an image that closely resembles the original image with some minor differences and reduction of sharpness. The adversarial images when $\epsilon = 0.1$ contain a very noticeable amount of noise and the purified images remove most of the noise but only to a certain extent with the images only being somewhat recognisable after purification.

The empirical performance of APE-GAN on the test set of the adversarial CIFAR-10 dataset is outlined in Table 6.4. APE-GAN was less effective on the CIFAR-10 images

Figure 6.11: CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.01$), APE-GAN Purified Adversarial

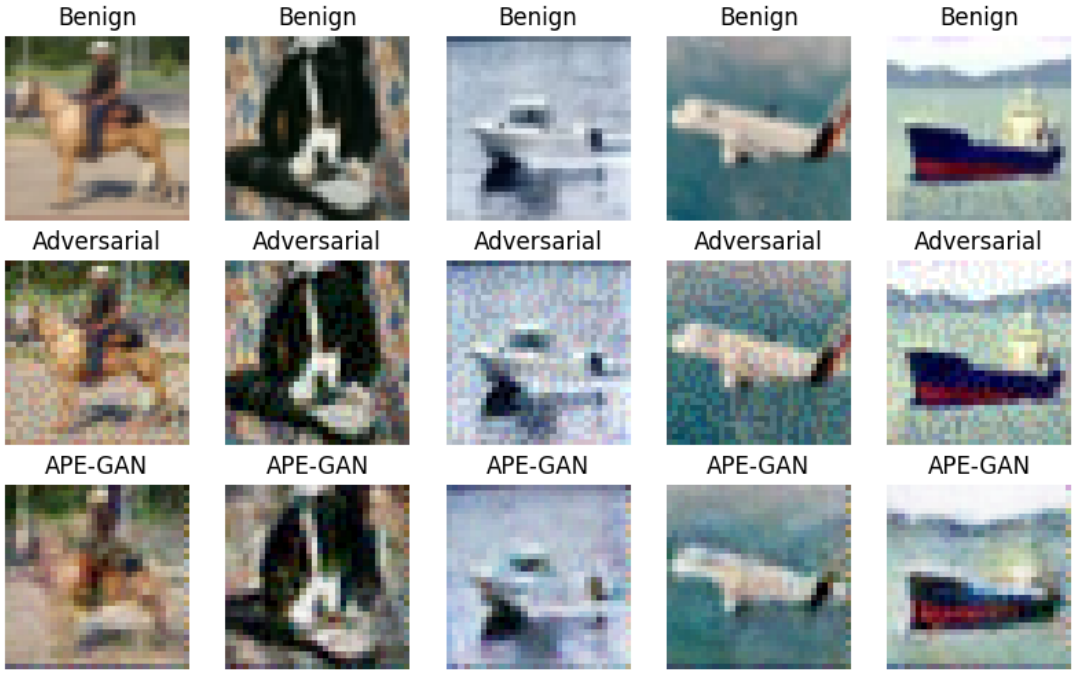


Figure 6.12: CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.05$), APE-GAN Purified Adversarial

Figure 6.13: CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.1$), APE-GAN Purified Adversarial

than it was on the simpler MNIST dataset, but was still able to recover the CNN classifier's accuracy up to approximately 70% when applied to the adversarial images.

APE-GAN was not able to perform with the same success on the CIFAR-10 dataset as it was on the MNIST dataset. APE-GAN was able to restore the CNN classifier's accuracy up to approximately 70% when applied to the adversarial images. The same model applied to the benign images managed to achieve an accuracy of 78.02% and 69.64% respectively. The performance of APE-GAN dropped when applied to the adversarial dataset with $\epsilon = 0.1$. The model performed with an accuracy of 66.55% on the adversarial images and 58.91% on the benign images.

Table 6.4: CIFAR-10 APE-GAN performance

ϵ	**Benign**	**Adversarial**	**APE (Adversarial)**	**APE (Benign)**
0.01	84.20	9.54	70.02	78.02
0.05	84.20	0.64	71.00	69.64
0.1	84.20	2.45	66.55	58.91

Figure 6.14 presents three benign images that APE-GAN has reconstructed at the various levels of ϵ. The generator from the APE-GAN model is required to remove a large amount of noise for more aggressive attacks. Applying the same transformation on images that do not contain harsh adversarial noise reduces the quality of the image and in turn, the classifier performs less optimally on reconstructed benign images.

(a) $\epsilon = 0.01$ (b) $\epsilon = 0.05$ (c) $\epsilon = 0.1$

Figure 6.14: Benign APE-GAN reconstructed CIFAR-10 images

6.2.2 MagNet

The autoencoder architecture used for the MagNet approach was the same as provided in the original paper [Meng and Chen 2017]. Table 6.5 shows the architecture of the autoencoder, it just contained 3 layers with Convolutional layers with the Sigmoid activation function. All convolutional layers have 3 input and output channels, a kernel size of 3 and a stride and padding of 1.

Table 6.5: MagNet CIFAR-10 autoencoder architecture

MagNet Reformer Architecture	
Conv2D Sigmoid	$3 \times 3 \times 3$
Conv2D Sigmoid	$3 \times 3 \times 3$
Conv2D Sigmoid	$3 \times 3 \times 1$

Figures 6.15 to 6.17 contain sample images from the CIFAR-10 test dataset. The plots compare the benign, adversarial and purified images by MagNet. Figure 6.15 shows that the adversarial perturbations when ϵ is set to 0.01 are minimal and that it is not noticeable to the human eye. It also shows that the reformed images by MagNet closely resemble the benign images. Table 6.6 shows that MagNet was able to achieve an accuracy of 60.93% when applied to the adversarial images and that the accuracy achieved when applied to benign images is 79.16%.

When ϵ is set to 0.05 there is significant adversarial noise present in the image as shown in Figure 6.16. The reconstructed images seem to look quite similar to the benign but the accuracy of the underlying model still suffers. Table 6.6 shows that the accuracy drops down to 44.61% when reconstructing from the adversarial images. The model's classification accuracy when applied to the benign images still remains high at 79.21%.

Lastly, Figure 6.17 shows some example images when ϵ is 0.1 which is high in the context of this dataset and classifier. The adversarial noise is very aggressive making the underlying image very unclear. MagNet is able to remove some of the noise but the resultant image does lack sharpness which in turn results in the CNN classifying poorly on these reconstructed images. Table 6.6 shows that the accuracy drops all the way down to 14.75% on the purified adversarial images. The accuracy when applied to the benign images also drops slightly, down to 67.21% compared to the original accuracy of 84.20%.

Table 6.6: CIFAR MagNet Performance

ϵ	**Benign**	**Adversarial**	**MagNet (Adversarial)**	**MagNet (Benign)**
0.01	84.20	9.54	60.93	79.16
0.05	84.20	0.64	44.61	79.27
0.1	84.20	2.45	14.75	67.21

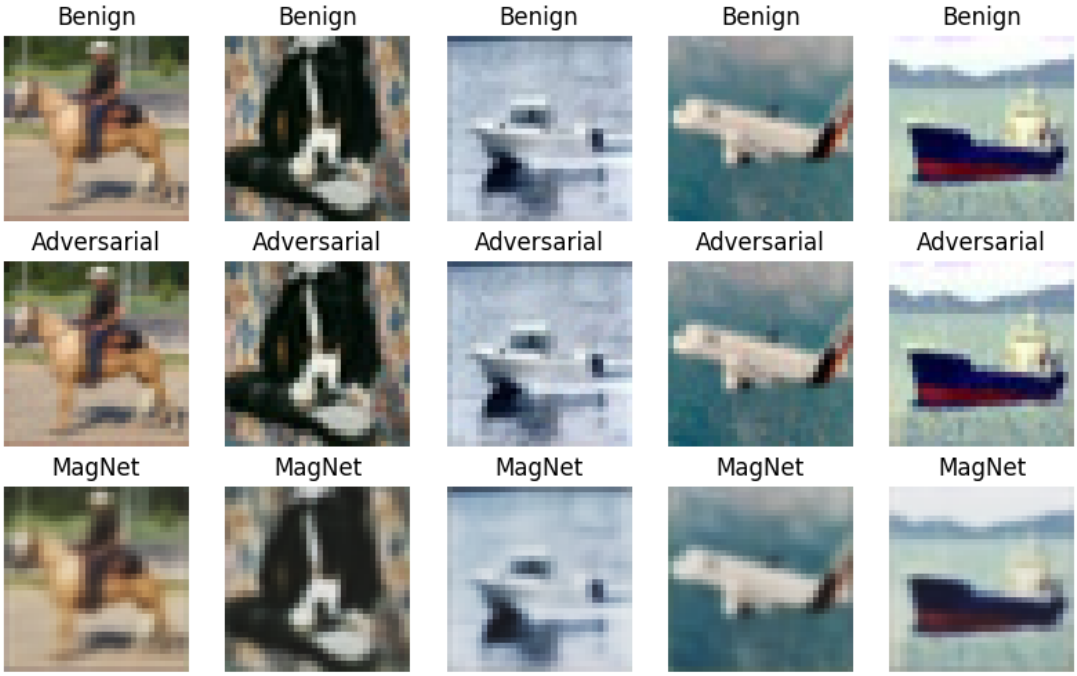


Figure 6.15: CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.01$), MagNet Purified Adversarial

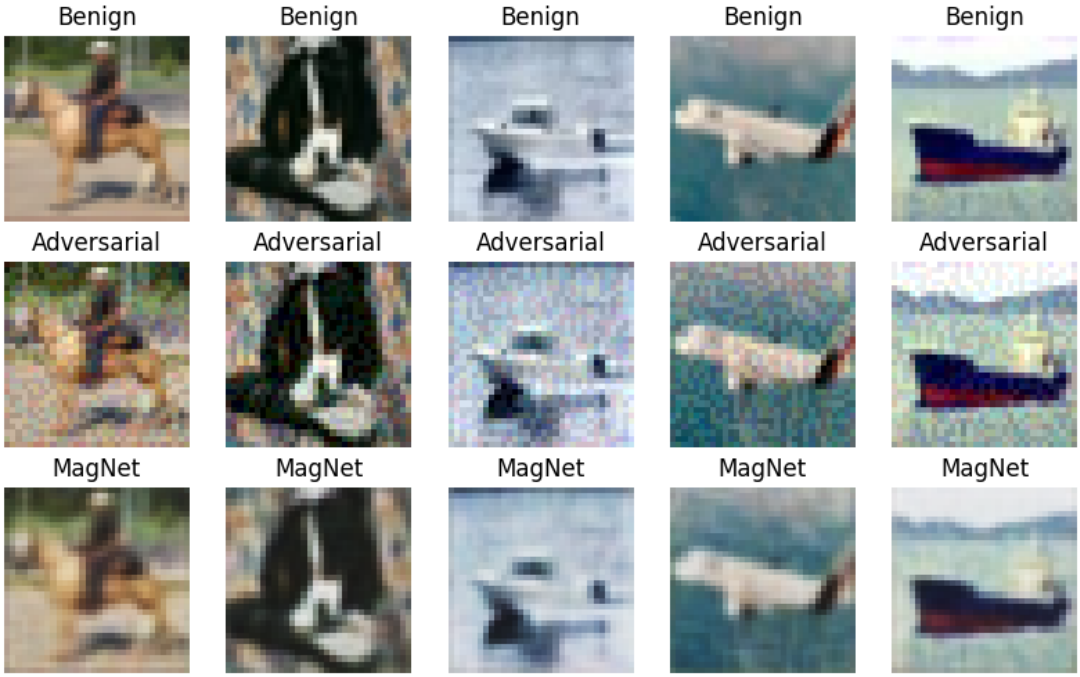


Figure 6.16: CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.05$), MagNet Purified Adversarial

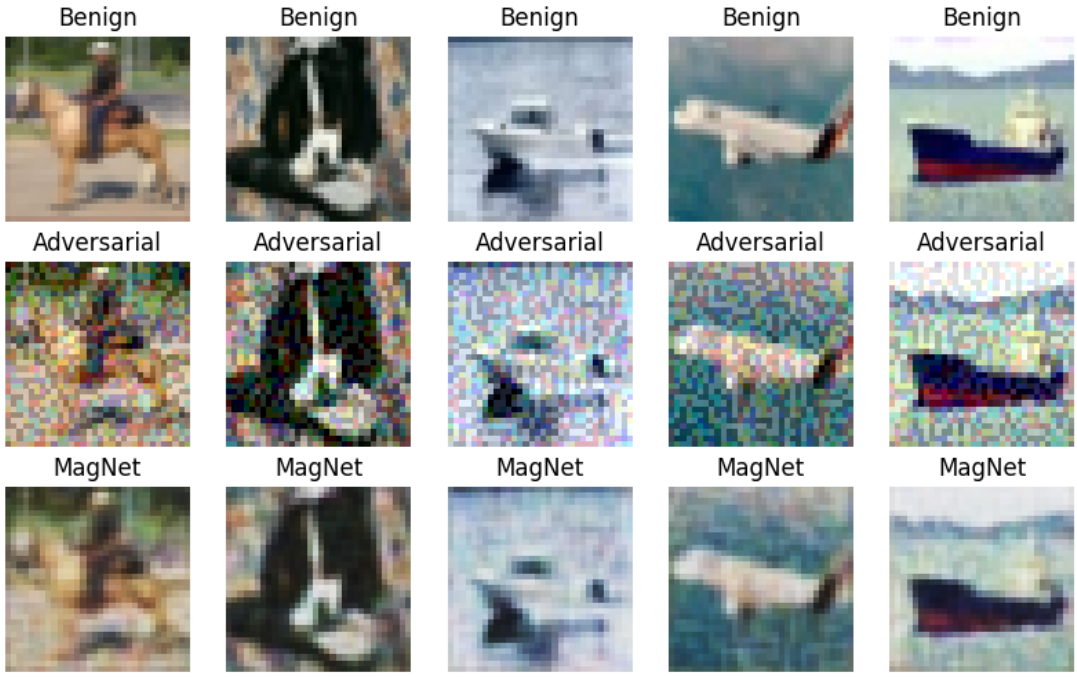


Figure 6.17: CIFAR-10 Benign, Adversarial (FGSM $\epsilon = 0.1$), MagNet Purified Adversarial

Figure 6.18 contain examples of the MagNet reformer model being applied to benign images. The reconstructed images contain little added noise and do not lose a great deal of sharpness compared to the reconstructed APE-GAN variants found in Figure 6.14.

(a) $\epsilon = 0.01$ (b) $\epsilon = 0.05$ (c) $\epsilon = 0.1$

Figure 6.18: Benign MagNet CIFAR-10 reconstructed images

6.2.3 Analysis

The APE-GAN method was effective at drastically improving the classification accuracy of the model when applied to the adversarial CIFAR-10 dataset. It was unable to fully restore the accuracy but was effective at improving it.

MagNet was unable to fully restore the performance of the CIFAR-10 classifier irrespective of how aggressive the adversarial attack was. The defense method was only able to restore the accuracy of the model back to approximately 61% compared to the original accuracy of 84.20% with the least aggressive attack. The model failed to defend successfully at higher levels of ϵ. APE-GAN was able to perform with higher accuracy on the adversarial CIFAR-10 dataset.

6.3 FaceForensics++

The evaluation on the FaceForensics++ dataset is more thorough that the other datasets. In addition to the accuracies of the victim model shown in the previous chapter, the precision, recall and F1-score of the victim model on the benign dataset is shown in Table 6.7. The NeuralTextures subclass has been omitted when evaluating the precision, recall and F1-score as it performs poorly and would skew the results unfairly. The implemented defense model results should not be hindered by the failure of the underlying victim model. It is therefore important to always compare the metrics to the victim model's metrics on the benign inputs without the adversarial defense in place.

Table 6.7: XceptionNet evaluation metrics on benign dataset

	Precision	**Recall**	**F1-score**
Real	0.87	0.83	0.85
Fake	0.94	0.96	0.95
Weighted	0.92	0.93	0.93

6.3.1 APE-GAN

The APE-GAN defense is a GAN based method. A GAN consists of two networks, the generator and the discriminator network. The architecture of the artificial neural networks used for the APE-GAN implementation are shown in Table 6.8 and contains the same architecture used for the CIFAR-10 dataset as presented in the original paper by Jin *et al.* [2019]. Variations of this network were implemented and tested but the original architecture proved to be the most effective. The models were trained for 200 epochs, using the Adam Optimiser with a learning rate of 0.001, a batch size of 16 and a gradient penalty of 10.

Table 6.8: APE-GAN FaceForensics++ architecture

Generator Architecture	**Discriminator Architecture**
Conv2D + Batch Norm	Conv2D + Batch Norm
Leaky Relu	Leak Relu
Conv2D + Batch Norm	Conv2d + Batch Norm
Leaky Relu	Leaky Relu
Conv2D + Batch Norm	Conv2D + Batch Norm
Leaky Relu	Leaky Relu
Deconv2D + Batch Norm	Conv2D + Batch Norm
Leaky Relu	Leaky Relu
Deconv2D + Batch Norm	FC + Batch Norm
Leaky Relu	Leak Relu
Deconv2D	FC
Sigmoid	Sigmoid

APE-GAN: I-FGSM-0.0001

Table 6.9 contains the precision, recall and F1-score of the model across the adversarial real and fake videos in the FaceForensics++ dataset.

The fake class of videos contains three different deepfake generation methods, DeepFakes, Face2Face and FaceSwap, it, therefore, contains three times more videos than the real class. As a result, the weighted measures are closer to the values achieved on the fake video sub-datasets.

The victim model with no defense performed with an F1-score of 0.85 on the benign real class and now performs with an F1-score of 0.67 on the purified adversarial real class, which is lower and shows a noticeable drop in performance. Initially, the victim model with no defense, performed with an F1-score of 0.95 on the benign fake class. APE-GAN was able to restore the victim model's F1-score to 0.86 which is lower than the original value but can still be considered effective.

The recall for the real class is significantly lower than the fake class. This can be attributed to two things, the victim model is worse at predicting real videos accurately and APE-GAN could be introducing artefacts into the images that further reduce its performance.

Table 6.9: APE-GAN I-FGSM-0.0001 evaluation metrics

	Precision	Recall	F1-score
Real	0.76	0.61	0.67
Fake	0.82	0.90	0.86
Weighted	0.80	0.80	0.80

Table 6.10 contains a detailed breakdown of the model's accuracy performance across the different FaceForensics++ subsets. The benign and adversarial accuracies for each subset are shown and additionally, the accuracies for when the APE-GAN defense is used to reconstruct the adversarial examples and benign examples are shown.

Table 6.10: APE-GAN I-FGSM-0.0001 accuracies per FaceForensics++ subset

	Benign	Adversarial	APE-GAN (Adversarial)	APE-GAN (Benign)
Real	83.44	1.02	60.54	61.11
DeepFakes	96.94	22.69	94.81	94.92
Face2Face	96.23	5.02	85.60	86.03
FaceSwap	93.79	4.84	86.01	86.24
NeuralTextures	31.75	0.00	47.76	48.23

The first row shows that APE-GAN can improve the model's accuracy when predicting adversarial real videos to around 61%. This is a large improvement from the adversarial performance but is significantly lower than the model's original accuracy. When the trained APE-GAN generator is applied to the real benign images to reconstruct them, it also reduces the performance down to around 61%. This indicates that the reconstructed images do have some artefacts that are introduced that confuse the victim model.

APE-GAN is able to almost fully restore the performance of the model for the DeepFakes subset as it brings the accuracy results up to 94.81% which is only slightly lower than the benign accuracy. Furthermore, APE-GAN can reconstruct benign DeepFakes to around 95% which is very close to the original performance. This shows that APE-GAN performed successfully for this particular FaceForensics++ subset.

APE-GAN was able to drastically improve the performance for both adversarial Face2Face and FaceSwap subsets. It restored their accuracy to approximately 86%. This can also be said for their performance when reconstructing the benign examples. These accuracies are 10% lower than the accuracies achieved on the benign subsets, but are still relatively high and could be considered usable.

The victim model performs extremely poorly on the NeuralTextures subset. When APE-GAN is applied to reconstruct both adversarial and benign examples, it does result in a slight increase in performance. This could be a result of APE-GAN introducing some artefacts that results in the underlying classifier predicting the images as fake.

Figure 6.19 provides a grid of images that shows some insight into what the reconstructed APE-GAN images look like and if there are any artefacts that are being intro-

duced. A single frame from a video contained in the test set has been chosen for visualisation and comparison. Each column represents a subset of the FaceForensics++ dataset which are real, DeepFakes, Face2Face, FaceSwap and NeuralTextures. Each row represents a different variant of the image. The first row contains a benign, unaltered image, the second row contains the adversarial image, the third row contains the APE-GAN purified adversarial image and the final row contains the APE-GAN reconstructed benign image.

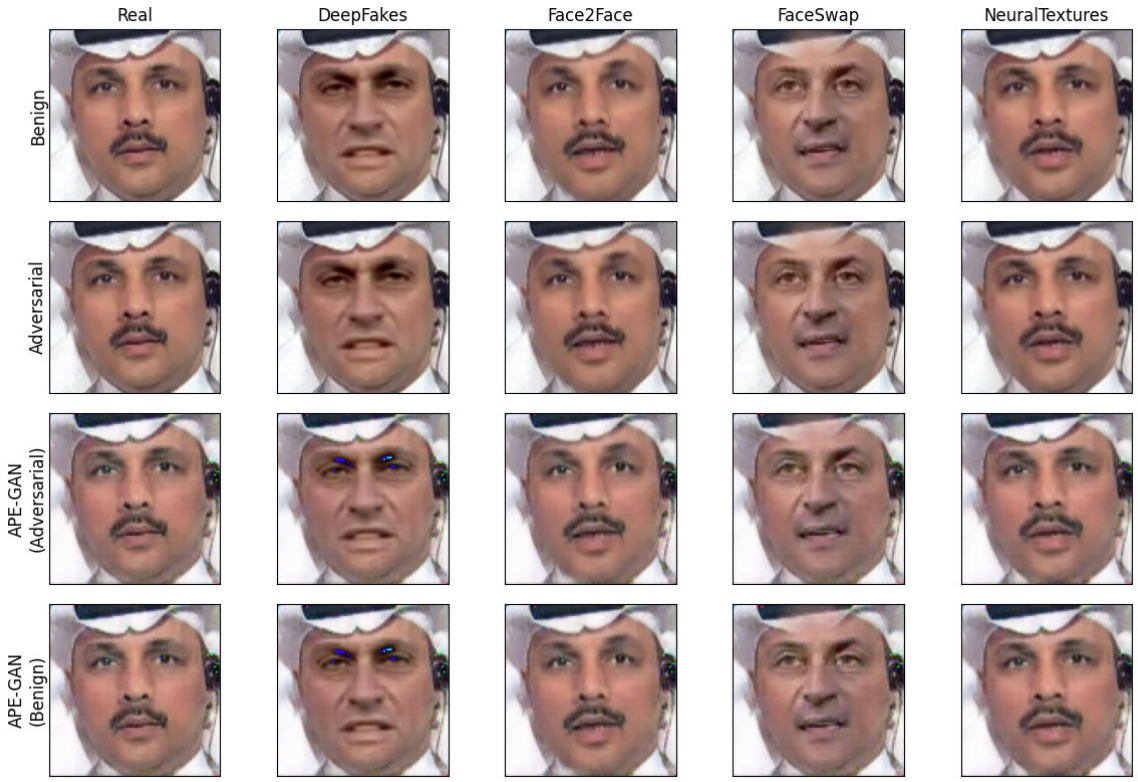


Figure 6.19: Benign, Adversarial (I-FGSM-0.0001), APE-GAN Purified Adversarial, APE-GAN Reconstructed Benign example frame comparison

The image also shows that the Face2Face and NeuralTextures methods make use of a reenactment technique which maintains the features of the real image. The DeepFakes and FaceSwap methods utilise replacement techniques, resulting in facial features that match the source video and not the target video.

When comparing the images in the first and second rows, the added adversarial noise cannot be seen through visual inspection, however, it results in a drastic drop off in performance for the underlying victim model. The last two rows contain the reconstructed images that have been produced by the generator from the APE-GAN model. It is clear from these images that the generator does introduce some artefacts. Brightly coloured pixelated spots can be seen in the dark regions of the image by the person's right eyebrow and ear. It is hypothesised that these artefacts are picked up by the victim model and causes it to predict the images as fake, further reducing its performance on the real class.

APE-GAN: I-FGSM-0.01

This subsection details the performance of APE-GAN when applied to the adversarial dataset that has been crafted with $\epsilon = 0.01$. This is a more aggressive attack and is able to fool the victim model on every frame.

Table 6.11 contains the precision, recall and F1-score for APE-GAN when applied to the adversarial dataset. The F1-score of the model on the real class is low at 0.26. It was able to achieve an F1-score of 0.62 when applied to the fake subsets which is higher than the performance on the real subset. The victim model performed with an F1-score of 0.85 on the benign real subset and 0.95 on the benign fake subset with no defense applied. This indicates that APE-GAN has failed to restore the performance of the victim model.

Table 6.11: APE-GAN I-FGSM-0.01 evaluation metrics

	Precision	**Recall**	**F1-score**
Real	0.20	0.34	0.26
Fake	0.71	0.55	0.62
Weighted	0.59	0.50	0.53

Table 6.12 contains a breakdown of the model's accuracy performance across the different FaceForensics++ subsets. The benign and adversarial accuracies for each subclass are shown and additionally, the accuracies for when the APE-GAN defense is used to reconstruct the adversarial examples and benign examples. The adversarial column shows that the adversarial attack is powerful, as it is able to completely fool the victim model across all FaceForensics++ subsets.

The trained APE-GAN model performs poorly across most adversarial subsets, but it is able to restore the classification accuracy to 70% for the DeepFakes subset. This is however much lower than the initial 97% accuracy. When APE-GAN is used to reconstruct the benign images, it is able to achieve decent results for the DeepFakes, Face2Face and FaceSwap subsets. It does however struggle to reconstruct the real and NeuralTextures subset.

Table 6.12: APE-GAN I-FGSM-0.01 accuracies per FaceForensics++ subset

	Benign	**Adversarial**	**APE-GAN (Adversarial)**	**APE-GAN (Benign)**
Real	83.44	0.0	34.47	56.31
DeepFakes	96.94	0.0	70.37	94.13
Face2Face	96.23	0.0	38.39	81.36
FaceSwap	93.79	0.0	55.58	85.69
NeuralTextures	31.75	0.0	19.94	52.11

Figure 6.20 contains a grid of images that provides some insight into what the reconstructed APE-GAN images look like and if there are any artefacts that are being introduced. The same frames that were used in Figure 6.19 are used again for comparison.

This attack is more aggressive and the images contain an average l_∞ norm that is approximately 100 times more than the attack when $\epsilon = 0.0001$. The difference between the benign and adversarial frames is still unnoticeable upon visual inspection. This indicates that only a slight nudge in the pixel values can cause the model to perform extremely poorly.

The third and fourth rows in Figure 6.20 show that both the purified adversarial and reconstructed benign images introduced bright pixel artefacts in the dark regions of the image. The black band at the top of the image contains more of these artefacts when compared to the previous results. Different architectures for APE-GAN were trained and investigated but the introduced artefacts could not be removed from the reconstructed output.

Figure 6.20: Benign, Adversarial (I-FGSM-0.01), APE-GAN Purified Adversarial, APE-GAN Reconstructed Benign example frame comparison

The APE-GAN defense method performs much worse on the more aggressive attack and is deemed unsuccessful when $\epsilon = 0.01$.

6.3.2 MagNet

The MagNet reformer architecture used for the deepfake results is the same architecture used on the CIFAR-10 dataset and is shown below in Table 6.13. The models were trained for 50 epochs with batch size of 16 using the Adam optimiser with a learning rate of 0.001 and the mean squared error loss function was used.

Table 6.13: MagNet FaceForensics++ autoencoder architecture

MagNet Reformer Architecture	
Conv2D Sigmoid	$3 \times 3 \times 3$
Conv2D Sigmoid	$3 \times 3 \times 3$
Conv2D Sigmoid	$3 \times 3 \times 1$

MagNet: I-FGSM-0.0001

Table 6.14 contains the precision, recall and F1-score of the defense model evaluated on the adversarial real and fake datasets. As previously stated, the fake class of videos for these metrics contain three different deepfake generation methods and therefore contain three times more videos than the real class. As a result, the weighted measures are closer to the values achieved on the fake video sub-datasets.

Table 6.14: MagNet I-FGSM-0.0001 evaluation metrics

	Precision	Recall	F1-score
Real	0.63	0.69	0.66
Fake	0.83	0.86	0.88
Weighted	0.83	0.82	0.82

The MagNet defense was able to achieve an F1-score of 0.66 on the real class and an F1-score of 0.88 on the fake class. These results are very similar to those achieved by the APE-GAN model, with MagNet performing with an F1-score that is 0.02 higher on the fake class and only 0.01 lower on the real class.

The victim model performed with an F1-score of 0.85 on the benign real dataset. The defense model is able to perform with an F1-score of 0.66 on the adversarial real dataset which is much lower. In contrast, the victim model performed with an F1-score of 0.95 on benign fake dataset and the defense model now performs with an F1-score of 0.88 on the adversarial fake dataset. This is lower but can still be considered good with a result close to 0.9.

Table 6.15 contains a breakdown of the model's accuracy performance across the different FaceForensics++ subsets. The benign and adversarial accuracies for each subclass is shown and furthermore, the accuracies for when the APE-GAN defense is used to reconstruct the adversarial examples and benign examples.

Table 6.15: MagNet I-FGSM-0.0001 accuracies per FaceForensics++ subset

	Benign	Adversarial	MagNet (Adversarial)	MagNet (Benign)
Real	83.44	1.02	70.50	69.40
DeepFakes	96.94	22.69	95.21	95.01
Face2Face	96.23	5.02	87.71	88.42
FaceSwap	93.79	4.84	76.14	77.14
NeuralTextures	31.75	0.00	39.76	40.99

The first row shows that the MagNet model is able to almost restore the accuracy performance on the real subset of the FaceForensics++ dataset. It achieves an accuracy of around 70% for both purifying the adversarial input and reconstructing the benign input with the model. This is higher than what the equivalent APE-GAN model was able to achieve. 70% is considered somewhat decent for a binary classifier.

The defense model performed particularly well on the DeepFakes subset as it was able to perform with an accuracy of 95% on both adversarial and benign inputs. This is only 2% lower than the original accuracy and is considered a successful defense technique. The defense model performs slightly worse on the Face2Face subset, where it is only able to achieve accuracies of around 88% when it is used to reconstruct the adversarial and benign inputs. The performance on the FaceSwap subset is slightly worse, where it performs with an accuracy of 76% and 77% when applied to the adversarial and benign inputs respectively. The underlying victim model only performs with an accuracy of 32% on the NeuralTextures subset and the defense technique is able to improve this accuracy to about 40% but this is still poor performance for a binary classifier.

Figure 6.21 provides a grid of sample images that show some insight into what the reconstructed MagNet images look like. A visual inspection can be done to determine if the model is incorrectly reconstructing the images. The same single frame from the prior Figures 6.19 and 6.20 is shown again but with the reconstructed images using the MagNet model. The adversarial noise is not discernible to a human when comparing the images in row one and row two.

The reconstructed images in rows three and four do seem to maintain the majority of the input image's integrity. The only difference noticeable is the slight difference in colours. The white in the benign and adversarial images now contains a slight pink hue after reconstruction. This slight distortion could be a contributing factor to the model not fully restoring the victim model's initial accuracy.

MagNet: I-FGSM-0.01

This section details the performance of the MagNet defense when applied to the adversarial dataset that has been crafted with $\epsilon = 0.01$. This makes for a more aggressive and stronger attack that is able to fool the victim model on every frame.

Table 6.16 contains the precision, recall and F1-score for MagNet when applied to the adversarial dataset. The defense is only able to achieve an F1-score of 0.21 and 0.45 on the real and fake subsets respectively. This is considerably lower than what the defense was able to achieve when the adversarial attack was less aggressive. Furthermore, these results are poor when compared to the F1-scores of 0.95 and 0.85 that the victim model is able to achieve on the benign examples respectively. This indicates that the defense is not successful when defending against more aggressive attacks.

Table 6.17 provides insight into the model's accuracy across the different FaceForensics++ subsets. The benign and adversarial accuracies for each subclass are shown. The accuracies for when MagNet purifies the adversarial input and reconstructs the benign input are also provided. The attack is more aggressive and this can be seen by the adversarial accuracies all being reduced to 0%.

Figure 6.21: Benign, Adversarial (I-FGSM-0.0001), MagNet Purified Adversarial, MagNet Reconstructed Benign example frame comparison

Table 6.16: MagNet I-FGSM-0.01 evaluation metrics

	Precision	**Recall**	**F1-score**
Real	0.15	0.35	0.21
Fake	0.62	0.35	0.45
Weighted	0.50	0.39	0.39

Table 6.17: MagNet I-FGSM-0.01 accuracies per FaceForensics++ subset

	Benign	**Adversarial**	**MagNet (Adversarial)**	**MagNet (Benign)**
Real	83.44	0.0	34.90	67.50
DeepFakes	96.94	0.0	59.53	89.85
Face2Face	96.23	0.0	23.26	78.02
FaceSwap	93.79	0.0	22.60	65.41
NeuralTextures	31.75	0.0	7.75	42.32

The accuracies achieved across all subsets on the adversarial inputs are low. The model was only able to restore the accuracy for the DeepFakes subset to around 60% with all other subsets falling below 35%. This further indicates the failure of the defense across all FaceForensics++ subsets.

The model does perform better when reconstructing benign inputs compared to the adversarial inputs. This implies that the MagNet model did not learn to remove the adversarial noise and does not alter the input images significantly.

Lastly, Figure 6.22 contains a grid of the same frame shown previously along with the reconstructed images. The adversarial noise is not identifiable when comparing the benign and adversarial images. The only difference in the reconstructed images is the slight colour distortion between the images. Since the underlying classifier still performs poorly, we can conclude that the adversarial noise is still present after reconstruction using MagNet.

Figure 6.22: Benign, Adversarial (I-FGSM-0.01), MagNet Purified Adversarial, MagNet Reconstructed Benign example frame comparison

The MagNet model performs significantly worse than on the more aggressive adversarial dataset and is considered unsuccessful when $\epsilon = 0.01$.

6.3.3 Analysis

Table 6.18 provides a summary of the results presented in this chapter. The first row provides the F1-scores for the victim model with no defense on the benign dataset. The

following rows contain the achieved F1-scores across both the real and fake subsets for the defense models.

Table 6.18: Defense model F1-score summary

	Real	**Fake**
No Defense - Benign	0.85	0.95
APE-GAN I-FGSM-0.0001	0.67	0.86
MagNet I-FGSM-0.0001	0.66	0.88
APE-GAN I-FGSM-0.01	0.26	0.62
MagNet I-FGSM-0.01	0.21	0.39

The first row indicates the victim model does perform worse on the real subset compared to the fake subset. This trend continues when looking at the results achieved by the defence models.

Both APE-GAN and MagNet defenses achieve respectable F1-scores when applied to the adversarial dataset where ϵ is set to 0.0001. The two defenses are not able to fully restore the performance of the underlying victim model but do achieve reasonable results in the context of binary classification. The F1-scores on the real class are lower than desired but it can be argued that it is still usable. The defense models can also be applied to reconstruct benign inputs without a severe drop in accuracy as shown in Tables 6.10 and 6.15. This means that the model can be used to reconstruct both benign and adversarial images. If this was not true, another model will be needed to detect if the input is adversarial. This introduces another model that could potentially be attacked and bypassed completely which is not desirable.

In contrast, the defense models do not succeed in restoring the F1-scores of the underlying model when the attack is more aggressive with ϵ set to 0.01. Visual inspection of the reconstructed adversarial and benign images for the APE-GAN defense shows that bright pixel spot artefacts are introduced into regions of the image that are close to black in colour. These artefacts could be the cause for the underlying victim models' inability to completely restore the performance of the classifier. The MagNet model doesn't introduce these bright pixel artefacts but does reconstruct the images with slightly altered colours.

In the real-world scenario, the value of ϵ used for adversarial training and testing can vary and may not be known in advance. Understanding how a defense model, trained on a specific value of ϵ, performs against different epsilon values is crucial for assessing its robustness.

In our experiments, we trained APE-GAN and MagNet defense models using a more aggressive adversarial dataset with $\epsilon = 0.01$. Subsequently, we evaluated these models on a less aggressive attack using $\epsilon = 0.0001$, as shown in table 6.19. As expected, we observed a slight drop in the F1-score across both classes when the models were tested on a different value of ϵ. Nonetheless, the results demonstrate that models trained on more aggressive attacks can still be effective in countering less aggressive attacks.

Table 6.19: F1-score summary when applying the models trained with $\epsilon = 0.01$ on the $\epsilon = 0.0001$ test set

	Real	Fake
APE-GAN I-FGSM-0.0001	0.62	0.79
MagNet I-FGSM-0.0001	0.58	0.81

On the other hand, we also conducted a reverse experiment, where defense models trained on the less aggressive attack with $\epsilon = 0.0001$ were evaluated on a more aggressive test dataset with $\epsilon = 0.01$. The results, as shown in table 6.20, were worse compared to the performance of the models originally trained on the adversarial dataset with $\epsilon = 0.0001$. This suggests that the original performance of defense models on more aggressive attacks was already low, and their performance degraded further when applied to more challenging scenarios.

Table 6.20: F1-score summary when applying the models trained with $\epsilon = 0.0001$ on the $\epsilon = 0.01$ test set

	Real	Fake
APE-GAN I-FGSM-0.01	0.22	0.55
MagNet I-FGSM-0.01	0.18	0.33

The experiments indicate that it is acceptable but not ideal to use a model trained on more aggressive attacks to defend against less aggressive attacks. It is crucial to carefully choose and fine-tune the defense model based on the expected threat level (epsilon value) to achieve better performance and robustness against adversarial attacks.

In summary, the APE-GAN and MagNet defense models can be successful when the adversarial attack is mild. These models, however, cannot be guaranteed to work when the attacks are more aggressive.

6.4 Future recommendations

The research conducted only made use of the iterative fast gradient sign method. The defense models should be tested against the Carlini Wagner L2 attack which is considered a harder attack to defend against [Carlini and Wagner 2017].

There have also been great advancements in the field of deepfake creation. These newer more sophisticated methods should be tested along with the new state-of-the-art detection methods. This will provide further insight if these techniques could potentially be deployed to social media platforms.

The chosen victim model is a CNN model that only evaluates the input on a frame-by-frame basis. Further research should look into possibly using 3D convolutional networks that can evaluate multiple frames at a time and detect both spatial and temporal features. The adversarial noise contained from frame to frame may be different enough to pick up on and identify the presence of an attack.

GANs and autoencoders have proven to be very useful generative neural network models across a vast variety of applications. Diffusion models are a new class of model that has been shown to outperform GANs and autoencoders [Dhariwal and Nichol 2021]. Nie *et al.* [2022] have introduced a method known as DiffPure which used diffusion models for adversarial purification. This technique should be applied to the context of deepfake detectors and evaluated.

Chapter 7

Conclusion

Deepfake detection is a critical task that needs to be addressed to curb the spread of misinformation. Although deep learning models based on convolutional neural networks have shown high accuracy in detecting deepfakes, they are susceptible to adversarial attacks. While there has been some research on adversarial defense mechanisms, little exists in the context of deepfakes.

Two different generative based adversarial defense mechanisms were investigated in this report, namely APE-GAN, a GAN based model and MagNet an autoencoder based model. These models can be combined with deepfake detectors to potentially create a classification pipeline that is robust against adversarial attacks. The FaceForensics++ deepfake dataset which contains 4 different deepfake creation techniques has been used. The victim model chosen for experimentation is a CNN model based on the XceptionNet architecture which has been pretrained on the ImageNet dataset. The iterative fast gradient sign adversarial attack method was used to attack the model and form the basis of the experimentation.

The victim model was attacked using the iterative fast gradient sign method at two different levels of ϵ, $\epsilon = 0.0001$ and $\epsilon = 0.01$. The first attack reduces the performance of the victim model to an average accuracy of 6% across all the FaceForensics++ subsets. In comparison, the more aggressive attack reduces the performance of the victim model to 0%.

Both APE-GAN and MagNet defenses achieved respectable F1-scores when applied to the adversarial dataset when ϵ is set to 0.0001. The two defenses are not able to fully restore the performance of the underlying victim model but do drastically improve the performance and achieve reasonable results in the context of binary classification. The victim model originally performed with an F1-score of 0.85 on the real class and 0.95 on the fake class. The defense models were able to restore the F1-score of the classifier to 0.67 for the real class and 0.88 for the fake class. In addition to this, the models can be used to reconstruct benign input images without having a significant impact on the overall performance. This is desirable as the models can be used without the need for an adversarial detector network to determine if the images need to be purified or not. It can pass all inputs through the pipeline for classification.

On the other hand, when the attack is more aggressive with ϵ set to 0.01, the defense models fail to restore the F1-scores of the underlying model. Upon visual inspection of the reconstructed adversarial and benign images for the APE-GAN defense, it is observed that bright pixel spot artefacts are introduced into regions of the image that are close to black in colour. These artefacts could be the reason behind the underlying victim model's inability to completely restore the performance of the classifier. The MagNet model does not introduce such bright pixel artefacts, but it does reconstruct the images with slightly altered colours that could also hinder performance.

In summary, the APE-GAN and MagNet defense models can be effective against mild adversarial attacks, their success cannot be guaranteed when the attacks become more aggressive. Therefore, it is advisable to use an adversarial defense method alongside a deepfake detector to ensure accurate predictions, especially given the vulnerability of CNN-based deepfake detectors. Despite their limitations, APE-GAN and MagNet remain effective methods for deepfake detection, provided that the adversarial attacks are not aggressive.

References

[Afchar *et al.* 2018] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec 2018.

[Arjovsky *et al.* 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[Carlini and Wagner 2017] Nicholas Carlini and David Wagner. *Towards Evaluating the Robustness of Neural Networks*, 2017.

[Carreira and Zisserman 2018] Joao Carreira and Andrew Zisserman. *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset*, 2018.

[Chollet 2017] François Chollet. *Xception: Deep Learning with Depthwise Separable Convolutions*, 2017.

[Dhariwal and Nichol 2021] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[Goodfellow *et al.* 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*, 2014.

[Goodfellow *et al.* 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*, 2015.

[Gulrajani *et al.* 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[He *et al.* 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*, 2015.

[Hinton and Salakhutdinov 2006] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[Hosseini *et al.* 2017] Hossein Hosseini, Yize Chen, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. *Blocking Transferability of Adversarial Examples in Black-Box Learning Systems*, 2017.

[Jin *et al.* 2019] Guoqing Jin, Shiwei Shen, Dongming Zhang, Feng Dai, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3842–3846, 2019.

[King 2017] Davis King. *High quality face recognition with deep learning*, Feb 2017.

[Kingma and Welling 2014] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*, 2014.

[Krizhevsky *et al.* 2009] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. *Learning Multiple Layers of Features from Tiny Images*. Technical report, Technical Report, 2009.

[Krizhevsky *et al.* 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[Kurakin *et al.* 2017] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. *Adversarial Machine Learning at Scale*, 2017.

[Laskov and Lippmann 2010] Pavel Laskov and Richard Lippmann. *Machine learning in adversarial environments*, 2010.

[Lecun *et al.* 1998] 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.

[LeCun *et al.* 2010] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[Meng and Chen 2017] Dongyu Meng and Hao Chen. *MagNet: a Two-Pronged Defense against Adversarial Examples*, 2017.

[Narodytska and Kasiviswanathan 2017] Nina Narodytska and Shiva Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1310–1318, 2017.

[Neekhara *et al.* 2020] Paarth Neekhara, Shehzeen Hussain, Malhar Jere, Farinaz Koushanfar, and Julian J. McAuley. Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples. *CoRR*, abs/2002.12749, 2020.

[Nie *et al.* 2022] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.

[Ozdag 2018] Mesut Ozdag. Adversarial attacks and defenses against deep neural networks: a survey. *Procedia Computer Science*, 140:152–161, 2018.

[Papernot *et al.* 2016] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[Papernot *et al.* 2017] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, page 506–519, New York, NY, USA, 2017. Association for Computing Machinery.

[Pérez *et al.* 2003] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, July 2003.

[Radford *et al.* 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Ronneberger *et al.* 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015.

[Rössler *et al.* 2018] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. *FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces*, 2018.

[Samangouei *et al.* 2018] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models*, 2018.

[Sitawarin *et al.* 2018] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430*, 2018.

[Szegedy *et al.* 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. *Intriguing properties of neural networks*, 2014.

[Thies *et al.* 2016] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Niessner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Thies *et al.* 2019] Justus Thies, Michael Zollhöfer, and Matthias Nießner. *Deferred Neural Rendering: Image Synthesis using Neural Textures*, 2019.

[Tran *et al.* 2015] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. *Learning Spatiotemporal Features with 3D Convolutional Networks*, 2015.

[Wang and Dantcheva 2020] Yaohui Wang and Antitza Dantcheva. A video is worth more than 1000 lies. comparing 3dcnn approaches for detecting deepfakes. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 515–519, 2020.

[Wierstra *et al.* 2011] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, and Jürgen Schmidhuber. *Natural Evolution Strategies*, 2011.

[Wu *et al.* 2021] Jing Wu, Mingyi Zhou, Ce Zhu, Yipeng Liu, Mehrtash Harandi, and Li Li. Performance evaluation of adversarial attacks: Discrepancies and solutions. *arXiv preprint arXiv:2104.11103*, 2021.

[Xiao *et al.* 2018] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

[Yang *et al.* 2021] Rui Yang, Xiu-Qing Chen, and Tian-Jie Cao. Ape-gan++: An improved ape-gan to eliminate adversarial perturbations. *IAENG International Journal of Computer Science*, 48(3):827–844, 2021.