TARGETED MULTI-VIEW ADVERSARIAL ATTACKS WITH UNIVERSAL PERTURBATION

A Thesis

Presented to the

Faculty of

Wentworth Institute of Technology

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Applied Computer Science

---

by

Christian Green.

Spring 2024

WENTWORTH INSTITUTE OF TECHNOLOGY

The Undersigned Committee Approves the

Thesis of Christian Green.:

Targeted Multi-View Adversarial Attacks

with Universal Perturbation

Mehmet Ergezer, Chair
School of Computing and Data Science

Frank Kreimendahl
School of Computing and Data Science

Youssef Qranfal
School of Computing and Data Science

DEDICATION

This thesis is dedicated to my Mother, Patricia Green and my Father, Milton Green, whose unwavering support, love, and encouragement have been my greatest source of strength throughout the adventure that this research has become.

The mystery of life isn't a problem to solve, but a reality to experience.

– Frank Herbert

ABSTRACT OF THE THESIS

Targeted Multi-View Adversarial Attacks
with Universal Perturbation
by
Christian Green.
Master of Science in Applied Computer Science
Wentworth Institute of Technology, 2024

This paper extends a recent novel universal perturbation method designed to generate robust multi-view adversarial examples in 3D object recognition. Our approach focuses on enhancing the proposed algorithm to incorporate targeted use environments, in turn broadening its applicability in real-world scenarios. The initial motivation for our work stemmed from the concept of adversarial architecture—a novel and intriguing area of study that explores the application of adversarial images to various structures in our environment. The extensions we make on the Universal method are not only a technical advancement but also a step towards realizing the broader vision of adversarial architecture.

Popular adversarial techniques are currently susceptible to any form of transformation to an image, whether it be deformations or shifts in the view points. Resulting alterations to the image can lead to attack perturbations being rendered ineffective. The universal perturbation method was presented as a direct solution to this problem. The method proved to be successful in creating a singular perturbation which could attack a myriad of images. Our targeted universal perturbation maintains the same advantages as the aforementioned untargeted method while also having the added benefit of targeting specific class labels. We continue to offer a single perturbation that remains effective across multiple angles of an object. Additionally, this work introduces novel alterations to the untargeted universal perturbation algorithm that improves the robustness and versatility of adversarial attacks, enabling precise and controlled adversarial manipulation.

Utilizing 1,210 images from 121 diverse rendered 3D objects, experiments emphasize the effectiveness of the proposed method in both targeted and untargeted instances. Our untargeted developments fall in line with conclusions drawn in previous adversarial research. The universal perturbation successfully identifies a single adversarial noise for each given set of 3D renders from multiple viewpoints. Our targeted results indicate that targeted universal attacks effectively demonstrate the most potential for practical adversarial distortions over the widest range of epsilon values. Targeted attacks' top-5 accuracies exceed 95% for test images evaluated on the majority of epsilon values tested. Comparatively, the untargeted perturbation was successful in misclassifying the test images below a top-5 accuracy of 10% for epsilon values below 5.

TABLE OF CONTENTS

PAGE

ABSTRACT vi
LIST OF TABLES ix
LIST OF FIGURES x
ACKNOWLEDGMENTS xii
CHAPTER
1 INTRODUCTION 1
1.1 Problem Statement 1
1.2 Definitions 2
1.2.1 White Box Attack 2
1.2.2 Black Box Attack 2
1.2.3 ImageNet 3
1.2.4 OmniObject3D 3
1.2.5 Label Leaking Effect 3
1.3 Objective 3
2 RELATED WORK 5
2.1 Fast Gradient Sign Method 5
2.2 Basic Iterative Method 5
2.3 Carlini-Wagner 6
2.4 Expectation Over Transformation 7
2.5 Momentum Iterative Fast Gradient Sign Method 8
3 METHODS 9
3.1 Untargeted Universal Perturbation 9
3.2 Targeted Universal Perturbation 12
3.3 Classifier Selection 13
3.4 Experimental Setup 13
3.5 3D Model Creation and Rendering in Blender from Multiple Views 14
3.6 Object Selection 15
3.7 Perturbation Based FGSM and BIM 18

3.8 Experimental Processes 18
4 Results 20
4.1 Untargeted Attack Accuracies and Confidences 20
4.2 Targeted Attack Accuracies and Confidences 24
4.3 Discussion and Limitations 27
5 Conclusion 30
BIBLIOGRAPHY 32
APPENDICES
A Github Repository Link 35

LIST OF TABLES

PAGE

4.1 Average soft max predictions of the true label from MobileNetV2 under different untargeted adversarial attacks at various epsilon values for both train and test images. $\epsilon = 0.00$ indicates clean images without an attack. The lower the result, the more successful the attack. 20

4.2 Average softmax predictions of MobileNetV2 under different targeted adversarial attacks at various epsilon values for both train and test images. $\epsilon = 0.00$ indicates clean images without an attack. The higher the result, the more successful the attack. 26

LIST OF FIGURES

PAGE

3.1 The figure provides a visual representation of how common adversarial attacks fail to incorporate multiple viewpoints. 10

3.2 The figure provides a visual representation of how the Universal perturbation can incorporate multiple viewpoints. 10

3.3 This Flow chart displays the process of obtaining train and test images from the 3D object dataset and Blender. The bottom portion of the chart shows how we took a 3D object and rendered it in Blender to gain 10 images. The top portion of the chart depicts the process of gathering the 10 images directly from the dataset. 17

3.4 The top row consists of three rendered images of a pineapple from the 3D object dataset. The bottom row consists of three rendered images of a tractor from Blender. 17

4.1 Top-1 and top-5 *untargeted* accuracies of MobileNetV2, after adversarial attacks with ϵ values ranging from 0.5 to 50 were compared with those for clean Images—unmodified images from the dataset. The accuracies were calculated using a set of 605 *train* images, which were rendered from 121 distinct 3D object models. The figure on the left displays the top-1 accuracies while the figure on the right shows the top-5. 21

4.2 Top-1 and top-5 accuracies of MobileNetV2, after adversarial attacks with epsilon ϵ values ranging from 0.5 to 50 were compared with those for clean images—unmodified images from our dataset. The accuracies were calculated using a set of 605 *test* images, which were rendered from 121 distinct 3D object models. The figure on the left displays the top-1 accuracies while the figure on the right shows the top-5. 22

4.3 Each FGSM attack at different epsilon values. 23

4.4 Figure consists of three rendered images of a strawberry with each adversarial attack at the same epsilon value. At $\epsilon = 3.0$ the perturbation is still quite transparent. At higher epsilon levels, the noise becomes more prominent. 24

4.5 Top-1 and top-5 accuracies of MobileNetV2, after **targeted** adversarial attacks with epsilon ϵ values ranging from 0.5 to 50, were compared with those for clean images—unmodified images from our dataset. The accuracies were calculated using a set of 605 **train** images, which were rendered from 121 distinct 3D object models. 25

4.6 Top-1 and top-5 accuracies of MobileNetV2, after **targeted** adversarial attacks with epsilon ϵ values ranging from 0.5 to 50, were compared with those for clean images—unmodified images from our dataset. The accuracies were calculated using a set of 605 **test** images, which were rendered from 121 distinct 3D object models. 25

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my thesis advisor, Professor Mehmet Ergezer. His consistent support, and unwavering patience have been invaluable throughout this journey. Our weekly meetings were a cornerstone of my progress, providing not only guidance but also motivation. Professor Ergezer's dedication to my success and his willingness to share his extensive knowledge have made a significant impact on my work. I would also like to thank my family and friends for their endless encouragement and understanding. Their belief in my abilities kept me going even during the most challenging times.

Finally, I extend my appreciation to Professor Schuster, Professor Kreimendahl, and Professor Qranfal who have offered their endless support and patience in this long process. It has felt like an eternity since I started this research and their nonstop encouragement has played a crucial role in shaping this thesis.

Thank you all for your contributions and support.

CHAPTER 1
INTRODUCTION

Adversarial Images are a vital topic within the field of artificial intelligence and machine learning. These distinctively designed images are made to manipulate a machine learning model by exploiting vulnerabilities in the model's decision making process. Specifically, these images are deliberately crafted so that they appear harmless to a human observer, while additionally leading a model to make incorrect predictions. Adversarial images accentuate the susceptibility that machine learning algorithms have from a security perspective. Due to this, it is important to analyze the methods that can be used to deceive the quickly evolving models of today.

Previous research submitted to the International Conference on Artificial Intelligence and Applications focused on crafting a single noise perturbation that could be applied to various views of the same object [1]. This method demonstrated the feasibility of creating adversarial images that remain effective across different perspectives, thus addressing a critical challenge in the field. However, the scope of this method was limited to non-targeted adversarial attacks, which, while impactful, did not fully explore the potential applications of adversarial images in more specific and controlled scenarios.

We also highlight that motivations for this research and the Universal method derived from previous studies done to apply adversarial techniques to the world of architecture [2]. At a high level, Adversarial architecture explores scalable approaches to design adversarial surfaces for physical objects. The long-term vision of this work being to embed layers of information onto physical surfaces that are only perceptible to machines. This idea presents exciting possibilities for enhancing security, privacy, and communication in ways that were previously unimaginable. We hope that the work done in this paper can serve as a building block for the purpose of realizing adversarial architecture's true potential.

1.1 Problem Statement

In recent years, deep learning has revolutionized various fields, including 3D object recognition [3, 4, 5, 6, 7], where it plays a crucial role in applications like augmented reality [8, 9, 10], autonomous driving [11, 12, 13], and robotics [14, 15, 16]. However, a major challenge lies in the vulnerability of deep learning models to

adversarial attacks [17]. These attacks entail generating imperceptible noise that, when added to a model's input, leads the model to make incorrect predictions. This poses serious security and safety concerns, particularly in critical applications like self-driving cars where misclassification can have catastrophic consequences [18, 19]. Ensuring adversarial robustness, the ability of models to resist such attacks is crucial for real-world deployments.

Conventional adversarial attacks primarily focus on crafting perturbations for single views of objects, exploiting limitations in 2D image understanding [20, 21]. These attacks face limitations when applied to 3D objects, as they often fail to transfer across different viewing angles and lack robustness to real-world variations in perspective. This obstacle motivated recent research aimed at creating multi-view adversarial attacks, capable of fooling object recognition models across diverse poses and viewpoints. The Universal perturbation method was proposed as a potential solution this problem and found substantial success on a limited amount of data [1]. There is an essential need to assess the algorithm's performance on significantly more objects to validate the credibility of such adversarial techniques. While the Universal method can seemingly achieve robust noise transferable across numerous angles, we extend the technique to function in targeted scenarios. This addition will provide us with the ability to better manipulate the perturbation by guiding it to a specific target label.

1.2 Definitions

1.2.1 White Box Attack

A white-box adversarial attack is one in which the attacker possesses full knowledge of the target model's architecture, parameters, and training data. The attack is considered a white box because of the internal information (i.e. weights, gradients) that is accessible for the perpetrator to thoroughly analyze. The main benefit of using white-box attacks is the ability to utilize information such as gradients with respect to the input images to steer the perturbation process towards a specified target label. This enables the production of very effective and tailored adversarial attacks for a variety of machine learning models. Some examples of popular white-box attacks include the Fast Gradient Sign Method (FGSM), the Basic Iterative Method (BIM), and the Carlini-Wagner L2 attack.

1.2.2 Black Box Attack

A black-box adversarial attack means the attacker has little or no information of the target model. In this configuration, the attacker conceptualizes the model as a

"black box" and has no access to important details such as model architecture, parameters, or training data. Often times, black-box attacks will rely on approaches that aim to estimate gradients or other information about the target model without direct access to them.

1.2.3 ImageNet

ImageNet is one of the largest labeled datasets utilized for object recognition, containing millions of labeled images. The dataset originally had more than 20,000 categories, but most machine learning models are trained on a subset of about 1,000 categories for classification tasks. The MobileNetV2 model used for the experiments in this paper was directly trained with this data.

1.2.4 OmniObject3D

OmniObject3D is a large 3D object dataset consisting of 6,000 objects in 190 different class labels. The dataset has multiview 2D rendered images along with 3D point clouds and real captured videos [22].

1.2.5 Label Leaking Effect

Label leaking is a particular phenomenon in adversarial machine learning where adversarial attacks perform better on adversarial examples than clean images. The problem is commonly seen in one-step attacks such as FGSM and BIM. It is discussed further in many past experiments in the field [23].

1.3 Objective

The objective of this thesis is two fold. First, to undertake an in-depth analysis of popular adversarial methods [18, 24, 25, 23]. This study will obtain insights into the strengths, flaws, and potential countermeasures associated with these attacks by evaluating their performance on images derived from 3D rendered objects.

Secondly, we propose an extension to the novel "Universal perturbation" method for generating robust multi-view adversarial images. Our approach alters the existing algorithm to allow the perturbation to excel in targeted environments. Similarly to the untargeted Universal method, our process departs from traditional per-view attacks by crafting a single noise perturbation applicable to various views of the same object. This single-noise, multi-view attack offers the same advantages to the referenced adversarial technique [1]. Furthermore, the main contributions of this paper include:

Targeted Application: By slightly tweaking the Universal algorithm, we allow far more control over the model's output. The updated method will be able to target any class that is present in our classifier.

Robustness: The trained noise is effective across diverse viewing angles, allowing more robust adversarial attacks compared to single-view methods. The untargeted Universal perturbation was originally validated on five objects, consisting of 5 train and 5 testing images. This paper will extend previous research to encompass 121 objects for a total of 605 train and 605 test images.

This paper investigates the effectiveness of our Universal perturbation method in comparison to conventional single-view attacks in both the targeted and untargeted sense. We conduct comprehensive experiments on various 3D object datasets, evaluating the attack success rates, and transferability across different views. We believe this work represents a significant step towards developing efficient and robust adversarial attacks for 3D object recognition, paving the way for improved model security and robustness in real-world applications.

The paper is organized as follows: Section 2 introduces related adversarial attack methods, focusing on single-view attacks. The Universal perturbation algorithm is formulated in Chapter 3 along with our proposed targeted modifications. Section 3.4 and Section 4 present the setup for our experiments, including object selection and rendering, adversarial attacks, related metrics, and results. Section 4.3 discusses the potential and limitations of Universal perturbation. Chapter 5 provides conclusion remarks.

CHAPTER 2
RELATED WORK

In this section, we provide an overview of the existing adversarial algorithms that are designed for single-view attacks.

2.1 Fast Gradient Sign Method

One of the earliest adversarial attack methods, Fast Gradient Sign Method (FGSM), was proposed in 2014 [18]. The method was popular for forging adversarial examples due to its simplicity in generation and implementation and relatively low cost of execution. FGSM attacks rely heavily on knowing the architecture and the weight of the victim model—known as a white-box attack—which in turn may make the method poor when it comes to transferability and in black-box settings where only the input features are known.

FGSM works by first calculating the loss based on the predicted class after performing forward propagation. We determine the gradients with respect to the input image and update the image's pixels in the direction that maximizes the loss. In Equation 2.1, we let X and y be the input image and true label respectively. We assume X as a 3-D matrix (width $\times$ height $\times$ color). ϵ represents a constant that determines the strength of the perturbation. $\nabla_X J(X, y_{true})$ is the gradient of the model's loss with respect to X. The perturbation is calculated by taking the sign of this gradient and adding it to the original image. We initialize $X_0^{adv} = X$.

$$X^{adv} = X + \epsilon \cdot \text{sign}(\nabla_X J(X, y_{true})) \tag{2.1}$$

While in this study we focus primarily on untargeted adversarial attacks, the FGSM algorithm can be updated to target specific class labels. This can be accomplished by simply changing the direction of the gradients to minimize the loss between the targeted and predicted labels as shown in Equation 2.1.

$$X^{adv} = X + \epsilon \cdot \text{sign}(\nabla_X J(X, y_{target})) \tag{2.2}$$

2.2 Basic Iterative Method

Kurakin et al. proposed several new iterative versions of FGSM, including the Basic Iterative Method (BIM) and iterative least-likely class method (ILCM) [23]. Both of these methods brought adversarial attacks to the physical world by inputting captures from a mobile phone camera, instead of feeding source images directly to the AI model. ILCM extends BIM by allowing the attacker to specify a desired target label for the attacked object.

The Basic Iterative Method (BIM) extends the idea of FGSM by applying it iteratively, allowing for a more fine-tuned manipulation of the input image. This iterative approach often results in more effective adversarial examples than single-step perturbation. Note that intermediate results are clipped at a pixel level after each step. Just like with FGSM, we initialize BIM, $X_0^{adv} = X$. N in Equation 2.2 represents the iteration count.

$$X_{N+1}^{adv} = \text{Clip}_{X,\epsilon}\{X_N^{adv} + \epsilon \cdot \text{sign}(\nabla_X J(X_N^{adv}, y_{true}))\} \tag{2.3}$$

2.3 Carlini-Wagner

The Carlini and Wagner attack treats the means of creating adversarial examples as an optimization issue [24]. It was developed by Nicholas Carlini and David Wagner. The goal is to identify a perturbation to the input data that causes the target neural network to misclassify while limiting the disturbance's perceptibility. The assault consists of solving an optimization problem with numerous objectives and constraints.

The model revolves around two different loss functions: An adversarial loss capable of making the input image adversarial and an image distance loss restrict the perturbation from becoming too apparent. Our goal is to minimize:

$$D(x, x + \sigma) \tag{2.4}$$

such that

$$C(x + \sigma) = t \tag{2.5}$$

and

$$x + \sigma \in [0, 1]^n \tag{2.6}$$

Where x is the input image, σ is the perturbation, n is the dimension of the input image, and t is the class we are targeting. D is a function that determines the distance between the input and adversarial image while C serves as a classifier function.

The traditional method for solving this optimization problem is to create an objective function and then use gradient descent to take us to an optimal point.

However, the aforementioned formula is difficult to solve since the classifier portion of the formula $C(x + \sigma)$ is significantly non-linear.

We use the objective function to inform us on how close the adversarial image is to the target class. Through various forms of testing, Carlini and Wagner found the most optimal function to be:

$$f(x') = max(max\{Z(x')_i : i \neq t\} - Z(x')_t, -k) \tag{2.7}$$

where $Z(x')$ is the probability predictions for each class for the inputted adversarial image x'. $max\{Z(x')_i : i \neq t\}$ tells us the probability of the targeted class. So as follows, $max\{Z(x')_i : i \neq t\} - Z(x')_t$ is the difference between two probabilities: what the model thinks the adversarial image is and what we want the model to think it is. The value of $-k$ is just used as a lower bound for the loss. Carlini and Wagner also introduced a substitution of variables to handle the constraint of $x + \sigma \in [0, 1]^n$. They let:

$$x + \sigma = \frac{1}{2}(\tanh(w) + 1) \tag{2.8}$$

This means that the when $x + \sigma$ is between 0 and 1, $\tanh(w)$ will be between -1 and 1. Finally, we get our final optimization problem which simplifies to:

$$D(\frac{1}{2}(\tanh(w) + 1), x) + c * f(\frac{1}{2}(\tanh(w) + 1)) \tag{2.9}$$

Where $f(x') = max(max\{Z(x')_i : i \neq t\} - Z(x')_t, -k)$

The Carlini and Wagner attack is the solution to the above optimization function with respect to w. They utilize the Adam Optimizer to solve the given problem.

2.4 Expectation Over Transformation

The purpose of the Expectation Over Transformation (EOT) framework is to evaluate and enhance a model's resilience by examining the expectation of its predictions over a distribution of potential transformations. It was first proposed by Anish Athalye and from a high level, this algorithm was developed to aid in the goal of creating adversarial examples that are adversarial over multiple transformations [25].

Instead of minimizing the distance between adversarial examples and the original input image, EOT takes the expected distance between the two and constrains it below a particular threshold. We also try to find an adversarial example x' that maximizes the probability for the targeted class across the distribution of potential

transformations of the input image x. Just as in Carlini and Wagner's work, we frame this as an optimization problem that encapsulates all of the defined constraints:

$$\arg\max_{x'} \mathbb{E}_{t\sim T}[\log P(y_t|t(x'))]$$

Where:

$$\mathbb{E}[D(t(x'), t(x))] < \epsilon$$

and $x \in [0,1]^n$

2.5 Momentum Iterative Fast Gradient Sign Method

Due to low success rates with adversarial attacks in black-box settings, Dong et al. proposed a class of momentum-based iterative algorithms to boost adversarial attacks called MI-FGSM [26]. This class of adversarial attacks can be used for white-box attacks while also outperforming other one-step gradient methods in a black-box setting.

The primary motivation behind this method is to memorize previously seen gradients at each iteration of the attack in hopes of stabilizing the optimization process. Across iterations, retaining gradients allows for accelerated performance through the deviation from local minima. At a low level, the algorithm can be summarized by solving the following optimization problem where x is the original image, x^* is the perturbed image, and ϵ is the size of the perturbation:

$$\arg\max_{x^*} J(x^*, y), s.t. \qquad ||x^* - x||_\infty \leq \epsilon \tag{2.10}$$

The momentum portion of the iterative attack is implemented by dividing the gradients with respect to the image by the L1 norm of that same vector. We let g_t gather the previous gradients and multiply it by a decay factor of μ before adding it to the former result.

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x_t^{\text{adv}}, y^{\text{true})}}{||\nabla_x J(x_t^{\text{adv}}, y^{\text{true})}||_1} \tag{2.11}$$

The adversarial image is updated through a similar process to 2.1. We use the resulting sign of g_{t+1} to perturb the example for numerous iterations.

$$x_{t+1}^{adv} = \text{Clip}_{x,\epsilon}\{x_t^{adv} + \epsilon \cdot \text{sign}(g_{t+1})\} \tag{2.12}$$

CHAPTER 3
METHODS

In this chapter we reintroduce the Universal perturbation method and detail the enhancements made to existing algorithm which allow it to perform in targeted environments. We first describe the fundamental principles of the original work, highlighting how the untargeted Universal perturbation overcomes common limitations of popular adversarial attacks. Building upon this we establish our novel modifications that adapt the model for targeted use cases.

Our methodology is then validated through a series of experiments in the following chapter. These tests are designed to assess the performance of the improved algorithm on a significantly larger amount of 3D objects than those done in previous research. In doing so we provide a comprehensive analysis of the Universal method's effectiveness and applicability.

3.1 Untargeted Universal Perturbation

Figure 3.2 illustrates how the typical process of implementing adversarial noise involves crafting a unique perturbation for a particular image [1]. The subsequent noise is added to the image and fed through a classifier leading to an incorrect prediction of the class label. As seen in figure 3.1, attempting to apply this perturbation to separate images leads to inconsistent outputs by our classifier, rendering the noise useless.

The Universal perturbation was originally proposed to overcome this obstacle. The goal was to develop one distinct perturbation that can be applied to numerous images. A single noise was created that could be robust against a variety of object transformations including rotations, changes in lighting, and potential deformations.

The methodologies outlined in Section 2 are all tailored to attack a solitary image, denoted as X, generating a corresponding adversarial noise, X^{adv}, for each individual input. The Universal perturbation's goal was to extend single-view attacks by devising a singular perturbation that can be Universally applied across various perspectives of one or more objects, $\mathbb{X}^{\text{adv}}$.

To achieve this, a modification was made to the Basic Iterative Method, outlined in section 2.2. Instead of computing gradients with respect to the input image, researchers decided to take the gradients with respect to the adversarial noise itself [1]. This adjustment separated the number of input images from the shape of the generated

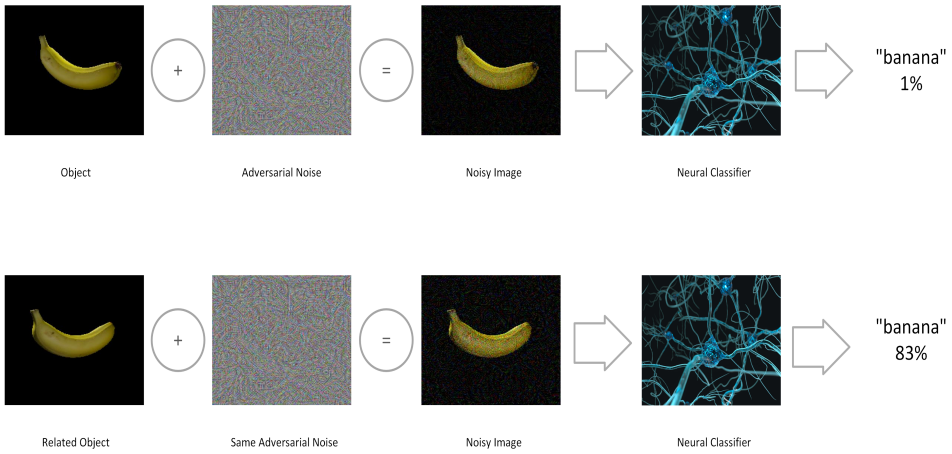

Figure 3.1. The figure provides a visual representation of how common adversarial attacks fail to incorporate multiple viewpoints.

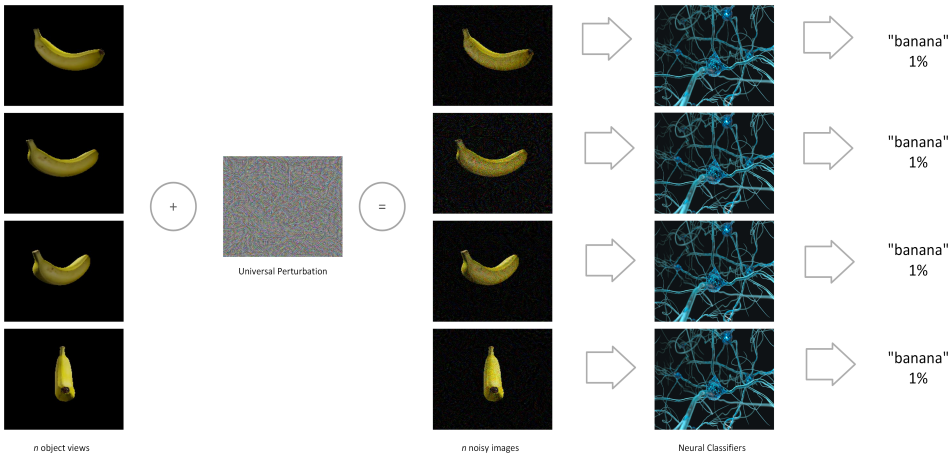

Figure 3.2. The figure provides a visual representation of how the Universal perturbation can incorporate multiple viewpoints.

adversarial noise. As a result, it allowed the attacker to control both the input image's dimensions and the associated noise, independently. This update allowed simultaneous inputs of various views of the same object. The new algorithm provided solitary adversarial noise capable of collectively compromising the recognition of all these distinct perspectives at once.

Equation 3.1 re-introduces the Universal perturbation calculation where X is stacked input images as a 4-D tensor (image count $\times$ width $\times$ height $\times$ color), $\mathbb{X}$ is the calculated perturbation as a 3-D matrix (width $\times$ height $\times$ color), matching the shape of an individual image in X. N is the number of desired iterations as defined in BIM, costing calls to the backpropagation. ϵ is a hyperparameter controlling the scale of the attack as used in FGSM. $\nabla_{\mathbb{X}_N} J(\mathbb{X}_N^{\text{adv}}, y_{\text{true}})$ is the gradient of the cross-entropy loss function of the model with respect to the calculated perturbation.

$$\mathbb{X}_{N+1}^{\text{adv}} = \text{Clip}_{X,\epsilon}\{\mathbb{X}_N^{\text{adv}} + \epsilon \cdot \text{sign}(\nabla_{\mathbb{X}_N} J(\mathbb{X}_N^{\text{adv}}, y_{\text{true}}))\} \tag{3.1}$$

Unlike BIM, the perturbation is initialized as $\mathbb{X}_0^{adv} = X + r$ where $r \sim U(-0.01, 0.01)$. Section 4.3 discusses the reasoning and intricacies associated with this initialization.

Algorithm 1 Generate Untargeted Universal Perturbation

Input: images of object(s) from different views

Parameter: target_model, $\epsilon = 0.01$, num_iterations=1, regularization=1., clip_value=1.

Output: Adversarial noise

perturbation $\leftarrow$ random.uniform(shape=images[0].shape, range= $\pm$0.01)
for *it* in range(num_iterations) **do**
    Compute gradient(*perturbation*)
        adv_images $\leftarrow$ *images* + *perturbation*
        predictions $\leftarrow$ *target_model*(*adv_images*)
        loss $\leftarrow$ crossentropy(predictions, true label)
        loss += *regularization* $\cdot$ norm(*perturbation*)
        gradients $\leftarrow$ gradient(*loss*, *perturbation*)
        gradients $\leftarrow$ clip(*gradients*, *clip_value*)
        perturbation += $\epsilon \cdot$ sign(*gradients*)
        perturbation $\leftarrow$ clip(*perturbation*, *clip_value*)
end for
return *perturbation*

3.2 Targeted Universal Perturbation

In this section we extend our previously proposed Universal perturbation method to account for targeted use cases. Similarly to Eq. 3.1 we make the same alterations to the Basic Iterative Method. This involves computing gradients with respect to the adversarial noise itself rather than computing them with respect to the input image.

To accomplish this goal we alter Eq. 3.1 by computing the cross-entropy loss between the target label and the predicted adversarial example. We take the gradient of the loss with respect to the calculated perturbation and multiply it by negative one. This ensures that through each training iteration the perturbation loss is being minimized. Algorithm 2 details the steps taken to implement our targeted Universal perturbation method.

$$\mathbb{X}_{N+1}^{\text{adv}} = \text{Clip}_{X,\epsilon}\{\mathbb{X}_N^{\text{adv}} - \epsilon \cdot \text{sign}(\nabla_{\mathbb{X}_N} J(\mathbb{X}_N^{\text{adv}}, y_{\text{target}}))\} \tag{3.2}$$

The approach departs from traditional solitary attacks by crafting a single noise perturbation applicable to various views of the same object. In addition to the benefits explained in Section 1.3 that this approach provides, there are further advantages we can add due to the extension into targeted environments.

Precise Label Control: In targeted adversarial attacks, the goal is to alter the input images in a way that the model outputs a distinct, predefined incorrect prediction. This gives us precise control over the classifier's output which is useful in situations where a specific target label is desired in the model's prediction.

Applications in Fine Tuning: Targeted adversarial attacks can aid in fine-tuning machine learning models to be robust against perturbations that might be more likely in real-world scenarios.

Algorithm 2 Generate Targeted Universal Perturbation

Input: images of object(s) from different views

Parameter: target_model, $\epsilon = 0.01$, num_iterations=1, regularization=1., clip_value=1.

Output: Adversarial noise

perturbation ← random.uniform(shape=images[0].shape, range= ±0.01)
for *it* in range(num_iterations) **do**
    Compute gradient(*perturbation*)
        adv_images ← *images* + *perturbation*
        predictions ← *target_model*(*adv_images*)
        loss ← crossentropy(predictions, target label)
        loss += *regularization* · norm(*perturbation*)
        gradients ← gradient(*loss*, *perturbation*)
        gradients ← clip(*gradients*, *clip_value*)
        perturbation += $-\epsilon \cdot$ sign(*gradients*)
        perturbation ← clip(*perturbation*, *clip_value*)
end for
return *perturbation*

3.3 Classifier Selection

MobileNetV2 is a convolutional neural network architecture that utilizes lightweight depthwise convolutions to filter out features through the intermediate layers [27]. This proves advantageous in forming a balance between having a model that is both computationally scarce and accurate. MobileNetV2 is often chosen as an architecture implemented in mobile and embedded devices due to its proven effectiveness in image classification tasks. This model was chosen as our neural classifier used for evaluating the performance of the adversarial perturbation. Due to its efficient architecture composing of convolutional and bottleneck layers along with its prominent adoption in real-world applications, we felt that it made a logical choice to use in our experiments.

3.4 Experimental Setup

In this section, we describe the system setup employed to automatically generate multi-view renders of objects and how we evaluate the FGSM, BIM, untargeted, and targeted Universal perturbation attacks. The analysis of our results will be given in the next chapter.

The experiments detailed in the following sections will substantiate our approach by assessing its decrease in true class accuracy, increase in target class accuracy, transferability across angles, and noise imperceptibility, while comparing the outcomes with those obtained using FGSM and BIM. Additionally, we will showcase the computational efficiency of our model in comparison to FGSM and BIM for generating transferable multi-angle noise on an expansive dataset. We also detail the same experiments with our targeted perturbation by assessing its performance through the target label's accuracy.

To showcase the effectiveness of our model in generating robust noise across multiple angles, we conducted experiments comparing our Universal perturbation method with common adversarial attack methods: FGSM and BIM. The objective was to identify the optimal noise level that significantly reduces a classifier's confidence in the object's true class, while also increasing the target class' confidence when necessary. By accomplishing these tasks we aimed to validate the previously proposed Universal perturbation method on a vast number of 3D objects, while also improving the given algorithm to allow control of the attack's output.

Algorithm 3 provides an overview of our experimental process. The Universal perturbation approach generates a single noise pattern applicable to all angles, resulting in consistent adversarial outcomes. Our process resembles that of the original Universal method [1]. We start by rendering our 3D objects in Blender to gain multiple 2D images at different viewpoints. Depending on the object, we take 2D images straight from a 3D rendered dataset. The images are split into two sets, one training and one test. A single perturbation is developed, applicable to all images in the training set. The noise is added to all images before being fed through MobileNetV2. All steps are repeated for the BIM and FGSM attacks. Each experiment is done twice to incorporate both targeted and untargeted use cases. It is important to note that, similarly to popular methods such as Expectation Over Transformation [25], the Universal Perturbation bridges the gap between 2D and 3D spaces. To extend the Universal attack, we evaluate our targeted method alongside FGSM and BIM We conducted multiple runs of FGSM and BIM to generate noise for various angles of our objects. Sec 3.5 provides more insights to the means of rendering our 3D objects.

3.5 3D Model Creation and Rendering in Blender from Multiple Views.

To generate diverse images of objects, we employed Blender and various 3D models. For exactly eight of the object we experimented on, we rendered images from

Algorithm 3 Experimental System

Input: 3D Blender model with texture or 2D images from dataset

Parameter: ϵ, *n_angles* to render, num iterations

Output: Adversarial image and its classification

1: Render object from multiple angles, *n_angles*, in Blender or directly from dataset.
2: Split *n_angles* images to two: tr_n to generate noise and ts_n to test the generate noise(s)
3: Identify Universal perturbation for tr_n views to find a single *perturbation* for all tr_n
4: Generate adversarial images: $ts_n^{adv} \leftarrow ts_n + (\epsilon \cdot perturbation)$
5: Classify the adversarial ts_n^{adv} images with MobileNetV2
6: Repeat above for FGSM and BIM, except each tr_n gets a unique *perturbation* calculated.
7: **return** top-1 and top-5 accuracy for these attacks
8: Repeat experiment for target label

ten distinct viewing angles, ensuring consistent recognition by our classification model across perspectives. The remaining objects were gathered directly from 3D render datasets consisting of multiple images of an object at different angles. Prior to running any experiments we ran sanity checks on all rendered images to ensure that the true label was correctly classified by MobileNetV2. For the rendered images gathered through Blender, camera angles were randomly selected based on a spherical coordinate system centered around the object, incorporating a 15% random deviation for robustness. The slight changes in the coordinates mimicked natural variations in viewing angles encountered in real-world environments. Consequently, constant lighting positions relative to the camera angles, generated shadowed areas that hindered initial MobileNetV2 recognition. To combat such challenges it was necessary to restrict the 15% deviation on certain objects to one axis.

3.6 Object Selection

To validate our methods, we randomly selected eight diverse 3D objects: baseball [28], snail [29], acorn [30], conch [31], pretzel [32], lemon [33], broccoli [34], and tractor [35], that were readily renderable on Blender from various viewpoints. These objects offered varying shapes, sizes, and textures, representing a range of potential real-world applications. While the 8 objects were randomly selected, many of the initial objects found in the dataset had to be removed because they did not have a label found

in Imagenet which was the dataset our classifier was trained on [36]. We proceeded to randomly sample each object until we reached a total of 8 classifiable figures with labels found in the Imagenet.

Our pipeline utilizes Blender to generate multi-angle views of objects, at 224 by 224 resolution, suitable for our image classification model, MobileNetV2. We rendered ten images of each object from different angles. These angles were determined using sinusoidal and cosinusoidal functions, ensuring even distribution across the viewing sphere. Additionally, random rotations were introduced to simulate natural object orientations. It is important to note that not all of the images produced by our Blender script were able to be classified properly by MobilenetV2. For certain objects, Imagenet's training data was not robust enough to allow the rendered images to be classified at any angle [36]. Due to this issue, on some of the objects we had to restrict the angles to a specific axis. This allowed the rendered images to be classified as the true label prior to the experimentation process.

To increase the number of test objects and further validate the conclusions from our results, we utilized a 3D dataset consisting of multiple 3D objects and rendered images at various angles of each figure [22]. 113 objects were selected from this dataset consisting of 6 distinct class labels: backpack, banana, dumbbell, pineapple, strawberry, teddy bear. Each object in the dataset had 100 images taken at various angles circling the object. Much like the renders generated from Blender, these images were rendered using the same sine and cosine functions across spherical view points. For this experiment, we selected the first ten of the one hundred images that classified as the true label. 3.4 shows examples of the images captured from blender renderings of a tractor and a pineapple from the 3D object dataset.

Figure 3.3 summarizes the process used to retrieve the images used for our experiments. One path of the flowchart depicts how we first selected a 3D object before rendering it in Blender to 2D images from 100 different angles. Another path illustrates the process of collecting images directly from the rendered image dataset. Both paths lead to us verifying that the images classified correctly through MobileNetV2. The last stage of the process was to select the first 10 images that got predicted as the true label. The 10 images got split into two sets of five: one for training the perturbation and one for testing its effectiveness. This process was repeated for all 121 objects until a total of 1,210 images were generated.

In cases of targeted attacks, we included an extra step in the pipeline that involves randomly selecting a label from the 1000 classes available in Imagenet. By doing this our results can avoid unnecessary variance while also avoiding the "label

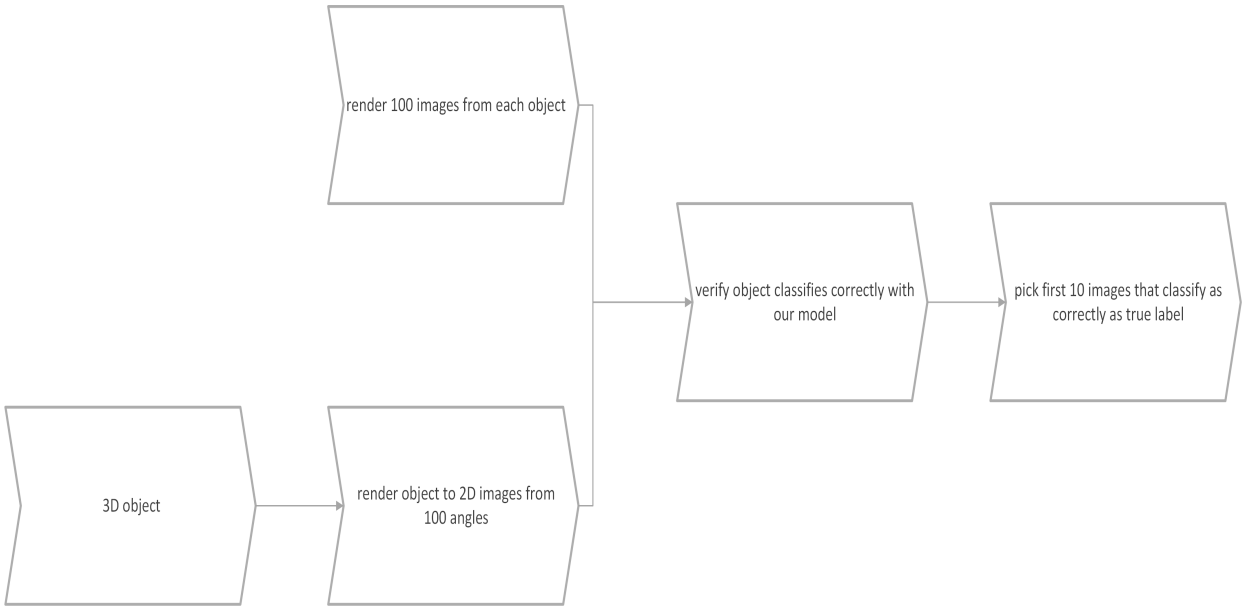

Figure 3.3. This Flow chart displays the process of obtaining train and test images from the 3D object dataset and Blender. The bottom portion of the chart shows how we took a 3D object and rendered it in Blender to gain 10 images. The top portion of the chart depicts the process of gathering the 10 images directly from the dataset.

Figure 3.4. The top row consists of three rendered images of a pineapple from the 3D object dataset. The bottom row consists of three rendered images of a tractor from Blender.

leaking" effect found in previous adversarial machine learning studies [23]. If the randomized target label happened to be equal to the true label, we randomly sampled another label using the same criteria.

3.7 Perturbation Based FGSM and BIM

In our own extension of the fast gradient sign method we build off the foundational formula 2.1. Similar to our Universal perturbation, we focus on taking the gradients of the loss function with respect to the perturbation rather than the image itself. In untargeted instances we aim to maximize the loss, while in targeted spaces we flip the sign of the gradients to minimize. It is important to note that in this version of FGSM we are initializing the perturbation following a uniform distribution.

The same process is applied to our version of the Basic Iterative Method. We iteratively perform FGSM steps while continuously clipping the perturbations between -1 and 1.

3.8 Experimental Processes

When performing our experiments we opted to split the data into train and test sets of 5 images each. This decision was motivated by the limitations in computing resources available for the experimentation processes. The task of training a singular perturbation took well over 40 minutes and had to be repeated for both untargeted and targeted test cases. The time required to execute such tasks on over 120 objects was far beyond the scope of this work. Future studies may involve training each Universal perturbation on significantly more images to aid in creating adversarial noise effective against any transformation. Analyzing results with a variety of train-test splits can also potentially lead to an increase in testing efficiency.

The metric used for evaluating our experiments consisted of top-1 and top-5 accuracy for the train and test sets of each attack. These methods of evaluation provided clear and easy to understand measures of performance. Due to the robustness of classes that MobileNetV2 was trained on, we decided to include Top-5 accuracy to offer additional insight to each attack's performance. Top-5 accuracy is essentially equivalent to to top-1 accuracy. We just extended the definition of a successful classification to any prediction where the true label was in the top-5 predictions. Targeted experiments followed the same metrics as untargeted ones. The only difference in the targeted cases was we swapped out the true label for the target label.

$$\text{Top-1 Accuracy} = \frac{\text{\# of Top-1 True Label Classifications}}{\text{Total \# of Classifications}}$$

$$\text{Top-1 Target Accuracy} = \frac{\text{\# of Top-1 Target Label Classifications}}{\text{Total \# of Classifications}}$$

CHAPTER 4

Results

ϵ	Train Images $\times 10^{-6}$			Test Images $\times 10^{-6}$		
	FGSM	BIM	Universal	FGSM	BIM	Universal
0.00	509757.92	509757.92	496621.40	**519231.59**	**519231.59**	**507621.61**
0.50	78091.40	0.01	16.58	491976.42	258959.30	**78269.06**
1.00	35697.12	0.00	0.04	472773.68	143625.36	**26230.52**
3.00	21658.82	0.00	0.00	446958.38	10037.76	**2386.95**
5.00	20657.97	0.00	0.00	394593.91	444.74	**3.03**
10.00	17980.33	0.00	0.00	230679.69	**2.33**	262.84
15.00	12265.05	0.00	0.00	109324.64	**38.94**	5403.15
30.00	1651.91	0.22	0.24	4924.44	**1.99**	49.57
50.00	464.07	3.36	1.69	674.42	**0.11**	3.36
Mean	77580.51	56640.17	55182.22	296793.02	103590.48	**68279.45**
Std (±)	163718.35	169919.16	165539.70	213028.22	180558.74	**166786.07**

Table 4.1. Average soft max predictions of the true label from MobileNetV2 under different untargeted adversarial attacks at various epsilon values for both train and test images. $\epsilon = 0.00$ indicates clean images without an attack. The lower the result, the more successful the attack.

4.1 Untargeted Attack Accuracies and Confidences

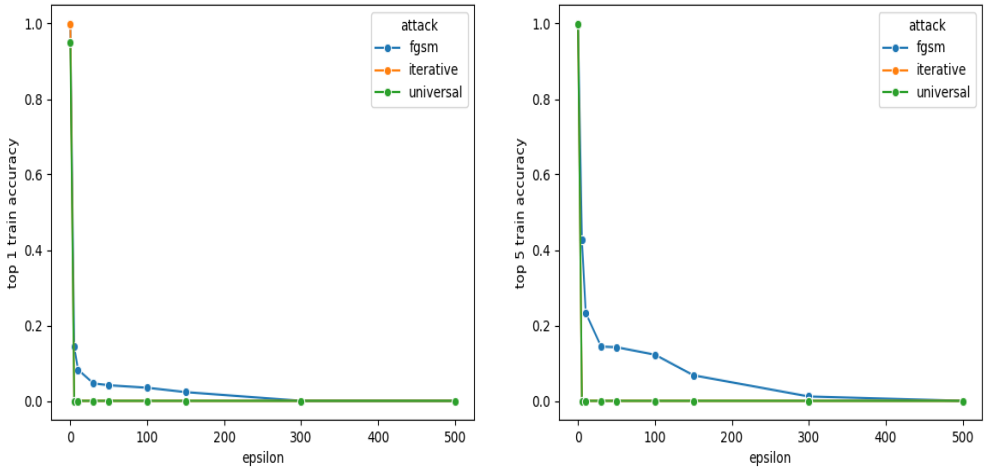

Figure 4.1. Top-1 and top-5 *untargeted* accuracies of MobileNetV2, after adversarial attacks with ϵ values ranging from 0.5 to 50 were compared with those for clean Images—unmodified images from the dataset. The accuracies were calculated using a set of 605 *train* images, which were rendered from 121 distinct 3D object models. The figure on the left displays the top-1 accuracies while the figure on the right shows the top-5.

Table 4.1 summarizes the average softmax predictions of the true object label when images are subjected to different adversarial attacks. The comparison encompasses three attack methods: FGSM, BIM, and our proposed Universal perturbation method. We split the table in two to analyze the robustness of these attacks across 605 training and 605 unseen images corresponding to multiple poses of our 121 objects. BIM and our Universal Perturbation method both adhere to 20 iterations while FGSM is limited to one iteration.

In Table 4.1 we note that the Universal and basic iterative attacks provide the highest success at the lowest noise levels. At ϵ= .50 all three attacks exhibit impressive performance on their training images. Although the difference is marginal, both the BIM and Universal attacks have the lowest average soft-max predictions. This pattern can be seen through all epsilon values from .5 to 30. At $\epsilon = 30$ each attack sees a slight shift in performance on the training images. BIM and Universal jump from 0.00 to 0.22E-06 and 0.00 to 0.24E-06 respectively. FGSM also sees a decline in its average softmax prediction dropping from 12265.05E-06 to 1651.91E-06. Similar changes are reflected at $\epsilon = 50$.

For the test images, the best attack result for each ϵ, corresponding to the lowest soft-max prediction of the true label, is highlighted in bold. While the performance on the training set is exceptional, when we look at the results of our test images, the

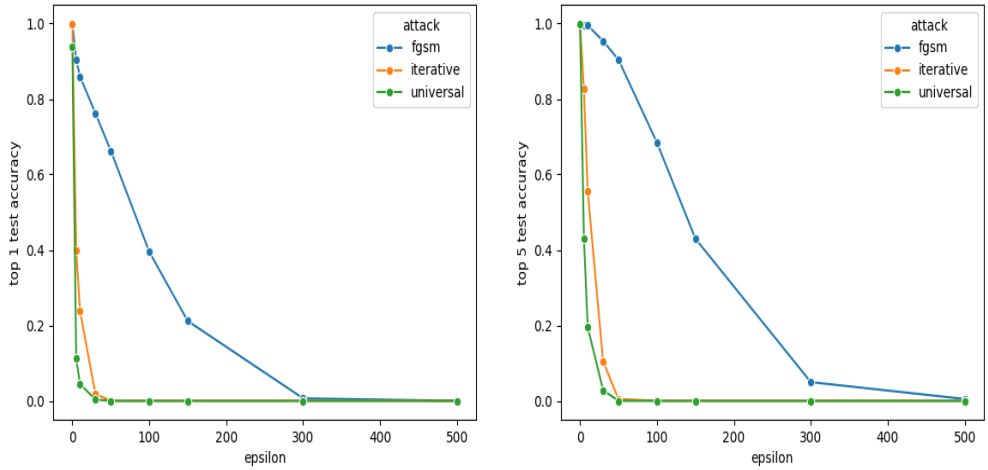

Figure 4.2. Top-1 and top-5 accuracies of MobileNetV2, after adversarial attacks with epsilon ϵ values ranging from 0.5 to 50 were compared with those for clean images—unmodified images from our dataset. The accuracies were calculated using a set of 605 *test* images, which were rendered from 121 distinct 3D object models. The figure on the left displays the top-1 accuracies while the figure on the right shows the top-5.

effectiveness of each attack degrades significantly. From $\epsilon = .50$ to $\epsilon = 5.00$ our Universal method displays the most success out of each attack. At $\epsilon = 10$ and on, the BIM proves to provide the most effective perturbation. The decline in adversarial performance between train and test data is to be expected since the noise is being evaluated on images it hasn't seen before. Despite the changes in results, our Universal attack is still more than adequate for practical applications. Even at its least effective epsilon value ($\epsilon = .50$) the Universal attack is producing an average soft-max prediction of $\tilde{8}\%$ which is enough to deceive a majority neural classifiers. We note that the achievements shown through the Universal attack's results are extended through its robustness to different views.

We also list the mean and standard deviation of true label prediction rates where lower values indicate a more successful attack. Universal perturbation provides the minimum average true level probability across all ϵ values. It is important to emphasize the influence that epsilon has on an image at the highest values. At a certain point, the noise becomes so substantial that the characteristics of the image are unrecognizable by the classifier. This depicts the effectiveness of the proposed algorithm on unseen images at the most challenging ϵ levels. This point further adds to the significance that our Universal method provides at lower epsilon levels.

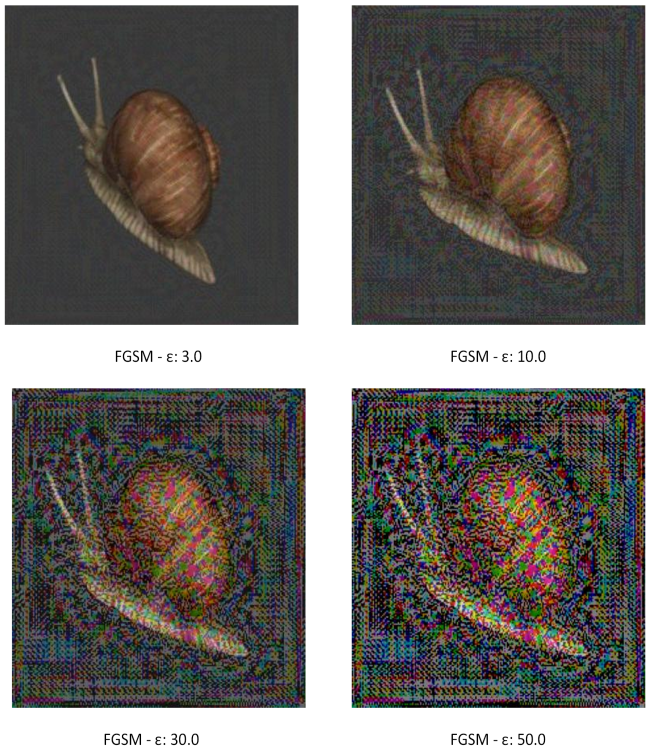

Figure 4.3. Each FGSM attack at different epsilon values.

In Fig. 4.1 and Fig. 4.2, we evaluate the top-1 and top-5 classification accuracies for training and test images, respectively. Sample images of one of our strawberry objects, perturbed at $\epsilon = 3$ or a 3% noise level, are shown in Fig.4.4 to demonstrate the different noise types generated by each algorithm. In figures 4.1 and 4.2 we observe that as noise level, ϵ increases, we achieve successful misclassifications across all models. Both the basic iterative method and Universal attacks exhibited faster degradation in top-1 and top-5 accuracies compared to FGSM.

Fig. 4.1 shows the effectiveness of the FGSM, BIM, and Universal attack on training images, achieving success at particularly low epsilon values. By $\epsilon = 5.0$, all attacks are well below 20% for both top-1 and top-5 accuracies.This suggests that during the training process, the perturbations become exceedingly accustomed to the model, a vulnerability that these attacks are able to exploit efficiently. In Fig. 4.2 we can see that when evaluated on test images, the overall performance of each attack slightly declines. By $\epsilon = 5.0$, the BIM and Universal attacks both converge at a top-1 and top-5 accuracy threshold below 20%. The rate that these two attacks converge at is similar to the training experiments. However, the FGSM attack does not manage to converge to the same threshold until an epsilon of 30. The decrease in performance

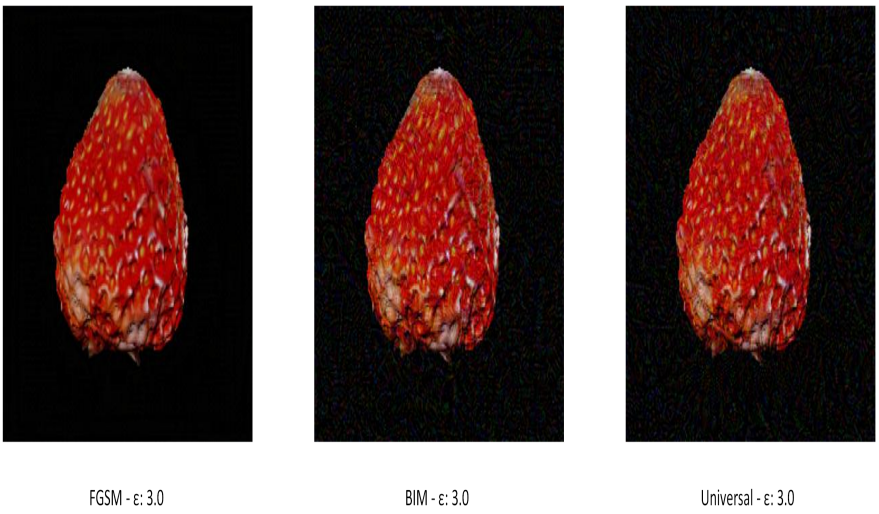

Figure 4.4. Figure consists of three rendered images of a strawberry with each adversarial attack at the same epsilon value. At $\epsilon = 3.0$ the perturbation is still quite transparent. At higher epsilon levels, the noise becomes more prominent.

between train and test data is mainly due to the perturbation being evaluated on images it hasn't seen before.

Similarily, we should note that for all three attacks, as the ϵ increases to 30, they begin to demonstrate marked success on the test images with accuracies plummeting to below 1%. This indicates that as the ϵ value increases, the adversarial modifications become so sufficiently pronounced that they compromise the model's ability to generalize, leading to a significant degradation in performance on unseen data.

4.2 Targeted Attack Accuracies and Confidences

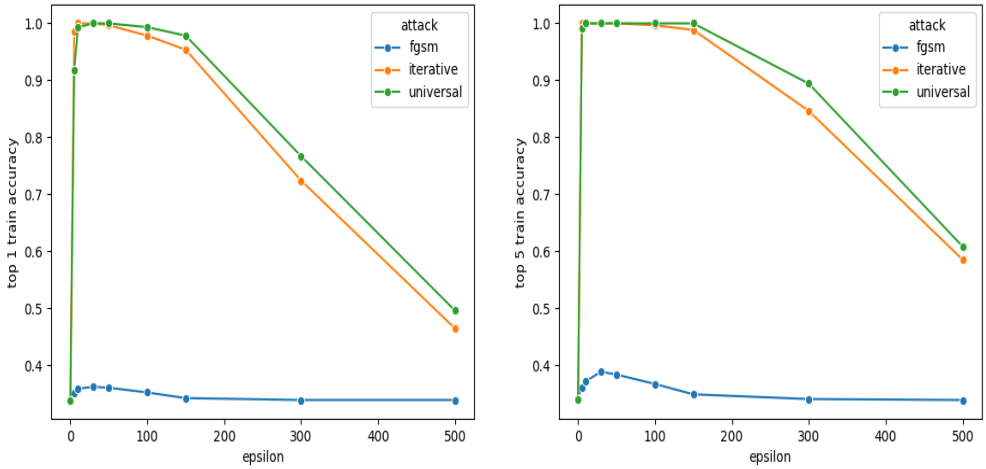

Figure 4.5. Top-1 and top-5 accuracies of MobileNetV2, after targeted adversarial attacks with epsilon ϵ values ranging from 0.5 to 50, were compared with those for clean images—unmodified images from our dataset. The accuracies were calculated using a set of 605 train images, which were rendered from 121 distinct 3D object models.

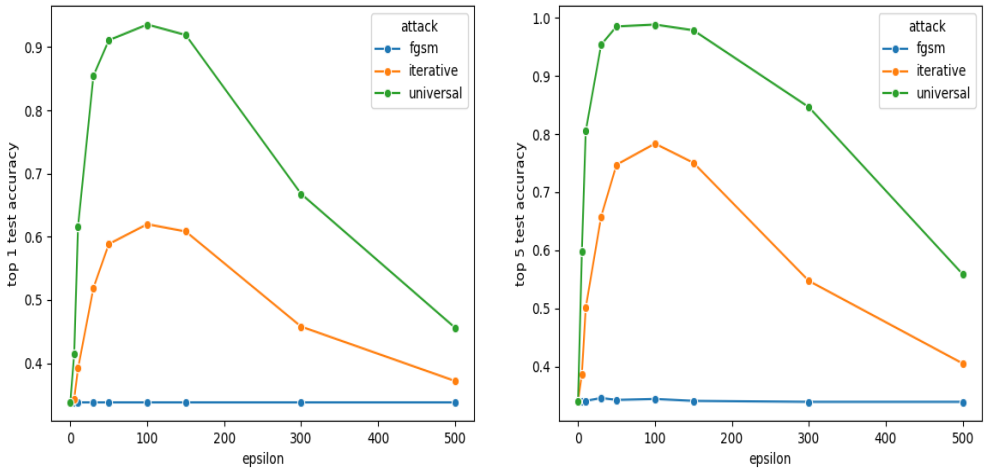

Figure 4.6. Top-1 and top-5 accuracies of MobileNetV2, after targeted adversarial attacks with epsilon ϵ values ranging from 0.5 to 50, were compared with those for clean images—unmodified images from our dataset. The accuracies were calculated using a set of 605 test images, which were rendered from 121 distinct 3D object models.

ϵ	Train Images			Test Images		
	FGSM	BIM	Universal	FGSM	BIM	Universal
0.00	0.00	0.00	0.00	**0.00**	**0.00**	**0.00**
0.50	0.00	0.91	0.73	0.00	0.01	**0.09**
1.00	0.01	0.97	0.94	0.00	0.04	**0.31**
3.00	0.01	0.97	0.97	0.00	0.18	**0.61**
5.00	0.01	0.97	0.97	0.00	0.25	**0.74**
10.00	0.00	0.91	0.94	0.00	0.29	**0.76**
15.00	0.00	0.81	0.88	0.00	0.27	**0.71**
30.00	0.00	0.40	0.45	0.00	0.01	**0.32**
50.00	0.00	0.12	0.16	0.00	0.03	**0.11**
Mean	0.01	0.67	0.67	0.00	0.13	**0.41**
Std (±)	0.01	0.39	0.38	0.00	0.12	**0.30**

Table 4.2. Average softmax predictions of MobileNetV2 under different targeted adversarial attacks at various epsilon values for both train and test images. $\epsilon = 0.00$ **indicates clean images without an attack. The higher the result, the more successful the attack.**

Table 4.2 summarizes the average soft-max predictions of the **target label** when images are subjected to different adversarial attacks. The comparison encompasses three attack methods: FGSM, BIM, and our proposed targeted Universal method. We selected a random number between the values of 0 to 999 to determine the target label that our perturbation would be attacking. We allotted one side of the table to training data and the other to test data. This allows us to analyze the robustness of these attacks across 605 training and 605 unseen images corresponding to multiple poses of our 121 objects. BIM and our Universal Perturbation method both adhere to 20 iterations while FGSM is limited to one iteration.

We note that the Universal and Basic Iterative attacks provide the highest success on both train and test images. At $\epsilon = 0.00$ we can see that every attack has a soft-max value of 0 for train and test images. Due to there being no noise added to the image, MobileNetV2 proceeds to classify the image as the true label instead of the target label. For train images at $\epsilon = .5$, the BIM and Universal perturbations exhibit noteworthy performance with average predictions of .91 and .73 respectively. The soft-max prediction of both attacks gradually increase until reaching a maximum value of .97 at $\epsilon = 5.0$. Past that point, both attacks decline until reaching a minimum of .12 for BIM and .16 for Universal. FGSM does not successfully match up with the other attacks at any ϵ value having a maximum average prediction of .01 between $\epsilon = 1.0$ and $\epsilon = 10.0$. At $\epsilon = 30.0$ there is quite a decrease in the soft-max prediction for BIM and Universal attacks, going from .81 and .88 to .40 and .45 respectively. Similar to our

untargeted experiments, we should note this decrease is most likely due to the influence that epsilon has on an image at the highest values. The lower soft-max values at high epsilons across all attacks is indicative of the amount of distortion caused by the noise. The soft-max predictions quickly increase and then gradually diminish as the image becomes more significantly altered.

For the test images, the best attack result for each ϵ, corresponding to the highest soft-max prediction of the target label, is highlighted in bold. While the experiments done on the training data are satisfactory, when we look at the results of our test images, the drop in effectiveness of each attack is noticeable. The Universal method displays the most success out of each attack on all epsilon values. At $\epsilon = 10$, Universal and BIM have their respective maximum values of .76 and .29. Prior to this point, both attacks progressively improve before deteriorating after it. Once again the FGSM attack remains unaffected, having a soft-max prediction of 0 throughout all epsilons. The decline in adversarial performance between train and test data is understandable considering the noise is being evaluated on new images. Despite the changes in results, our Universal attack is still more than adequate for practical applications.

In Fig. 4.5 and Fig. 4.6, we evaluate the top-1 and top-5 classification accuracies for training and test images, respectively. We observe that the accuracies all begin at 0.0, representative of a clean image. Since we are using the perturbation to target a specific label, an image without noise should have a soft-max prediction of 0.0. As the noise level, ϵ increases, we achieve successful target label classifications across BIM and Universal attacks. Both the basic iterative method and Universal attacks exhibited faster growth in top-1 and top-5 accuracies compared to FGSM, which is not successful at any epsilon value.

Fig 4.6 also shows the effectiveness of the FGSM, basic iterative, and Universal attack on testing images, achieving success values. For BIM and our Universal method, as the ϵ increases to 10, these attacks demonstrate success on the test images with accuracies increasing to above 90%. From the figure we can observe that there is a clear epsilon threshold of 15 where the adversarial modifications are so excessively pronounced that they compromise the model's ability to generalize, leading to a significant degradation in performance.

4.3 Discussion and Limitations

The extension of the previously proposed method for generating robust multi-view adversarial images represents a significant advancement in the field of adversarial machine learning. This research builds on the foundational work of crafting

a single noise perturbation applicable to various views of the same object, enhancing its practical applications. The experiments performed on 121 different 3D objects indicate the impact of our adversarial techniques. Our extended method has shown promising results in creating both targeted and untargeted adversarial attacks for our train and test data. By extending the previously proposed method to enable targeted use cases, this research provides a more versatile and powerful tool for generating adversarial images that can be tailored to specific image altering tasks. This advancement not only enhances the robustness and effectiveness of adversarial attacks but also aligns with the broader vision of adversarial architecture.

While the experimentation has produced favorable results, it also raises important considerations. In this paper, we do not analyze techniques for accelerated optimization. As discussed in Sec. 2, incorporating momentum gradient techniques into the iterative processes can aid in escaping poor local maxima and minima. This leaves potential for having a more efficient Universal method that can converge at quicker rates.

Targeted use cases provide a vast landscape for unstudied research. In our work we do not analyze methods for selecting the target label outside of using random distributions. Future work can be done to determine the potential bias that is involved in selecting target labels which are closely related to our true ones. For example, if we were running experiments on an orange, which had a true label index of 10, and we wanted to target a lemon which had a target label index of 11. We could potentially research the possible bias involved in choosing similar labels.

Another potential improvement can be made in the early stages of creating our Universal perturbation. Key differences between Universal and FGSM/BIM arise in how we initialize the adversarial noise. Compared to BIM and FGSM perturbations which initialize at zero, our Universal method may be sensitive to the initialization of adversarial noise. Our extension of the Universal method involves numerous levels of clipping, so it's possible that the initialization process has yet to be optimized. Since we calculate the gradients with respect to the noise, instead of the image, it is highly likely that initial values of noise can cause numerical instabilities, including initializing noise to zero. Consequently, initial adversarial noise could be considered another parameter of the algorithm to be assessed and tuned in the future. Currently, we first construct the noise as a uniformly distributed 224 x 224 matrix between values $\{-0.01, 0.01\}$. Future endeavors may involve developing an algorithm to determine the optimal initialization or providing a default initialization that is effective across different objects.

Moreover, while the Universal method aims to bridge the gap between 2D and 3D spaces, the algorithm currently inputs and outputs 2D images. The main drawback here is that similarly to FGSM and BIM techniques, we are applying noise to the entirety of the image, including the object and background. In scenarios when we only want our attack to focus on one of many objects within an image, perturbation is rendered useless. Coupling our targeted attack with segmentation techniques could potentially alleviate this issue and provide another layer of control in addition our targeted modification. This limitation, coupled with our reliance on existing 3D objects, which may not accurately represent real-world objects, hinders the applicability of our research. To address this, future efforts could incorporate radiance fields-based 3D modeling techniques like NeRF and 3D Gaussian Splatting to produce 3D models, thereby expanding our research beyond the confines of 2D space.

Our work is further limited by the classier involved in the evaluation process. Like many other pretrained classifiers, MobileNetV2 is not robust enough to be used on any object. We are restricted to experimenting on 1000 predefined classes. Supplementary research can involve utilizing transformer based architecture and Large Language Models to increase the robustness of the neural classifier and give us a more expansive pool of classes to choose from.

While further work is needed to address these points, our targeted method offers distinct advantages to previous adversarial techniques like efficiency, multi-view applicability, and additional perturbation control. Future research could explore techniques to enhance efficiency, investigate increased targeted perturbation manipulation, and incorporate more expansive classification models for broader applicability.

CHAPTER 5

Conclusion

In this work, we validated the previously proposed "Universal perturbation" method for generating robust multi-view adversarial examples on more robust data. Furthermore, we succeed in extending the applicability of our purposed algorithm to targeted use cases, establishing its capability for more controlled adversarial alterations. After running numerous experiments, our results indicated that the Universal method outperformed existing techniques in terms of both effectiveness and efficiency across different angles. In untargeted experiments, the mean soft-max probability for the Universal attack was the lowest out of all tested methods. Comparatively, in targeted use cases the Universal attack also maintained the highest average soft-max probability on both train and test data. The importance of the Universal Perturbation Method is highlighted by it's ability to operate exclusively on 2D images, offering a practical and scalable alternative to computationally costly 3D adversarial attacks. We build off these benefits by offering additional advantages involved with targeted attacks and further validation of existing methods.

Experiments on 121 diverse 3D objects emphasize the effectiveness of our approach. In comparison to other techniques, the untargeted Universal attacks successfully identified single noise perturbations with higher destruction rates across multiple viewing angles, particularly at low ϵ levels. Results showed that between $\epsilon = 0.5$ and $\epsilon = 5.0$, the untargeted Universal attack displayed the lowest soft-max predictions. In contrast, targeted Universal attacks effectively exhibited its potential for practical adversarial manipulation over a wide range of epsilon values. Targeted Universal attacks exhibited the lowest average soft-max values for test images at .41 all while requiring the least amount of computational effort. This performance emphasizes the improvements over standard single-view attacks, which may struggle with viewpoint variations and deformations while also requiring a noise for each unique viewpoint. Our experiments also substantiates previous research done on the Universal method, validating that the algorithm remains successful on more diverse data.

Beyond its effectiveness, our targeted Universal method offers key advantages in terms of efficiency and control. Current attacks require a computation to be done at each view point while our targeted Universal approach keeps the same advantage of only needing one perturbation. As a result, our approach reduces computational

resources compared to other methods. This efficiency along with the added control one receives through targeted attacks makes it particularly well-suited for real-world applications where computational constraints or label restrictions are present.

While the work done in this paper represents a significant step forward, we acknowledge the possible limitations and opportunities for future research within adversarial machine learning. Further investigations into combining potential perturbation methods with image segmentation techniques may be necessary area of improvement. Incorporating systems for implementing a wider range of object categories and classification models could enhance the significance of our findings. We hope that the improvements covered in this paper can be seen as a step in the right direction in terms of the advancement of the Universal method.

BIBLIOGRAPHY

[1] M. Ergezer, P. Duong, C. Green, T. Nguyen, and A. Zeybey, "One noise to rule them all: Multi-view adversarial attacks with universal perturbation," 2024.

[2] M. Ergezer, A. Furgiuele, and C. H. Zaman, "Towards an adversarial architecture," 2022.

[3] R. Klokov and V. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3d point cloud models," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 863–872.

[4] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[5] T. Le and Y. Duan, "Pointgrid: A deep network for 3d shape understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9204–9214.

[6] C. Wang, M. Cheng, F. Sohel, M. Bennamoun, and J. Li, "Normalnet: A voxel-based cnn for 3d object classification and retrieval," *Neurocomputing*, vol. 323, pp. 139–147, 2019.

[7] S. Zhi, Y. Liu, X. Li, and Y. Guo, "Toward real-time 3d object recognition: A lightweight volumetric cnn framework using multitask learning," *Computers & Graphics*, vol. 71, pp. 199–207, 2018.

[8] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *The 25th annual international conference on mobile computing and networking*, 2019, pp. 1–16.

[9] K. Apicharttrisorn, X. Ran, J. Chen, S. V. Krishnamurthy, and A. K. Roy-Chowdhury, "Frugal following: Power thrifty object detection and tracking for mobile augmented reality," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, 2019, pp. 96–109.

[10] X. Li, Y. Tian, F. Zhang, S. Quan, and Y. Xu, "Object detection in the context of mobile augmented reality," in *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2020, pp. 156–163.

[11] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7644–7652.

[12] R. Ravindran, M. J. Santora, and M. M. Jamali, "Multi-object detection and tracking, based on dnn, for autonomous vehicles: A review," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 5668–5677, 2020.

[13] K. Zhang, S. Wang, L. Ji, and C. Wang, "Dnn based camera and lidar fusion framework for 3d object recognition," in *Journal of Physics: Conference Series*, vol. 1518, no. 1. IOP Publishing, 2020, p. 012044.

[14] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha, "3-d deformable object manipulation using deep neural networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4255–4261, 2019.

[15] D. Hossain, G. Capi, and M. Jindai, "Object recognition and robot grasping: A deep learning based approach," in *The 34th Annual Conference of the Robotics Society of Japan (RSJ 2016), Yamagata, Japan*, 2016.

[16] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Learning visual feature spaces for robotic manipulation with deep spatial autoencoders," *arXiv preprint arXiv:1509.06113*, vol. 25, p. 2, 2015.

[17] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[19] X.-Y. Luo, Y. Yu, J.-L. Liu, M.-Y. Zheng, C.-Y. Wang, B. Wang, J. Li, X. Jiang, X.-P. Xie, Q. Zhang *et al.*, "Entangling metropolitan-distance separated quantum memories," *arXiv preprint arXiv:2201.11953*, 2022.

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

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

[22] T. Wu, J. Zhang, X. Fu, Y. Wang, L. P. Jiawei Ren, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu, "Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2017.

[24] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," 2017.

[25] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," 2018.

[26] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," 2018.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019.

[28] A. Bes, "Worn baseball ball," June 2017, last accessed 10 February 2024. [Online]. Available: https://sketchfab.com/3d-models/worn-baseball-ball-fdf3de6ae225421ea78961b897b9608a

[29] soufiane oujihi, "Snail," January 2024, last accessed 22 April 2024. [Online]. Available: https://sketchfab.com/3d-models/snail-3d-69f9a70968a64e7396115247d0db3b2d

[30] Europac3d, "Acorn," Janurary 2024, last accessed 8 April 2024. [Online]. Available: https://sketchfab.com/3d-models/acorn-84ad040b23644329aa142e502a1d4b14

[31] KathrinChristian, "Conch," June 2019, last accessed 23 April 2024. [Online]. Available: https://sketchfab.com/3d-models/conch-3858726afd4444b88952049d1c68e2fa

[32] C. CGI, "Pretzel," February 2023, last accessed 23 April 2024. [Online]. Available: https://sketchfab.com/3d-models/pretzel-6e9a5edb16bf41f28c5dcf42865e5837

[33] dannyboy70000, "lemon 3d model," October 2014, last accessed 10 February 2024. [Online]. Available: https://free3d.com/3d-model/lemon-72357.html

[34] A. Alexandrescu, "Broccoli," Janurary 2023, last accessed 9 April 2024. [Online]. Available: https://sketchfab.com/3d-models/broccoli-d837fc2625234363a34e48a4d5f3e099

[35] selfie 3D scan, "Tractor," January 2019, last accessed 10 February 2024. [Online]. Available: https://sketchfab.com/3d-models/tractor-1b258bcc01bf4ed0935ef73e80442c30

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

APPENDIX A

Github Repository Link

Github Repository Link

Any further research done on the Universal Perturbation and the code that supported this work can be found at the following Github Repository:

https://github.com/memoatwit/UniversalPerturbation