Kim André Brunstad Midtlid & Johannes Åsheim

Magnitude Adversarial Spectrum Search-based Black-box Attack against Image Classification

Master's thesis in Computer Science
Supervisor: Jingyue Li
June 2022

NTNU
Norwegian University of Science and Technology

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Kim André Brunstad Midtlid & Johannes Åsheim

Magnitude Adversarial Spectrum Search-based Black-box Attack against Image Classification

Master’s thesis in Computer Science
Supervisor: Jingyue Li
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Recent development has revealed that deep neural networks used in image classification systems are vulnerable to adversarial attacks. In this thesis, we design an untargeted query-efficient decision-based black-box attack against robust image classification models that produce imperceptible adversarial examples. The proposed attack method, Magnitude Adversarial Spectrum Search-based Attack (MASSA), includes two novel components to generate the initial noise and reduce the noise in the frequency domain. Our experiments show that MASSA requires significantly fewer queries than the state-of-the-art HopSkipJumpAttack (HSJA). In addition, MASSA can create adversarial examples with $74,16\%$ lower l_2 distance than HSJA after only 250 queries. Finally, we demonstrate that MASSA bypasses two defense mechanisms and should be used to evaluate the robustness of future defenses.

Sammendrag

De siste årene har forskning vist at dype nevrale nettverk som brukes i bildeklassifiseringssystemer er sårbare mot fiendtlige angrep. I denne oppgaven utformer vi et umålrettet søkeeffektivt beslutningsbasert svart-boks angrep mot robuste bildeklassifiseringsmodeller som produserer skjulte endringer i bilder. Den utviklede angrepsmetoden, Magnitude Adversarial Spectrum Search-based Attack (MASSA), inkluderer blant annet to nyskapende komponenter for å generere den initielle støyen og redusere støyen i frekvensdomenet. Eksperimentene våre viser at MASSA krever betydlig færre spørringer enn dagens ledende angrep HopSkipJumpAttack (HSJA). I tillegg er MASSA i stand til å produsere fiendtlige bilder med $74,16\%$ lavere avstand enn HSJA etter kun 250 spørringer. Til slutt demonstrerer vi at MASSA slår to forsvarsmekanismer og bør brukes til å evaluere robustheten til fremtidige forsvar.

Preface

This is the Master Thesis written by Kim André B. Midtlid and Johannes Åsheim. The thesis was conducted during the spring semester of 2022 at NTNU. Our work was supervised by Professor Jingyue Li at the Institute of Computer Science and Informatics at NTNU. We would like to thank Li for his continuous assistance and helpful insights. Additionally, we would like to thank friends and family for their motivation and ongoing support throughout our years of study.

Contents

Figures

Tables

Acronyms

CNN Convolutional Neural Network. 3, 19, 33–35

DC Direct Current. 14

DFT Discrete Fourier Transform. 12

DNN Deep Neural Network. 1, 3

F-mixup F-mixup. 20, 21, 23, 33, 51, 52

FFT Fast Fourier Transform. 12, 13, 20

FGSM Fast Gradient Sign Method. 1

HSJA HopSkipJumpAttack. iii, v, xi, 16–18, 33, 37, 39, 41, 42, 45–49, 51–53, 55

IDFT Inverse Discrete Fourier Transform. 12, 13

IFFT Inverse Fast Fourier Transform. 13, 20, 21, 28

MASSA Magnitude Adversarial Spectrum Search-based Attack. iii, v, xi, 2, 24, 33, 37–42, 44–49, 51–53, 55, 62

NSFW Not-Safe-For-Work. 8, 9

PAR Patch-wise Adversarial Removal. xi, 18, 19, 33, 51, 52

SLR Structured Literature Review. 16, 38

UAP Universal Adversarial Perturbations. 10

ViTs Vision Transformers. 19, 33

Chapter 1

Introduction

In recent years computing power has become more and more powerful, paving the way for Deep Neural Networks (DNNs) to be used in computer vision tasks such as image classification. The remarkable results of DNNs have led to their use in various safety-critical tasks such as autonomous driving [1–5] and facial biometric systems, including surveillance and access control [6, 7]. These safety-critical systems require certain robustness from the DNNs, where failure can lead to severe consequences.

Despite their exceptional performance, Goodfellow *et al.* [8] have demonstrated that DNNs are vulnerable to adversarial attacks. They present Fast Gradient Sign Method (FGSM), which can generate imperceptible adversarial examples to impact the predictions of image classification models. Since then, the research community has published more adversarial attacks to shed light on the vulnerabilities of DNNs to evaluate the robustness of image classification models. Further development of new attack methods is vital to evaluate and strengthen the robustness of these models, which can be implemented in safety-critical systems.

Adversarial attacks are conducted under a particular threat model. The white-box threat model assumes internal knowledge of the target model, while the black-box assumes no knowledge. In real-world applications, an adversary cannot expect to obtain knowledge of the target model, making the black-box setting more realistic [9]. The most realistic black-box setting is when the adversary only has access to the output labels alone, known as decision-based attacks. Decision-based attacks are usually iterative and query the target model repeatedly to gradually lower the perceptibility of the adversarial example.

The first proposed decision-based attack methods required hundreds of thousands of model queries to create imperceptible adversarial examples [10], i.e., with a minimal l_2 distance to the original image. Even though the current state-of-the-art decision-based attack methods are more query-efficient, they still require thousands of model queries to achieve imperceptibility. Hence, robust classification systems can detect a large number of queries to the target model and expose the adversary [11]. The main challenge of the decision-based attack field is to lower the query budget for adversarial attacks.

In this thesis, we aim to construct a query-efficient attack method that generates imperceptible adversarial examples in just hundreds of queries. Additionally, we investigate how models with implemented defense mechanisms are robust against our attacks. We want to answer the following research question:

RQ: *How to create an untargeted state-of-the-art query-efficient decision-based black-box attack against robust image classification models to produce less perceptible adversarial examples?*

The main contribution of this thesis is Magnitude Adversarial Spectrum Search-based Attack (MASSA), a novel decision-based black-box adversarial attack method. To the best of our knowledge, we are the first to design an attack method that addresses query-efficiency and imperceptibility by modifying all frequency components in the magnitude spectrum. First, we generate noise in low, medium, and high frequencies instead of sampling noise in the spatial domain. Then we reduce the size of the perturbation by minimizing the frequency noise through a binary search in each frequency component. Finally, we increase the imperceptibility by conducting a patch-wise removal of redundant noise. In summary, our contributions are as follows:

- Our attack method contains two novel parts. The first part creates initial noise in the frequency domain. The second minimizes the distance between the original and adversarial magnitude spectrums through a binary search in each frequency component.
- We design an attack method, MASSA, based on the proposed initiation method and *Frequency Spectrum Binary Search*.
- We demonstrate empirically that MASSA achieves superior query-efficiency and imperceptibility over a state-of-the-art decision-based attack through extensive experiments with an unprecedented low number of model queries.
- We evaluate defense mechanisms against our attack method and propose that our attack method can be used to assess the robustness of defense mechanisms.

The structure of the thesis is as follows: First, in chapter 2, we introduce the background theory necessary to understand the concepts for our proposed attack method. Second, in chapter 3, we discuss the related work and state-of-the-art decision-based attack methods, where we highlight the current limitations. In chapter 4, we propose our novel attack method and how we mitigate the limitations of the related work. In chapter 5, we detail our evaluation procedure followed by the results from our comprehensive experiments. Furthermore, the results are discussed regarding related work, academia, and industry in chapter 6. Finally, we conclude our work and propose directions for future work in chapter 7.

Chapter 2

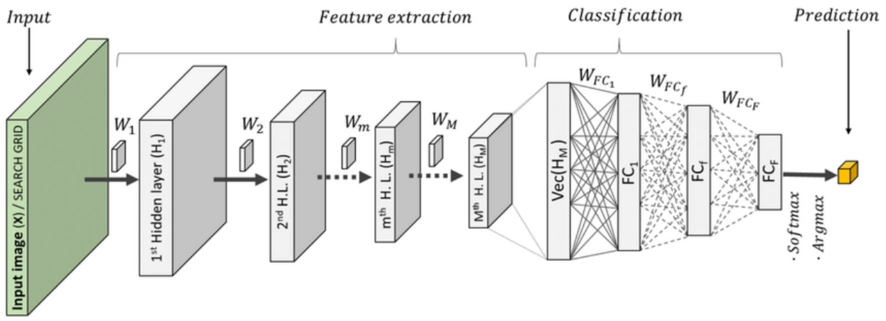
Background

The following chapter introduces the basis for the thesis. First, computer vision and image classification are detailed to understand the application fields for adversarial attacks that target image classifiers. To know how adversarial attacks are generated and executed, their taxonomy is denoted, and a high-level explanation of adversarial examples is detailed. Then, the threat models are introduced to explain the differences between conditions adversarial attacks require to be executed. Lastly, an overview of the different image domains is given.

2.1 Computer Vision and Image Classification

Computer vision is the field of artificial intelligence where systems can extract information from visual inputs, e.g., images, and videos, to perform tasks based on the information. Computer vision is implemented in industries such as energy, utilities, healthcare, manufacturing, and transportation [12]. For example, a computer vision system could be a form of intelligent image processing of the outputs from cameras and other sensors of self-driving vehicles. Then the goal of the system would be to identify the surroundings, such as other cars, traffic signs, and pedestrians, based on the visual inputs.

A sub-domain of computer vision is the task of classifying images. Image classification consists of assigning an input image to a specific label [13]. The advancements of DNNs have enabled the development of state-of-the-art image classifiers [14, 15]. The image classifiers utilize a sub-group of DNNs known as Convolutional Neural Networks (CNNs) to perform the task of recognizing images. We denote an image as an array comprised of three dimensions $width \times height \times channels$ where $width$ and $height$ is the size of the image in pixels, and $channels$ often refer to the values in the *red*, *green*, and *blue* channel of a colored image. The image format is complex for a DNN, but a CNN reduces the complexity and structures the information into feature maps. The feature maps are the extracted features of the image and are computed by a filter applied to pixel blocks of the image. The feature extraction is shown in Figure 2.1 as the first part of the CNN architecture.



Source: Avalos and Ortiz [16]

Figure 2.1: CNN architecture showing the feature extraction and classification.

After the feature extraction, an artificial neural network combines the features into attributes. Then, both features and attributes are used to predict the labels of the input images. During training, the CNN will calculate its error through a loss function and use backpropagation [17] to adjust its internal weights and biases to minimize the error.

2.2 Adversarial Attacks targeting Image Classification

Image classification unravels large opportunities for the industries with grounds for computer vision. As more and more systems implement image classifiers, it is essential to have robust models against adversarial attacks. The following section introduces the taxonomy and details how adversarial examples can be generated.

2.2.1 Taxonomy

To understand the context of adversarial attacks, the following terms are denoted.

Target model: The target model is the system that performs image classification. The system consists of a discriminant function $F : \mathbb{R}^D \to \mathbb{R}^m$ that produces the output $y = [0,1]^m$ such that $\sum_{c=1}^{m} y_c = 1$ given an input image $x \in [0,1]^D$, where $D = (\text{width} \times \text{height} \times \text{channels})$ is the dimensions of the image. Intuitively, y can be interpreted as a probability distribution over the set of labels $[m] = \{1,\ldots,m\}$, where y_i is the probability of input x belonging to class $i \in [1,m]$. The target model then uses a classifier $C(x) := \text{argmax}_{c\in[m]} F_c(x)$ which allocates x to the maximum probability in y. The target model is the victim of the attack. A target model can be based on any given architecture, and target models can vary from system to system.

Adversarial example: An adversarial example $x' \in [0,1]^D$ is a modified version of the original image x that the target model misclassifies. The adversarial example is commonly the output of the attack method.

Perturbation: A perturbation δ is a change to the original image $x' = x + \delta$.

To generate an adversarial example, the adversary adds a large enough perturbation to the original image such that the target model misclassifies it. How this perturbation is generated and modified depends on the attack method. Generally, an attack method aims to minimize the size of the perturbation δ while still keeping the modified image x' adversarial.

Adversary: An adversary is an entity that executes an attack to generate the adversarial example x' and achieve the *attack goal*. The adversary chooses which attack method to use and which goal to achieve.

Untargeted Attack: Untargeted attack is a type of attack that aims to change the original classifier decision $c := C(x)$ into any other decision $c' \in [m] \setminus \{c\}$. This can be formulated as $S_x(x') := \max_{c' \neq c} F_{c'}(x') - F_c(x')$, which tells us that the untargeted attack is successful if and only if $S_x(x') > 0$. Simply put, the *attack goal* is to change the originally predicted class of the target model into any other class.

Targeted Attack: Targeted attack is a type of attack that aims to change the original classifier decision $c := C(x)$ into a pre-determined other decision $c_t \in [m] \setminus \{c\}$. As with untargeted attacks, this can be formulated as $T_x(x') := \max_{c_t \neq c} F_{c_t}(x') - F_c(x')$, which tells us that the targeted attack is successful if and only if $T_x(x') > 0$. The *attack goal* is to change the originally predicted class into a chosen target class determined by the adversary.

Query: A query is the action of asking the target model to classify an image. The adversary uses queries to gain information about the target model which can improve the adversarial example x'. A query consists of an input-output pair: the input x is an image to be classified by the target model. The output is usually a probability distribution y over the set of labels, or just the maximum probability depending on the threat model, as explained later in section 2.3.

Transferability: The transferability of an attack method is the ability to achieve the *attack goal* on several target models with different architectures. Usually, an identical adversarial example is given to each of the target models. The ability to attack a higher number of target models implies higher transferability of the attack method.

2.2.2 Generating Adversarial Perturbations

A generic attack method starts by obtaining the input denoted original image. How the input is obtained is beyond the scope of the thesis, but we give two examples of attack contexts in subsection 2.2.5. Given an original image x and initial perturbation δ, the attack method can construct an adversarial example $x' = x + \delta$, i.e. the perturbation appended to the original image. Attack methods have different approaches, and some rely on iteratively querying the target model to optimize the perturbation and construct the final adversarial example. Figure 2.2 illustrates how a generic attack method can query the target model to optimize the perturbation and minimize perceptibility.

Since each attack method uses a different approach, we used Boundary Attack [10] to explain how an attack method can be constructed. Brendel *et al.* [10] presents an iterative attack algorithm that gradually lowers the imperceptibility of the initial perturbation. The attack method performs random sampling of adversarial and non-adversarial images to calculate the direction towards the original image. The method relies on querying the target model to conduct the sampling. Figure 2.3 shows the adversarial perturbations created by Boundary Attack during an untargeted attack. The adversarial input is created after 200 667 queries to the target model.

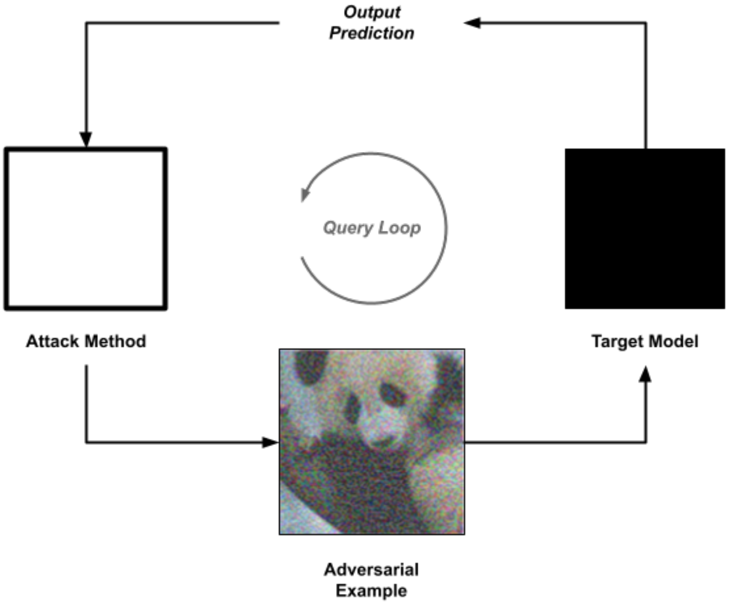
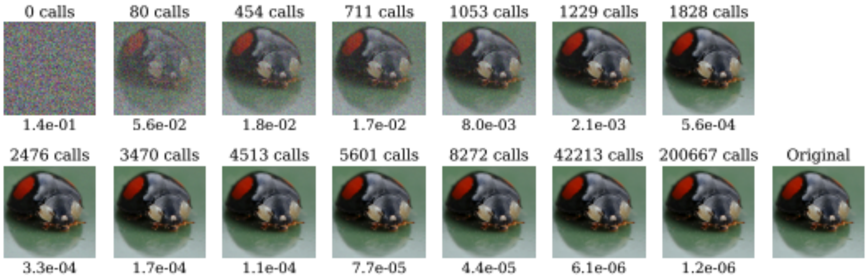


Figure 2.2: The query loop for a generic attack method.



Source: Brendel *et al.* [10]

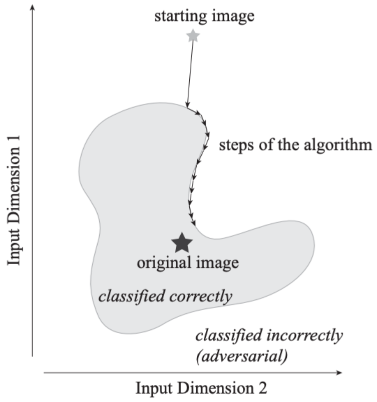
Figure 2.3: Adversarial example generation by Boundary Attack.

2.2.3 Distance Metrics

An adversary wants the adversarial input to be similar to the original image. To quantify this similarity, generally, three common distance metrics are used [18, 19]. The common distance metrics are based on l_p norms where $p \in \{0, 2, \infty\}$. l_0 distance quantifies the total number of perturbed pixels in an image. l_2 is the most common distance metric and measures the Euclidean distance between two images. Given many small perturbations, the l_2 distance metric can be small even if the number of perturbed pixels is high. l_∞ measures the largest perturbation to a single pixel among all pixels in the image.

2.2.4 Decision Boundary

The decision boundary is the border where an image on one side is classified correctly and on the other side misclassified. The misclassified side is referred to as adversarial. Attack methods aim to lower the perceptibility of the adversarial perturbation, i.e., be as close as possible to the original image while staying on the adversarial side of the boundary. Ideally, the final adversarial example has the minimum distance to the original image x while still adversarial. Figure 2.4 shows the steps along the decision boundary of Boundary Attack [10] to reach the ideal perturbation.



Source: Brendel *et al.* [10]

Figure 2.4: Illustration of the steps along the decision boundary of Boundary Attack [10].

When an attack method is at the decision boundary, the direction for the next step has to be decided. Usually, information about the boundary is obtained by *random sampling* near the image on the boundary [10, 19–23]. This is done by querying a set of random samples around the image, where one subset is adversarial, and the other is not adversarial. By calculating the distance to the random samples, the shape of the boundary in that area can be estimated. The attack method can move closer to the original image while remaining adversarial based on the estimated boundary.

2.2.5 Attack Context

The attack context explains how the adversary conducts the attack against the target model in a real-world use case. We discuss two relevant attack contexts and describe how they could be performed from an adversarial viewpoint.

One attack context can be a man-in-the-middle attack, where an entity intercepts the input and adjusts it before delivering it to the final destination. Figure 2.5 illustrates the man-in-the-middle attack context for a generic untargeted attack method. An attack method can be executed against a system where an adversary has access to the input before it is delivered to the target model. An attack can be conducted if the adversary has access to the camera sensors, e.g., autonomous vehicles or facial biometric systems.

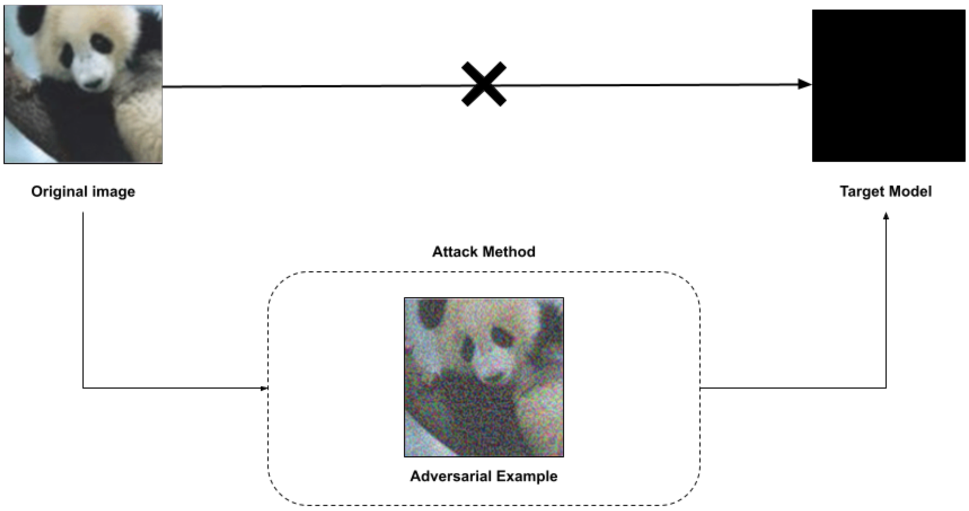


Figure 2.5: The attack context for a generic untargeted attack method. The degree of perturbation illustrates the adjustment to the original image and does not represent the actual output of the attack method.

Another real-world use case for adversarial attacks against image classification models is a Not-Safe-For-Work (NSFW) filter [24]. NSFW filters use image classification to detect and filter out explicit images and NSFW content. An adversary may bypass the NSFW filter, making the image classifier misclassify explicit content as safe-for-work, thus displaying NSFW content for users. To achieve this, an adversary starts with a random initial perturbation classified as safe-for-work. The goal of the adversary will be to get the initial perturbation as close to a NSFW image as possible while keeping the classification safe-for-work by not crossing the decision boundary. The attack method used by the adversary will iteratively move the adversarial example closer to the explicit image by reducing the perturbation based on the query information. In a real-life scenario, a query could be to upload an adversarial example and see if it is classified as NSFW or not. Based on this information the attack method would ideally reduce the perturbation iteratively until it is imperceptible to human beings. This final adversarial example would bypass the NSFW filter while still appearing as explicit content for humans.

2.3 Threat Model

The threat model can be viewed as the setting surrounding the attack and the restrictions which are imposed on the adversary. The setting refers to information known to the adversary, while the restrictions refer to possible actions of the adversary. The literature mainly divide the threat models into white-box and black-box. These two high-level threat models can be defined in greater detail by focusing on what information is accessible to the adversary. This section first explains the high-level white-box and black-box threat models before diving into the additional threat models.

2.3.1 White-box and Black-box Settings

White-box Attacks

White-box attacks assume an adversary has access to any information about the target model and datasets used during the training of the target model. Existing white-box attack methods such as Goodfellow *et al.* [8] are based on gradient descent which requires information about the model weights and internals. With this information Goodfellow *et al.* [8] can generate efficient adversarial examples with a high success rate. Computer vision systems may use different hyperparameters, datasets, and model architectures to fit their specific application needs. Therefore internal information about the target model is not available to an adversary, making white-box attacks impractical in real-world applications.

Black-box Attacks

Black-box attacks assume no information about the target model, which aligns better with real-world applications than white-box attacks [9]. Black-box attacks only assume the ability to observe the output of the target model for a given input. Figure 2.6 illustrates the difference between white-box attacks and black-box attacks where the latter does not have access to anything inside the target model.

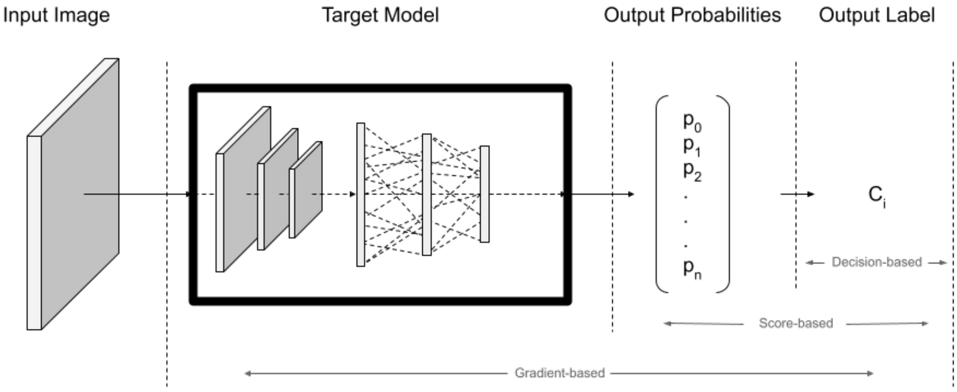


Figure 2.6: A visual representation of white-box and black-box threat model. The black-box model cannot access internal information about the target model (inside the black-box).

2.3.2 Another approach to classifying threat models

Gradient-based

The gradient-based threat model includes attacks that assume knowledge of the internal gradients of the target model. The knowledge is similar to the white-box threat model. Based on the gradient knowledge, the adversary can use attack methods that backpropagate the target model weights to generate adversarial perturbations [8].

Transfer-based

The transfer-based threat model includes attacks that assume the knowledge of transferability [23]. Often the attack methods generate a surrogate model similar to the target model. Then the adversary can use attack methods with high transferability to attack the surrogate model and transfer the attack to the target model. Even though the attack method does not require internal information about the target model, the attack method needs to make some assumptions about the target model to generate a similar surrogate model.

Universal Adversarial Perturbations (UAP) aims to create a single perturbation that can be applied universally, i.e., to different images, and still fool the target model [25]. To generate a universal perturbation, most UAP methods rely on the training data, model architecture, and target model parameters. They add various perturbations to the training data to determine an optimal universal perturbation.

Score-based

The score-based threat model includes attacks that assume no internal knowledge about the target model [26], but can access the output probability as shown in Figure 2.6. The scores are denoted as the output probabilities and allow the adversary to use attack methods that modify the perturbation based on the scores of other classes. The modification can then be determined based on the changes in the probability scores.

Decision-based

The decision-based threat model includes attacks that only assume the output label [27], as shown in Figure 2.6. Different from score-based attacks, decision-based will not have the information about other classes but solely rely on whether the input is adversarial or not. This is the most restricted threat model. An adversary has to use attack methods that navigate the decision boundary of the target model to find the optimal adversarial perturbation. The decision-based attack methods require the initial perturbation to be adversarial to find the decision boundary. From this point, the attack method always keeps the perturbation on the adversarial side of the boundary.

2.4 Image Domains

There are multiple ways to represent an image digitally. This section briefly introduces the two domains related to the thesis. First, we present the spatial domain, which is the most common way to represent an image, then we introduce the frequency domain.

2.4.1 Spatial Domain

Digitally, an image is represented in the form of pixel values. For grayscale images each pixel has an intensity value associated with it, usually in the range $[0, 255]$. This value indicates how bright the pixel is: 0 is entirely black, 255 is completely white, and everything in between is some shade of gray. For colored images, this intensity value is represented as a vector $[R, G, B]$. Each pixel in the image has its own RGB vector. The first element in the vector represents the contribution of red in the given pixel, the second element represents green, and the last element represents blue. Figure 2.7 illustrates this concept. The colored image can thus be represented as a 3D vector of 2D matrices, resulting in a shape of (w, h, c) where w and h are the width and height of the image, and c is the number of channels (3 for RGB images). This representation of an image is called the spatial domain.

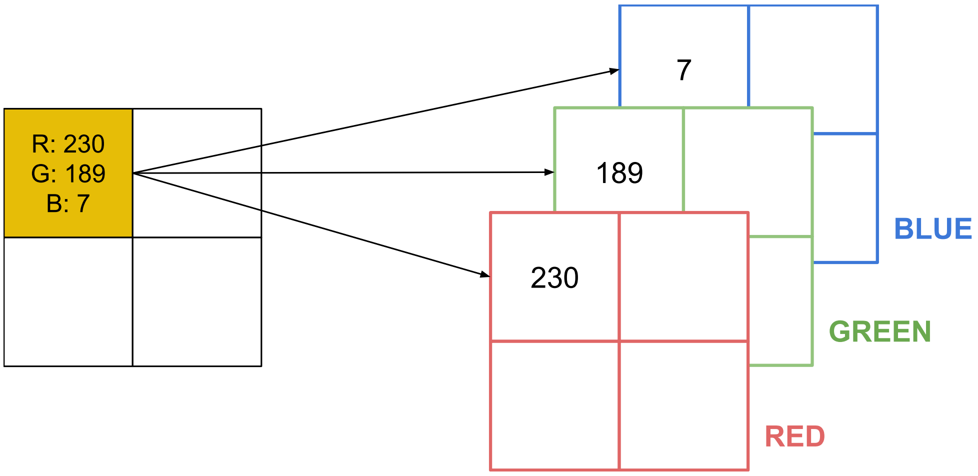


Figure 2.7: A visual representation of a $2x2$ colored image. A colored pixel can be split into red, green, and blue channel contributions.

2.4.2 Frequency Domain

Another image representation method is through the Fourier domain, which we denote as the *frequency domain*. In this domain each point represents a specific frequency contained in the spatial image. We can think of the frequency domain as a set of components consisting of sine and cosine waves. Each point in the frequency domain $F(u,v)$ represents a certain combination of magnitude and phase of these sinusoidal components, making it possible to represent any image. In this subsection, we give a detailed explanation of the frequency domain based on the description provided by Fisher *et al.* [28].

Going from the spatial domain into the frequency domain is known as *decomposing*, and going back to the spatial domain from the frequency domain is known as *synthesizing*. Both decomposing and synthesizing are straightforward processes using Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT), respectively. DFT is a sampled Fourier Transform which means it uses a large enough set of samples to represent a spatial image but does not contain all frequencies in the image. In our approach, we use a fast implementation of DFT known as Fast Fourier Transform (FFT), and we use these terms interchangeably throughout the thesis. For a given image of size $d \times d$, the two-dimensional DFT is given by

$$F(u,v) = \sum_{u=0}^{d-1}\sum_{v=0}^{d-1} f(x,y)e^{2\pi\frac{ux+vy}{d}j}. \tag{2.1}$$

Here, $f(x,y)$ represents the pixel value at position (x,y) in the spatial image, and the exponential term is the sinusoidal component corresponding to each point (u,v) in the frequency spectrum. This means that each point $F(u,v)$ in the frequency spectrum is obtained by summing the product between the spatial image and the correlated sinusoidal component. The synthesizing process is performed using IDFT. This two-dimensional inverse transformation is given by

$$f(x,y) = \frac{1}{d^2} \sum_{x=0}^{d-1} \sum_{y=0}^{d-1} F(u,v) e^{-2\pi \frac{ux+vy}{d} j}, \tag{2.2}$$

which is very similar to Equation 2.1. The only difference between the two is that IDFT introduces a normalization term $\frac{1}{d^2}$ [1] and changes the sign of the sinusoidal components.

Figure 2.8 intuitively illustrate the decomposing and synthesizing processes [2]. Notice how FFT produces two images: one for the magnitude and one for the phase. This results from FFT producing a complex number at each point $F(u,v)$. A complex number $z = x + iy$ can be written in the polar form $z = r(\cos\theta + i\sin\theta)$. Since the complex number can be split into its real and imaginary parts we can view the log-scaled magnitude r and phase θ images separately, as illustrated in Figure 2.8. We apply a logarithmic transformation [30] to log-scale the values in the frequency domain as their value range is too large to visualize. As explained by Fisher *et al.* [28], the magnitude spectrum contains most of the geometry in the spatial image, while the phase does not contribute much new information. Hence, we only talk about the magnitude spectrum when referring to the frequency domain from this point on. Still, IFFT requires the phase in the synthesizing process from the frequency domain back to the spatial domain, so we cannot completely discard the phase spectrum.

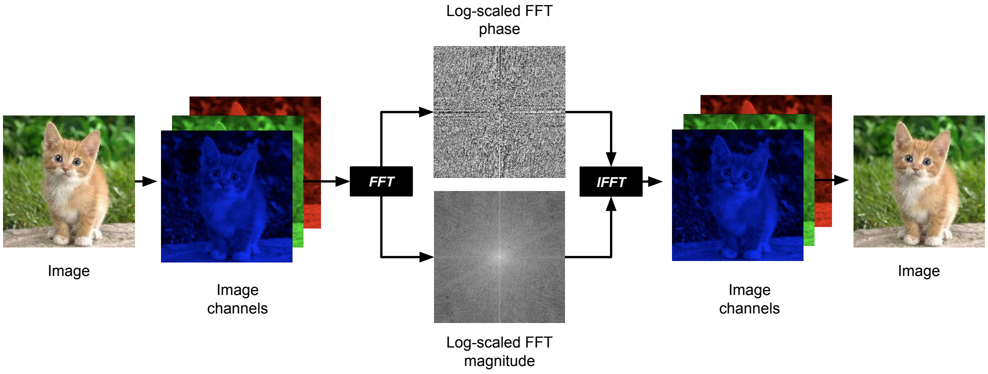


Figure 2.8: Visualization of decomposing (FFT) and synthesizing (IFFT). The original image is split into its RGB channels, and FFT is used channel-wise to obtain the phase and magnitude spectrum (log-scaled for visualization purposes). IFFT combines these spectrums back into the original image.

[1] The normalization term can be applied to the decomposition process instead, but should not be used in both decomposing and synthesizing.

[2] For visualization purposes, the Fourier transform is only applied to a single channel of the colored image. Shukla *et al.* [29] illustrates that FFT and IFFT can be independently applied channel-wise.

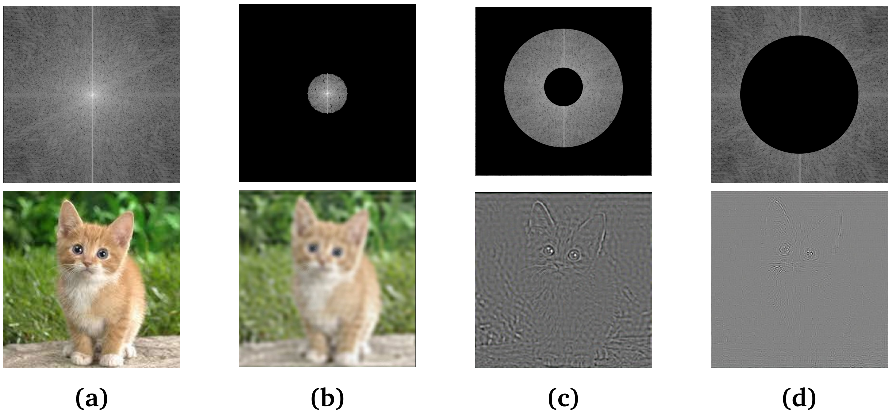


Figure 2.9: Each column represents a magnitude spectrum and its corresponding spatial image. Observe how most of the spatial information is contained in the low-frequency band. (a) The original image. (b) Only low-frequency band. (c) Only medium-frequency band. (d) Only high-frequency band.

As we can see from the magnitude spectrum in Figure 2.8 the largest values (light) are concentrated in the center of the image. The center point is known as the Direct Current (DC) component and is by far the largest component in the magnitude spectrum. The DC-component got its name from signal analysis in electrical engineering and represents an average brightness of the spatial information, which means that a small change to this value has major effects on the corresponding spatial image obtained from synthesizing. Other high-valued components in the frequency domain also demonstrate this property. From Figure 2.8, we can see that these components are located in the center of the magnitude spectrum. These components in the center of the image make up the *low frequencies* of the frequency domain. As we saw with the DC-component, the low frequencies contain most of the spatial information. As we move away from the center in the magnitude spectrum, the component values decrease, meaning less and less spatial image information is contained in these points. Outside the low frequencies, we find the *medium frequencies*, and outside that, we get to the *high frequencies*. Throughout this thesis, we use the term *frequency band* to refer to the areas of different frequencies.

We demonstrate how the different frequency bands relate to the spatial image information through Figure 2.9 where each column represents a magnitude spectrum and spatial image pair. The magnitude spectrums in Figure 2.9 have certain frequency bands masked (black) to show what spatial information is contained in the remaining band. From Figure 2.9b we see that the low frequencies are still able to recover most of the spatial image even without the medium and high frequencies, only losing some color intensity and sharpness. The medium and high frequencies do not contain as much spatial information as the low frequencies. We can see this from Figure 2.9c and Figure 2.9d, where the spatial image of the medium and high frequencies look very different from the original image.

Chapter 3

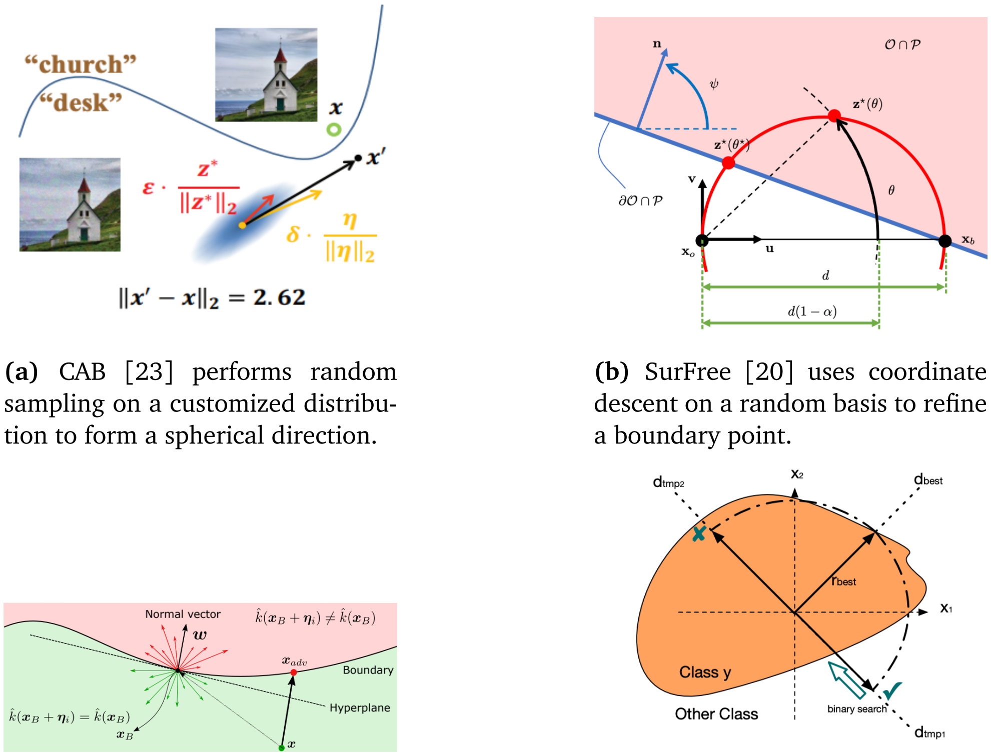
Related Work

In this chapter, we discuss related work in the field of decision-based black-box adversarial attacks. The chapter is divided into attacks in the spatial and frequency domain. First, we address the spatial domain with a brief summary of the state-of-the-art before two attack methods are detailed. Then a state-of-the-art summary and one attack method in the frequency domain are described. We denote decision-based attacks on the spatial and frequency domain as *Spatial Attacks* and *Frequency Attacks* respectively. This chapter highlights the limitations of the current attack methods.

3.1 Decision-based Attacks in Spatial Domain

The general approach of spatial attacks [20–23] is to traverse the decision boundary of the target model to minimize the distance between the original image and the adversarial example. The traversal of the decision boundary is mainly approached in a geometric manner, as illustrated in Figure 3.1. We see how different methods move the adversarial example towards the original image without crossing the decision boundary. The spatial attacks are initiated with an adversarial noise sampled from a spatial distribution, i.e., random noise values from a given distribution in the spatial domain. Then, a binary search is usually performed to approach the decision boundary. At the boundary, the methods differ in ways to calculate the next step closer to the original image. They all have individual geometrical approaches for moving towards the original image. Still, it is common to perform random sampling, i.e., sample a set of random perturbations in the local area. The number of sampled perturbations varies from method to method but usually consists of hundreds of samples due to a high number of search dimensions in images. The spatial attacks end when the query budget is reached, or the distance metric is below a set threshold.

A limitation of spatial attacks is the number of queries required at the boundary for random sampling. This leads to a larger number of queries needed to produce the final adversarial examples. The high amount of queries is a result of the large image dimensions. An RGB image of size (224 × 224 × 3) has 150 528 dimensions, resulting in a large search space.



(a) CAB [23] performs random sampling on a customized distribution to form a spherical direction.

(b) SurFree [20] uses coordinate descent on a random basis to refine a boundary point.

(c) GeoDA [21] estimates the normal vector *w* to the decision boundary hyperplane.

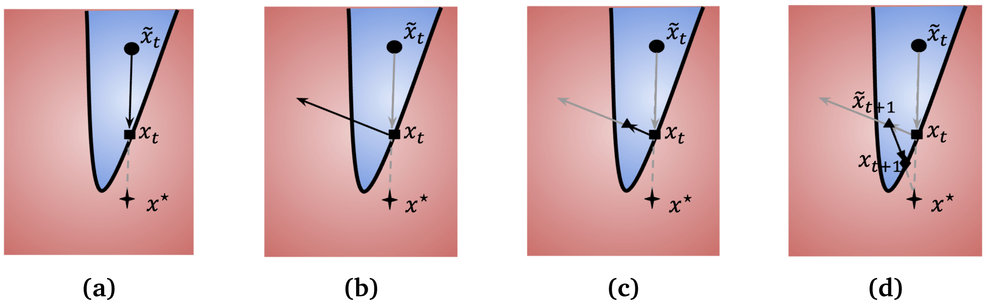
(d) RayS [22] directly search for the closest point decision boundary along a discrete set of ray directions.

Figure 3.1: The geometric approaches of spatial attacks.

3.1.1 HopSkipJumpAttack (HSJA)

As preliminary work for this thesis, we performed an Structured Literature Review (SLR) that compared black-box attacks against computer vision models. The SLR covers 29 distinct state-of-the-art attacks and can be found in Appendix A. Among these attacks, we find the work of Chen *et al.* [19] known as HSJA. HSJA is a recently published (2020) attack method in the decision-based threat model. Based on their state-of-the-art performance, up-to-date results, relevance, and publicly available code, we chose HSJA as a baseline for our results.

Chen *et al.* [19] presents HSJA, a hyperparameter-free and query-efficient decision-based black-box attack for both targeted and untargeted attack settings. The proposed algorithm is iteration-based with three components to an iteration and is intuitively explained in Figure 3.2. Because the attack method op-



Source: Chen *et al.* [19]

Figure 3.2: Intuitive explanation of a single iteration t of the HSJA algorithm. (a) Binary search to approach the decision boundary. (b) Gradient direction estimation. (c) Geometric progression to produce a valid step size. (d) Binary search back to the decision boundary.

erates in the decision-based threat model the attackers only have access to the output label. From this information, Chen *et al.* [19] defines a boolean function $\phi_{x^*} : [0,1]^d \rightarrow \{-1,1\}$ where $\phi_{x^*}(x) = 1$ if and only if x is adversarial. The overall goal of the attack method can then be summarized as generating an adversarial example x' such that $\phi_{x^*}(x') = 1$ while minimizing the distance between x' and the original image.

As with other decision-based attacks, HSJA requires an initial adversarial example $\tilde{x}_t$ usually sampled from a Gaussian distribution such that $\phi_{x^*}(\tilde{x}_t) = 1$. The first component in HSJA (Figure 3.2a) moves the initial adversarial example $\tilde{x}_t$ to the decision boundary, resulting in the image x_t. This operation is done through a binary search between the original image x^* and the adversarial example $\tilde{x}_t$. The binary search is performed over a blending factor $\alpha \in [0,1]$ to determine how much the initial adversarial example $\tilde{x}_t$ can be blended with the original image x^* while still satisfying $\phi_{x^*}(\tilde{x}_t) = 1$. When the binary search reaches a predetermined threshold HSJA updates the adversarial example $\tilde{x}_t \rightarrow x_t$.

The second component of HSJA (Figure 3.2b) uses a novel approach to estimate the gradient direction at the decision boundary by using binary information acquired from unbiased sampling. The gradient estimation is done by sampling B independent and identically distributed vectors $\{u_b\}_{b=1}^{B}$ from a uniform distribution over the d-dimensional sphere. Then, the direction of the gradient $\nabla S_{x^*}(x_t)$ is approximated via the Monte Carlo estimate

$$\widetilde{\nabla S}(x_t, \delta) := \frac{1}{B} \sum_{b=1}^{B} \phi_{x^*}(x_t + \delta u_b) u_b \tag{3.1}$$

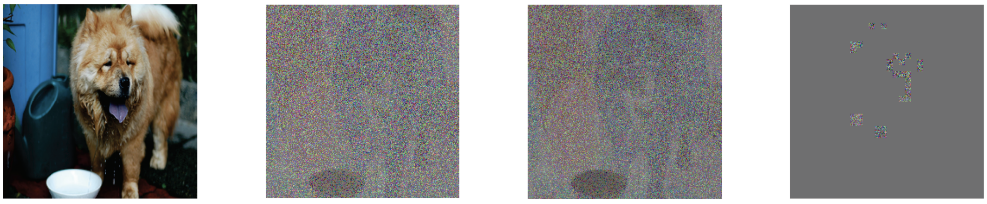
where δ is a small positive parameter. The novel gradient direction estimation makes HSJA require significantly fewer model queries than previous state-of-the-art methods [10, 31, 32], and Chen *et al.* [19] also demonstrates lower l_2 and l_∞ distances compared to other methods across multiple datasets and models.

The third component in HSJA (Figure 3.2c) uses geometric progression of a step size $\xi_t := \|x_t - x^*\|_2/\sqrt{t}$ to identify a valid step size along the gradient direction. ξ_t is decreased by half until it satisfies $\phi_{x^*}(\tilde{x}_t) = 1$. The geometric progression gives us the adversarial image $\tilde{x}_{t+1}$ which then can be moved to the decision boundary again using a binary search as illustrated in Figure 3.2d. This binary search concludes the t-th iteration of HSJA and prepares the attack method for another iteration.

Despite its novelty, the bulk of model queries used in HSJA comes from gradient direction estimation. The gradient estimation is performed in the spatial domain which requires more samples in order to produce a gradient estimate, due to its high dimensionality. HSJA performs this step because the algorithm requires evaluation of the target model when near the decision boundary. In fact, Chen *et al.* [19] mentions this as a limitation of all decision-based attack methods, and that decision-based attacks may not be effective when limiting the number of queries used at the boundary.

3.1.2 Patch-wise Adversarial Removal (PAR)

Shi and Han [33] presents Patch-wise Adversarial Removal (PAR), an attack method that removes redundant noise of adversarial examples. Rather than traversing the decision boundary, PAR divides the input image into coarse-to-fine patches. Shi and Han [33] explore how sensitivity and magnitude on each patch can be used to remove redundant noise, thus reducing the overall degree of the perturbation. The authors state that initial noise can be compressed due to different noise sensitivities in the image. As shown in Figure 3.3, PAR compresses the initial noise to a lower l_2 distance than Boundary Attack [10] in the same amount of queries.



Original image

Initial noise $l_2 = 99.60$

Boundary noise $l_2 = 72.60$

PAR noise $l_2 = 12.62$

Source: Shi and Han [33]

Figure 3.3: Noise comparison between Boundary Attack by Brendel *et al.* [10] and PAR by Shi and Han [33] after 100 model queries from the same initial noise.

PAR works by iteratively querying the target model to see if a certain part of the initial noise is redundant or necessary. First, the attack method divides the initial noise into coarse patches of size $PS = PS_0 \times PS_0$, which defines the initial patch size based on a hyperparameter PS_0. Then, the noise magnitude of each patch is recorded in a noise magnitude mask M_N as the l_2 distance between the original image and the adversarial example in that particular patch. In addition to the noise magnitude mask, PAR also keeps track of a noise sensitivity mask M_S. This mask is a binary mask where 1 indicates that the noise in this patch has been successfully removed or has not yet tried to remove the noise. A 0 indicates that the noise removal failed for this patch, meaning the noise in this patch is essential for keeping the image adversarial. PAR combines M_N and M_S through an element-wise product to obtain a query-value mask $M_Q = M_N \odot M_S$. Due to the properties of the element-wise product operator and the binary nature of M_S, PAR can sort the values of M_Q in descending order and remove the noise in the patch with the highest value in M_Q. This patch will have the highest noise magnitude and has not yet been queried to the target model. If the query result of the updated adversarial example is adversarial, it indicates that the noise sensitivity in that patch is low and that the noise is redundant and can be removed. If it is not adversarial, the corresponding value in M_S is set to 0. Preferably, the attack method should first remove noise in the patches with low noise sensitivity and high noise magnitude. Sorting M_Q in a descending order ensures that this happens, which can significantly reduce the overall l_2 distance between the original image and the adversarial example if successful. If all patches of patch size PS have removed their noise or performed an unsuccessful query, the sum of M_Q reaches 0. When this happens PAR halves the patch size $PS_{i+1} = PS_i/2$. The two masks M_N and M_S are then reinitialized before a new iteration of the attack method is conducted on the patches that still contain noise. PAR stops when it reaches one of two predefined values: the query budget or the minimum patch size.

The noise compression of PAR greatly reduces the initial noise, which can speed up the process of a subsequent attack method [23]. This property makes PAR particularly suitable as an initialization method for other attack methods. Shi and Han [33] illustrate this through their results, which show that other attack methods achieve better results if initiated with PAR in the same amount of queries. Even though Shi and Han [33] presents a query-efficient method to compress initial noise, PAR is more powerful in combination with existing decision-based attack methods. Additionally, Shi and Han [33] mainly targets Vision Transformers (ViTs) [34] which can be a limitation to attack methods against CNN based image classifiers.

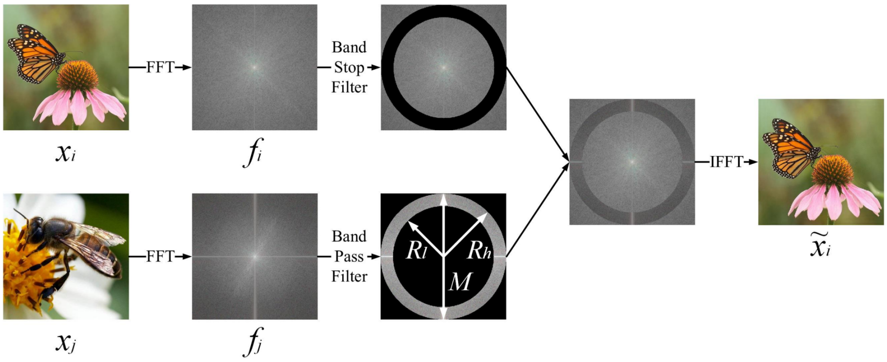
3.2 Decision-based Attacks in Frequency Domain

The general approach of frequency attacks [27, 29, 35–38] addresses a limitation of spatial attacks, which is to reduce the search space of adversarial perturbations. Most frequency attacks initiate, traverse the decision-boundary, and finish similar to spatial attacks. The main difference between the spatial and frequency attacks lies in the dimensionality of the space used for sampling random perturbations. In the high-dimensional spatial domain, an attack method can sample many unnecessary non-adversarial directions causing a higher number of required queries. Guo *et al.* [37] shows that adversarial examples exist abundantly in a very low-dimensional low-frequency subspace, meaning that adversarial directions occur much more often than in the high-dimensional spatial domain. Thus, adversarial perturbations sampled from the low-frequency subspace have a significantly lower number of required queries. This property allows for more query-efficient attack methods by sampling from the low dimensional frequency domain.

3.2.1 F-mixup

Li *et al.* [39] differentiate themselves from general frequency attacks mentioned previously by introducing a novel attack method F-mixup in the high frequencies of the magnitude spectrum, as opposed to performing random sampling in the low frequencies. Therefore F-mixup introduces a new idea for adversarial attacks — to explore how modification of frequency components can create adversarial examples.

F-mixup is a targeted attack that consists of mixing up the low-frequency component of an image x and the high-frequency component of the target image x^*. The result of the mixup is a new example x' which looks like x to a human, but is classified as x^* by the target model. The attack algorithm is illustrated in Figure 3.4, where the magnitude spectrums of x and x' are obtained with FFT.



Source: Li *et al.* [39]

Figure 3.4: F-mixup algorithm. Frequency spectrums are obtained through FFT and combined through IFFT to produce an adversarial example.

The magnitude spectrums are combined with the band stop filter from x and band pass filter from x' with parameters R_l and fixed R_h. To find the optimal band pass and stop filter, the algorithm performs a sampling of m random values R_l, where m is the query budget. The combined magnitude spectrums are converted back to spatial domain with IFFT, and queried to the target model for evaluation. The m adversarial examples are evaluated on the l_2 distance to the original image x. If an adversarial example is found, the example with the lowest l_2 distance is chosen.

Even though Li *et al.* [39] does not claim state-of-the-art performance with F-mixup, their contributions reveal a large potential to create imperceptible adversarial examples in the magnitude spectrum. They show that adversarial examples can lie in the high-frequency component of natural images. The main limitation of F-mixup is the exclusion of medium and low-frequency components. Chen *et al.* [40] argue that CNNs extract features from different frequencies and Guo *et al.* [37] show that adversarial perturbations also lie in the low frequencies. Additionally, F-mixup does not implement untargeted attacks but can only perform targeted attacks. Another limitation is the static R_h variable in the algorithm, which forces the method to sample in the high frequencies. A dynamic approach to selecting R_h and R_l could improve the ability to search in all frequency components.

Chapter 4

Methodology

In this chapter, we explain our methodology. We first discuss our motivation for producing a novel black-box attack by contextualizing the limitations of current black-box attacks. Then we propose a research question based on the discussed motivation, before explaining our novel attack method and how it will answer the research question.

4.1 Motivation

The necessity to query hundreds of perturbations at the decision boundary remains the main limitation for all decision-based attack methods [19], both spatial and frequency attacks. For that reason, state-of-the-art attack methods require thousands of queries to create imperceptible adversarial examples. The queries needed could significantly be reduced by circumventing the need for sampling at the decision boundary.

Li *et al.* [39] reveals a new idea for targeted adversarial attack methods by proposing to utilize the magnitude spectrum to create imperceptible adversarial examples. F-mixup inserts the high frequencies of one image into another image, demonstrating the potential of adversarial perturbations in the high-frequency components. However, Li *et al.* [39] do not investigate the potential of other frequency bands, even though Guo *et al.* [37] shows that adversarial perturbations also exist in the low frequencies. A limitation of all existing untargeted attack methods is the sampling of initial noise. The initial noise is normally sampled from a Gaussian distribution in the spatial domain [19], but could benefit from being sampled from the low-dimensional frequency domain [37].

To fill these gaps in decision-based attacks, we perform novel research exploring query-efficiency and imperceptibility in generating adversarial examples. This research aims to circumvent the need for sampling at the decision boundary and utilizes the frequency domain to produce a query-efficient attack method.

4.2 Research Questions

To create a clear goal for the thesis and to summarize our motivation, we propose the following research question as presented in chapter 1:

> **RQ:** *How to create an untargeted state-of-the-art query-efficient decision-based black-box attack against robust image classification models to produce less perceptible adversarial examples?*

The following section describes our contribution through our proposed attack method. Our contribution consists of various novel methods utilizing the frequency domain information to form a novel decision-based black-box attack method addressing the current limitations of the state-of-the-art.

4.3 Attack Design

We propose an answer to our research question through Magnitude Adversarial Spectrum Search-based Attack (MASSA), a novel untargeted decision-based black-box attack that directly modifies the entire frequency spectrum of an image to produce adversarial examples efficiently. The following points can characterize the proposed attack method:

1. The attack samples initial noise in the frequency domain.
2. The attack reduces the perturbation size in each frequency band.
3. The attack removes redundant noise.

These points categorize MASSA into three main components: noise generation, noise reduction, and removal of redundant noise. An illustration of these components and the high-level attack pipeline is found in Figure 4.1. This section first describes how we divide the frequency spectrum into separate frequency bands before explaining each part of the attack pipeline in detail.

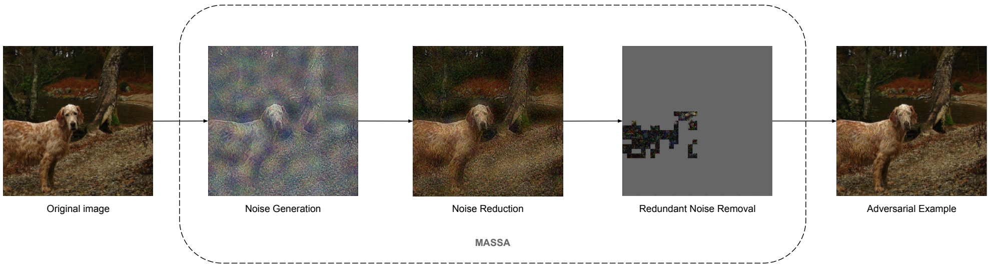


Figure 4.1: The overall attack pipeline of the proposed MASSA attack.

4.3.1 Creating frequency bands

We recall from subsection 2.4.2 that the frequency spectrum can be divided into three bands: low-, medium-, and high frequencies. We also recall that each frequency band affects the spatial image differently and that, ideally, we want to modify each band separately. Therefore, we separate the frequency spectrum into these three bands. Each spatial image decomposes into a different frequency spectrum with different frequency bands, which means finding the thresholds dividing the frequency spectrum into different bands is not a straightforward task. We use statistical analysis of the frequency spectrum for each image channel to calculate r_1 and r_2, the two radiuses which divide the frequency spectrum into low-, medium-, and high-frequency bands. Exactly how r_1 and r_2 divide the frequency spectrum is illustrated in Figure 4.2, where the innermost circle with radius r_1 contains most of the low frequencies, the annulus between r_1 and r_2 contains mostly medium frequencies, and everything outside the circle with radius r_2 contains high frequencies.

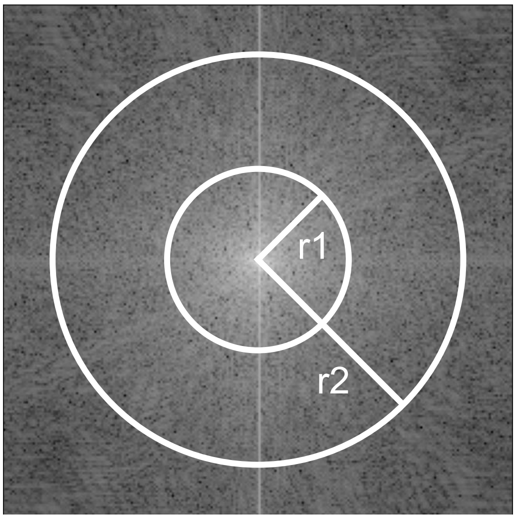
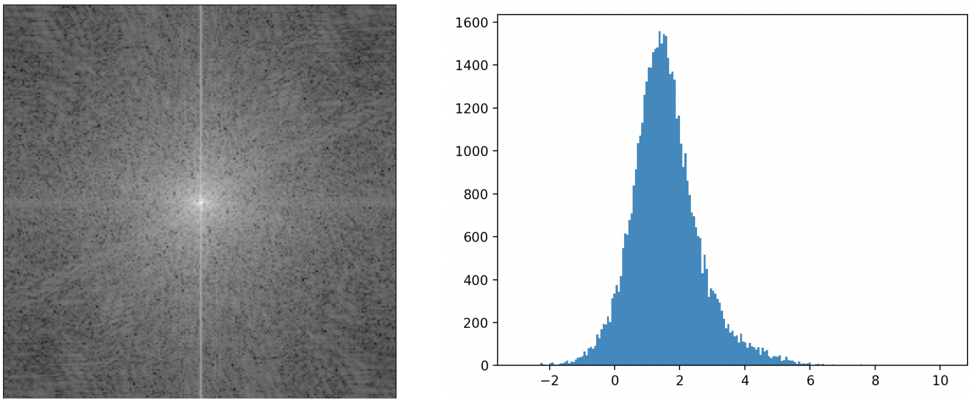


Figure 4.2: Illustration of the different frequency bands. Most low frequencies are located in the circle with radius r_1, the medium frequencies are mostly located in the circle with radius r_2, while the high frequencies are mostly contained outside this circle.

To determine r_1 and r_2, we study the value range of the logarithmically scaled frequency spectrum of an image. Recall from subsection 2.4.2 that we log-scale the values for visualization because of their large value range. Figure 4.3a illustrates the frequency spectrum of an image and Figure 4.3b its corresponding histogram of values. We can see from the histogram that most of the values in the frequency spectrum range between 0-3. Additionally, we have some smaller values towards -3 and some larger ones towards 10. The histogram in Figure 4.3b somewhat resembles a normal distribution with a mean of about $\mu = 1.5$ and standard deviation $\sigma = 1$. The values for μ and σ depend on the frequency spectrum, but the key takeaway is that the frequency spectrum values loosely resemble a normal distribution. Because of this property, we continue our statistical analysis based on the assumption that the values in the frequency spectrum follow the described normal distribution, with the exception of a longer right tail than a left tail.



(a) Frequency spectrum

(b) Distribution of values in the frequency spectrum. $\mu = 1.5$

Figure 4.3: Frequency spectrum and its value distribution

From subsection 2.4.2, we know that the high-frequency band contains the smallest values in the frequency spectrum, represented by the left tail in Figure 4.3b. Similarly, the low-frequency band is represented by the right tail, with the medium frequencies between the two tails. In order to decide on values for r_1 and r_2 we first need to identify two tail-values t_l and t_r that divide the histogram into three parts, one for each frequency band. We define the left t_l and right tail t_r as

$$t_l = \mu - \alpha_l \sigma \quad \text{and} \quad t_r = \mu + \alpha_r \sigma, \tag{4.1}$$

where μ is the mean, σ is the standard deviation, and α_l and α_r are scaling factors for the left and right tails respectively. To translate the tail-values of the histogram to the 2D frequency domain we define a mask for each frequency band:

$$M_h = F_{i,j} < t_l$$

$$M_m = t_l < F_{i,j} < t_r$$

$$M_l = F_{i,j} > t_r$$

where $F_{i,j}$ is the value of the frequency spectrum at position (i, j), t_l and t_r are the left and right tail values respectively. Each 2D mask M will contain all values in the frequency domain belonging to that band, i.e. M_l will contain all low-frequency values, given the threshold-value t_r. Then, for each mask, we calculate the euclidean distance between each value $F_{i,j}$ in the mask and the center as $d_{i,j} = \sqrt{i^2 + j^2}$. We can then use the average euclidean distance to calculate r_1 and r_2 like so:

$$r_1 = \frac{1}{|M_l|} \sum_{i,j} \sqrt{i^2 + j^2}, \quad r_2 = \frac{1}{|M_m|} \sum_{i,j} \sqrt{i^2 + j^2},$$

where $|M_l|$ and $|M_m|$ denote the number of values in the low-frequency mask and medium-frequency mask respectively. In summary, the radiuses used to create the frequency bands are dependent on the tail-values used to divide the histogram in Figure 4.3b. Recalling that the distribution of magnitude values loosely resembles a normal distribution, we chose a scaling factor of $\alpha_l = 2$ for t_l, and a factor of $\alpha_r = 3$ for t_r in order to compensate for the longer right side tail.

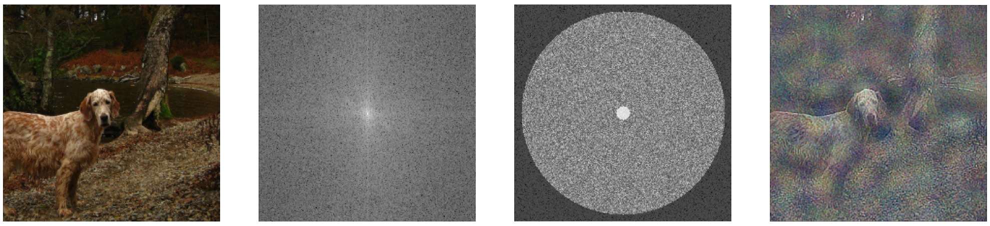
In summary, we use $t_l = \mu - 2\sigma$ and $t_r = \mu + 3\sigma$ as tail values, which are then used to calculate appropriate values for r_1 and r_2, allowing us to split the frequency spectrum into a low, medium, and high frequency band for separate modification at a later stage. μ and σ depend on the value distribution of the frequency spectrum, meaning we get unique tail values for each spatial image while also compensating for the long right tail with the α scaling values. These calculations are performed for each image channel to produce as accurate frequency bands as possible.

4.3.2 Initial Noise Generation

All decision-based attack methods in our related work sample the initial perturbation from the spatial domain. We propose to sample the initial perturbation from the frequency domain instead, such that we can generate an initial perturbation with noise in all frequency components.

Our method includes a novel initiation method to generate the initial perturbation. The goal is to create an adversarial perturbation as required by the subsequent *Noise Reduction* component. For simplicity, we only describe the process for a single channel, but it is easily extended to three-dimensional images channel-wise. Figure 4.4 illustrates each step in the noise generation component. We consider the original image x of size $d \times d$ and its frequency spectrum F of the same size. As performed by Li *et al.* [39], we shift the low frequencies of the frequency spectrum to the center and scale the values logarithmically, which results in Figure 4.4b. We use $F_{i,j}$ to index the magnitude values at position (i, j).

Unlike Chen *et al.* [19] which samples the noise from the normal distribution $N(0, 1)^D$ in the spatial domain, we directly perturb the frequency spectrum of x. To perturb the frequency spectrum, we divide the spectrum into three frequency bands: high, medium, and low. As explained in subsection 4.3.1, we use r_1 and r_2 to divide the frequency spectrum into three frequency bands. The low-frequency band given by r_1 covers a centered circle, the medium frequency band given by r_2 covers an annulus around the low-frequency band, and the high frequencies



(a) Original image

(b) Frequency spectrum

(c) Modified frequency spectrum

(d) Adversarial example

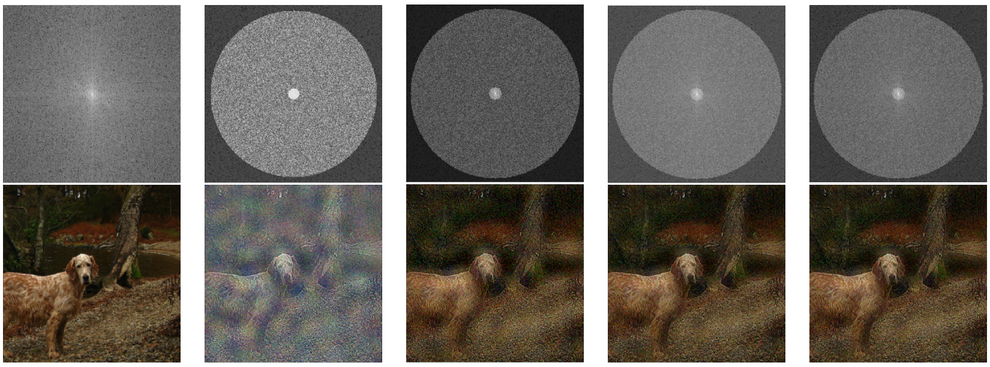
Figure 4.4: Visualization of the initiation process. (a) The original image in the spatial domain. (b) The frequency spectrum of the original image. (c) The frequency spectrum after insertion of noise in the frequency bands. (d) The perturbed image in the spatial domain.

cover everything else. Figure 4.4c illustrate these frequency bands. Based on r_1 and r_2 we can determine which band $F_{i,j}$ belongs to.

To create the perturbation seen in Figure 4.4d, we directly modify the values in the frequency spectrum. First, we use the mean μ and standard deviation σFigure 4.3b to replicate the distribution based on the assumption that it resembles a normal distribution. We use N^* to denote this replicated distribution. Then, for each band, we replace each value $F_{i,j}$ with a sample from N^*. For the low frequency band, we only sample values in the range $[F_{min}, t_l]$, where F_{min} is the lowest value in the frequency spectrum and t_l is the left tail value calculated by Equation 4.1 with $\alpha = 2$. Values in the medium frequency band are replaced with values sampled from N^* in the range $[t_l, t_r]$. Lastly, the values in the high-frequency band are replaced with values from N^* in the range $[t_r, F_{max}]$ where F_{max} is the largest value in the frequency spectrum. This process results in a perturbed frequency spectrum F', illustrated in Figure 4.4c, which consists of random values for all $F_{i,j}$ where the values in each band remain in their original range and with a similar distribution. Lastly, IFFT transforms the perturbed frequency spectrum back to the spatial domain resulting in Figure 4.4d. This adversarial example serves as the input to the noise reduction component of our attack method.

4.3.3 Noise Reduction

To circumvent the need to sample at the boundary, we design a novel reduction method to minimize the distance between the frequency spectrum F of the original image and the frequency spectrum F' of the initial perturbation. We call this method *Frequency Spectrum Binary Search*. Inspired by the use of binary search to efficiently minimize the distance between two images in the spatial domain, we redesign the binary search to minimize the distance between two frequency spectrums while only modifying values in a given band. This binary search method is then conducted separately for each frequency band. Since the most important features of an image is located in the low frequencies [41], we perform the first



(a) $l_2 = 0.0$ **(b)** $l_2 = 92.76$ **(c)** $l_2 = 27.00$ **(d)** $l_2 = 21.40$ **(e)** $l_2 = 21.39$

Figure 4.5: Visualization of the reduction process. The top row is the frequency spectrum in each step, and the bottom row is the corresponding image in the spatial domain. (a) The original frequency spectrum F and image x. (b) The initial perturbation before the reduction method. (c) After binary search in the low-frequency band. (d) After binary search in the medium-frequency band. (e) After binary search in the high-frequency band.

binary search in this band. This allows the low-frequency values to move closer to their original values because the noise in the medium and high frequencies helps keep the image adversarial. We then move to the medium-frequency band for the same reason, and finally, binary search in the high-frequency band. The overall reduction process is illustrated in Figure 4.5 and shows how drastically the distance to the original image is reduced through binary searches in the low, medium, and high-frequency bands.

Chen *et al.* [19] propose a traditional binary search to approach the decision boundary as *Algorithm 1* in their paper. We apply this algorithm in the frequency spectrum and modify it to only adjust values for the given band b, either the low, medium, or high-frequency band. Our modified version aims to reduce the frequencies across bands proportionally, allowing each frequency band to stay in its original value range. Keeping the value ranges consistent is essential since each frequency band has a different impact on the spatial image. Our redesigned algorithm used for noise reduction is detailed in Algorithm 1.

4.3.4 Removal of Redundant Noise

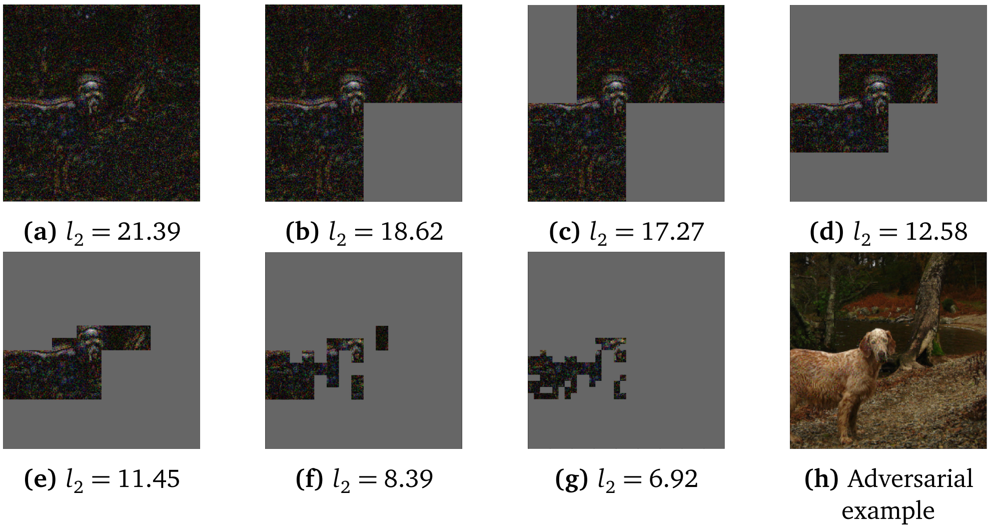
The last component of our attack method is the removal of redundant noise. Shi and Han [33] reveals that most noise in the initial adversarial example is redundant, and one can speed up subsequent decision-based attacks by removing the redundant noise [23]. Based on this discovery, we propose to remove redundant noise as the final step of our attack method. Similar to Shi and Han [33], we perform this removal through a coarse-to-fine patch-wise manner.

Our redundant noise removal process is based on a trial-and-error approach. Given an adversarial example x' of size $d \times d$ we first divide the image into four

Algorithm 1 Frequency Spectrum Binary Search

```
Require: Original frequency spectrum $F$ and adversarial frequency spectrum $F'$.
  Binary function $\phi$, such that $\phi(F') = 1$ and $\phi(F) = 0$. Threshold $\theta$, and band
  $b$.
Ensure: An adversarial frequency spectrum $F''$ closer to $F$ in band $b$
  $\alpha_l \leftarrow 0$
  $\alpha_u \leftarrow 1$
  while $|\alpha_u - \alpha_l| > \theta$ do
      $\alpha_m \leftarrow \frac{\alpha_l + \alpha_u}{2}$
      $F'' = \Pi^b_{F,\alpha_m}(F')$          ▷ Average frequencies between $F$ and $F'$ in band $b$
      if $\phi(F'') = 1$ then                  ▷ Check if $F''$ is adversarial
          $\alpha_u \leftarrow \alpha_m$
      else
          $\alpha_l \leftarrow \alpha_m$
      end if
  end while
  Output $F'' = \Pi^b_{F,\alpha_u}(F')$.
```

coarse patches of size $\frac{d}{2} \times \frac{d}{2}$. We then iteratively remove the noise in each patch and query the model to see if that noise patch is necessary for misclassification. In this way, we are able to remove large parts of unnecessary noise, which further decreases the l_2 distance. After performing this on each patch, we recursively perform the same steps on each patch where the noise was not removed. This gradually moves from coarse patches to finer patches as the size of each patch decreases in each iteration. The iterative process is visualized in Figure 4.6, which shows how the redundant noise is removed from the adversarial perturbation. For visualization purposes, we subtracted the original image from the adversarial examples after the noise reduction component, meaning Figure 4.6a depicts only the changes made to the original image. Each step after that removes patches of different sizes, clearly illustrating that a lot of noise is redundant. The noise removal process ends when the minimum patch size is reached. To finish of this process we perform a binary search to ensure that the final adversarial example is close to the decision boundary. The result of this component gives us the final adversarial example, which contains minimal redundant noise, i.e., a low l_2 distance.



(a) $l_2 = 21.39$ **(b)** $l_2 = 18.62$ **(c)** $l_2 = 17.27$ **(d)** $l_2 = 12.58$

(e) $l_2 = 11.45$ **(f)** $l_2 = 8.39$ **(g)** $l_2 = 6.92$ **(h)** Adversarial example

Figure 4.6: Visualization of the noise removal process. For each image, the unnecessary noise patches have been removed, here replaced by gray for visualization purposes. Between each image, the patch size is halved as we move from coarse to fine patches. (a) The noise appended to the original image. (b-g) Noise removed with patch sizes 112 × 112 to 7 × 7 respectively. (h) The final adversarial example.

Chapter 5

Evaluation & Results

This section presents the evaluation of MASSA. First, we describe the evaluation procedure for our experimental analysis. Then, we present the results for evaluating efficiency and defense mechanisms compared to HSJA, as previously introduced in subsection 3.1.1.

5.1 Evaluation Procedure

The evaluation procedure is used to ensure the quality of the results to compare our attack method to the state-of-the-art. The following section describes the evaluation methods used consistently throughout our experiments. First, we introduce how we evaluate the efficiency of MASSA against a carefully chosen baseline. Then we detail the evaluation procedure of MASSA against target models which have implemented a defense mechanism.

We compare the evaluation procedures of our related work from chapter 3 in order to design our own evaluation procedure. A summary of the evaluation procedures of our related work is shown in Table 5.1, which describes the included datasets, models, and metrics. The related work has overlapping evaluation procedures but differs mainly in the datasets and models used in the experiments.

Among the related work in chapter 3, we chose HSJA by Chen *et al.* [19] as a baseline for our evaluations. PAR introduces a powerful redundant noise removal approach but achieves the best performance as an initiation method instead of a standalone attack method. Additionally, PAR mainly focuses on attacking ViTs, even though some CNN models are included in their experiments. Therefore we chose to exclude PAR from our baseline. F-mixup is another powerful attack method, but it is only evaluated on the CIFAR-10 dataset. The fact that F-mixup is a targeted attack makes it incompatible for comparison with our untargeted attack method. F-mixup is not an iterative attack method because it samples a given amount of queries and chooses the best candidate. This makes the number of queries needed and l_2 distance metrics challenging to compare to an iterative attack method. Hence, F-mixup is also excluded from our baseline. Since HSJA is our only baseline, we mainly design our evaluation procedure around theirs in terms of Table 5.1, in order to make a reasonable performance comparison of the two attack methods.

		HSJA [19]	PAR [33]	f-mixup [39]
Dataset	ILSVRC-2012	✓	✓	
	ImageNet-21k		✓	
	Tiny-Imagenet		✓	
	MNIST	✓		
	CIFAR10	✓		✓
	CIFAR100	✓		
Models	AlexNet			✓
	ResNet-32			✓
	ResNet-50	✓	✓	
	ResNet-101		✓	
	DenseNet-121	✓	✓	
	Simple CNN	✓		
	VGG16			✓
	VGG19		✓	
	SeNet		✓	
	ViTs		✓	
Metrics	Number of Queries	✓	✓	✓
	Success Rate	✓		✓
	Median l_2 distance	✓	✓	
	Average l_2 distance		✓	✓
	Median l_∞ distance	✓		

Table 5.1: Summarized evaluation procedures of related work. *ViTs* are various Vision Transformers different from CNN based target models.

5.1.1 Dataset

ImageNet [42] is an image database consisting of 14 197 122 images, where 21 841 classes are indexed. The data is available for free to researchers for non-commercial use. A subset of ImageNet is the ILSVRC2012 Challenge, which includes 1000 classes and consists of 1.28 million images for training, 50 000 images for validation, and 100 000 images for testing. The validation set has 50 images for each class. The first ten classes and corresponding labels are shown in Table 5.2. We use the ILSVRC2012 Challenge validation dataset for all our experiments as it was used for evaluation by Chen *et al.* [19] and Shi and Han [33]. ImageNet images are considered large with respect to other common datasets such as the CIFAR family and MNIST. In comparison, ImageNet offers images of varying sizes with an average size of 469 × 387, while the CIFAR datasets offer 32 × 32 images and MNIST 28 × 28 images. Larger image size makes the ImageNet dataset more relevant to real-world applications [43], while the CIFAR and MNIST datasets can be left out. Due to the varying image sizes of ImageNet, the classification models apply a preprocessing step for input images [14, 15]. The images are scaled to 256 pixels on the shortest side and then center-cropped to 224 × 224. The preprocessing guarantees a consistent image size of 224 × 224 and comparable results between attack methods. Figure 5.1 shows three images before and after the preprocessing step.

Class ID	Labels
0	tench, Tinca tinca
1	goldfish, Carassius auratus
2	great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias
3	tiger shark, Galeocerdo cuvieri
4	hammerhead, hammerhead shark
5	electric ray, crampfish, numbfish, torpedo
6	stingray
7	cock
8	hen
9	ostrich, Struthio camelus

Table 5.2: Class ID and labels of the first 10 ILSVRC2012 classes.

5.1.2 Classification Models

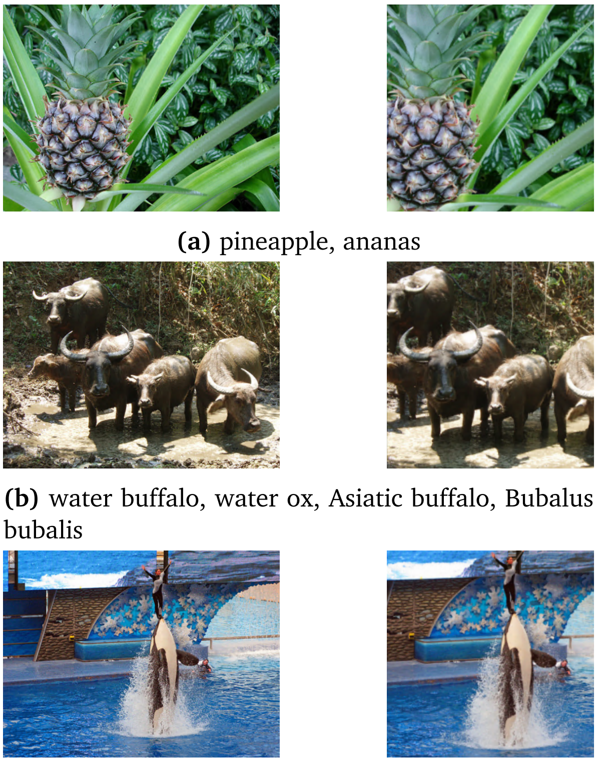
Ideally, a comprehensive evaluation consists of various models to ensure the results. In order to compare the results to the baseline, the experimental setup required models from the Tensorflow library version 1.2.1 [44]. Therefore, we include three different image classification models available in this library, which all achieve state-of-the-art performance on the ImageNet dataset. Each model is pre-trained on the ImageNet training dataset and requires image input of size 224 × 224.

ResNet-50

ResNet-50 was introduced by He *et al.* [14] and presented a residual learning framework to train deep neural networks. The ResNet-50 version of the model uses a 50-layer deep architecture and achieves an accuracy of 77.15% on the ImageNet test dataset.

VGG16

VGG16 was introduced by Simonyan and Zisserman [15] who investigated the increasing depth of CNNs. VGG16 uses a 16-layer deep architecture and achieves an accuracy of 72.7% on the ImageNet test dataset.



(a) pineapple, ananas

(b) water buffalo, water ox, Asiatic buffalo, Bubalus bubalis

(c) killer whale, killer, orca, grampus, sea wolf, Orcinus orca

Figure 5.1: Three images and corresponding classes from ImageNet. Left: Original size. Right: Scaled and center-cropped.

VGG19

VGG19 was also introduced by Simonyan and Zisserman [15]. In addition to investigating a 16-layer deep architecture, they also evaluated a 19-layer deep architecture which achieved an accuracy of 74.5% on the ImageNet test dataset.

5.1.3 Metrics

In our research question in section 4.2 we stated our focus on query-efficiency and less perceptible adversarial examples. We use four different metrics to evaluate these metrics properly: number of queries, l_2 distance, success rate, and query finish rate, which we present in this subsection. With the exception of query finish rate, all these metrics are also used by Chen *et al.* [19].

Number of queries

The number of queries represents how many times the attack method has to query the target model. The optimal attack method would require as few as possible queries to execute the attack. The more queries used during the attack, the larger possibility of detection by a defence mechanism as mentioned in Appendix A. We limit the maximum number of queries to 1000, in difference to Chen *et al.* [19] which limits to 25 000 queries.

l_2 distance

We use the l_2 distance metric to measure the imperceptibility of the adversarial perturbation. We described the l_2 distance more in detail in subsection 2.2.3. An ideal attack method should create a perturbation with a minimal l_2 distance between the original image and the adversarial example. We see from Table 5.1 how l_2 distance is more common between the related work, both as a median and average distance across multiple images. We chose to leave out the l_∞ metric as it is not as widely used among our related work.

Success rate

Another metric we use to evaluate attack efficiency is the success rate. We know from section 2.3.2 that most of all decision-based attack methods traverse the decision boundary on the adversarial side, meaning all perturbations during the attack are adversarial. Because of this, the success rate is measured at various l_2 distance thresholds where the attack methods are given a maximum query budget. The definition of success is that the l_2 distance does not exceed a given distance threshold. For example, the success rate can tell us how many of the adversarial examples produced by the attack method under a maximum query limit are below a certain l_2 distance.

Query finish rate

Lastly, we introduce a new metric separately from Table 5.1: Query Finish Rate. HSJA is an iterative algorithm that converges, i.e., it can run infinitely if it is not stopped by a query limit or target l_2 distance. Our attack method, MASSA, has a ceiling on its number of queries. Both the initial noise generation described in subsection 4.3.2 and the noise reduction step described in subsection 4.3.3 only use a (small) finite number of queries. And, given a fixed image input size of 224×224 and a minimum patch size of 7, the redundant noise removal described in subsection 4.3.4 also has an upper limit on its number of queries. In the worst case scenario where no noise is redundant it uses a maximum of $\sum_{i=1}^{5} 4^i = 1364$ queries. If there is any redundant noise, the algorithm will use fewer queries. Depending on how much noise is redundant, we get the number of queries ρ used to create an adversarial example by MASSA. The distribution of ρ over a set of images is what we define as Query Finish Rate.

5.1.4 Defense mechanisms

We want to investigate the robustness of various defense mechanisms under our proposed attack method. As a specialization project for this thesis, we performed an Structured Literature Review (SLR) on black-box attacks which can be found in Appendix A. From this work, we identified the following defense mechanisms:

- Input Transformations
- Adversarial Training
- Adversarial Detection
- Adversarial Distillation
- Region-based classification
- Rounded output probabilities
- Query-access prevention

Input Transformations cover multiple defense mechanisms such as clipping, median filtering, and JPEG compression. Dziugaite *et al.* [45] propose to use JPEG compression as a defense method. Their experiments show that JPEG compression can reverse adversarial perturbation on images modified by a small magnitude. Wallace [46] details the compression method with a transformation to the frequency domain. JPEG compression uses the properties of the frequency domain to remove irrelevant data from the image. The nature of JPEG compression in terms of perturbation reversal in the frequency domain has the potential to work as a resistant defense mechanism. It is interesting to explore whether the input transformation with JPEG compression can defend against our attack method since we rely heavily on the frequency domain. The other input transformations are not included since they are less relevant to our attack method, and we chose to focus on JPEG compression as an input transformation.

Adversarial training [47] increases robustness by including adversarial examples in the training data. To the best of our efforts, we could only find one adversarially trained model compatible with our chosen classification models. Engstrom *et al.* [48] presents a robustness library that provides available weights for ResNet-50 adversarially trained on the CIFAR-10 and ImageNet dataset. The adversarial examples are generated with PGD [49], a state-of-the-art white-box attack method. These robust models are trained on different degrees of perturbations denoted by ϵ. A model without defense has a robustness level $\epsilon = 0$. On the other hand, a highly robust model has a robustness level $\epsilon = 3$. The benefits of adversarial training come at the cost of accuracy. The adversarial trained ResNet-50 with $\epsilon = 3$ has accuracy 57.90%, compared to ResNet-50 with $\epsilon = 0$ which has accuracy 77.15%. Even though black-box attacks have previously been shown to evade adversarial training [9], we want to investigate how adversarial training affects our frequency-based attack MASSA.

Adversarial detection [50], such as feature squeezing [51] and MagNet [52], uses statistical testing to detect adversarial examples based on their properties. They train a model with an additional outlier class to detect the adversarial examples based on the fact that the distribution of adversarial examples are different to training examples. Similarly, adversarial distillation [53] is a training procedure that uses transferred knowledge from a different model for gradient masking. To the best of our efforts, we could not find trained models for detection or distillation for the ImageNet dataset. Therefore, evaluating these defense mechanisms would require us to train a model ourselves on the ImageNet dataset. This would directly influence our results as the experiments would only depict data which directly depend on the performance of our trained model. In other words, our evaluation of the defense mechanism would only be as good as our trained model. To avoid this pitfall, we chose not to include adversarial detection and adversarial distillation.

Region-based classification [54] samples points from a hypercube with side-length *r* centered at the input image. Then it predicts based on which label appears most frequently among the sampled points. The defense mechanism estimates the hyperparameter *r* based on a validation dataset and is only calculated for MNIST and CIFAR-10. We chose to exclude this defense mechanism as it would require us to calculate a new *r* for the ImageNet validation dataset, which gives us the same problem as adversarial detection and distillation.

Our research question clearly defines the threat model we assume for our attack method, namely a decision-based attack. Rounded output probabilities only affect score-based attacks since they rely directly on the output probabilities. In our case, this defense mechanism would not affect our results. The same argument can be used for query-access prevention, as the decision-based threat model already assumes query-access. Based on this, we chose to exclude these defense mechanisms.

5.2 Efficiency Evaluation

We ran multiple experiments following the evaluation procedure as described in section 5.1 to evaluate our approach. We implemented a system to carry out adversarial attacks on different image classification models. As a baseline for comparison, we use the implementation of HSJA from their publicly available code [55]. All experiments were carried out on a Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz with 32.0GB of RAM. Each experiment uses a set of 500 correctly classified random images from the ILSVRC2012 Challenge validation dataset. Our code is publicly available online on GitHub [56].

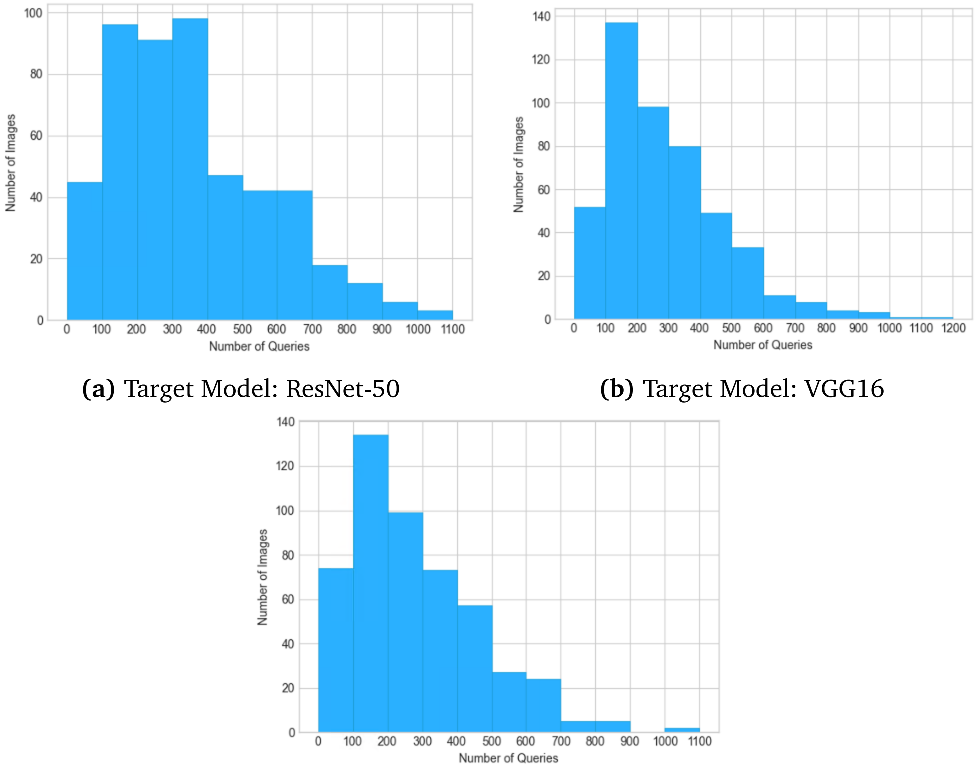
In this section, we present our results from the evaluation process. First, we discuss the Query Finish Rate before comparing the median and average l_2 distances of our approach with HSJA. Then we present the success rates, and finally, we demonstrate how the defense mechanisms affect the performance of the attack methods.

5.2.1 Query Finish Rate

Our first experiment explores the Query Finish Rate for MASSA. Table 5.3 illustrate how many of the 500 adversarial examples our approach produced were

Models	Model Queries			
	< 250	< 500	< 750	< 1000
ResNet-50	38.20%	75.40%	93.80%	99.40%
VGG16	49.48%	87.21%	97.48%	99.58%
VGG19	52.68%	87.58%	98.32%	99.66%

Table 5.3: The Query Finish Rate for each target model. It shows the percentage of how many MASSA executions finish with less than 250, 500, 750, and 1000 queries, respectively.



(a) Target Model: ResNet-50

(b) Target Model: VGG16

(c) Target Model: VGG19

Figure 5.2: Histogram of query finish rate for each target model.

created using less than a given query interval. For example, 75.4% of our adversarial examples were created using less than 500 queries on the ResNet-50 target model. Additionally, the Query Finish Rate is illustrated as a histogram in Figure 5.2. Each bin has a size of 100, where the ranges are [0, 100), [100, 200), and so on.

The results show that MASSA easily generates adversarial examples in less than 1000 queries. In fact, 50% of the adversarial examples are created using less than 250 queries for VGG16 and VGG19. This shows how MASSA is able to conduct a powerful attack under a very limited query budget. To further support this result, more than 90% of adversarial examples are created using less than 750 queries for all models. This demonstrates that there is no need to push a query budget of over 1000 queries. We see the same trends in Figure 5.2, where very few adversarial examples require 1000 queries. It is also interesting to notice how MASSA is able to produce a significant amount of adversarial examples in less than 100 queries. For instance, MASSA generates more than 40 adversarial images against ResNet-50 in less than 100 queries, demonstrating the effectiveness of our approach. From the high query usage in Table 5.3 and Figure 5.2 we observe that ResNet-50 is more challenging to create adversarial examples against compared to the VGG models. Although this is the case for small query budgets, we notice the difference between the models becomes negligible when approaching a query budget of 1000. In subsection 5.1.3 we mentioned the worst-case scenario for MASSA in terms of the number of queries. The results show that this is a rare occurrence as less than 1% of adversarial examples produced by MASSA require 1000 queries or more.

5.2.2 l_2 Distance

In the next experiment, we investigate the performance of MASSA in terms of l_2 distance compared to the baseline. Table B.1 in Appendix B contains a summary of all l_2 distance experiments for comparison across all experiments. Table 5.4 summarizes the median and average l_2 distances for both attack methods with query budgets of 250, 500, 750, and 1000 queries. We also illustrate the median distance results in Figure 5.3. The spikes for HSJA in Figure 5.3 comes from the geometric progression, as explained in subsection 3.1.1, where the adversarial example is moved away from the decision boundary. This step causes the adversarial example to move away from the original image, hence the sudden spikes. The plateaus for HSJA come from the gradient direction estimation step, also explained in subsection 3.1.1. Here, HSJA samples and queries hundreds of adversarial examples around the boundary, where all samples have approximately the same l_2 distance to the original image. For simplicity, we plot this as a straight line since the differences are insignificant. The HSJA plot in Figure 5.3 also helps visualize each iteration of the attack method. For MASSA in Figure 5.3 we see some plateaus in the first 50 queries. These come from the noise reduction component of our attack method, as explained in subsection 4.3.3, where each plateau corresponds to a different frequency band being moved closer to its original values. As mentioned in subsection 5.2.1 our attack method rarely uses 1000 queries, so in Figure 5.3 we have padded each result from their stopping point with their respective end l_2 distance up to 1000 queries in order to compute the median.

Table 5.4 clearly shows that MASSA can create adversarial examples with a significantly smaller l_2 distance than the corresponding adversarial examples created by HSJA. For all comparisons made in Table 5.4, MASSA beats HSJA across all models and query budgets. The decrease in l_2 distance is given in percentages for each comparison between the two attacks. The differences are most apparent on ResNet-50 and VGG16 at a query budget of 250, where MASSA achieves approximately 74% lower median l_2 distance than HSJA. Even after 1000 queries MASSA still beats HSJA with a $41,77\%$ lower median l_2 distance on ResNet-50. On top of that, we see MASSA, on a query budget of 250, beats HSJA on a 1000 query budget with a $33,18\%$ lower median l_2 distance. This demonstrates the efficiency of MASSA under a very limited query budget. Both attacks show better performance against the VGG models than ResNet-50, which may be caused by ResNet-50 having a higher accuracy score on the ImageNet test dataset.

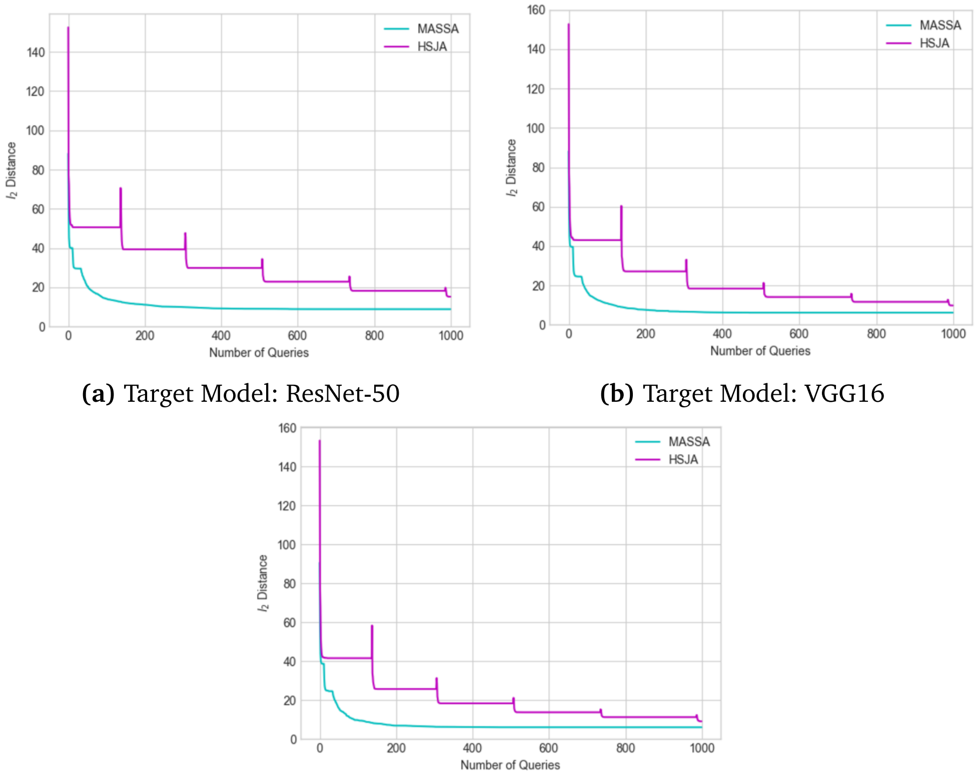
Models	l_2 distance	Model Queries							
		250		500		750		1000	
		HSJA	MASSA	HSJA	MASSA	HSJA	MASSA	HSJA	MASSA
ResNet50	Median	39.33	**10.19** (-74.09%)	29.85	**9.06** (-69.65%)	18.23	**8.88** (-51.29%)	15.25	**8.88** (-41.77%)
	Average	40.24	**12.66** (-68.54%)	32.29	**11.07** (-65.72%)	22.04	**10.66** (-51.63%)	18.54	**10.58** (-33.21%)
VGG16	Median	27.09	**7.00** (-74.16%)	18.37	**6.12** (-66.68%)	11.60	**6.12** (-47.24%)	9.77	**6.12** (-37.36%)
	Average	31.87	**9.66** (-69.69%)	23.91	**8.56** (-64.20%)	15.91	**8.35** (-47.52%)	13.71	**8.32** (-39.31%)
VGG19	Median	25.69	**7.00** (-72.75%)	18.37	**6.05** (-67.07%)	11.26	**6.05** (-46.27%)	9.10	**6.05** (-33.52%)
	Average	29.61	**9.10** (-69.27%)	21.85	**8.08** (-63.02%)	14.40	**7.86** (-45.42%)	12.25	**7.84** (-36.00%)

Table 5.4: Median and average distance at various model queries for each target model. The smaller distance at a given model query is bold-faced.

Figure 5.3 shows that MASSA achieves a steeper decrease in l_2 distance than HSJA, where MASSA descend to an l_2 distance of 20-30 in the first 50 queries. This demonstrates the effectiveness of the frequency binary search explained in subsection 4.3.3. Although MASSA ends on a lower l_2 distance than HSJA for all target models, we can see a sign of HSJA catching up. If we let the experiments run past a query budget of 1000, HSJA may beat MASSA in l_2 distance, due to its convergence property. We still argue that this scenario is irrelevant, as a query budget of more than 1000 queries moves the attack outside the range of state-of-the-art performance and is less realistic in real-world situations. All-in-all, we observe that MASSA achieves better results than HSJA by generating more imperceptible adversarial examples with a significantly lower query budget.

We include visualized trajectories of MASSA in Figure 5.4. The trajectories are selected randomly from correctly classified images from the ILSVRC2012 validation set. The first column displays the initial perturbation generated by Initial

Noise Generation described in subsection 4.3.2. Columns 2 - 5 display perturbations at 25, 50, 100 and 200 queries, respectively. Lastly, column 6 displays the original image. We observe from column 5 that the final adversarial examples are similar to the original image.



(a) Target Model: ResNet-50

(b) Target Model: VGG16

(c) Target Model: VGG19

Figure 5.3: Median l_2 distances versus number of queries for each target model.

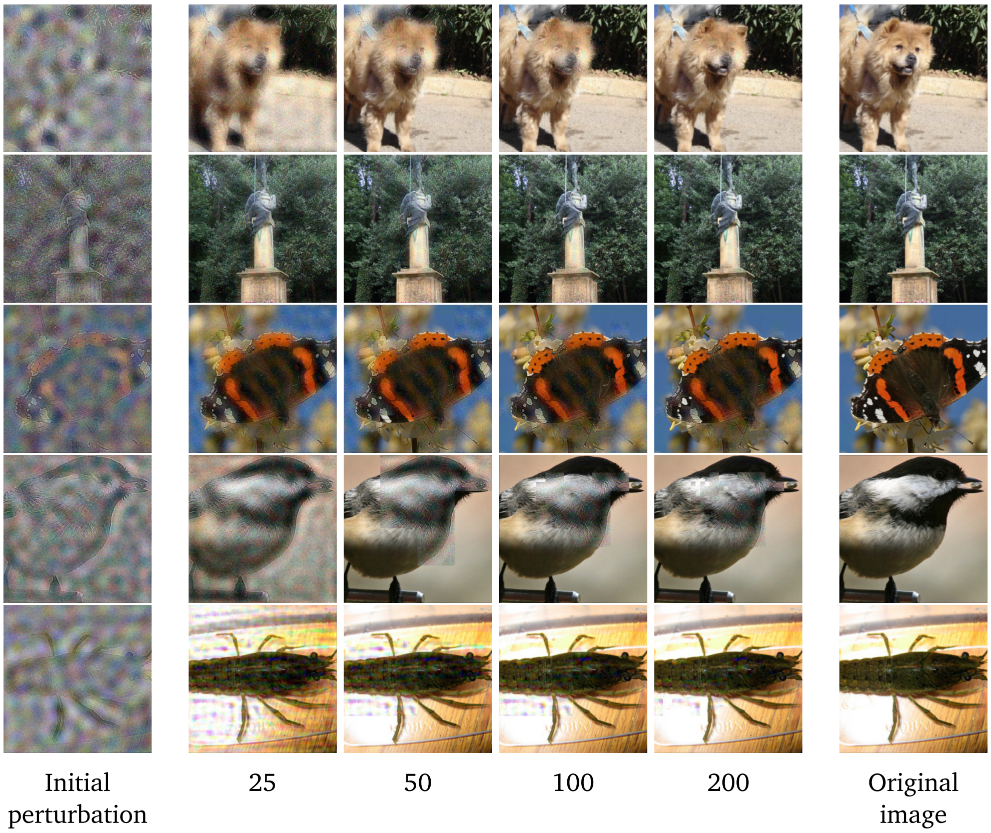
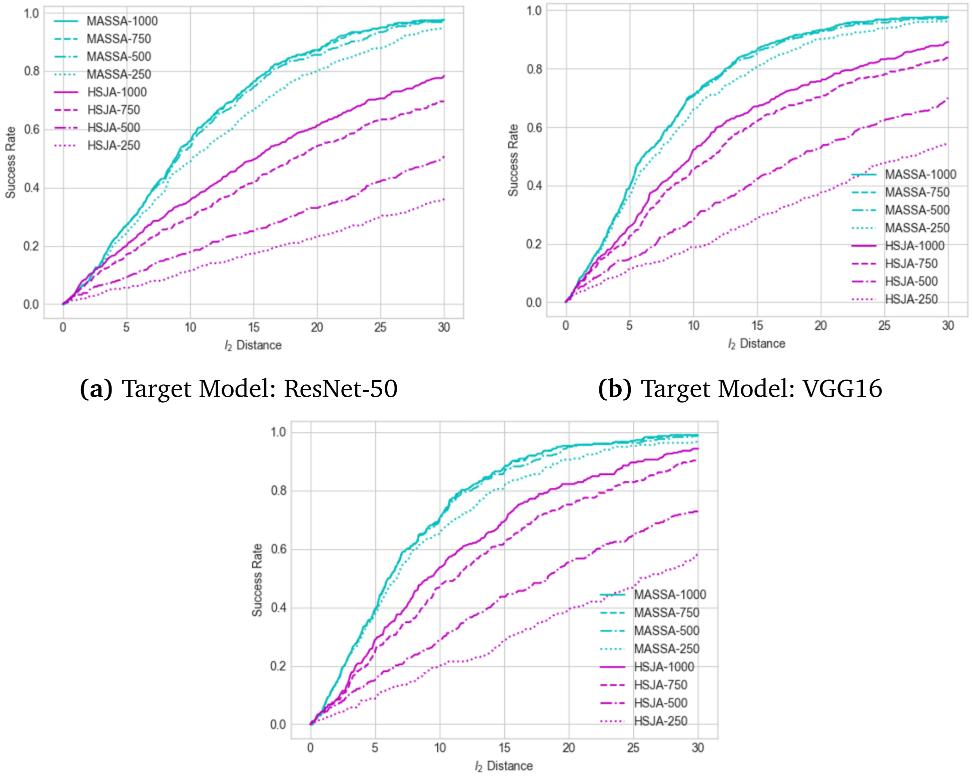


Figure 5.4: Visualized trajectories of MASSA for 5 images from ILSVRC2012 validation dataset. 1st column: initial perturbation. Columns 2-5: adversarial examples at 25, 50, 100, and 200 queries, respectively. Last column: Original image.

5.2.3 Success Rate

We also evaluate the success rate of MASSA compared to HSJA. Figure 5.5 illustrates the success rate at various thresholds between $[0, 30]$ in l_2 distance. We study each attack method under four different query budgets: 250, 500, 750, and 1000. The legend indicates the name of the attack method and the size of the query budget, e.g., MASSA with a budget of 750 queries is denoted *MASSA-750*. Figure 5.5 shows the superior performance of MASSA compared to HSJA. All query budgets of MASSA achieve a consistently higher success rate than the respective HSJA attack. It is also worth noting that *MASSA-250* achieves a significantly higher success rate than *HSJA-1000*, exemplifying how MASSA is a more query-efficient attack than HSJA. Additionally, all MASSA attacks are similar across models than HSJA, which might support that MASSA has higher transferability between models and is a more generalizable attack.



(a) Target Model: ResNet-50

(b) Target Model: VGG16

(c) Target Model: VGG19

Figure 5.5: Success rate for various l_2 distance thresholds for each target model.

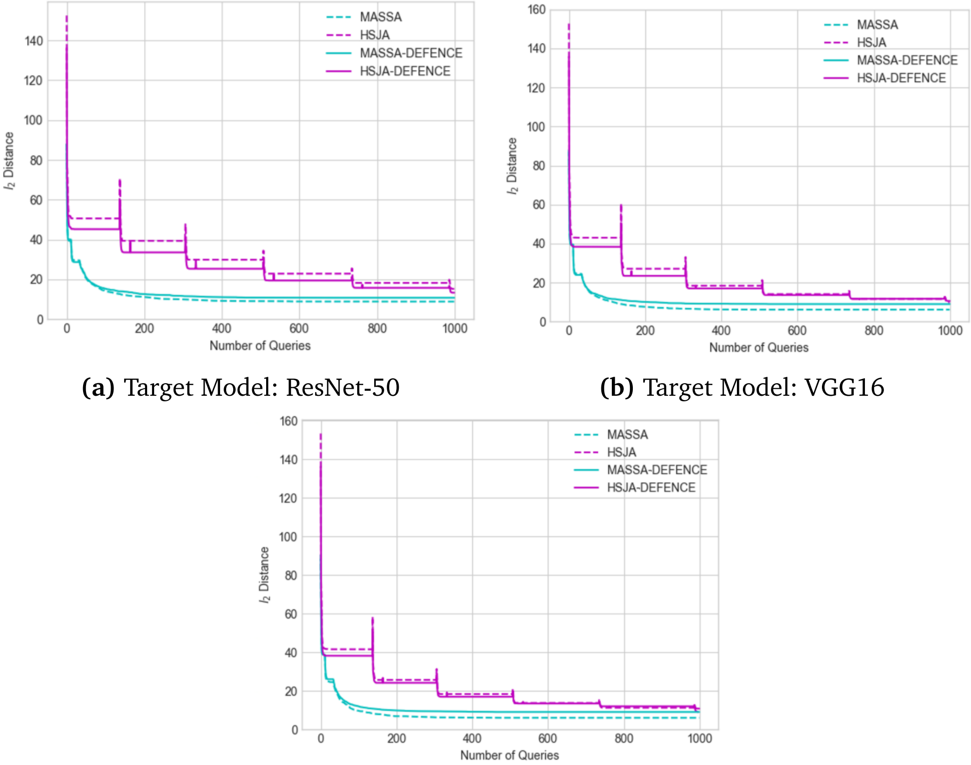
5.2.4 Evaluation Under JPEG Compression

To evaluate our attack method under defense mechanisms, we use l_2 distance and success rate as metrics, These metrics clearly illustrate how the performances of the attack methods are affected by a defense mechanism. We evaluate attack efficiency using JPEG compression as a defense mechanism where the adversarial example is compressed right before querying the target model. Table 5.5 summarizes the results of median and average l_2 distance for HSJA and MASSA across target models with JPEG compression. As shown in Table 5.4, we see that MASSA still beats HSJA in all comparisons, even with a defense mechanism implemented. Furthermore, MASSA under a 250 query budget still outperforms HSJA on a 1000 query budget across all models, although the improvement is slightly reduced under JPEG compression. Table 5.5 also reveals a slight increase in l_2 distance for MASSA, e.g. under a query budget of 500 we see a 19,2% increase in median l_2 distance on ResNet-50. Thus, JPEG compression slightly affects MASSA. Although the l_2 distances have slightly increased, MASSA still beats HSJA in all comparisons. It is also worth noting here that JPEG compression improves the results for HSJA. This is because JPEG compression removes the high frequencies in an image where noise is already located, meaning JPEG compression can further reduce the l_2 distance of adversarial examples from HSJA. This is not the case for MASSA, as we actively modify and use the values in the high frequencies to create an adversarial example. However, since we also modify the medium and low frequencies, the defense mechanism has small impact on the performance of MASSA.

Models	l_2 distance	Model Queries							
		250		500		750		1000	
		HSJA	MASSA	HSJA	MASSA	HSJA	MASSA	HSJA	MASSA
ResNet50	Median	33.56	**12.16**	25.34	**10.80**	15.78	**10.77**	13.32	**10.77**
	Average	35.90	**14.08**	28.88	**12.69**	19.87	**12.37**	17.21	**12.31**
VGG16	Median	23.42	**9.67**	16.99	**9.04**	11.68	**8.96**	10.58	**8.96**
	Average	27.92	**11.80**	21.37	**10.89**	15.12	**10.71**	13.51	**10.70**
VGG19	Median	24.20	**9.51**	16.97	**9.08**	12.02	**9.08**	10.81	**9.08**
	Average	27.78	**11.90**	21.12	**10.94**	14.73	**10.76**	13.08	**10.75**

Table 5.5: Median and average distance at various model queries for each target model with JPEG compression. The smaller distance at a given model query is bold-faced.

We show the median l_2 distance and success rate under JPEG-compression in Figure 5.6 and Figure 5.7, respectively. In Figure 5.6, we show each attack method both with JPEG compression (denoted with DEFENCE) and without JPEG compression for ease of comparison. Figure 5.7 only includes the attack methods under JPEG compression. In Figure 5.6 we see that the early decrease in l_2 distance for MASSA remains unchanged after JPEG compression. The noise reduction step in subsection 4.3.3 first modifies the low and medium frequencies, which JPEG compression never modifies. After about 50 queries we start to see the impact of JPEG compression on MASSA. We also see how HSJA improves during its first iterations but evens out towards 1000 queries because JPEG compression increases the l_2 distance HSJA converges to.



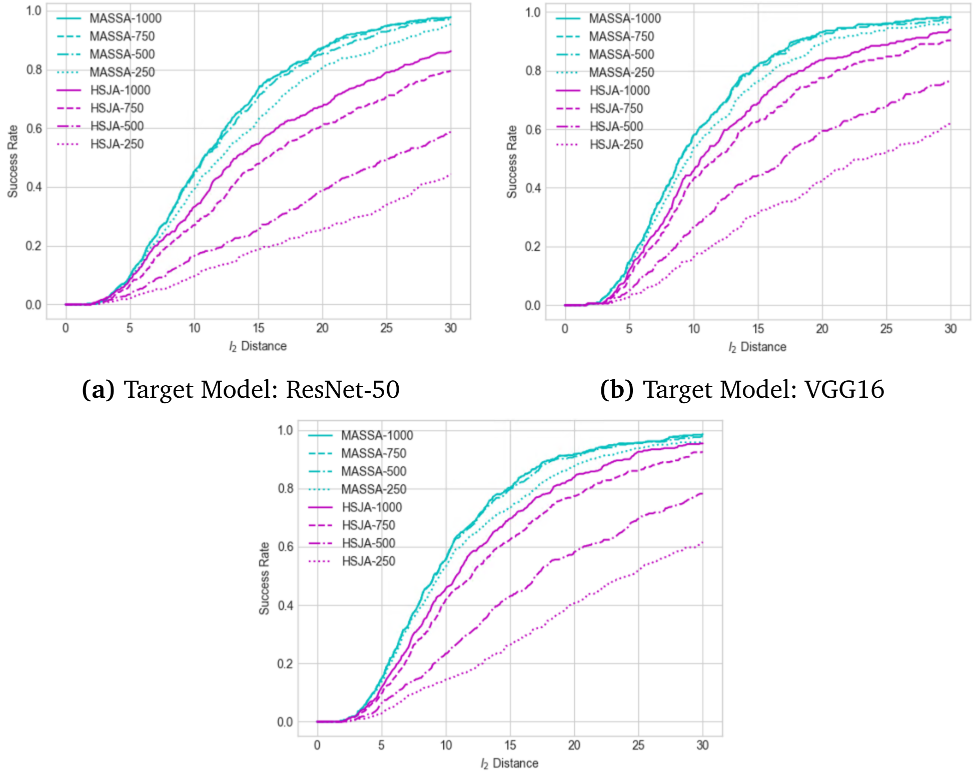
(a) Target Model: ResNet-50

(b) Target Model: VGG16

(c) Target Model: VGG19

Figure 5.6: Median l_2 distances versus number of queries for each target model with JPEG compression.

Figure 5.7 shows that MASSA achieves a higher success rate than HSJA under JPEG compression. We observe that even *MASSA-250* still outperforms *HSJA-1000*. In difference to the success rate without JPEG compression, both attack methods struggle to generate adversarial examples with a l_2 distance below 5 in this case. An explanation for this could be that adversarial examples with $l_2 < 5$ already have a very small perturbation, which will be mostly located in the high frequencies of the frequency spectrum. Since JPEG compression removes high frequencies, the defense mechanism can convert the adversarial examples into non-adversarial images, resulting in a low success rate. In adversarial examples with distances $l_2 > 5$, the perturbations are located in the high frequencies and in the medium and low frequencies. Since JPEG compression never touches these frequencies, the attack methods can still produce powerful results against the defense mechanism.



(a) Target Model: ResNet-50

(b) Target Model: VGG16

(c) Target Model: VGG19

Figure 5.7: Success rate for various thresholds of l_2 distance for each target model with JPEG compression.

5.2.5 Evaluation Under Adversarial Training

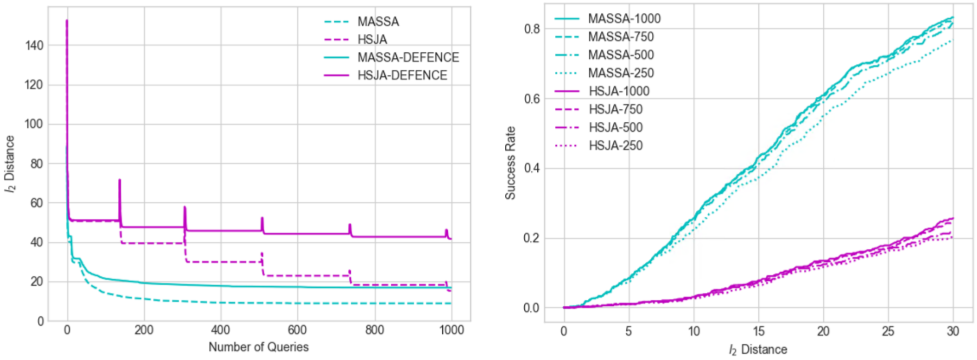
We conduct experiments to evaluate MASSA and HSJA under adversarial training. Table 5.6 summarizes the results of median and average l_2 distance for the attack methods against an adversarial trained ResNet-50 with $\epsilon = 3$. MASSA still beats HSJA in every comparison. At most MASSA beats HSJA by 61.97% at 500 model queries. Compared to the results without any defense mechanisms in Table 5.4, for median distance at 500 queries, we see an increase of 89.53% for MASSA and 172.98% for HSJA. This indicates that adversarial training is a valid defense mechanism against adversarial attacks. However, we see that the results for MASSA under adversarial training are still comparable to HSJA without any defense mechanisms.

We illustrate the median distance and success rate for adversarial trained ResNet-50 in Figure 5.8. We include both MASSA and HSJA with and without defense for comparison in Figure 5.8a, where the adversarial model is denoted with *DEFENCE*. Figure 5.8b shows the success rate of each attack method under various query budgets. We observe both attack methods achieve similar median

l_2 distance in the first 50-100 queries, but HSJA struggles to decrease the l_2 distance throughout the attack. MASSA is able to decrease the l_2 distance, but ends with a slightly higher l_2 distance. Figure 5.8b clearly shows how HSJA is affected by adversarial training. The success rate is significantly lower under adversarial training, in contrast to MASSA, which remains at a similar performance despite adversarial training.

Models	l_2 distance	Model Queries							
		250		500		750		1000	
		HSJA	MASSA	HSJA	MASSA	HSJA	MASSA	HSJA	MASSA
ResNet50	Median	47.55	**18.70**	45.70	**17.38**	42.65	**16.91**	41.63	**16.83**
	Average	48.22	**20.60**	46.83	**19.21**	44.73	**18.69**	43.80	**18.53**

Table 5.6: Median and average distance at various model queries for adversarial trained ResNet-50 model with $\epsilon = 3$. The smaller distance at a given model query is bold-faced.



(a) Median Distance **(b)** Success Rate

Figure 5.8: Median distance and success rate for ResNet-50 under adversarial training.

Chapter 6

Discussion

The following chapter discusses the results presented in chapter 5, and outlines how the results answer the research question proposed in section 4.2. First, we reflect on the position of our work regarding the related work presented in chapter 3. Then we consider the implications our work has for academia and industry. Lastly, we discuss the threats to the validity of the thesis.

6.1 Comparison to Related Work

Our thesis aims to create a query-efficient untargeted attack method that generates imperceptible adversarial examples. Li *et al.* [39] presents a targeted attack method in the frequency domain, but no study has explored an untargeted approach in the frequency domain to generate imperceptible perturbations. Shi and Han [33] argues that adversarial perturbations consist of redundant noise. Instead of using PAR as an initiation method, we implement the removal of redundant noise inspired by PAR as the final part of our attack method to reduce the imperceptibility. Unlike F-mixup, MASSA directly modifies all frequency components through the proposed *Frequency Spectrum Binary Search* to create imperceptible adversarial examples.

A binary search that moves the adversarial example closer to the original image is quite often used to approach the decision boundary [19]. The binary search itself is cheap to perform in terms of queries used. What is usually expensive is sampling at the decision boundary. HSJA and related work use hundreds of queries at this step to determine the gradient direction of the boundary. We circumvent the need to sample at the boundary and utilize only the cheap binary search to produce our adversarial example. We can see the effectiveness of this approach in our results, demonstrated by the steep decrease in l_2 distance in Figure 5.3. This makes our approach far more query-efficient than any related work. While other state-of-the-art attack methods operate with thousands of queries, we only need a few hundred. Additionally, the results show that the low query number does not affect the performance of our approach. We still achieve a significantly lower l_2 distance than HSJA, translating to less perceptible adversarial perturbations.

From our related work, only Chen *et al.* [19] includes evaluation under defense mechanisms. HSJA evaluates adversarial distillation and training on the MNIST dataset but does not include an evaluation on the ImageNet dataset. Our experiments evaluate both MASSA and HSJA against JPEG compression and adversarial training on the ImageNet dataset. Our results show that MASSA mitigates the defense mechanisms to a certain degree and outperforms HSJA in all experiments. Chen *et al.* [19] claims that HSJA bypasses adversarial training on MNIST, but we see from our results that HSJA struggles significantly under adversarial training on ImageNet.

To summarize, we propose an attack method that creates adversarial examples with higher imperceptibility and better query-efficiency than state-of-the-art HSJA. The results also show that MASSA bypasses the defense mechanisms and outperforms HSJA.

6.2 Academic Implications

Our approach operates under a significantly lower query budget than current state-of-the-art decision-based methods, representing a new effective attack type. It might not be sufficient to examine and defend against attacks with a scope of thousands of queries anymore. Defenses based on thousands of queries are easily bypassed by our new attack. Future attacks should aim to use this limited query budget to push the robustness of computer vision systems further.

We are also the first to directly modify all frequency components of an image to create adversarial examples. This demonstrates another gap in the community where the frequency domain may not be getting enough focus. We clearly illustrate the potential of frequency-based attacks for future research with our promising results.

Our method uses a redundant noise removal step inspired by PAR [33] which clearly shows how much of the generated noise is redundant. Other attacks might benefit from the same noise removal process, so further research into noise removal might be a smart approach moving forward. Noise removal processes have the potential to become add-ons which further reduces the l_2 distance of adversarial examples.

Based on our results on JPEG compression, we see the unrealized possibilities in other defense mechanisms targeting the frequency domain. As illustrated in Figure 4.5 and in F-mixup by Li *et al.* [39], the frequency-based attacks result in unnatural frequency spectrums. New defense mechanisms might detect these abnormal spectrums, making computer vision systems more robust against future attacks.

The evaluation results under adversarial training show the effectiveness of the defense mechanism against HSJA. It also indicates that frequency-based attack methods can bypass mitigate adversarial training. Therefore, we argue that research should focus on adversarial training with adversarial examples from the frequency domain to further increase robustness.

6.3 Implications for Industry

Our results also have some practical implications for the industry. We have demonstrated the potential to craft imperceptible adversarial examples in just hundreds of queries. This poses a more significant threat to the industry than current state-of-the-art attack methods because it is a more realistic approach. As we push the query budget lower, it might become more challenging for defense mechanisms to detect an attack. From the perspective of the target model, this might seem like a standard request, meaning a small enough query budget can make it difficult to separate an attack from normal behavior. Consequently, this can have severe implications for safety-critical computer vision systems.

6.4 Threats to Validity

For the purpose of integrity, we highlight some threats to the validity of our thesis. First, the removal of redundant noise step of our approach inspired by [33] greatly contributes to our results. This might indicate that there is a more optimal method for reducing the l_2 distance before removing redundant noise. Second, the experiments could include a more extensive set of images. Due to limited time and resources, we only included 500 images for each experiment. Third, the experiments could include a larger variety of classification models and datasets. Even though we included three models with state-of-the-art performance, many models with different architecture remain untested. A larger variety in model architectures could strengthen our results and demonstrate the attack transferability more clearly. More datasets would also benefit our experiments. Datasets with higher resolution than ImageNet could validate whether the frequency domain is favorable over the spatial domain regarding their large search space. Fourth, our baseline for comparison consists only of HSJA, which we argued in subsection 3.1.1. The results would be more comprehensive if we compared MASSA against several attack methods with state-of-the-art performance. Lastly, we have based the similarity of images on the l_2 distance. Other metrics which measure the similarity of images could be included in the experiments. For example, a similarity metric between images that favors human perceptibility over mathematical perceptibility could affect our results differently.

Chapter 7

Conclusion and Future Work

Inspired by the potential of frequency components, we propose a new decision-based black-box attack method, MASSA, which generates imperceptible adversarial examples under a strict query budget. The method includes two novel components, an initiation method that samples noise from the frequency domain and a *Frequency Spectrum Binary Search* to minimize the distance between two magnitude spectrums. We conduct a comprehensive evaluation of the efficiency under various defense mechanisms. The results demonstrate that MASSA achieves superior performance over the state-of-the-art attack HSJA across all classification models and defense settings. Furthermore, we show that MASSA can generate adversarial examples in less than a hundred queries and usually only require a few hundred queries. This demonstrates a significant leap from previous state-of-the-art attacks, which required thousands of queries. Additionally, MASSA bypasses two defense mechanisms with comparable results to HSJA without defense mechanisms.

Future work should focus on expanding the proposed attack method to the targeted attack setting by inserting different frequency components of the target image into the original image. This would no longer limit the attack to a single purpose but allow for a broader attack purpose. The attack method could be tested against unexplored defense mechanisms to evaluate their robustness. Future defense mechanisms should also explore the frequency domain to better understand adversarial examples and strengthen their overall robustness.

Bibliography

[1] H. A. Najada and I. Mahgoub, 'Autonomous vehicles safe-optimal trajectory selection based on big data analysis and predefined user preferences,' in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2016, pp. 1–6. DOI: `10.1109/UEMCON.2016.7777922`.

[2] B. M. Elbagoury, A.-B. M. Salem and L. Vladareanu, 'Intelligent adaptive precrash control for autonmous vehicle agents (cbr engine amp; hybrid a path planner),' in *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, Nov. 2016, pp. 429–436. DOI: `10.1109/ICAMechS.2016.7813486`.

[3] Z. Chen and X. Huang, 'Accurate and reliable detection of traffic lights using multiclass learning and multiobject tracking,' eng, *IEEE intelligent transportation systems magazine*, vol. 8, no. 4, pp. 28–42, 2016, ISSN: 1939-1390. DOI: `10.1109/MITS.2016.2605381`.

[4] F. Ebadi and M. Norouzi, 'Road terrain detection and classification algorithm based on the color feature extraction,' in *2017 Artificial Intelligence and Robotics (IRANOPEN)*, IEEE, 2017, pp. 139–146. DOI: `10.1109/RIOS.2017.7956457`.

[5] S. Hamdi, H. Faiedh, C. Souani and K. Besbes, 'Road signs classification by ann for real-time implementation,' in *2017 International Conference on Control, Automation and Diagnosis (ICCAD)*, IEEE, 2017, pp. 328–332. DOI: `10.1109/CADIAG.2017.8075679`.

[6] N. Technology. 'Sentiveillance sdk.' (2022), [Online]. Available: `https://www.neurotechnology.com/sentiveillance.html` (visited on 26/05/2022).

[7] MobileSec. 'Mobilesec android authentication framework.' (2022), [Online]. Available: `https://github.com/mobilesec/authentication-framework-module-face` (visited on 26/05/2022).

[8] I. J. Goodfellow, J. Shlens and C. Szegedy, *Explaining and harnessing adversarial examples*, 2014. DOI: `10.48550/ARXIV.1412.6572`. [Online]. Available: `https://arxiv.org/abs/1412.6572`.

[9] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik and A. Swami, *Practical black-box attacks against machine learning*, 2016. DOI: 10.48550/ARXIV.1602.02697. [Online]. Available: https://arxiv.org/abs/1602.02697.

[10] W. Brendel, J. Rauber and M. Bethge, *Decision-based adversarial attacks: Reliable attacks against black-box machine learning models*, 2017. DOI: 10.48550/ARXIV.1712.04248. [Online]. Available: https://arxiv.org/abs/1712.04248.

[11] P. Zhao, P.-Y. Chen, S. Wang and X. Lin, *Towards query-efficient black-box adversary with zeroth-order natural gradient descent*, 2020. DOI: 10.48550/ARXIV.2002.07891. [Online]. Available: https://arxiv.org/abs/2002.07891.

[12] RevolveAI. 'Computer vision applications in industry across sectors.' (2022), [Online]. Available: https://revolveai.com/computer-vision-applications/ (visited on 26/05/2022).

[13] S. Wang and Z. Su, *Metamorphic testing for object detection systems*, 2019. DOI: 10.48550/ARXIV.1912.12162. [Online]. Available: https://arxiv.org/abs/1912.12162.

[14] K. He, X. Zhang, S. Ren and J. Sun, *Deep residual learning for image recognition*, 2015. DOI: 10.48550/ARXIV.1512.03385. [Online]. Available: https://arxiv.org/abs/1512.03385.

[15] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2014. DOI: 10.48550/ARXIV.1409.1556. [Online]. Available: https://arxiv.org/abs/1409.1556.

[16] S. Avalos and J. M. Ortiz, *Geological modeling using a recursive convolutional neural networks approach*, 2019. DOI: 10.48550/ARXIV.1904.12190. [Online]. Available: https://arxiv.org/abs/1904.12190.

[17] D. E. Rumelhart, G. E. Hinton and R. J. Williams, *Learning representations by back-propagating errors*, 1986. DOI: 10.1038/323533a0. [Online]. Available: https://www.nature.com/articles/323533a0.

[18] S. Bhambri, S. Muku, A. Tulasi and A. B. Buduru, *A survey of black-box adversarial attacks on computer vision models*, 2019. DOI: 10.48550/ARXIV.1912.01667. [Online]. Available: https://arxiv.org/abs/1912.01667.

[19] J. Chen, M. I. Jordan and M. J. Wainwright, *Hopskipjumpattack: A query-efficient decision-based attack*, 2019. DOI: 10.48550/ARXIV.1904.02144. [Online]. Available: https://arxiv.org/abs/1904.02144.

[20] T. Maho, T. Furon and E. L. Merrer, 'Surfree: A fast surrogate-free black-box attack,' 2020. DOI: 10.48550/ARXIV.2011.12807. [Online]. Available: https://arxiv.org/abs/2011.12807.

[21] A. Rahmati, S.-M. Moosavi-Dezfooli, P. Frossard and H. Dai, *Geoda: A geometric framework for black-box adversarial attacks*, 2020. DOI: 10.48550/ARXIV.2003.06468. [Online]. Available: https://arxiv.org/abs/2003.06468.

[22] J. Chen and Q. Gu, *Rays: A ray searching method for hard-label adversarial attack*, 2020. DOI: 10.48550/ARXIV.2006.12792. [Online]. Available: https://arxiv.org/abs/2006.12792.

[23] Y. Shi, Y. Han and Q. Tian, 'Polishing decision-based adversarial noise with a customized sampling,' in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2020, pp. 1027–1035. DOI: 10.1109/CVPR42600.2020.00111.

[24] N. Pottekkat. 'Nsfw filter.' (2020), [Online]. Available: https://github.com/nsfw-filter/nsfw-filter (visited on 01/06/2022).

[25] H. Liu, R. Ji, J. Li, B. Zhang, Y. Gao, Y. Wu and F. Huang, 'Universal adversarial perturbation via prior driven uncertainty approximation,' in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, 2019, pp. 2941–2949. DOI: 10.1109/ICCV.2019.00303.

[26] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi and C.-J. Hsieh, 'ZOO,' in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ACM, Nov. 2017. DOI: 10.1145/3128572.3140448. [Online]. Available: https://doi.org/10.1145%2F3128572.3140448.

[27] X. Wang, Z. Zhang, K. Tong, D. Gong, K. He, Z. Li and W. Liu, *Triangle attack: A query-efficient decision-based adversarial attack*, 2021. DOI: 10.48550/ARXIV.2112.06569. [Online]. Available: https://arxiv.org/abs/2112.06569.

[28] R. Fisher, S. Perkins, A. Walker and E. Wolfart. 'Fourier transform.' ©HIPR. (2003), [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm (visited on 27/04/2022).

[29] S. N. Shukla, A. K. Sahu, D. Willmott and J. Z. Kolter, *Simple and efficient hard label black-box adversarial attacks in low query budget regimes*, 2020. DOI: 10.48550/ARXIV.2007.07210. [Online]. Available: https://arxiv.org/abs/2007.07210.

[30] R. Fisher, S. Perkins, A. Walker and E. Wolfart. 'Fourier transform.' ©HIPR. (2003), [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixlog.htm (visited on 27/04/2022).

[31] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang and C.-J. Hsieh, *Query-efficient hard-label black-box attack:an optimization-based approach*, 2018. DOI: 10.48550/ARXIV.1807.04457. [Online]. Available: https://arxiv.org/abs/1807.04457.

[32] A. Ilyas, L. Engstrom, A. Athalye and J. Lin, *Black-box adversarial attacks with limited queries and information*, 2018. DOI: 10.48550/ARXIV.1804.08598. [Online]. Available: https://arxiv.org/abs/1804.08598.

[33] Y. Shi and Y. Han, *Decision-based black-box attack against vision transformers via patch-wise adversarial removal*, 2021. DOI: 10.48550/ARXIV.2112.03492. [Online]. Available: https://arxiv.org/abs/2112.03492.

[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2020. DOI: 10.48550/ARXIV.2010.11929. [Online]. Available: https://arxiv.org/abs/2010.11929.

[35] S. Liu, J. Sun and J. Li, 'Query-efficient hard-label black-box attacks using biased sampling,' in *2020 Chinese Automation Congress (CAC)*, IEEE, 2020, pp. 3872–3877. DOI: 10.1109/CAC51589.2020.9326734.

[36] W. Zhao and Z. Zeng, 'Improved black-box attack based on query and perturbation distribution,' in *2021 13th International Conference on Advanced Computational Intelligence (ICACI)*, IEEE, 2021, pp. 117–125. DOI: 10.1109/ICACI52617.2021.9435907.

[37] C. Guo, J. S. Frank and K. Q. Weinberger, 'Low frequency adversarial perturbation,' *CoRR*, vol. abs/1809.08758, 2018. arXiv: 1809.08758. [Online]. Available: http://arxiv.org/abs/1809.08758.

[38] H. Li, X. Xu, X. Zhang, S. Yang and B. Li, 'QEBA: query-efficient boundary-based blackbox attack,' *CoRR*, vol. abs/2005.14137, 2020. arXiv: 2005.14137. [Online]. Available: https://arxiv.org/abs/2005.14137.

[39] X. Li, X. Zhang, F. Yin and C. Liu, 'F-mixup: Attack cnns from fourier perspective,' in *2020 25th International Conference on Pattern Recognition (ICPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jan. 2021, pp. 541–548. DOI: 10.1109/ICPR48806.2021.9412611. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9412611.

[40] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, Y. Shuicheng and J. Feng, 'Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution,' eng, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, vol. 2019-, IEEE, 2019, pp. 3434–3443, ISBN: 9781728148038.

[41] D. J. Field, 'Relations between the statistics of natural images and the response properties of cortical cells,' *J. Opt. Soc. Am. A*, vol. 4, no. 12, pp. 2379–2394, Dec. 1987. DOI: 10.1364/JOSAA.4.002379. [Online]. Available: http://opg.optica.org/josaa/abstract.cfm?URI=josaa-4-12-2379.

[42] P. U. Stanford Vision Lab Stanford University. 'Imagenet.' ©2020 Stanford Vision Lab, Stanford University, Princeton University. (2015), [Online]. Available: `https://www.image-net.org` (visited on 08/05/2022).

[43] A. Geiger, P. Lenz and R. Urtasun, 'Are we ready for autonomous driving? the kitti vision benchmark suite,' in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361. DOI: `10.1109/CVPR.2012.6248074`.

[44] Google. 'Tensorflow documentation.' (2022), [Online]. Available: `https://www.tensorflow.org/api_docs/python/tf/keras/applications` (visited on 13/05/2022).

[45] G. K. Dziugaite, Z. Ghahramani and D. M. Roy, 'A study of the effect of JPG compression on adversarial images,' *CoRR*, vol. abs/1608.00853, 2016. arXiv: `1608.00853`. [Online]. Available: `http://arxiv.org/abs/1608.00853`.

[46] G. Wallace, 'The jpeg still picture compression standard,' *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992. DOI: `10.1109/30.125072`.

[47] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh and P. McDaniel, *Ensemble adversarial training: Attacks and defenses*, 2017. DOI: `10.48550/ARXIV.1705.07204`. [Online]. Available: `https://arxiv.org/abs/1705.07204`.

[48] L. Engstrom, A. Ilyas, H. Salman, S. Santurkar and D. Tsipras, *Robustness (python library)*, 2019. [Online]. Available: `https://github.com/MadryLab/robustness`.

[49] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, *Towards deep learning models resistant to adversarial attacks*, 2017. DOI: `10.48550/ARXIV.1706.06083`. [Online]. Available: `https://arxiv.org/abs/1706.06083`.

[50] K. Grosse, P. Manoharan, N. Papernot, M. Backes and P. McDaniel, *On the (statistical) detection of adversarial examples*, 2017. DOI: `10.48550/ARXIV.1702.06280`. [Online]. Available: `https://arxiv.org/abs/1702.06280`.

[51] W. Xu, D. Evans and Y. Qi, 'Feature squeezing: Detecting adversarial examples in deep neural networks,' in *Proceedings 2018 Network and Distributed System Security Symposium*, Internet Society, 2018. DOI: `10.14722/ndss.2018.23198`. [Online]. Available: `https://doi.org/10.14722%2Fndss.2018.23198`.

[52] D. Meng and H. Chen, *Magnet: A two-pronged defense against adversarial examples*, 2017. DOI: `10.48550/ARXIV.1705.09064`. [Online]. Available: `https://arxiv.org/abs/1705.09064`.

[53] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami, *Distillation as a defense to adversarial perturbations against deep neural networks*, 2015. DOI: `10.48550/ARXIV.1511.04508`. [Online]. Available: `https://arxiv.org/abs/1511.04508`.

[54] X. Cao and N. Z. Gong, *Mitigating evasion attacks to deep neural networks via region-based classification*, 2017. DOI: `10.48550/ARXIV.1709.05583`. [Online]. Available: `https://arxiv.org/abs/1709.05583`.

[55] J. Chen, M. I. Jordan and M. J. Wainwright. 'Hopskipjump github repository.' (2019), [Online]. Available: `https://github.com/Jianbo-Lab/HSJA/` (visited on 25/05/2022).

[56] J. Åsheim and K. A. B. Midtlid. 'MASSA github repository.' (2022), [Online]. Available: `https://github.com/johanaas/master` (visited on 25/05/2022).

Appendix A

Structured Literature Review

Understanding Black-Box Attacks Against Object Detectors From a User's Perspective

Kim André Midtlid, Johannes Åsheim, and Jingyue Li[0000--0002-7958-391X]

Norwegian University of Science and Technology, Trondheim, Norway
kamidtli@stud.ntnu.no, johannes.asheim@ntnu.no, jingyue.li@ntnu.no

Abstract. Due to recent developments in object detection systems, and the realistic threat of black-box adversarial attacks on object detector models, we argue the need for a contextual understanding of the attacks from the users' perspective. Existing literature reviews either do not provide complete and up-to-date summaries of such attacks or focus on the knowledge from the researchers' perspective. In this research, we conducted a systematic literature review to identify state-of-the-art black-box attacks and extract the information to help users evaluate and mitigate the risks. The literature review resulted in 29 black-box attack methods. We analyzed each attack from the following main aspects: attackers' knowledge needed to perform the attack, attack consequences, attack generalizability, and strategies to mitigate the attacks. Our results demonstrate an emerging increase in highly generalizable attacks, which now make up more than 50% of the landscape. We also reveal that more than 50% of recent attacks remain untested against mitigation strategies.

Keywords: artificial intelligence · object detection · image classification · adversarial attacks

1 Introduction

As Deep Neural Networks (DNNs) becomes more and more pertinent in image recognition and object detection tasks, their robustness also becomes more of a concern. Goodfellow et al. [14] have shown that the robustness of these models is susceptible to adversarial attacks. Such vulnerabilities have motivated researchers to develop adversarial attacks to exploit the object detection systems and contribute to improving their robustness. White-box attacks that assume knowledge about the target model continue to dominate the adversarial attack landscape, but there is an increase in black-box attacks. Black-box attacks assume no or very limited knowledge about the target model and are, therefore, more realistic approaches to adversarial attacks [34]. We argue that the increase in black-box attacks should be followed by a contextual understanding of the attacks from a user perspective. We define a user as a person who wants to know the risk and impact of adversarial attacks and how to defend against these attacks without knowing specific attack implementation details. Therefore, this paper omit the technical properties of the attacks for the traditional researcher perspective. Existing surveys and reviews of adversarial attacks on image classification and object detection, e.g., [6, 20], focus mostly on the information needed by researchers and do not cover sufficient up-to-date black-box attacks. Our research motivation is to summarize the state-of-the-art black-box attacks targeting object detection models to help users evaluate and mitigate the risks. We focus on answering the following research questions.

- **RQ1:** What does the attacker need to know about the target model?
- **RQ2:** How generalizable is the attack?
- **RQ3:** What are the consequences of the attack?
- **RQ4:** Which mitigation strategies have been tested against the attack?

We performed a systematic literature review on articles published between 2017 and 2021 to collect state-of-the-art black-box attacks. Through the systematic literature review and snowballing, we uncovered 29 state-of-the-art attack methods, which we analyze and present in this paper. Our study benefits industrial practitioners and scientists. The contributions of the study are twofold.

- We provide comprehensive and up-to-date consolidated knowledge about black-box attacks targeting object detection models to help users to evaluate the risks and choose effective mitigation solutions.
- We identify the trends and weaknesses of existing studies in this field, which may inspire researchers' future work.

The rest of the paper is organized as follows: Section 2 introduces the background. Section 3 presents the related work. Section 4 explains our research methods, and Section 5 presents the results. We then discuss our results in Section 6. Conclusions and future work are in Section 7.

2 Background

Object detection is the field of Artificial Intelligence (AI) that uses deep learning to extract high-dimensional information from images and videos. An autonomous car with camera sensors uses image processing to navigate the road and detect obstacles.

2.1 Object Detection and Image Classification

Image classification is the task of classifying an input image by assigning it to a specific label [42], while object detection is the task of localizing and classifying distinct objects in an image or video. Current object detectors can be split into two main categories: two-stage and one-stage detectors. Two-stage detectors consist of two main parts. First, the detector uses a Region Proposal Network (RPN) to calculate proposed regions for objects. The RPN uses a set of predefined *anchor boxes* uniformly placed over the image to calculate proposed regions before outputting a predefined number of proposed bounding boxes with a corresponding objectiveness score. The objectiveness score indicates whether the proposed region belongs to an object class or the background. These proposed regions significantly reduce the computational complexity needed to localize and classify an object. In the second stage, the proposed regions from the RPN are passed to a high-quality image classifier to recognize objects. One-stage detectors aim to improve the inference speed while still achieving acceptable accuracy. One-stage detectors achieve this goal by removing the region proposal stage required by the two-stage detectors. Instead, they run detection on a dense sampling of predefined default boxes. The ability to skip the region proposal step significantly decreases inference time and has led to the development of many one-stage detectors, e.g., [30, 38].

2.2 Threat Models

The threat model of an attack is based on what the adversary knows about the target model, thus we can categorize the attacks into three threat models. *White-box attacks*, e.g., FGSM [14], assume the adversary has complete knowledge of the target model , which include the model's internal structure, such as weights and parameters of the target model, and knowledge of the output given an input. In some cases, the adversary knows the training data distribution. This allows the adversary to construct attack methods specific to the given model. *Black-box attacks*, assume no internal information of the target model, but the ability to observe the output for a given input. Usually, black-box attack methods are constructed based on querying the target model [5, 8, 9]. Han Xu et al. [46] introduce *grey-box attacks* as a hybrid of white-box attacks and black-box attacks, where the attacker trains a generative model to create adversarial examples in white-box setting. Then the target model is attacked in the black-box setting with adversarial examples from the trained generative model.

3 Related Work

Bhambri et al. [6] performed a survey focusing on adversarial black-box attacks. The paper aims to conduct a comparative study of both adversarial attacks and defenses. Nineteen black-box attacks were compared on the number of queries, success rate, and perturbation norm. The survey categorizes the attacks based on gradient estimation, transferability, local search and combinatorics. Shilin Qiu et al. [37] presents a comprehensive study of the research of adversarial attack and defenses. The paper details white-box and black-box attack methods but mainly focuses on defense strategies. Kong et al. [25] reviewed adversarial attack literature in the different application fields of AI security. The fields include images, texts and malicious code. The paper presents attack algorithms for the different application domains and includes 13 attacks for the image domain, five of which are black-box attacks. The survey further elaborates on defense methods and how they affect the presented attacks. In order to help new researchers in the field, the paper introduces and discusses the different datasets and tools available. There are other surveys and articles, i.e., [1, 27, 46, 48], which discuss adversarial attacks and defenses. The common limitation of these studies are the low number of included black-box attacks. In addition, the studies focus on consolidating information from the researchers' perspective.

4 Research Design and Implementation

We performed a Systematic Literature Review (SLR) and followed the SLR guidelines proposed by Kitchenham and Charters [24]. After analyzing the terms related to our research questions and their synonyms, we chose to use the search query: *Adversarial* AND *Attack* AND (*"Object detection"* OR *"Object detector"*).

We chose `oria.no`, a search engine that covers many scientific databases, including IEEE Xplore, Springer, ACM Digital library, and Scopus. To include only recent literature and to reduce the scope, we used the advanced search functionality in `oria.no`, and included only peer-reviewed and published scientific papers from the last 5 years back from 2021. The identified articles were filtered mainly based on their relevance to the research questions by reading their abstract, introduction, and, in some cases, methodology. After filtering, we identified 11 relevant primary studies. Then, we performed a snowballing search following the process proposed by [45], with the exception that forward and backward snowballing searches were limited to a single iteration each. The forward snowballing was performed using Google Scholar. The snowballing identified 16 more papers, resulting in 27 primary studies.

5 Research Results

In this section, we present our answers to each research question. Attack names preceded by asterisks (*) were not presented with a name in their corresponding paper. Therefore, a descriptive name is given based on the attack method.

5.1 RQ1—Attacker's Knowledge

How much information the attacker requires from the output labels varies across the identified papers but can be split into three categories: **Soft-labels** refer to the threat model where an attacker accesses the output probabilities $P(y|x)$ for y in the top k classes. Soft-labels also might include the label for each of the output probabilities. For object detectors, information about the bounding boxes indicates soft-labels. **Hard-labels** refer to a more restricted threat model where an attacker only has access to a list of $k \in \mathbb{Z}^+$ output labels. Different attacks make different assumptions about k. For $k = 1$, the attacker only has access to the single predicted class. In the case of $k > 1$, the list of classes is often ordered by decreasing probabilities but does not include the probabilities. For object detectors, the hard-label category signifies no information about the bounding boxes. Some attacks assume the target model outputs $k = 1$ or $k > 1$

Table 1: Attacks grouped by attacker knowledge

Attack Name	Year	Knowledge
NRDM [33]	2018	No-labels
DaST [51]	2020	Hard-labels and Soft-labels
HopSkipJumpAttack [9]	2020	Hard-labels
*Partial-retraining [36]	2020	Hard-labels
*Evolutionary Attack [13]	2019	Hard-labels
Label-Only Attack [20]	2018	Hard-labels
Opt-Attack [11]	2018	Hard-labels
Boundary Attack [8]	2017	Hard-labels
CMA-ES [19]	2021	Soft-labels
Simple Transparent Adversarial Examples [7]	2021	Soft-labels
*Discrete Cosine Transform Attack [26]	2021	Soft-labels
*Differential Evolution Attack [44]	2021	Soft-labels
BMI-FGSM [29]	2020	Soft-labels
*Transferable Universal Perturbation Attack [49]	2020	Soft-labels
Adv-watermark [23]	2020	Soft-labels
Evaporate Attack [43]	2020	Soft-labels
Daedalus [41]	2019	Soft-labels
One-Pixel-Attack [39]	2019	Soft-labels
Single Scratch attack [22]	2019	Soft-labels
GenAttack [2]	2019	Soft-labels
Universal perturbation attack [50]	2019	Soft-labels
Query-Limited Attack [20]	2018	Soft-labels
Partial-Info Attack [20]	2018	Soft-labels
Bandits [21]	2018	Soft-labels
Gradient Estimation Attacks [5]	2018	Soft-labels
R-AP [28]	2018	Soft-labels
ZOO [10]	2017	Soft-labels
LocSearchAdv [32]	2016	Soft-labels
*Substitute Attack [34]	2016	Soft-labels

labels. **No-labels** refer to the most restricted threat model, where an attacker requires no access to the output of the target model.

Table 1 presents the attacks grouped by the required attacker knowledge. We notice that more than 75% of the discussed attacks use the soft-labels approach. Table 1 also illustrates that about 25% of the discussed attacks use hard-labels as part of their method. We can also see that the number of hard-label attacks has tripled from 2017 to 2020, which might indicate that hard-label attacks are becoming more popular. The new trend might suggest that hard-label attacks have room for improvement in the coming years and should be investigated further. It is also worth noting **DaST** [51], which can be used in both a soft- and hard-label scenario because the attack is customizable. This might be an indication of a new type of attack that can be modified based on the target model. **NRDM** [33] requires no labels at all. These two attacks illustrate a possibility in the landscape, as attacks can become more applicable to any target model and more independent of the attacker's knowledge.

5.2 RQ2—Attack Generalizability

The generalization of adversarial black-box attacks examines the number of different types of object detection models which are claimed to have been successfully attacked. We have defined four categories of generalization and present the results in Table 2. The categories are *None*, *Low*, *High* and *Very High*. The presented attack is tested on and successful against either one, two, three to five or six or more target models respectively. The term generalizability is only determined based on the number of attacked target models, and do not include datasets, model accuracy, attack hyperparameters and model hyperparameters. It is important to note that the generalizability is derived from the number of models claimed by the authors of the primary studies. Therefore an attack with *None* may be generalizable, but the authors only includes experiments against one target model.

Most of the attacks only target image classifiers, but the focus could be on one-stage models, two-stage models, or a combination of both for object detectors. An attack targeting both types of object detectors poses a significant threat, as it generalizes to most model architectures. This aspect is captured in the *target architecture* column in Table 2. Attacks targeting object detectors are labeled with one-stage, two-stage, or both, while attacks targeting image classifiers are labeled correspondingly.

From Table 2, we observe a balanced distribution between high and low generalizability. Both attack types show promising results, but the ones with high generalizability might be more interesting to be studied further, as they are successful across a broader range of object detectors. The number of highly generalizable attacks has increased from 2019, as shown in Figure 1. From Table 2, we also notice that **R-AP** [28] and **NRDM** [33] stand out. They are both classified as very high, meaning they have been tested and exhibited promising performance on six or more different models. Additionally, **NRDM** has been tested against both image classifiers and object detectors, demonstrating notable generalizability . It is also worth noting that [28] and [9] mention the possibility of combining **R-AP** and **HopSkipJumpAttack**, respectively, with other adversarial attacks as areas for future work. This combination demonstrates a potential to improve attacks through amalgamation, which is worth considering in future research. Many of the discussed attacks have also been tested on real-world APIs, which are listed in Table 3. From a user perspective, this illustrates a potential area of focus and risks to consider in the future.

Table 2: Attacks grouped by their level of generalizability

Attack Name	**Year**	**Generalization**	**Target Architecture**
NRDM [33]	2018	Very High	Image classifiers
R-AP [28]	2018	Very High	Two-stage
CMA-ES [19]	2021	High	One-stage and two-stage
*Differential Evolution Attack [44]	2021	High	Image classifiers
Adv-watermark [23]	2020	High	Image classifiers
Evaporate Attack [43]	2020	High	One-stage and two-stage
HopSkipJumpAttack [9]	2020	High	Image classifiers
*Partial-retraining [36]	2020	High	Image classifiers
*Transferable Universal Perturbation Attack [49]	2020	High	One-stage and two-stage
Daedalus [41]	2019	High	One-stage
One-Pixel-Attack [39]	2019	High	Image classifiers
Universal perturbation attack [50]	2019	High	Image classifiers
Single Scratch attack [22]	2019	High	Image classifiers
Bandits [21]	2018	High	Image classifiers
Gradient Estimation Attacks [5]	2018	High	Image classifiers
Boundary Attack [8]	2017	High	Image classifiers
*Substitute Attack [34]	2016	High	Image classifiers
*Discrete Cosine Transform Attack [26]	2021	Low	Image classifiers
BMI-FGSM [29]	2020	Low	Image classifiers
DaST [51]	2020	Low	Image classifiers
*Evolutionary Attack [13]	2019	Low	Image classifiers
GenAttack [2]	2019	Low	Image classifiers
Opt-Attack [11]	2018	Low	Image classifiers
Query-Limited Attack [20]	2018	Low	Image classifiers
Partial-Info Attack [20]	2018	Low	Image classifiers
Label-Only Attack [20]	2018	Low	Image classifiers
LocSearchAdv [32]	2016	Low	Image classifiers
Simple Transparent Adversarial Examples [7]	2021	None	Image classifiers
ZOO [10]	2017	None	Image classifiers

Table 3: Attacks against real-world APIs

Attack Name	**Year**	**Real-World API**
*Discrete Cosine Transform Attack [26]	2021	AWS Rekognition [4]
*Partial retraining [36]	2020	Google AutoML Vision [15]
Partial-Info Attack [20]	2018	Google Cloud Vision [16]
Gradient Estimation Attacks [5]	2018	Clarifai [12]
Boundary Attack [8]	2017	Clarifai [12]
*Substitute Attack [34]	2016	Amazon and Google Oracles [3, 16]

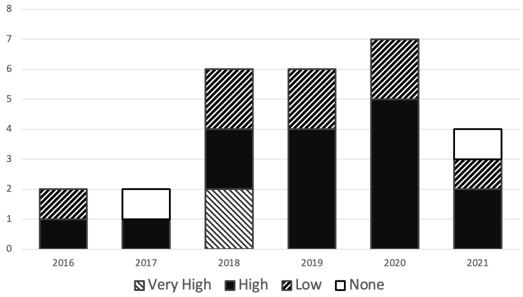


Fig. 1: The ratio of generalization levels for each year

5.3 RQ3—Attack Consequences

Classification attack is divided into targeted and untargeted attacks. Targeted attacks aim to misclassify a adversarial input image i' of class c', where the the target model would have classified input image i in to class c. In other words, the attacker wants to force the target model to predict a chosen class. Untargeted attacks aim to misclassify an adversarial input image i' in to any class c', where $c' \neq c$. Object detection attack can lead to object vanishing and object population. An object vanishing attack aims to suppress all object detection in a input image, while an object population attack aims to fabricate false objects in a predicted image.

Table 4 shows the consequences of each attack. Untargeted attacks are the most common, making up more than 75% of the discussed attacks. Even though these attacks make up the majority and pose a significant threat, targeted attacks might be more dangerous from a defender's perspective. Targeted attacks still make up about 65% of discussed attacks, and it is worth noting that most image classification attacks provide both targeted and untargeted versions. This trend suggests that attacks are not limited to a single purpose but can achieve multiple goals. In the realm of object detection attacks, we have looked at five attacks. Four of them exploit the object vanishing vulnerability, while only one focuses on object population. **CMA-ES** [19] stands out because it combines object detection and image classification attacks. **CMA-ES** is a very recently developed attack that could hint at a change of focus in the landscape. Additionally, **Daedalus** [41] is the only attack that can execute object population. Results in Table 4 also shows the emerging focus on attacks against object detectors from 2018.

5.4 RQ4—Mitigation Strategies

Table 5 contains a summary of all the mitigation strategies an attack is claimed to have been tested against. The *Vulnerable Mitigations* column lists all tested mitigation strategies where the attack is still able to reduce the overall accuracy of the system significantly. The definition of a significant drop in accuracy is claimed by each paper. The *Robust Mitigations* column lists all mitigation

Table 4: Attacks grouped by their consequences

Attack Name	**Year**	**Target Architecture**	**Consequenses**
CMA-ES [19]	2021	One-stage and two-stage	Vanishing, Targeted, and Untargeted
Evaporate Attack [43]	2020	One-stage and two-stage	Vanishing
*Transferable Universal Perturbation Attack [49]	2020	One-stage and two-stage	Vanishing
R-AP [28]	2018	Two-stage	Vanishing
Daedalus [41]	2019	One-stage	Population
*Differential Evolution Attack [44]	2021	Image classifiers	Targeted and Untargeted
BMI-FGSM [29]	2020	Image classifiers	Targeted and Untargeted
DaST [51]	2020	Image classifiers	Targeted and Untargeted
HopSkipJumpAttack [9]	2020	Image classifiers	Targeted and Untargeted
One-Pixel-Attack [39]	2019	Image classifiers	Targeted and Untargeted
Single Scratch attack [22]	2019	Image classifiers	Targeted and Untargeted
Gradient Estimation Attacks [5]	2018	Image classifiers	Targeted and Untargeted
Query-Limited Attack [20]	2018	Image classifiers	Targeted and Untargeted
Partial-Info Attack [20]	2018	Image classifiers	Targeted and Untargeted
Label-Only Attack [20]	2018	Image classifiers	Targeted and Untargeted
Bandits [21]	2018	Image classifiers	Targeted and Untargeted
Opt-Attack [11]	2018	Image classifiers	Targeted and Untargeted
Boundary Attack [8]	2017	Image classifiers	Targeted and Untargeted
ZOO [10]	2017	Image classifiers	Targeted and Untargeted
LocSearchAdv [32]	2016	Image classifiers	Targeted and Untargeted
*Discrete Cosine Transform Attack [26]	2021	Image classifiers	Targeted
*Partial-retraining [36]	2020	Image classifiers	Targeted
GenAttack [2]	2019	Image classifiers	Targeted
Simple Transparent Adversarial Examples [7]	2021	Image classifiers	Untargeted
Adv-watermark [23]	2020	Image classifiers	Untargeted
*Evolutionary Attack [13]	2019	Image classifiers	Untargeted
Universal perturbation attack [50]	2019	Image classifiers	Untargeted
NRDM [33]	2018	Image classifiers	Untargeted
*Substitute Attack [34]	2016	Image classifiers	Untargeted

strategies where the attack cannot reduce the overall accuracy of the system significantly. It is worth noting that *None tested* in the *Robust Mitigations* column only means that the attack has not been tested on any mitigation strategy. It does not mean that the attack is able to bypass all defense strategies. This also applies to the *Vulnerable Mitigations* column. A cell with "-" means that none of the tested mitigation strategies applies to that column. A list of defenses in the *Vulnerable Mitigations* column and "-" in the *Robust Mitigations* column means that none of the tested defenses successfully defended against the attack.

From Table 5, we notice that more than half of the discussed attacks have not been tested against any mitigation strategies. This illustrates that mitigation strategies have not been given enough attention. We also notice that Adversarial Training and Input Transformations repeat across different attacks in the *Vulnerable Mitigations* column. The repetition indicates that no single mitigation strategy works for all attacks, and that most modern mitigation strategies struggle to defend against the discussed attacks. It is worth noting that many of the mitigation strategies listed are umbrella terms, covering multiple defense implementations. For example, input transformations [18] cover multiple defense mechanisms such as JPEG-compression, clipping and median filtering. Although Figure 2 shows an increase in the number of mitigation strategies evaluated, we can also see a large emerging ratio of untested attacks from 2018.

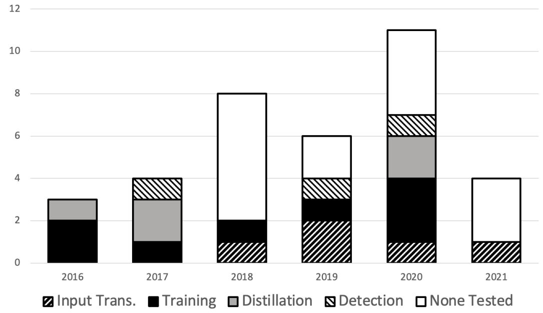


Fig. 2: The ratio of mitigation strategies each year

6 Discussion

The aim of our work is to summarize the state-of-the-art black-box attacks targeting object detectors to help users evaluate and mitigate the risks. No related work outlined in Section 3 takes the user's perspective but rather explains black-box attacks from a researcher's perspective and focuses on explaining the attack methods. For example, Kong et al. [25] and Bhambri et al. [6] provide categories of black-box attacks, but the categorization is based on the attack method. Understanding a black-box attack method requires a high level of competence in a user. Our study does not focus on the attack methods because they are not the most relevant information for a user. The main focuses from a user perspective are covered in our research questions. Results of RQ1 (Knowledge) can inform a user of the attacks which can and cannot be executed on a system. Results of RQ2 (Generalization) warns the user of which attacks have a large impact area and could affect the system. Results of RQ3 (Consequences) give the user insight into the attacks' results. Results of RQ4 (Mitigation strategies) are highly important to the user because they contain information that can help the user implement relevant defenses to the system.

Table 5: Attacks grouped by mitigation strategies they have been tested against

Attack	**Year**	**Vulnerable Mitigations**	**Robust Mitigations**
*Differential Evolution Attack [44]	2021	Feature squeezing [47] Input Transformations [18]	-
Adv-watermark [23]	2020	Adversarial Training [40] Input Transformations [18]	-
HopSkipJumpAttack [9]	2020	Adversarial Distillation [35], Region-based classification	Adversarial Training [40]
*Partial-retraining [36]	2020	Adversarial Detection [17] Adversarial Distillation [35] Adversarial Training [40] Feature squeezing [47]	-
GenAttack [2]	2019	Adversarial Training [40], Input Transformations [18]	-
One-Pixel-Attack [39]	2019	-	Adversarial Detection [17]
Daedalus [41]	2019	MagNet [31] Minimize bounding box size	-
Single Scratch attack [22]	2019	Input Transformations (JPEG-compression) [18] Input Transformations (Clipping) [18]	Input Transformations (Median Filtering) [18]
Gradient Estimation Attacks [5]	2018	Adversarial Training [40]	Rounded output probabilities
NRDM [33]	2018	Input Transformations [18]	-
Boundary Attack [8]	2017	Adversarial Distillation [35]	-
ZOO [10]	2017	Adversarial Detection [17] Adversarial Distillation [35]	Adversarial Training [40]
LocSearchAdv [32]	2016	Adversarial Training [40]	Query-access prevention
*Substitute Attack [34]	2016	Adversarial Distillation [35] Adversarial Training [40]	-
CMA-ES [19]	2021	None tested	None tested
*Discrete Cosine Transform Attack [26]	2021	None tested	None tested
Simple Transparent Adversarial Examples [7]	2021	None tested	None tested
DaST [51]	2020	None tested	None tested
Evaporate Attack [43]	2020	None tested	None tested
BMI-FGSM [29]	2020	None tested	None tested
*Transferable Universal Perturbation Attack [49]	2020	None tested	None tested
*Evolutionary Attack [13]	2019	None tested	None tested
Universal perturbation attack [50]	2019	None tested	None tested
Bandits [21]	2018	None tested	None tested
Label-Only Attack [20]	2018	None tested	None tested
Opt-Attack [11]	2018	None tested	None tested
R-AP [28]	2018	None tested	None tested
Query-Limited Attack [20]	2018	None tested	None tested
Partial-Info Attack [20]	2018	None tested	None tested

The results of the survey show that many modern adversarial attack studies have not focused on testing mitigation strategies, as shown in Table 5. Eighty percent of the discussed attacks against object detectors have not been tested against any mitigation strategies. Our study shows that the generalizability of recent attacks is increasing, which poses a more significant threat to the industry. No longer do the attacks focus on a single objective or target model, but rather, they combine all these goals into broader attacks. This means that modern attacks can bypass more defenses and achieve multiple attack objectives.

7 Conclusion and Future Work

We conducted a systematic literature review in order to summarize state-of-the-art black-box attacks targeting object detection models to help users evaluate and mitigate the risks. The literature review resulted in 29 unique black-box attack methods from 27 papers. Our analyses summarized the status and trends regarding attackers' knowledge needed to perform the attack, consequences, generalizability, and current mitigation strategies for each attack. We acknowledge that the SLR may have left out some papers due to missing search queries and limited database coverage. One finding from our study is that mitigation strategies should be comprehensively tested on the identified black-box attacks to find out which defenses are robust and which could be improved. We plan to focus on evaluating and improving different mitigation strategies as our future work.

References

1. Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access **6**, 14410–14430 (2018). https://doi.org/10.1109/ACCESS.2018.2807385
2. Alzantot, M., Sharma, Y., Chakraborty, S., Zhang, H., Hsieh, C.J., Srivastava, M.: Genattack: Practical black-box attacks with gradient-free optimization (2018). https://doi.org/10.48550/ARXIV.1805.11090, https://arxiv.org/abs/1805.11090
3. Amazon: Aws machine learning (2021), https://aws.amazon.com/machine-learning
4. Amazon: Aws rekognition (2021), https://aws.amazon.com/rekognition/
5. Bhagoji, A.N., He, W., Li, B., Song, D.: Practical black-box attacks on deep neural networks using efficient query mechanisms. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 158–174. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-030-01258-8_10
6. Bhambri, S., Muku, S., Tulasi, A., Buduru, A.B.: A survey of black-box adversarial attacks on computer vision models (2019). https://doi.org/10.48550/ARXIV.1912.01667, https://arxiv.org/abs/1912.01667
7. Borkar, J., Chen, P.Y.: Simple transparent adversarial examples (2021). https://doi.org/10.48550/ARXIV.2105.09685, https://arxiv.org/abs/2105.09685

8. Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: Reliable attacks against black-box machine learning models (2017). https://doi.org/10.48550/ARXIV.1712.04248, https://arxiv.org/abs/1712.04248
9. Chen, J., Jordan, M.I., Wainwright, M.J.: Hopskipjumpattack: A query-efficient decision-based attack. In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 1277–1294 (2020). https://doi.org/10.1109/SP40000.2020.00045
10. Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.J.: ZOO. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM (nov 2017). https://doi.org/10.1145/3128572.3140448, https://doi.org/10.1145%2F3128572.3140448
11. Cheng, M., Le, T., Chen, P.Y., Yi, J., Zhang, H., Hsieh, C.J.: Query-efficient hard-label black-box attack:an optimization-based approach (2018). https://doi.org/10.48550/ARXIV.1807.04457, https://arxiv.org/abs/1807.04457
12. Clarifai: The world's ai (2021), https://www.clarifai.com/
13. Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., Zhu, J.: Efficient decision-based black-box adversarial attacks on face recognition (2019). https://doi.org/10.48550/ARXIV.1904.04433, https://arxiv.org/abs/1904.04433
14. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2014). https://doi.org/10.48550/ARXIV.1412.6572, https://arxiv.org/abs/1412.6572
15. Google: Automl (2021), https://cloud.google.com/automl
16. Google: Vision ai (2021), https://cloud.google.com/vision
17. Grosse, K., Manoharan, P., Papernot, N., Backes, M., McDaniel, P.: On the (statistical) detection of adversarial examples (2017). https://doi.org/10.48550/ARXIV.1702.06280, https://arxiv.org/abs/1702.06280
18. Guo, C., Rana, M., Cisse, M., van der Maaten, L.: Countering adversarial images using input transformations (2017). https://doi.org/10.48550/ARXIV.1711.00117, https://arxiv.org/abs/1711.00117
19. Haoran, L., Yu'an, T., Yuan, X., Yajie, W., Jingfeng, X.: A cma-es-based adversarial attack against black-box object detectors. Chinese Journal of Electronics **30**(3), 406–412 (2021). https://doi.org/https://doi.org/10.1049/cje.2021.03.003, https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cje.2021.03.003
20. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information (2018). https://doi.org/10.48550/ARXIV.1804.08598, https://arxiv.org/abs/1804.08598
21. Ilyas, A., Engstrom, L., Madry, A.: Prior convictions: Black-box adversarial attacks with bandits and priors (2018). https://doi.org/10.48550/ARXIV.1807.07978, https://arxiv.org/abs/1807.07978
22. Jere, M., Rossi, L., Hitaj, B., Ciocarlie, G., Boracchi, G., Koushanfar, F.: Scratch that! an evolution-based adversarial attack against neural networks (2019). https://doi.org/10.48550/ARXIV.1912.02316, https://arxiv.org/abs/1912.02316
23. Jia, X., Wei, X., Cao, X., Han, X.: Adv-watermark: A novel watermark perturbation for adversarial examples (2020). https://doi.org/10.48550/ARXIV.2008.01919, https://arxiv.org/abs/2008.01919

24. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering **2** (2007)
25. Kong, Z., Xue, J., Wang, Y., Huang, L., Niu, Z., Li, F., Meng, W.: A survey on adversarial attack in the age of artificial intelligence. Wirel. Commun. Mob. Comput. **2021** (jan 2021). https://doi.org/10.1155/2021/4907754, `https://doi.org/10.1155/2021/4907754`
26. Kuang, X., Gao, X., Wang, L., Zhao, G., Ke, L., Zhang, Q.: A discrete cosine transform-based query efficient attack on black-box object detectors. Information Sciences **546**, 596–607 (2021). https://doi.org/https://doi.org/10.1016/j.ins.2020.05.089, `https://www.sciencedirect.com/science/article/pii/S0020025520305077`
27. Li, G., Zhu, P., Li, J., Yang, Z., Cao, N., Chen, Z.: Security matters: A survey on adversarial machine learning (2018). https://doi.org/10.48550/ARXIV.1810.07339, `https://arxiv.org/abs/1810.07339`
28. Li, Y., Tian, D., Chang, M.C., Bian, X., Lyu, S.: Robust adversarial perturbation on deep proposal-based models (2018). https://doi.org/10.48550/ARXIV.1809.05962, `https://arxiv.org/abs/1809.05962`
29. Lin, J., Xu, L., Liu, Y., Zhang, X.: Black-box adversarial sample generation based on differential evolution (2020). https://doi.org/10.48550/ARXIV.2007.15310, `https://arxiv.org/abs/2007.15310`
30. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. Lecture Notes in Computer Science p. 21–37 (2016). https://doi.org/10.1007/978-3-319-46448-0$_2$, `http://dx.doi.org/10.1007/978-3-319-46448-0_2`
31. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples (2017). https://doi.org/10.48550/ARXIV.1705.09064, `https://arxiv.org/abs/1705.09064`
32. Narodytska, N., Kasiviswanathan, S.: Simple black-box adversarial attacks on deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1310–1318 (2017). https://doi.org/10.1109/CVPRW.2017.172
33. Naseer, M., Khan, S.H., Rahman, S., Porikli, F.: Task-generalizable adversarial attack based on perceptual metric (2018). https://doi.org/10.48550/ARXIV.1811.09020, `https://arxiv.org/abs/1811.09020`
34. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning (2016). https://doi.org/10.48550/ARXIV.1602.02697, `https://arxiv.org/abs/1602.02697`
35. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks (2015). https://doi.org/10.48550/ARXIV.1511.04508, `https://arxiv.org/abs/1511.04508`
36. Park, H., Ryu, G., Choi, D.: Partial retraining substitute model for query-limited black-box attacks. Applied sciences **10**(20), 1–19 (2020). https://doi.org/10.3390/app10207168
37. Qiu, S., Liu, Q., Zhou, S., Wu, C.: Review of artificial intelligence adversarial attack and defense technologies. Applied sciences **9**(5), 909 (2019). https://doi.org/10.3390/app9050909

38. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement (2018). https://doi.org/10.48550/ARXIV.1804.02767, https://arxiv.org/abs/1804.02767
39. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE transactions on evolutionary computation **23**(5), 828–841 (2019). https://doi.org/10.1109/TEVC.2019.2890858
40. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses (2017). https://doi.org/10.48550/ARXIV.1705.07204, https://arxiv.org/abs/1705.07204
41. Wang, D., Li, C., Wen, S., Han, Q.L., Nepal, S., Zhang, X., Xiang, Y.: Daedalus: Breaking nonmaximum suppression in object detection via adversarial examples. IEEE Transactions on Cybernetics pp. 1–14 (2021). https://doi.org/10.1109/TCYB.2020.3041481
42. Wang, S., Su, Z.: Metamorphic testing for object detection systems (2019). https://doi.org/10.48550/ARXIV.1912.12162, https://arxiv.org/abs/1912.12162
43. Wang, Y., Tan, Y.a., Zhang, W., Zhao, Y., Kuang, X.: An adversarial attack on dnn-based black-box object detectors. Journal of network and computer applications **161**, 102634 (2020). https://doi.org/10.1016/j.jnca.2020.102634
44. Wei, X., Guo, Y., Li, B.: Black-box adversarial attacks by manipulating image attributes. Information sciences **550**, 285–296 (2021). https://doi.org/10.1016/j.ins.2020.10.028
45. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14. ACM Press (2014). https://doi.org/10.1145/2601248.2601268
46. Xu, H., Ma, Y., Liu, H.C., Deb, D., Liu, H., Tang, J.L., Jain, A.K.: Adversarial attacks and defenses in images, graphs and text: A review. International journal of automation and computing **17**(2), 151–178 (2020). https://doi.org/10.1007/s11633-019-1211-x
47. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. Proceedings 2018 Network and Distributed System Security Symposium (2018). https://doi.org/10.14722/ndss.2018.23198, http://dx.doi.org/10.14722/ndss.2018.23198
48. Zhang, J., Li, C.: Adversarial examples: Opportunities and challenges. IEEE Transactions on Neural Networks and Learning Systems **31**(7), 2578–2593 (2020). https://doi.org/10.1109/TNNLS.2019.2933524
49. Zhang, Q., Zhao, Y., Wang, Y., Baker, T., Zhang, J., Hu, J.: Towards cross-task universal perturbation against black-box object detectors in autonomous driving. Computer Networks **180**, 107388 (2020). https://doi.org/10.1016/j.comnet.2020.107388, https://www.sciencedirect.com/science/article/pii/S138912862030606X
50. Zhao, Y., Wang, K., Xue, Y., Zhang, Q., Zhang, X.: An universal perturbation generator for black-box attacks against object detectors. In: Qiu, M. (ed.) Smart Computing and Communication. pp. 63–72. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-34139-8$_7$
51. Zhou, M., Wu, J., Liu, Y., Liu, S., Zhu, C.: Dast: Data-free substitute training for adversarial attacks (2020). https://doi.org/10.48550/ARXIV.2003.12703, https://arxiv.org/abs/2003.12703

Appendix B

Collection of all l_2 Distance Result Tables

Defence	Models	l_2 distance	Model Queries							
			250		500		750		1000	
			HSJA	MASSA	HSJA	MASSA	HSJA	MASSA	HSJA	MASSA
None	ResNet50	Median	39.33	**10.19**	29.85	**9.06**	18.23	**8.88**	15.25	**8.88**
		Average	40.24	**12.66**	32.29	**11.07**	22.04	**10.66**	18.54	**10.58**
	VGG16	Median	27.09	**7.00**	18.37	**6.12**	11.60	**6.12**	9.77	**6.12**
		Average	31.87	**9.66**	23.91	**8.56**	15.91	**8.35**	13.71	**8.32**
	VGG19	Median	25.69	**7.00**	18.37	**6.05**	11.26	**6.05**	9.10	**6.05**
		Average	29.61	**9.10**	21.85	**8.08**	14.40	**7.86**	12.25	**7.84**
JPEG compression	ResNet50	Median	33.56	**12.16**	25.34	**10.80**	15.78	**10.77**	13.32	**10.77**
		Average	35.90	**14.08**	28.88	**12.69**	19.87	**12.37**	17.21	**12.31**
	VGG16	Median	23.42	**9.67**	16.99	**9.04**	11.68	**8.96**	10.58	**8.96**
		Average	27.92	**11.80**	21.37	**10.89**	15.12	**10.71**	13.51	**10.70**
	VGG19	Median	24.20	**9.51**	16.97	**9.08**	12.02	**9.08**	10.81	**9.08**
		Average	27.78	**11.90**	21.12	**10.94**	14.73	**10.76**	13.08	**10.75**
Adversarial Training	ResNet50	Median	47.55	**18.70**	45.70	**17.38**	42.65	**16.91**	41.63	**16.83**
		Average	48.22	**20.60**	46.83	**19.21**	44.73	**18.69**	43.80	**18.53**

Table B.1: Median and average distance at various model queries for each target model. The smaller distance at a given model query is bold-faced.

NTNU

Norwegian University of
Science and Technology