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

Kim A. B. Midtlid*
Norwegian University of Science and Technology
Trondheim, Norway
kamidtli@stud.ntnu.no

Johannes Åsheim*
Norwegian University of Science and Technology
Trondheim, Norway
johannes.asheim@ntnu.no

Jingyue Li
Norwegian University of Science and Technology
Trondheim, Norway
jingyue.li@ntnu.no

ABSTRACT

Recent development has revealed that deep neural networks used in image classification systems are vulnerable to adversarial attacks. Thus, it is critical to understand the possible adversarial attacks to develop effective defense mechanisms. In this study, we designed an untargeted query-efficient decision-based black-box attack against 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. The evaluation results show that MASSA requires significantly fewer queries than the state-of-the-art decision-based black-box attack, i.e., HSJA. In addition, MASSA can create adversarial examples with 74, 16% lower l_2 distance than HSJA after only 250 queries. We also demonstrate that two existing defense mechanisms, namely, JPEG compression and adversarial training, are not effective in defending against MASSA. Results of the study give new insights into the potential risks of using deep neural networks in critical systems and encourage the community to study improved defense approaches to mitigate the risks.

CCS CONCEPTS

• **Security and privacy → Software and application security**.

KEYWORDS

artificial intelligence, image classification, adversarial attacks

ACM Reference Format:
Kim A. B. Midtlid, Johannes Åsheim, and Jingyue Li. 2022. Magnitude Adversarial Spectrum Search-based Black-box Attack against Image Classification. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security (AISec '22), November 11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3560830.3563723

1 INTRODUCTION

The remarkable results of deep neural network (DNN) have led to their use in various safety-critical tasks such as autonomous driving [27, 12, 8, 11, 19] and facial biometric systems, including surveillance and access control [34, 26]. These safety-critical systems require certain robustness from the DNNs, where failure can lead to severe consequences.

Goodfellow, Shlens, and Szegedy [17] have demonstrated that DNNs are vulnerable to adversarial attacks. 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.

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 [28]. The most realistic black-box setting is when the adversary only has access to the output labels alone, known as decision-based attacks, e.g., [4]. 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 [1], 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 [38]. The main challenge of the decision-based attack field is to lower the query budget for adversarial attacks.

Thus, we want to study if it is possible to construct a query-efficient attack method that generates imperceptible adversarial examples in just hundreds of queries. Additionally, we aim to investigate whether existing defense mechanisms are robust against the potential new attacks. Our research question is: *Is it possible to create an untargeted query-efficient decision-based black-box attack against robust image classification models to produce less perceptible adversarial examples?*

We designed Magnitude Adversarial Spectrum Search-based Attack (MASSA), a novel decision-based black-box adversarial attack method. 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 demonstrate empirically that MASSA achieves superior query-efficiency and imperceptibility over the state-of-the-art decision-based attack. We also evaluate defense mechanisms against our attack method and show the weaknesses of the evaluated defense mechanisms. To the best of our knowledge, we are the first to design

*Both authors contributed equally to this research.

AISec '22, November 11, 2022, Los Angeles, CA, USA

ACM ISBN 978-1-4503-9880-0/22/11.
https://doi.org/10.1145/3560830.3563723

an attack method that addresses query-efficiency and imperceptibility by modifying all frequency components in the magnitude spectrum.

The rest of the paper is organized as follows. Section 2 introduces the necessary theory to understand our proposed method. Section 3 presents the related work. Section 4 explains our method. Section 5 shows the evaluation results. Section 6 discusses the results and Section 7 concludes.

2 FREQUENCY DOMAIN

An image can be presented in spatial domain, in which an image is represented in the form of pixel values. Another way of representing an image is through the Fourier domain, which we denote as the *frequency domain* [15]. Each point in the frequency domain $F(u, v)$ represents a certain combination of magnitude and phase of sinusoidal components, making it possible to represent any image.

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*. Decomposing and synthesizing use 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 paper. For a given image of size $d \times d$, the two-dimensional DFT is given by

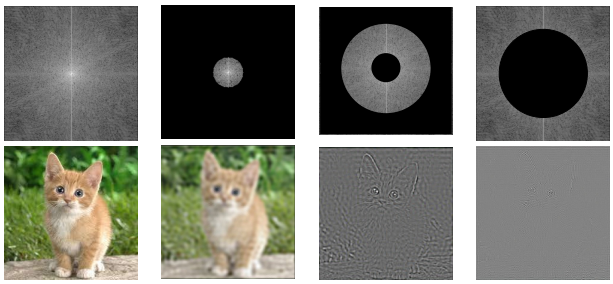
$$F(u, v) = \sum_{u=0}^{d-1} \sum_{v=0}^{d-1} f(x, y) e^{2\pi \frac{ux+vy}{d} j}. \quad (1)$$

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

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

which is very similar to Equation 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.

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. We apply a logarithmic transformation [16] to log-scale the values in the frequency domain as their value range is too large to visualize. As explained in [15], 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.



(a) (b) (c) (d)

Figure 1: Each column represents a magnitude spectrum and its corresponding spatial image. The figure shows that 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 shown in the magnitude spectrum in Figure 1a, 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 the magnitude spectrum in Figure 1a, 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 see 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 paper, we use the term *frequency band* to refer to the areas of different frequencies.

3 RELATED WORK

The malicious attacks targeting image classifiers can be categorized as white-box and black-box attacks. White-box attacks, e.g., [17], assume an adversary has access to any information about the target model and datasets used during the training of the target model. Black-box attacks, e.g.,[28], assume no information about the target model, which aligns better with real-world applications than white-box attacks. Black-box attacks can be transfer-based, score-based, or decision-based. The transfer-based attacks, e.g.,[23], assume the knowledge of transferability [32]. The attack methods usually

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

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. The score-based attacks assume no internal knowledge about the target model [6], but can access the output probabilities. 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. The decision-based attacks only assume the output label [37]. 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.

The decision-based attack methods require the initial perturbation to be adversarial to find the decision boundary. From this point, the adversary needs to navigate the decision boundary of the target model to find the optimal adversarial perturbation. A possible real-world use case for decision-based adversarial attacks against image classification models is a Not-Safe-For-Work (NSFW) filter [29]. 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. The decision-based attack can happen in spatial and frequency domains.

3.1 Decision-based Attack in Spatial Domain

The general approach of spatial attacks [32, 5, 25, 30] is to traverse the decision boundary of the target model to minimize the distance between the original image and the adversarial example. HSJA [4] is a state-of-the-art hyperparameter-free and query-efficient decision-based black-box attack in spatial domain for both targeted and untargeted attack settings. HSJA [4] 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 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 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

$$\widetilde{\nabla S}(x_t, \delta) := \frac{1}{B} \sum_{b=1}^{B} \phi_{x^*}(x_t + \delta u_b) u_b \tag{3}$$

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 [1, 9, 20], and [4] also demonstrates lower l_2 and l_∞ distances compared to other methods across multiple datasets and models.

The third component in HSJA 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 produces the adversarial image $\tilde{x}_{t+1}$ which then can be moved to the decision boundary again using a binary search. 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.

3.2 Decision-based Attacks in Frequency Domain

The general approach of frequency attacks [24, 39, 33, 18, 21, 37] 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, Frank, and Weinberger [18] show 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. The study [22] proposed a novel attack method, called F-mixup, in the high frequencies of the magnitude spectrum, as opposed to performing random sampling in the low frequencies. 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 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 Zhang et al. [22] 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. The study [7] argues that convolutional neural networks (CNNs) extract features from different frequencies and [18] 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 of F-mixup [22] 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.

4 METHODOLOGY

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 method has the following characters:

(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 in Figure 2. This section first describes how we divide the frequency spectrum into separate frequency bands before explaining each part of the attack pipeline in detail.

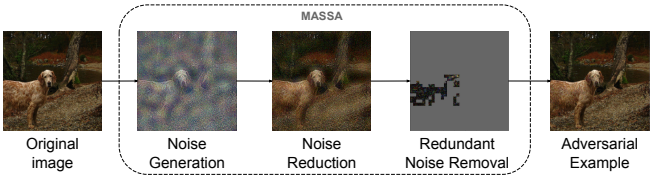
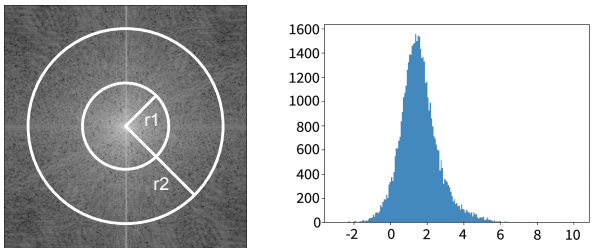


Figure 2: The overall attack pipeline of the proposed MASSA attack.

4.1 Creating frequency bands

The frequency spectrum can be divided into three bands: low-, medium-, and high frequencies. Each frequency band affects the spatial image differently. Ideally, we want to modify each band separately. Therefore, we separate the frequency spectrum into these three bands. Each spatial image needs to be decomposed into a different frequency spectrum with different frequency bands. Finding the thresholds to divide the frequency spectrum into different bands is the main challenge. 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 three bands. How r_1 and r_2 divide the frequency spectrum is illustrated in Figure 3a, 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.

To determine r_1 and r_2, we study the value range of the logarithmically scaled frequency spectrum of an image. We log-scale the values for visualization because of their large value range. Figure 3a illustrates the frequency spectrum of an image and Figure 3b shows its corresponding histogram of values. Figure 3b shows that most of the values in the frequency spectrum range are between 0-3. Additionally, we have some smaller values towards -3 and some larger ones towards 10. The histogram in Figure 3b 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. Here, we can see 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 with low, medium and high frequency bands.

(b) Distribution of values in the frequency spectrum.

Figure 3: Frequency spectrum and its value distribution

As explained in Section 2, the high-frequency band contains the smallest values in the frequency spectrum, represented by the left tail in Figure 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

$$t_l = \mu - \alpha_l \sigma \quad \text{and} \quad t_r = \mu + \alpha_r \sigma, \tag{4}$$

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 as follows:

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

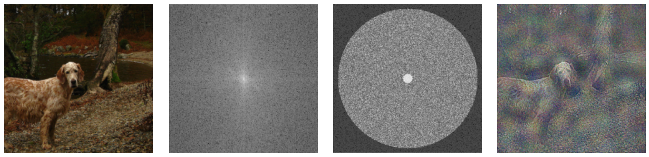
where $|M_l|$ and $|M_m|$ denote the number of values in the low-frequency mask and medium-frequency mask, respectively. In summary, the radiuses used to create the frequency bands are dependent on the tail-values used to divide the histogram in Figure 3b. The distribution of magnitude values loosely 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.

4.2 Initial Noise Generation

We propose to sample the initial perturbation from the frequency domain, such that we can generate an initial perturbation with noise in all frequency components. 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 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 in [22], we shift the low frequencies of the frequency spectrum to the center and scale the values logarithmically, which results in Figure 4b. We use $F_{i,j}$ to index the magnitude values at position (i, j).

Unlike [4], 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 4.1, 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 cover everything else. Figure 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 4d, we directly modify the values in the frequency spectrum. First, we use the mean μ and standard deviation σ shown in Figure 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 with $\alpha_l = 2$. 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 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 4d. At this point we query the target model to ensure the generated image is adversarial. This adversarial example serves as the input to the noise reduction component of our attack method.



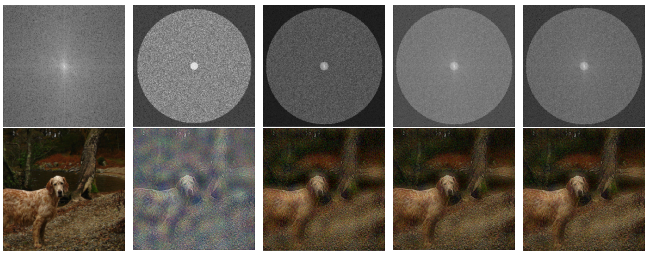
(a) (b) (c) (d)

Figure 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.

4.3 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 [14], we perform the first 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 5 and shows how drastically the distance to the original image is reduced through binary searches in the low, medium, and high-frequency bands. We apply the traditional binary search algorithm [4] in the frequency spectrum and modify it to only adjust values for the given band b. 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. During the binary search, the target model is queried to make sure we keep the frequency spectrum of the generated image adversarial.



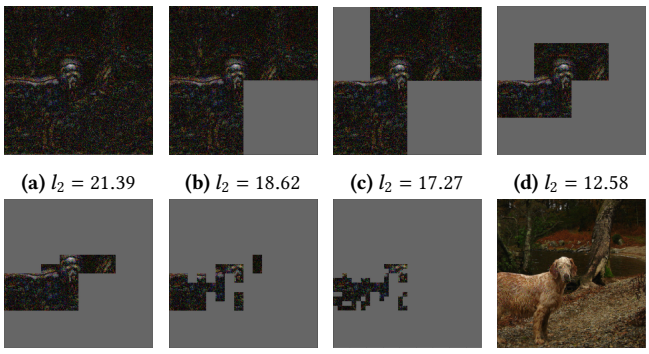
(a) $l_2 = 0.0$ **(b)** $l_2 = 92.76$ **(c)** $l_2 = 27.00$ **(d)** $l_2 = 21.40$ **(e)** $l_2 = 21.39$

Figure 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.

4.4 Removal of Redundant Noise

The last component of our attack method is the removal of redundant noise. Shi and Han [31] reveal that most noise in the initial adversarial example is redundant, and one can speed up subsequent decision-based attacks by removing the redundant noise [32]. Shi and Han [31] present Patch-wise Adversarial Removal (PAR), an attack method that removes redundant noise of adversarial examples. 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 one indicates that the noise in this patch has been successfully removed or has not yet tried to remove the noise. A zero 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. The noise compression of PAR greatly reduces the initial noise, which can speed up the process of a subsequent attack method [32]. Even though [31] presents a query-efficient method to compress initial noise, PAR is more powerful in combination with existing decision-based attack methods. Based on this discovery, we propose to remove redundant noise as the final step of our attack method. Similar to [31], 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 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 can remove large parts of unnecessary noise, which further decreases the l_2 distance. We recursively perform the same steps on each patch where the noise was not removed and 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 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 6a depicts only the changes made to the original image. Each step after that removes patches of different sizes, clearly illustrates that a lot of noise is redundant. The noise removal process ends when the minimum patch size is reached. To decide when to finish 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 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.

5 EVALUATION RESULTS

We implemented a system to carry out adversarial attacks on different image classification models. Our code is available at [2]. We measured the Query Finish Rate of MASSA, the success rates, and compare the median and average l_2 distances of our approach with

HSJA. As a baseline for comparison, we use the implementation of HSJA from their publicly available code [3]. 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. The l_2 distance comparisons are performed on normal target models, target models with JPEG compression as a defense mechanism, and target models with adversarial training as a defense mechanism. We chose JPEG compression as a defense mechanism with a quality factor of 0.8 because it directly targets the frequency domain by removing the high frequencies of an image [36], and, JPEG compression has been shown to reverse the adversarial perturbation on images modified by a small amount [10]. Adversarial training [35] increases robustness by including adversarial examples in the training data. Even though black-box attacks have previously been shown to evade adversarial training [28], we investigate how adversarial training affects our frequency-based attack MASSA. We use the ResNet-50 model from [13] with $\epsilon = 3$, which is adversarially trained on the ImageNet dataset.

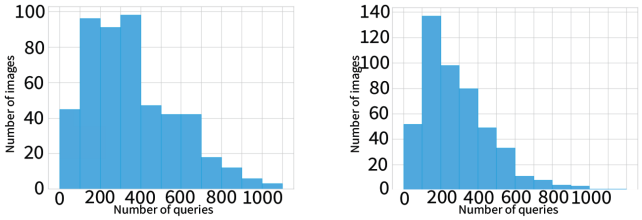
5.1 Query Finish Rate

Our first experiment explores the Query Finish Rate for MASSA. Table 1 illustrate how many of the 500 adversarial examples our approach produced were 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 7. Each bin has a size of 100, where the ranges are [0, 100), [100, 200), and so on.

Table 1: 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.

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%

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 MASSA can conduct a powerful attack under a very limited query budget. In addition, 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. Figure 7 shows that very few adversarial examples require 1000 queries. MASSA is also 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 1 and Figure 7 we observe that ResNet-50 is more challenging to create adversarial



(a) Target Model: ResNet-50 (b) Target Model: VGG16

(c) Target Model: VGG19

Figure 7: Histogram of query finish rate for each target model.

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. The results also show that the worst-case scenario for MASSA in terms of queries is a rare occurrence, as less than 1% of adversarial examples produced by MASSA require 1000 queries or more.

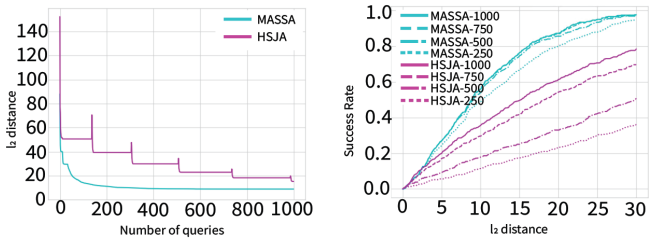
5.2 l_2 Distance

We investigated the performance of MASSA in terms of l_2 distance compared to the HSJA. Table 2 summarizes the median and average l_2 distances for both attack methods with query budgets of 250, 500, 750, and 1000 queries across all experiments. We also illustrate the median distance results in Figure 8. The spikes for HSJA comes from the geometric progression, where the adversarial example is moved away from the decision boundary. This step causes the increase of l_2 distance, hence the sudden spikes. The plateaus for HSJA come from the gradient direction estimation step, where HSJA samples and queries hundreds of adversarial examples around the boundary and 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. For MASSA in Figure 8, 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, where each plateau corresponds to a different frequency band being moved closer to its original values. As mentioned in subsection 5.1, our attack method rarely uses 1000 queries. So, in Figure 8, 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.

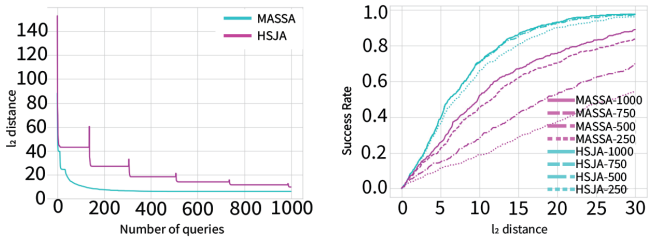
Table 2 clearly shows that MASSA can create adversarial examples with a significantly smaller l_2 distance than the corresponding adversarial examples created by HSJA. For all comparisons made in Table 2, MASSA beats HSJA across all models and query budgets. The differences are most apparent on ResNet-50 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. We also 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.

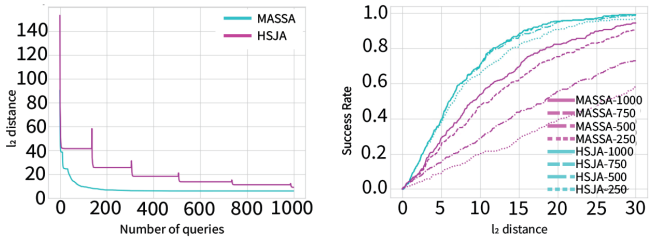
Figure 8 shows that MASSA achieves a steeper decrease in l_2 distance than HSJA, where MASSA descend to an l_2 distance of 20-30 in the first 50 queries. This demonstrates the effectiveness of the frequency binary search explained in subsection 4.3. Although MASSA ends on a lower l_2 distance than HSJA 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. However, we argue that this scenario is probably irrelevant, as a query budget of more than 1000 queries moves the attack outside the range of state-of-the-art performance. We include visualized trajectories of MASSA in Figure 9. The trajectories are selected randomly from correctly classified images from the ILSVRC2012 validation set.



(a) Target Model: ResNet-50



(b) Target Model: VGG16



(c) Target Model: VGG19

Figure 8: Left column: Median l_2 distances versus number of queries. Right column: Success rate for various l_2 distance thresholds.

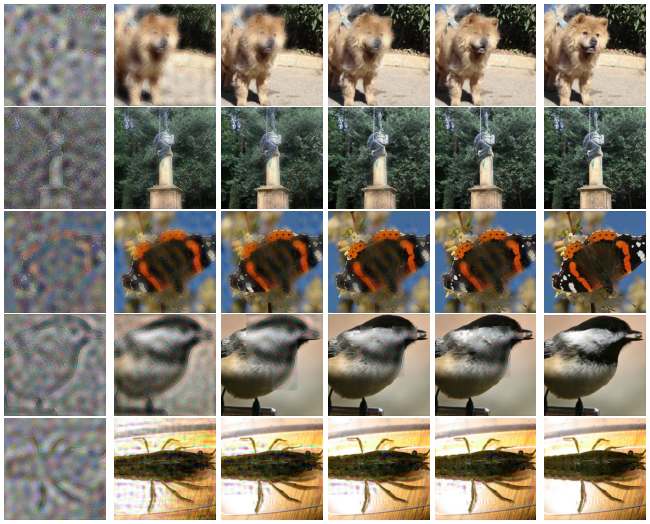


Figure 9: Visualized trajectories of MASSA for 5 images from ILSVRC2012 validation dataset. 1st column: initial perturbation. Columns 2-5: adversarial examples at 25, 50, 100, and 200 queries, respectively. Last column: Original image.

5.3 Success Rate

We also evaluate the success rate of MASSA compared to HSJA. Figure 8 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 8 shows the superior performance of MASSA compared to HSJA. All query budgets of MASSA achieve a consistently higher success rate than HSJA. *MASSA-250* also 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.

5.4 Evaluation Under Defense Mechanisms

To evaluate our attack method under defense mechanisms, we use l_2 distance and success rate as metrics. We first evaluate attack efficiency using JPEG compression as a defense mechanism where the adversarial example is compressed right before querying the target model. Table 2 summarizes the results of median and average l_2 distance for HSJA and MASSA across target models with JPEG compression. Table 2 shows 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 2 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

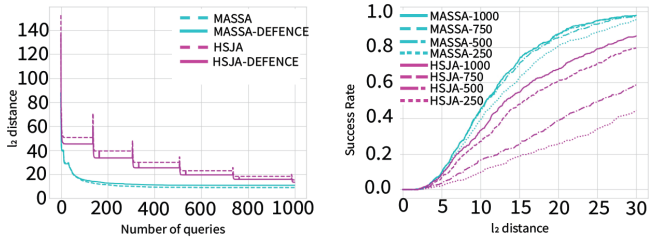
Table 2: Median and average distance at various model queries for each target model under different defense mechanisms. The smaller distance at a given model query is bold-faced.

Defense	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** (-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%)
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**

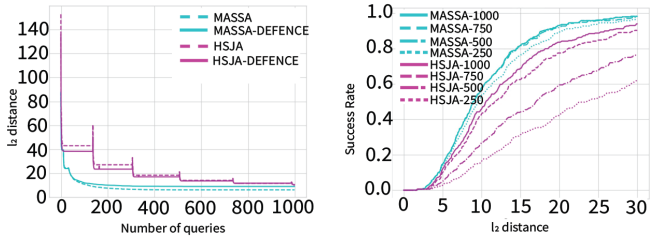
on ResNet-50. Thus, JPEG compression slightly affects MASSA. It is also worth noting here that JPEG compression improves the results for HSJA. This is because JPEG compression removes the high frequencies in an image where noise is already located, meaning JPEG compression can further reduce the l_2 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.

We show the median l_2 distance and success rate under JPEG-compression in Figure 10. For median l_2 distance, we show each attack method both with JPEG compression (denoted with DEFENCE) and without JPEG compression for ease of comparison. The success rate only includes the attack methods under JPEG compression. Figure 10 shows that the early decrease in l_2 distance for MASSA remains unchanged after JPEG compression. The noise reduction step of MASSA first modifies the low and medium frequencies. 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.

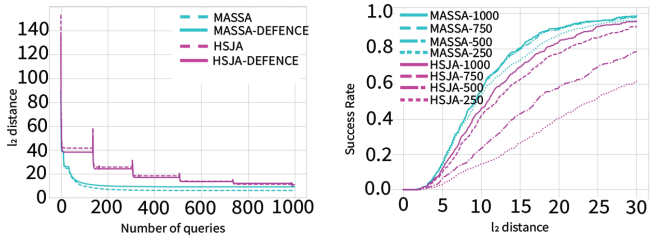
Figure 10 shows that MASSA achieves a higher success rate than HSJA under JPEG compression. We observe that even *MASSA-250* still outperforms *HSJA-1000*. In difference to the success rate without JPEG compression shown in Figure 8, both attack methods struggle to generate adversarial examples with 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.



(a) Target Model: ResNet-50



(b) Target Model: VGG16



(c) Target Model: VGG19

Figure 10: Under JPEG compression. Left column: Median l_2 distances versus number of queries. Right column: Success rate for various l_2 distance thresholds.

We also conduct experiments to evaluate MASSA and HSJA under adversarial training. Table 2 summarizes the results of median and average l_2 distance for the attack methods against an

adversarial trained ResNet-50 with $\epsilon = 3$. It is worth noting that the adversarially trained network is not optimized against MASSA or HSJA. In these experiments, 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 2, for median distance at 500 queries, we see an increase of 90% for MASSA and 173% 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 11. We include both MASSA and HSJA with and without defense for comparison, where the adversarial model is denoted with *DEFENCE*. The median distance versus number of queries in Figure 11 shows that both attack methods achieve similar slope 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. The success rate clearly shows how HSJA is affected by adversarial training, compared to the success rate in Figure 8. The success rate is significantly lower under adversarial training, in contrast to MASSA, which remains at a similar performance as without adversarial training in Figure 8.

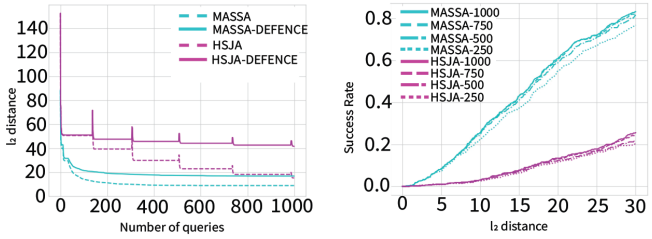


Figure 11: Target model: ResNet-50 under adversarial training. Left: Median l_2 distance versus number of queries. Right: Success rate for various l_2 distance thresholds.

6 DISCUSSION

This section will compare our approach with related work and discuss the results of our studies in academia and industry.

6.1 Comparison to Related Work

F-mixup [22] 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. Instead of using PAR [31] 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.

HSJA and related work use hundreds of queries at the binary search 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 demonstrated by the steep decrease for median l_2 distance. This also makes our approach far more query-efficient than any related work. 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.

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. HSJA can bypasses adversarial training on MNIST [4]. Our results show that HSJA struggles significantly under adversarial training on ImageNet.

6.2 Implications

Our approach operates under a significantly lower query budget than 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. New defense mechanism against few queries need to be developed. 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. Our method uses a redundant noise removal step inspired by PAR [31] which clearly shows how much of the generated noise is redundant. Other attacks might benefit from the same noise removal process, and the corresponding defense mechanisms need to be developed. As illustrated in Figure 5 and in [22], the frequency-based attacks result in unnatural frequency spectrums. New defense mechanisms need to detect these abnormal spectrums to make computer vision systems more robust.

We have demonstrated the potential to craft imperceptible adversarial examples in just hundreds of queries. This poses a more significant threat to the industry because it is a more realistic approach. As we push the query budget lower, it might become more challenging to detect an attack. From the perspective of the target model, 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.

7 CONCLUSION AND FUTURE WORK

We propose a new decision-based black-box attack method, MASSA, which generates imperceptible adversarial examples under a strict query budget. The evaluation results demonstrate that MASSA achieves superior performance over the state-of-the-art attack HSJA across all classification models and defense settings. Additionally, MASSA bypasses two defense mechanisms with comparable results to HSJA without defense mechanisms.

Many ideas of MASSA are derived from empirical experiments. We will advance its theoretical basement in the future. Another future work is to evaluate MASSA with other defenses, e.g., defense mechanisms in the frequency domain.

REFERENCES

[1] Wieland Brendel *, Jonas Rauber *, and Matthias Bethge. 2018. Decision-based adversarial attacks: reliable attacks against black-box machine learning models. In *International Conf. on Learning Representations*. https://openreview.net/forum?id=SyZI0GWCZ.

[2] Johannes Åsheim and Kim André Brunstad Midtlid. 2022. MASSA github repository. https://github.com/johanaas/master.

[3] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. 2019. Hopskipjump github repository. Retrieved 05/25/2022 from https://github.com/Jianbo-Lab/HSJA/.

[4] Jianbo Chen and Michael I. Jordan. 2020. Hopskipjumpattack: a query-efficient decision-based attack. *2020 IEEE Symp. on Security and Privacy (SP)*, 1277–1294.

[5] Jinghui Chen and Quanquan Gu. 2020. Rays: a ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conf. on Knowledge Discovery Data Mining* (KDD '20). ACM, Virtual Event, CA, USA, 1739–1747. ISBN: 9781450379984. DOI: 10.1145/3394486.3403225. https://doi.org/10.1145/3394486.3403225.

[6] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. ZOO. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, (November 2017). DOI: 10.1145/3128572.3140448. https://doi.org/10.1145%2F3128572.3140448.

[7] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Yan Shuicheng, and Jiashi Feng. 2019. Drop an octave: reducing spatial redundancy in convolutional neural networks with octave convolution. eng. In *2019 IEEE/CVF International Conf. on Computer Vision (ICCV)*. Volume 2019-. IEEE, 3434–3443. ISBN: 9781728148038.

[8] Zhilu Chen and Xinming Huang. 2016. Accurate and reliable detection of traffic lights using multiclass learning and multiobject tracking. eng. *IEEE intelligent transportation systems magazine*, 8, 4, 28–42. ISSN: 1939-1390. DOI: 10.1109/MITS.2016.2605381.

[9] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2019. Query-efficient hard-label black-box attack: an optimization-based approach. In *7th International Conf. on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rJlk6iRqKX.

[10] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. 2016. A study of the effect of JPG compression on adversarial images. *CoRR*, abs/1608.00853. arXiv: 1608.00853. http://arxiv.org/abs/1608.00853.

[11] Fatemeh Ebadi and Mohammad Norouzi. 2017. Road terrain detection and classification algorithm based on the color feature extraction. In *2017 Artificial Intelligence and Robotics (IRANOPEN)*. IEEE, 139–146. DOI: 10.1109/RIOS.2017.7956457.

[12] Bassant Mohamed Elbagoury, Abdel-Badeeh M. Salem, and Luige Vladareanu. 2016. Intelligent adaptive precrash control for autonmous vehicle agents (cbr engine amp; hybrid a path planner). In *2016 International Conf. on Advanced Mechatronic Systems (ICAMechS)*. (November 2016), 429–436. DOI: 10.1109/ICAMechS.2016.7813486.

[13] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. 2019. Robustness (python library). (2019). https://github.com/MadryLab/robustness.

[14] David J. Field. 1987. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, 4, 12, (December 1987), 2379–2394. DOI: 10.1364/JOSAA.4.002379. http://opg.optica.org/josaa/abstract.cfm?URI=josaa-4-12-2379.

[15] Robert Fisher, Simon Perkins, Ashley Walker, and Erik Wolfart. 2003. Fourier transform. ©HIPR. Retrieved 04/27/2022 from https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm.

[16] Robert Fisher, Simon Perkins, Ashley Walker, and Erik Wolfart. 2003. Fourier transform. ©HIPR. Retrieved 04/27/2022 from https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixlog.htm.

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conf. on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conf. Track Proceedings*. Yoshua Bengio and Yann LeCun, editors. http://arxiv.org/abs/1412.6572.

[18] Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. 2019. Low frequency adversarial perturbation. In *Proceedings of the Thirty-Fifth Conf. on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019* (Proceedings of Machine Learning Research). Amir Globerson and Ricardo Silva, editors. Volume 115. AUAI Press, 1127–1137. http://proceedings.mlr.press/v115/guo20a.html.

[19] Sabrine Hamdi, Hassene Faiedh, Chokri Souani, and Kamel Besbes. 2017. Road signs classification by ann for real-time implementation. In *2017 International Conf. on Control, Automation and Diagnosis (ICCAD)*. IEEE, 328–332. DOI: 10.1109/CADIAG.2017.8075679.

[20] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conf. on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018* (Proceedings of Machine Learning Research). Jennifer G. Dy and Andreas Krause, editors. Volume 80. PMLR, 2142–2151. http://proceedings.mlr.press/v80/ilyas18a.html.

[21] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. 2020. Qeba: query-efficient boundary-based blackbox attack. *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1218–1227.

[22] X. Li, X. Zhang, F. Yin, and C. Liu. 2021. F-mixup: attack cnns from fourier perspective. In *2020 25th International Conf. on Pattern Recognition (ICPR)*. IEEE Computer Society, Los Alamitos, CA, USA, (January 2021), 541–548. DOI: 10.1109/ICPR48806.2021.9412611. https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9412611.

[23] Hong Liu, Rongrong Ji, Jie Li, Baochang Zhang, Yue Gao, Yongjian Wu, and Feiyue Huang. 2019. Universal adversarial perturbation via prior driven uncertainty approximation. In *2019 IEEE/CVF International Conf. on Computer Vision (ICCV)*. IEEE, 2941–2949. DOI: 10.1109/ICCV.2019.00303.

[24] Sijia Liu, Jian Sun, and Jun Li. 2020. Query-efficient hard-label black-box attacks using biased sampling. In *2020 Chinese Automation Congress (CAC)*. IEEE, 3872–3877. DOI: 10.1109/CAC51589.2020.9326734.

[25] Thibault Maho, Teddy Furon, and Erwan Le Merrer. 2021. Surfree: A fast surrogate-free black-box attack. In *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 10430–10439. DOI: 10.1109/CVPR46437.2021.01029.

[26] MobileSec. 2022. Mobilesec android authentication framework. Retrieved 05/26/2022 from https://github.com/mobilesec/authentication-framework-module-face.

[27] Hamzah Al Najada and Imad Mahgoub. 2016. 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 Conf. (UEMCON)*, 1–6. DOI: 10.1109/UEMCON.2016.7777922.

[28] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conf. on Computer and Communications Security, AsiaCCS 2017*. ACM, 506–519. DOI: 10.1145/3052973.3053009. https://doi.org/10.1145/3052973.3053009.

[29] Navendu Pottekkat. 2020. Nsfw filter. Retrieved 06/01/2022 from https://github.com/nsfw-filter/nsfw-filter.

[30] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. 2020. Geoda: A geometric framework for black-box adversarial attacks. In *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 8443–8452. DOI: 10.1109/CVPR42600.2020.00847.

[31] Yucheng Shi and Yahong Han. 2021. Decision-based black-box attack against vision transformers via patch-wise adversarial removal. (2021). DOI: 10.48550/ARXIV.2112.03492. https://arxiv.org/abs/2112.03492.

[32] Yucheng Shi, Yahong Han, and Qi Tian. 2020. Polishing decision-based adversarial noise with a customized sampling. In *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1027–1035. DOI: 10.1109/CVPR42600.2020.00111.

[33] Satya Narayan Shukla, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. 2021. Simple and efficient hard label black-box adversarial attacks in low query budget regimes. In *KDD '21: The 27th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. Feida Zhu, Beng Chin Ooi, and Chunyan Miao, editors. ACM, 1461–1469. DOI: 10.1145/3447548.3467386. https://doi.org/10.1145/3447548.3467386.

[34] NEURO Technology. 2022. Sentiveillance sdk. Retrieved 05/26/2022 from https://www.neurotechnology.com/sentiveillance.html.

[35] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: attacks and defenses. (2017). DOI: 10.48550/ARXIV.1705.07204. https://arxiv.org/abs/1705.07204.

[36] G.K. Wallace. 1992. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38, 1, xviii–xxxiv. DOI: 10.1109/30.125072.

[37] Xiaosen Wang, Zeliang Zhang, Kangheng Tong, Dihong Gong, Kun He, Zhifeng Li, and Wei Liu. 2021. Triangle attack: a query-efficient decision-based adversarial attack. (2021). DOI: 10.48550/ARXIV.2112.06569. https://arxiv.org/abs/2112.06569.

[38] Pu Zhao, Pin-Yu Chen, Siyue Wang, and Xue Lin. 2020. Towards query-efficient black-box adversary with zeroth-order natural gradient descent. In *The Thirty-Fourth AAAI Conf. on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conf., IAAI 2020, The Tenth AAAI Symp. on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 6909–6916. https://ojs.aaai.org/index.php/AAAI/article/view/6173.

[39] Weiwei Zhao and Zhigang Zeng. 2021. Improved black-box attack based on query and perturbation distribution. In *2021 13th International Conf. on Advanced Computational Intelligence (ICACI)*. IEEE, 117–125. DOI: 10.1109/ICACI52617.2021.9435907.