HAW HAMBURG

MASTER THESIS
Kolja Luca Sielmann

Adversarial Attacks and Defense based on JPEG Coefficients

Faculty of Engineering and Computer Science
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Kolja Luca Sielmann

Adversarial Attacks and Defense based on JPEG Coefficients

Master thesis submitted for examination in Master´s degree
in the study course *Master of Science Informatik*
at the Department Computer Science
at the Faculty of Engineering and Computer Science
at University of Applied Science Hamburg

Supervisor: Prof. Dr. Peer Stelldinger
Supervisor: Prof. Dr. Marina Tropmann-Frick

Submitted on: July 13, 2023

Kolja Luca Sielmann

Title of Thesis

Adversarial Attacks and Defense based on JPEG Coefficients

Keywords

Adversarial Attacks, JPEG Coefficients, Perceptual Metrics, Neural Nets

Abstract

Neural networks have been shown to be vulnerable towards small, barely visible alterations of input images that lead to misclassifications, so-called adversarial examples. There has been a lot of research on creating adversarial examples and how to defend nets against them. Usually, those methods perturb images' RGB pixel representations. We propose applying perturbations straight on JPEG coefficients. Our method allows to control the perturbation applied on each YC_bC_r channel and each DCT frequency. We find that adversarial perturbation is often most efficient when it is applied on medium DCT frequencies, with efficiency being defined as the proportion of success rate and perceived distances. The superiority of medium-frequency perturbations is especially clear when JPEG compression is used in defense. We also show that, for maximum-confidence attacks, perturbing JPEG coefficients is more efficient than the state-of-the-art attacks that mainly apply the alterations in RGB pixel space, which is reasoned in using the YC_bC_r color model allowing us to limit the perturbation to the luma channel where it is more efficient but also controlling the perturbation applied on each frequency. By weighting multiple JPEG attacks that concentrate their perturbations on different parts of the DCT frequency spectrum during adversarial training, we are able to train a net that is robust against perturbations on the whole frequency spectrum and RGB and YC_bC_r pixel attacks as well which shows that JPEG coefficients are a representation that is well-suited to achieve more generalizing robustness against unforeseen threat models as well.

Kolja Luca Sielmann

Thema der Arbeit

Adversarial Attacks und Defenses auf Basis von JPEG Koeffizienten

Stichworte

Adversarial Attacks, JPEG Koeffizienten, Perzeptuelle Distanzmetriken, Neuronale Netze

Kurzzusammenfassung

Kleine, kaum sichtbare Veränderungen, sogenannte Adversarial Examples, können zur falschen Klassifikation durch neuronale Netze führen. Es wurden bereits viele Methoden zur Erstellung solcher Adversarial Examples und zur Verteidigung entworfen. Üblicherweise verändern diese die Bilder in ihrer RGB-Pixelrepräsentation. In dieser Thesis werden diese Veränderungen direkt auf JPEG Koeffizienten durchgeführt. Dabei kann die Stärke der Veränderung auf jedem YC_bC_r-Kanal sowie jeder DCT-Frequenz einzeln kontrolliert werden. Wir zeigen, dass Veränderungen auf mittleren Frequenzen am effizientesten sind, wobei die Effizienz als Verhältnis von Erfolgsrate und wahrgenommener Distanz definiert ist. Die Überlegenheit der Veränderungen auf mittleren Frequenzen gilt insbesondere dann, wenn JPEG compression zur Verteidigung genutzt wird. Zusätzlich zeigen wir, dass JPEG Koeffizienten grundsätzlich die effizientere Representation für Maximum-Confidence-Attacks als RGB-Pixel sind. Dies ist sowohl durch die Nutzung des YC_bC_r-Farbmodells begründet, was ermöglicht, nur Luminanz-Informationen zu verändern, als auch durch die Nutzung der DCT Koeffizienten, wodurch die Veränderungen manuell auf das Frequenzspektrum verteilt werden kann. Mithilfe der Gewichtung verschiedener solcher Angriffe, die jeweils unterschiedliche Teile des Frequenzspektrums anvisieren, trainieren wir mit Adversarial Training ein Netz, welches sowohl gegen JPEG Angriffe auf unterschiedlichen Frequenzen als auch gegen Angriffe auf RGB- und YC_bC_r-Pixeln robust ist.

Contents

List of Figures

List of Tables

1 Introduction

The performance of neural networks has significantly increased in recent years in many use cases in image processing, such as object detection [60, 103], generative methods [16, 49] or image classification where it reached similar accuracies as humans [37, 78]. However, Szegedy et al. [91] have shown in 2014 that a neural net's classification cannot be trusted as they are vulnerable against perturbation on the input that is specifically crafted to fool them. These malicious input images are called adversarial examples.

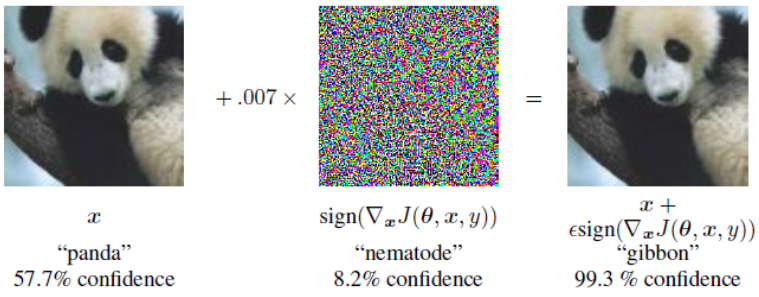


Figure 1.1: Example of a well-known adversarial image from [33]. The adversarial noise shown in the middle image is applied to the original image on the left to receive a misclassified image.

Figure 1.1 shows a well-known adversarial example where a panda from the IMAGENET [18] dataset is misclassified as a gibbon despite looking indistinguishable from the original image after adding the adversarial noise. This obviously has strong implications on the use of neural networks in a real-world scenario as the predictions of a net cannot be trusted.

Since the vulnerability towards such adversarial perturbations has been discovered, there has been a large amount of research on methods that create such perturbations, so-called adversarial attacks, and how to defend neural nets against them [5, 33, 45, 56, 55, 65, 80, 91, 93].

The vulnerability of neural networks shows that neural networks have not "*obtained a true human-level understanding*" [32, p. 265]. But how can a model be trained so that it aligns better with human perception and is thus robust against adversarial attacks? A well-known method to make neural networks more robust towards these attacks is adversarial training, where adversarial examples are added to the training set during training [33]. And while it has been shown that this method can significantly increase robustness [65] and make the nets use features that are more aligned with the human perception [93], the resulting nets still tend to be vulnerable towards threat models that are unseen during training and often do not generalize well [48, 58].

Usually, the images that are created by adversarial attacks and included in the training process, are represented as RGB pixels and the perturbations are limited by L_p distances measured in RGB pixel space, which results in a colored noise added to the image, as can be seen in fig. 1.1.

In contrast, we perturb the images straight on JPEG coefficients. The motivation is as follows: As neural networks trained on benign datasets use properties of the data that are different from those used by humans, it could be advantageous for adversarial attacks to use a data representation that separates perceptible parts from imperceptible parts of the data. From an offensive point of view, one could then exploit the neural network's reliance on imperceptible parts of the data by slightly perturbing exactly those imperceptible parts and thereby hiding the perturbation. From a defensive point of view, the net could be forced to use perceivable parts of the data. As a lossy compression algorithm, JPEG compression does exactly this by converting the images to frequency space, where the highest frequencies are usually assumed to be less perceivable for humans. Using our attacks, one can manually weight the perturbation applied on each frequency and thus, presumably force the model to use frequencies that are also used by the human perception. Thus, we believe that our JPEG attacks can be the basis for training a net that is better aligned with human perception and thus more robust against unseen threat models as well.

A similar idea is also used by methods that apply JPEG compression in defense. Here, the imperceptible parts are automatically removed during the compression, which tries to force the net to use features that are visible for humans as well. However, this is known to be a weak defense only [17, 22, 35, 77].

Additionally, allowing to distribute across frequencies and the three $\mathrm{YC_bC_r}$ color channels could lead to the JPEG attacks being more efficient than RGB attacks as well, where the

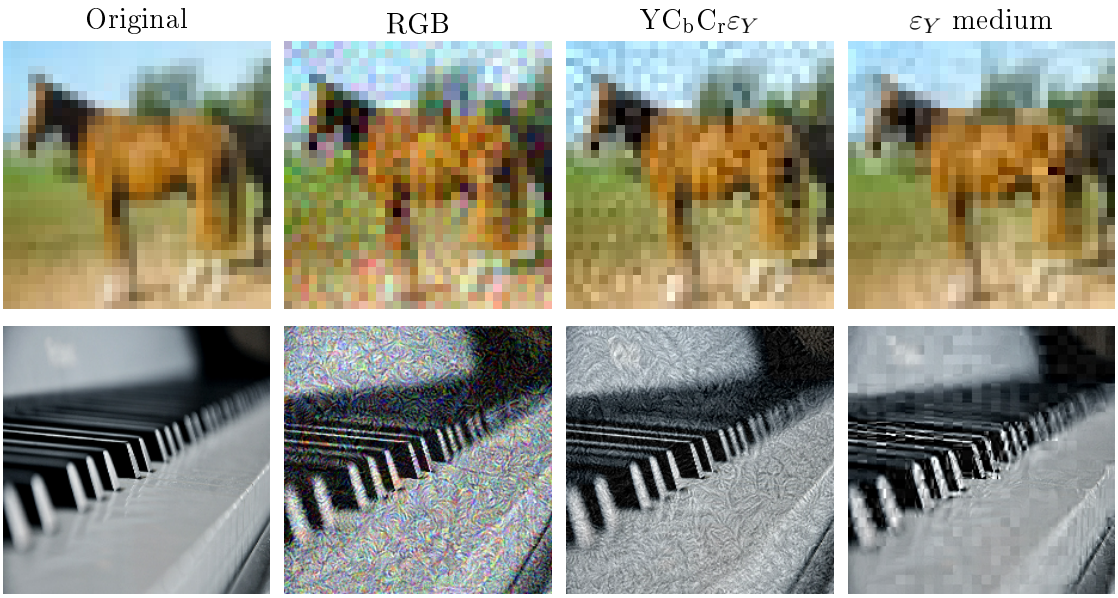


Figure 1.2: Minimum perturbation required for a misclassification by the Densenet^{jq50}. Images are created on a Resnet. The picture on the right was created using our JPEG attack with the perturbation concentrated in medium frequencies. Note that our attack as well as the YC_bC_r attack only perturbed the luma channel.

efficiency is measured as a rate of the attack's success and the created perceived distance. For example, by attacking medium frequencies one could avoid a visible high-frequency noise, or the noise can be forced to be grayscale which is usually less visible than colored noise, as illustrated in fig. 1.2.

This work's main purposes are:

- Developing an attack that perturbs the JPEG coefficients directly instead of relying on perturbations made in RGB representation, and exploring the efficiency of our attack for various parameters,
- examining whether, compared to other attacks, our attack's perturbations are more robust against JPEG compression used in the defense,
- determining on which part of the frequency spectrum our attacks are most efficient on different nets,
- analyzing how the defense method influences the net's vulnerability across the frequency spectrum,

- testing whether our JPEG attacks can indeed increase the robustness and generalization across the whole frequency spectrum and against RGB attacks as well,
- and discussing the state of current defenses and how our results could help improve them.

This thesis is structured as follows: First, all necessary background on adversarial attacks and defenses, JPEG compression and perceptual metrics will be provided in chapter 2. Then, we will describe our proposed method in chapter 3, followed by our experiments and results in chapter 4. At last, chapter 5 will summarize and discuss our results.

Some parts of this work have already been submitted as a conference paper simultaneously to the work on this thesis and are currently under review [89]. The conference paper includes the results on maximum-confidence attacks, the comparison between RGB and JPEG attacks and on adversarial training using our JPEG attacks.

2 Background

This chapter will cover all necessary background for our JPEG attacks. It starts with an overview on adversarial attacks, their explanations and implications in section 2.1, followed by a summary of JPEG compression and how it is related to adversarial attacks in section 2.2. As adversarial examples are intended to be close to the original input, measuring the perceived distortion using perceptual metrics is an important part of research on adversarial attacks. Some of these metrics will be detailed in section 2.3. Then, section 2.4 covers defense methods against adversarial perturbations. Finally, section 2.5 will discuss adversarial attacks and robustness from a frequency perspective.

2.1 Adversarial Attacks

Adversarial examples are input samples that have been altered with nearly imperceptible or natural looking perturbations to fool machine learning models such as neural networks. Adversarial examples can, for example, be textual [59], auditory [15], graph-structured [105] or visual data [33, 91] such as images which will be this work's focus. The vulnerability of neural networks to such adversarial perturbations was first identified by Szegedy et al. [91]. They showed that by maximizing the network's loss adversarial images that look very similar to benign samples can be found. Some examples have been shown in figs. 1.1 and 1.2.

The existence of adversarial examples has strong implications for the use of machine learning models in applications where correct classification is necessary for safety reasons, such as autonomous driving [19] or medical diagnosis [29]. While the safety risks can be prevented in some applications such as medical diagnosis, where the net's classification is only used to support the human expert's decision and are ideally explained using Explainable AI methods [41, 92], more automated domains such as autonomic driving require the classification to be correct or the classifier has to be able to state when its

prediction cannot be relied upon to enable the car to react accordingly in an automated process. Thus, detecting and preventing adversarial examples from being successful is a major task in machine learning research.

Mathematically, an adversarial example can be defined in several ways. First, one can limit the allowed perturbation of an adversarial example. Then, an image $x' = x + \delta$ is called an adversarial example, if and only if

$$D(x', x) \leq \varepsilon \wedge C(x') \neq y, \tag{2.1}$$

where D is some distance metric, $C(x)$ is the neural net's predicted class for input x, y is the image's ground-truth label and ε limits the allowed perturbation [80].

This definition is used in maximum-confidence [75] attacks, which are limited by the allowed perturbations ε and try to perturb the image in a way that it is misclassified by some neural net. The definition above describes an untargeted attack setting. A targeted attack would then be defined by replacing the second term by $C(x') \neq y_t$, where y_t is the target label. In this work, we will focus on untargeted attacks though.

Second, one can define the misclassification by a neural net as the only condition. Attacks that rely on this definition, usually minimize the distance measured using the distance metric D subject to the image being misclassified. Those attacks are called minimum-norm [75] attacks.

The first (targeted) adversarial attack, proposed by Szegedy et al. [91], follows the second definition. They formulate the following optimization problem:

$$\begin{aligned} &\text{minimize} && ||\delta||_2 \\ &\text{subject to} && C(x+\delta) = y_t \\ & && x_i + \delta_i \in [0, 255]\, \forall i \end{aligned} \tag{2.2}$$

However, they approximate the optimization problem by using a box-constrained L-BFGS [104] to

$$\begin{aligned} &\text{minimize} && c \cdot ||\delta||_2 + \mathrm{J}(x+\delta, y_t) \\ &\text{subject to} && x_i + \delta_i \in [0, 255]\, \forall i \end{aligned} \tag{2.3}$$

where J denotes the categorical crossentropy loss. While the attack is very successful, it is computationally expensive due to the L-BFGS optimization and thus, has soon been replaced by more efficient attacks for both attacking and defending neural nets.

In this work, we consider attacks that apply noise on the whole image. The considered attacks, which usually use RGB representations of images, will be described in section 2.1.4 for maximum-confidence and section 2.1.5 for minimum-norm attacks. The attacks that already use JPEG or perceptual metrics will be covered later, in sections 2.2 and 2.3, as they require knowledge of JPEG compression and perceptual metrics. First, however, we will give an overview on attacks, their applications and explain the implications for the usage of AI systems and for necessary research in section 2.1.1, and address possible explanations for the vulnerability of neural nets against adversarial perturbations in section 2.1.2.

2.1.1 Overview on Attacks, Applications and Implications

In order to understand the state-of-the-art of adversarial attacks and defenses and why research on both is important for being able to use machine learning models safely in a real-world scenario, it is important to give a short overview of attacks and applications in which adversarial attacks are problematic. We begin with an overview on the attacks that are not considered in detail in this work but help to understand why adversarial attacks are more than a theoretical problem.

Attacks

There are various types of adversarial attacks on images. In this work, we will focus on attacks that put some adversarial noise on the whole image. This has been defined by Laidlaw and Feizi [57] as the additive threat model, where the input data is perturbed by

$$(x_1, \ldots, x_n) \to (x_1 + \delta_1, \ldots, x_n + \delta_n). \tag{2.4}$$

There will be more information on those attacks later in sections 2.1.3 to 2.1.5.

Now, we will give a short overview on attacks that use different threat models but are not considered in detail in this work. For example, attacks can only be allowed to perturb some parts of images. These perturbations are called adversarial patches [9]. Sharif et al. [82] showed that adding eyeglasses to a portrait can prevent people from being recognized

Figure 2.1: Impersonation-Attack from [82]. The attack is performed by adding eyeglasses to the person on the left, who is then recognized as the person on the right.

Figure 2.2: Real graffiti and adversarial stickers from [27]. The perturbations are designed to imitate graffiti such that they are unsuspicious for humans.

or allows them to impersonate another person in a targeted attack. Figure 2.1 shows an example where an image of an actress that was perturbed by adding eyeglasses was misclassified as another actress. The attack was also successfully tested in a physical setting where eyeglasses were created using a 3D-printer. As they are often easy to apply in the physical world, adversarial patches can be considered a practical threat in security-critical applications, e.g. for face-recognition or autonomous driving systems. For the latter, there is a well-known example, shown in fig. 2.2 of physical perturbations on stop signs, where the targeted attack proposed by Eykholt et al. [27] achieved 100 % success rate on road sign classifiers by simply mounting white and black stickers on signs. Another example of adversarial attacks that only perturb certain parts of images are L_0 attacks. The L_0-norm measures the number of pixels that differ from the original

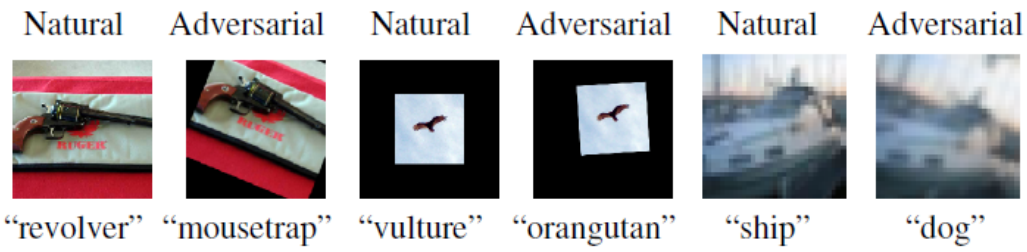


Figure 2.3: Adversarial geometric transformations from [25].

images, such that only a certain number of images is allowed to change. An example for such an attack has been proposed by Carlini and Wagner [13].

Some attacks using the spatial threat model [57] also try to force a misclassification by applying simple geometric transformations [25, 98]. E.g., Engstrom et al. [25] showed that transformations such as translations and rotations can fool neural networks. They define the transformation as moving the pixel (u, v) to the position

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \delta v \\ \delta v \end{bmatrix}, \tag{2.5}$$

where θ defines the degree by which the image is rotated and $\delta u, \delta v$ define the translation. Then, they try to find the parameters $\theta, \delta u, \delta v$ that maximize the model loss, for example by ascending the loss gradient. Examples for successful attacks are shown in fig. 2.3.

Laidlaw and Feizi [57] proposed functional adversarial attacks that limit the threat model by the condition that all pixels have to be perturbed by the same function h. In contrast to the additive threat model from eq. (2.4), they define the functional threat model as

$$(x_1, \ldots, x_n) \to (h(x_1), \ldots, h(x_n)). \tag{2.6}$$

Using this threat model, they define the ReColorAdv attack that transforms input colors to output colors. ReColorAdv regularizes the function h such that similar input colors result in similar output colors as well. Thereby, the neighbouring pixels in their adversarial examples tend to have similar colors, which leads to them looking very natural, as illustrated in fig. 2.4.

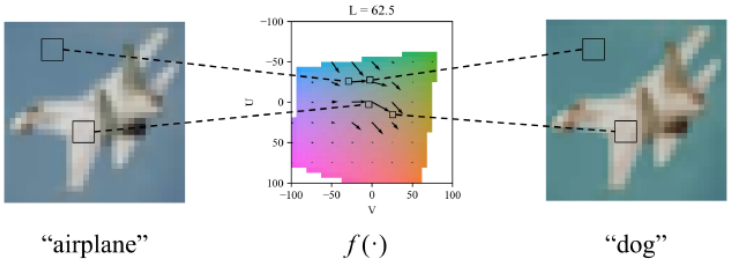


Figure 2.4: Adversarial Example created with the ReColorAdv. Figure from [57].

Applications and Implications

Applying adversarial attacks in a physical setting, where the input to the classifier can not be controlled directly, seems like a difficult task as the success depends on the camera's or, in general, the sensor's properties such as the resolution or the viewing angle. Correspondingly, Lu et al. [61] show that viewing the adversarial image from a different angle or distance can decrease the success rate significantly using the example of autonomous vehicles. They print and attach adversarial stop signs to real-world stop signs and try to detect them using a YOLO object detector [79]. They find that the adversarial images "*cannot reliably fool object detectors across a scale of different distances and angles*" [61] and state that "*we might not need to worry about it in many real circumstances*" [61]. They use simple gradient ascent methods like FGSM and BIM for their experiments. Both perturb the image by applying an adversarial noise to the whole image. They will be explained in detail in section 2.1.4. However, it has been shown that other adversarial attacks can indeed be successful in such physical settings. As explained above, Eykholt et al. [27] show that applying adversarial stickers to stop signs can successfully fool classification models, and their attacks worked for a variety of viewing angles and distances. In [26], Eykholt et al. extend their work by applying the attack on object detectors. They include epoch-wise randomly chosen object rotation and position in their attack to make the adversarial images more robust against changes in viewing angle and distance changes. They also propose a disappearance attack that tries to avoid that the signs are recognized by the object detector at all.

Another example of an attack that has been applied successfully in a physical setting is given by Kurakin et al. [56]. They print an adversarial image and use a smartphone camera to classify it. Indeed, the attack is successful as the classification was correct

for the original, but not for the adversarial image.[1] In this case, the viewing angle is unchanged, but it still shows that camera properties such as the resolution are not necessarily preventing the failure of the machine learning model and that adversarial attacks can indeed be applied physically. Athalye et al. [6] are able to use 3d-printers to create physical adversarial objects that can often fool IMAGENET [18] classifiers and their attack is robust against changes in viewing distance and angles. So, while the viewing angle and distance can decrease the success of some attacks, more sophisticated approaches are still able to fool object detectors and classifiers. In applications where the cost of a single failure is high, such as autonomous driving or medical diagnosis, the existence of adversarial examples can significantly compromise the safety of using machine learning models. For autonomous driving, there is a lot of further research on adversarial attacks: For example, Deng et al. [19] analyze multiple attacks and defenses when applied to autonomous driving models that computes the steering angle based on camera images. Kong et al. [52] propose an attack that is aimed at fooling autonomous steering systems. The attack perturbs advertising posters which are then printed and physically mounted in a real-world scenario. The computed steering angles differ significantly between benign and adversarial samples. Cao et al. [11] analyze the security of LiDAR based perception systems of autonomous vehicles.[2] They perform an attack that creates spoofed obstacles and leads to an emergency break by the simulated autonomous vehicle. In [12], Cao et al. create adversarial objects in a simulation that resulted in LiDAR sensors measuring point clouds that prevented the object from being detected by the ML model.

Another application where adversarial examples can have severe consequences is medical diagnostics. Several papers have focused on adversarial attacks on medical machine learning models. Finlayson et al. [29] show an example where an original image of a melanocytic nevus is classified as benign, while an indistinguishable adversarial example is classified as malignant, and discuss the implications of adversarial examples for medical diagnostics. Ma et al. [63] state that adversarial attacks are even easier to create in the medical domain. They argue that this is because of the characteristics of medical images, where small changes can change the model's prediction significantly. However, they argue that adversarial examples that are small and imperceptible to human observers, could naturally be unable to fool the human medical expert and state that more subtle

[1] A video of their experiment is available at https://www.youtube.com/watch?v=zQ_uMenoBCk.
[2] Light Detection And Ranging (LiDAR) is a sensor that measures the time of flight a laser needs to travel from the sensor to an object and back. Usually, the laser is redirected using a rotating mirror. Thereby, the distance of objects around the sensor can be measured and a 3D point cloud can be created [87].

perturbations would be required to also fool the expert. Thus, as long as a human expert is included in the process, an adversarial attack leading to an incorrect diagnosis is seen as unlikely. Currently, AI systems are only used to support the human expert in his decision such that the expert can still intervene. However, as the accuracy of AI systems as well as their explainability increases and medical experts' trust in the model's predictions increases, this might lead to over-reliance issues [10] which increases the importance of reliable predictions.

The two examples - autonomous driving and medical diagnostics - show that adversarial attacks are not just a problem in theory, but can also be used to cause significant damage in practice. This demonstrates the importance of research in the defence of machine learning models. In both fields of application, the aim is to develop a model that acts like a human would do. This raises the question to what extent neural networks use other information than humans and how this can be harmonized. In the following section, section 2.1.2, we will discuss explanations for the existence of adversarial examples which will lead us to the question of what neural networks learn and how this is different from human perception.

2.1.2 Explanations for the Vulnerability of Neural Nets

Explaining why adversarial examples exist is an important part of research on adversarial attacks. The explanations indicate how neural networks learn and how the learned classifiers are different from human perception. Thus, they allow to find methods that might overcome these differences to make the models more robust and more aligned to the human perception. We will discuss two well-known explanations in the following: The explanation of linearity from Goodfellow et al. [33] and the non-robust features hypothesis from Ilyas et al. [45].

Linearity of Neural Networks

Goodfellow et al. [33] name the linearity of neural networks as an explanation for the existence of adversarial examples. Depending on the activation function, every neuron in a neural network can be a linear component itself, so a neural network is build of a very high number of linear components. Usually, neural networks use activation functions like ReLU that are "*intentionally designed to behave in very linear ways, so that they are*

easier to optimize" [33]. So, while neural networks do not necessarily implicate linear behavior, and "*are able to represent functions that can range from nearly linear to nearly locally constant*" [32, p. 262], the choice of activation functions and the optimization often forces the linear behaviour. For ReLU networks, Hein et al. [39] show that they result in piecewise linear classification functions indeed.

The explanation of linearity now argues that even small perturbations can have a significant impact because of the very high number of dimensions. Goodfellow et al. [33] illustrate this by using an example: Let w be the n-dimensional vector of weights of a linear function and $x' = x + sign(w) \cdot \varepsilon$.[3] The linear function's output is determined by the dot product of inputs and weights. The output for the perturbed input,

$$w \circ x' = w \circ x + w \circ (sign(w) \cdot \varepsilon), \tag{2.7}$$

differs from the output for the original input $w \circ x$ by $w \circ (sign(w) \cdot \varepsilon)$. Assuming an average weight of m, this difference sums up to $nm\varepsilon$. So, even a small perturbation in the input can force a high difference in the output value, if the number of dimensions n is high. Thus, even a small perturbation on images can cause a misclassification.

The explanation of linearity implicates that adversarial examples can also be created in a simple, linear way. Goodfellow et al. designed such an attack, the Fast Gradient Sign Method that builds on the linearity of neural networks. It will be explained in detail in section 2.1.4.

Non-robust features

Another explanation for the existence of adversarial examples has been that neural networks rely on non-robust features for classfication. Ilyas et al. [45] distinguished between useful, robust and non-robust features. A useful feature is one that is correlated with the ground-truth label. In a binary classification problem, where $y \in \{-1, 1\}$, this is defined as

$$\mathbb{E}_{(x,y)\sim D}[y \cdot f(x)] \geq \rho, \tag{2.8}$$

where $f : X \to \mathbb{R}$ is the feature that maps the input space to real numbers.

A robust feature (γ-robustly useful feature) is a useful feature, if the feature remains γ-useful when adversarially perturbed within a set of valid perturbations Δ. This is

[3] The selection of $\delta = \text{sign}(w) \cdot \varepsilon$ maximizes the L_1 difference in the output.

mathematically defined as

$$\mathbb{E}_{(x,y)\sim D}[\inf_{\delta\in\Delta(x)} y \cdot f(x+\delta)] \geq y. \tag{2.9}$$

A non-robust feature, on the other hand, is defined as a useful feature that is not γ-robust for any $\gamma > 0$ though. These non-robust features can explain the existence of adversarial examples, as neural networks are trained to minimize the classification loss and will use any useful feature, independent of whether it is robust or not, and independent of whether the feature is useful or even visible for humans. So, when an adversarial attack perturbs exactly those non-robust features, the classifier can be fooled with only small perturbations. In a discussion[4] following the publication of the paper, the authors state that

> "*adversarial vulnerability can arise from flipping features in the data that are useful for classification of correct inputs.*" [24].

We do not see this explanation as contradictory to the linearity explanation, but as complementary. The explanation of non-robust features is more specific on where the linear behavior of neural networks can be exploited. When many non-robust features are slightly perturbed, this has a big impact on the output due to the linearity.

The existence of non-robust features also explains the transferability of adversarial examples: As similar optimization and loss functions are used, neural networks tend to learn the same non-robust features from the same or similar datasets. Ilyas et al. also argue that the existence of adversarial examples is thus a consequence of the used datasets. They collect datasets of CIFAR10 images that contain either preferably robust or non-robust features, respectively, and find that when training on robust features, the nets tend to be more robust against adversarial attacks. Examples of images that contain robust and non-robust features, respectively, can be found in fig. C.2.

There has been a very active discussion [24][4] on this paper following its publication in which Gilmer and Hendrycks [31] state that Ilyas et al.'s main point is a special case of the fact that a neural network "*latches onto superficial statistics in the data*" [31] that might be unintuitive for humans. As an example, they name a paper from Yin et al. [99] in which the authors analyze model robustness from a fourier perspective. They found that a neural network can be successfully trained when only using high-frequency

[4] https://distill.pub/2019/advex-bugs-discussion/

information that is barely visible to humans. Similar findings have been made by Abello et al. [2] and Wang et al. [96].

These findings suggest that to improve the adversarial robustness, neural networks should use similar features for classification as the humans perception does. If a classifier would use exactly the same, robust features as humans, this would contradict the existence of adversarial examples. We will discuss adversarial defenses that try to force a net to use similar features as humans in section 2.4. The robust features show much more abstract, coarse structures.

2.1.3 Terminology of Adversarial Attacks

Now, we will provide a brief terminology related to adversarial attacks. First, a distinction is made between different settings, which define what knowledge is available about the machine learning model targeted by the attack. In a white-box setting, the attacker has full access to and knowledge about the target model, including its architecture and parameters. The target model is generally defined as the model that is aimed to be fooled by the attacker. In a black-box setting, the target model's structure and its parameters are unknown. It is often assumed that the attacker still has access to the target model's predictions [80]. However, there are query-limited settings where the number of queries on the target model is limited, or label-only setting where only the label of the current classification is known, but probabilities are not [44]. In this work, we will generally assume a black-box setting with no knowledge about the target model and where the attack is not able to react to its predictions: The attack itself is usually performed on a ResNet [38], and then the success rate is measured on both the same ResNet (white-box-setting) and some (partially defended) DenseNets [42]. The success rate on the DenseNet is also referred to as transferability. We train both nets on the same, full dataset for CIFAR10 [53] or use pretrained models for IMAGENET [18]. The net on which the images are crafted, the ResNet in this case, is called the source model. In the black-box setting, this model is also called substitute [71] or surrogate [80] model. In the gray-box setting, which is not considered in this work, some knowledge about the target model, such as its archtitecture, is available, while other information, such as its weights, is not [80].

As mentioned before, we will limit our work to attacks that perturb the whole images using the additive threat model. They can be divided into two types: Maximum-confidence and minimum-norm attacks. Maximum-confidence attacks are adversarial attacks that

maximize the confidence of the misclassification given a budget ε of allowed perturbation measured with distance metric D [75]. Usually, L_p norms are used as distance metric. The L_p-distance between two images x, x' is defined as[5]

$$||x - x'||_p = (\sum_{i=1}^{n} |x_i - x'_i|^p)^{\frac{1}{p}}. \tag{2.10}$$

The perturbation budget is often described as an L_p-ball around the original image. In contrast, minimum-norm attacks try to find a misclassified image with the smallest perturbation [75]. Often, the confidence c with which the image has to be misclassified can also be determined, which is useful especially in black-box settings. The confidence is usually computed as the difference in logits of the ground-truth class and the highest logit of an incorrect class (in an untargeted setting). More details will follow in section 2.1.5. Figure 2.5 compares the settings of maximum-confidence and minimum-norm attacks. While minimum-norm attacks aim at finding a minimal perturbation that is leading to a wrong classification, maximum-confidence attacks try to maximize the model's loss while staying inside the L_p-ball. Maximum-confidence attacks are therefore easy to implement and usually very efficient, since they only have to maximize the model loss while staying inside the L_p-ball, which can easily be accomplished by selecting an appropriate step size or by clipping the image back inside the L_p-ball. It is noteworthy that the ideal adversarial image does not always have to be on the bounds of the L_p-ball, as is the case in the simplified 2D-illustration.

2.1.4 Maximum-Confidence-Attacks

Now, we will explain some well-known maximum-confidence attacks [75] that maximize the confidence of the misclassification given a budget ε of allowed perturbation measured with distance metric D. Here, we will only describe attacks that originally use RGB images. Our proposed JPEG versions will be introduced later in chapter 3.

Fast Gradient Sign Method

The Fast Gradient Sign Method (FGSM) was proposed by Goodfellow et al. [33]. The attack exploits the explanation of linearity of neural networks by performing a single

[5] For the L_0-norm, assume $\frac{1}{0} = 1$, such that the number of different entries is measured.

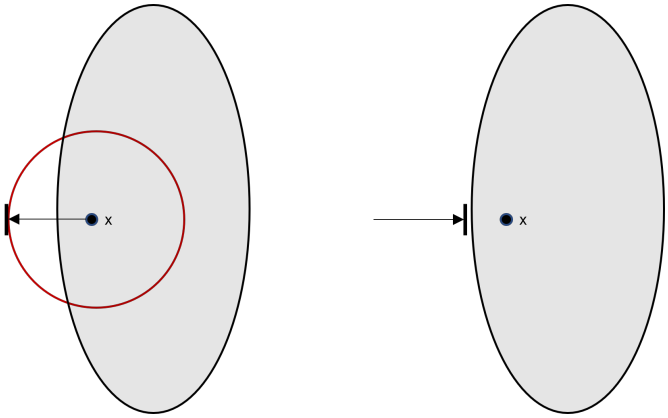


Figure 2.5: Illustation of maximum-confidence (left) and minimum-norm (right) attacks. The shaded areas represent the correctly classified input spaces, the red circle visualizes the L_p-ball, the arrows direct towards the attacks' targets.

step of gradient ascent on the model's loss gradients. Thereby, the source model's loss[6] is maximized and the image potentially misclassified. The potential adversarial example x' is obtained by

$$x' = x + \varepsilon \cdot \text{sign}(\nabla_x \, \text{J}(x, y)) \tag{2.11}$$

for original RGB image[7] x, ground-truth label y and step size and perturbation bound ε. The equation above defines the untargeted version. To perform a targeted attack, the loss would have to be computed for the target label y_t and would be descended by reversing the sign of the perturbation. In this case, every pixel value is altered by ε, the attack is therefore limited by the L_∞ norm that is used as distance metric D such that

$$||x' - x||_\infty \leq \varepsilon. \tag{2.12}$$

All the maximum-confidence attacks covered in this work, usually use the L_∞-norm.

FGSM is a simple and very effective attack but often leads to more perturbation than needed to force an incorrect classification as it always uses the full perturbation budget ε and, as only one gradient descent step is performed, the output is unlikely to be close to the loss function's optimum but given its simplicity it is surprising how successful the attack has been and

[6] usually, crossentropy is used
[7] For RGB images, we generally assume the pixel values to be in the interval $[0, 255]$.

> "*the fact that these simple, cheap algorithms are able to generate misclassified examples serves as evidence in favor of our interpretation of adversarial examples as a result of linearity*" [33].

Basic Iterative Method

Kurakin et al. [56] proposed an iterative version of FGSM, called Basic Iterative Method (BIM)[8]. Starting from $x'_0 = x$, the image gets perturbed by a smaller step size α repeatedly for T iterations by

$$\begin{aligned} x'_t &= x'_{t-1} + \alpha \cdot \text{sign}(\nabla_{x'_{t-1}} \text{J}(x'_{t-1}, y)) \\ x'_t &= \min(x + \varepsilon, \max(x - \varepsilon, x'_t)), \end{aligned} \tag{2.13}$$

where the first equation does the perturbation and the second clips the image back inside the L_∞-ball. In the original paper, they use $\alpha = 1$ and $T = \min(\varepsilon + 4, 1.25\varepsilon)$ to be able to reach the edge of the L_∞-ball but also keep the number of iterations small enough to receive a computationally efficient attack. Our selection of parameters will be detailed in the experimental setup in section 4.1.

BIM is known to be a very successful attack that uses the neural net's linearity in the same way that FGSM does but for multiple iterations. Because it does multiple iterations it usually gets closer to the loss function's optimum and does not perturb the image as much as FGSM does. However, this can also lead to overfitting to the source model which leads to great performance in the white-box setting but is often less successful than FGSM in the black-box setting as found by Kurakin et al. [55].

Momentum Iterative Fast Gradient Sign Method

To improve the transferability on black-box models Dong et al. [21] proposed the Momentum Iterative Fast Gradient Sign Method (MI-FGSM). The attack is very similar to BIM as it iteratively perturbs the image by a small step size α with the only difference that the direction of the perturbation is not determined by the gradient's sign,

[8] The attack is also known as Projected Gradient Descent (PGD) [65]

but by the sign of the momentum g_t to prevent overfitting on the source model and force a higher transferability. The momentum is computed by

$$g_t = \mu \cdot g_{t-1} + \frac{\nabla_x \mathrm{J}(x'_{t-1}, y)}{L_1(\nabla_x \mathrm{J}(x'_{t-1}, y))} \tag{2.14}$$

in every iteration $t = 1, \ldots, T$. The authors recommend to use $\mu = 1.0$ as the decay factor, which we follow. The perturbation itself is then applied by

$$x_t = x'_{t-1} + \alpha \cdot \mathrm{sign}(g_t). \tag{2.15}$$

The authors show that, in dependence of the perturbation budget ε, MI-FGSM indeed outperforms the Basic Iterative Method in the black-box setting as it prevents overfitting to the source model, as explained above [21].

2.1.5 Minimum-Norm-Attacks

Now, we will discuss minimum-norm attacks that try to find an adversarial image with a minimal perturbation that can be measured by any distance metric D. However, we will only cover attacks that minimize L_p norms of RGB representations in this section. In section 2.3, another minimum-norm attack that uses a perceptual color model will be described.

Note that all maximum-confidence attacks described above can be transformed to a minimum-norm attack by performing a binary search over ε in some interval $[\varepsilon_{\min}, \varepsilon_{\max}]$.

Carlini-Wagner Attacks

Carlini and Wagner [13] propose attacks that minimized the L_0-, L_2- and L_∞-norms. We will focus on the L_2-attack as it is their most popular attack and most suitable to minimize the perceptual distance compared to the L_0- and L_∞-norms. The Carlini-Wagner L_2 - Attack (C&W-L_2) was initially proposed as a targeted attack. However, we will describe the untargeted version here.

They formulate the optimization problem as[9]

$$\begin{aligned} &\text{minimize} \quad D(x, x+\delta) + c \cdot f(x+\delta) \\ &\text{subject to} \qquad x_i + \delta_i \in [0,1] \, \forall i. \end{aligned} \tag{2.16}$$

D is one of the distance norms mentioned above, $c > 0$ is some constant that is determined in a binary search during the attack and $f(x)$ is some objective function that uses the neural net's prediction for input x and meets the condition that $f(x+\delta) \leq 0$ if and only if $C(x+\delta) = t$. They propose and analyze a number of functions but decide to use

$$f(x) = (\max_{i \neq t}(Z(x)_i) - Z(x)_t)^+ \tag{2.17}$$

in the targeted version, where $Z(x)_i$ is defined as the model's logit for class i. The objective function basically checks whether the target class's logit is the highest and, if not, returns the difference. In the untargeted version, f has to be adopted such that $f(x+\delta) \leq 0$ if and only if $C(x+\delta) \neq y$. This can be accomplished by just replacing the target class t by ground-truth label y and swapping minuend and subtrahend:

$$f(x) = (Z(x)_y - \max_{i \neq y}(Z(x)_i))^+ \tag{2.18}$$

In fact, they use a slightly modified version of this attack function, as they add a confidence parameter κ that allows them to control the difference in logits required for the image to count as adversarial. Especially in the black-box setting, this is a very important extension because otherwise the adversarial images would be perturbed just enough to be misclassified on the source model, which leads to barely any success on black-box models. This confidence parameter is thus used in most minimum-norm attacks ever since. The modified version of the objective function is then defined by

$$f(x) = \max(Z(x)_y - \max_{i \neq y}(Z(x)_i), -\kappa) \tag{2.19}$$

in the untargeted case.

Carlini and Wagner [13] use tanh to eliminate the box-constraint from the initial optimization problem by defining the perturbation

$$\delta_i = \frac{1}{2}(\tanh(w_i) + 1) - x_i. \tag{2.20}$$

[9] In difference to our usual notation, this attack assumes pixel values between 0 and 1.

The optimization is then performed over w. Then, $x_i + \delta_i \in [0, 1]$ is always valid since $-1 \leq \tanh(w_i) \leq 1$. With this change, the authors were able to use optimization functions that do not support box-constraints such as the Adam [50] optimizer.

While C&W-L_2 guarantees success in the white-box setting when doing enough inner and outer iterations and is also transferable to black-box models using the confidence parameter, the optimization process is very time-consuming: In their experiments, Carlini and Wagner [13] search for the optimal c in 20 steps of binary search. For every c, they perform 10 000 steps with the Adam optimizer. Although the optimizer's number of iterations can be decreased with the attack still being very successful, the binary search over c still results in a very high runtime.

Decoupled Direction and Norm Attack

As explained in section 2.1.3, performing maximum-confidence attacks is usually easier than performing minimum-norm attacks, as the condition that the image lays within the L_p-ball can usually be satisfied by simply projecting the image back into the L_p-ball. For minimum-norm attacks, the condition that the image is adversarial is dependent on the neural network's output and cannot be easily forced. Therefore, minimum-norm attacks usually use a penalty in the optimization function instead of a condition, as we have seen for both L-BFGS and C&W-L_2. However, finding the optimal weight for the image's distance to the original and the model's loss is difficult and it varies at lot between images, as shown in [81], such that it requires time-consuming image-wise search, such as the binary search in C&W-L_2.

Thus, Rony et al. [81] propose the Decoupled Direction and Norm Attack (DDN). The attack does not impose a penalty on the optimization function but instead constraint the image by projecting it into the L_2-ball, similar to BIM, which uses L_∞ though. The second, and more important difference to BIM is that the allowed perturbation is adjusted in every iteration, depending on whether the current image is adversarial or not, as visualized in fig. 2.6. When the current image is adversarial, the current epsilon is decreased by

$$\varepsilon_t = (1 - \gamma) \cdot \varepsilon_{t-1}, \tag{2.21}$$

otherwise, it is increased by

$$\varepsilon_t = (1 + \gamma) \cdot \varepsilon_{t-1} \tag{2.22}$$

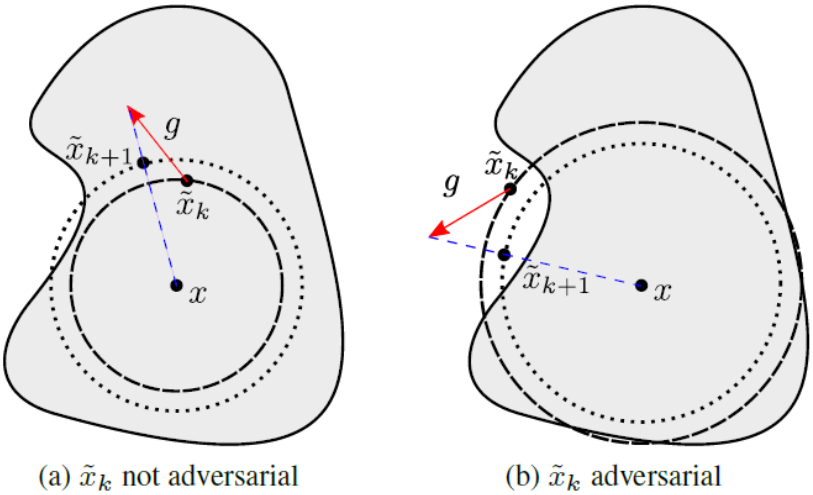


Figure 2.6: Illustration of the DDN attack [81]. The gray area denotes the correctly classified input space. When the current image is not adversarial, the norm is increased (a). When it is adversarial, the norm is decreased (b).

for some factor γ.

The image perturbation itself is then performed by updating the current δ with

$$\delta_t = \delta_{t-1} + \alpha \cdot \frac{\nabla_{x_{t-1}} \mathrm{J}(x_{t-1}, y)}{||\nabla_{x_{t-1}} \mathrm{J}(x_{t-1}, y)||_2} \tag{2.23}$$

and projecting the image back into the current L_2-ball using

$$x_t = x + \varepsilon_t \cdot \frac{\delta_t}{||\delta_t||_2}. \tag{2.24}$$

Out of all images that were adversarial, the attack returns the one with the smallest L_2 norm after T iterations. Whether an image is adversarial or not, can be determined using the confidence function from eq. (2.19), such that the adversarial images will be misclassified by the source model with some confidence κ. In their experiments, Rony et al. find that DDN results in perturbations of similar size as C&W-L_2 does, measured using the RGB L_2 distance, but in much fewer iterations, which allows the attack to be used in adversarial training [81].

The attacks mentioned here perform perturbations in the images' RGB pixel representation and are also limited by L_∞ norms or try to minimize the L_2 norm. Attacks

that partially perturb in frequency space or are related to JPEG compression, or try to minimize a perceptual distance, will be explained in the following sections.

2.2 JPEG Compression

In general, image compression algorithms aim to separate relevant from irrelevant and redundant information which can the be removed to reduce the amount of data. Plataniotis and Venetsanopoulos [76] distinguish between three types of compression algorithms: Lossless compression, lossy compression and perceptually lossless compression. While JPEG is a lossy compression algorithm that can lead to heavily perceptible alterations, especially for low JPEG qualities, it also uses some ideas from perceptually lossless compression algorithms to make the alterations as subtle as possible. It tries to remove both spatial redundancy that is described as the "*correlation among neighboring pixels*" [76, p. 280] and observable redundancy, i.e. the part of "*the visual data that is irrelevant from a perceptual point of view*" [76, p. 280]. In general, perceptually lossless compression algorithms "*make use of the properties of the human visual system to improve further the compression ratio*" [76, p. 281]. JPEG uses two properties of the human visual system to make the compression loss acceptable: First, due to the distribution of rods and cones on the retina, humans are more responsive to brightness than to color changes as the brightness perception is of higher resolution [76]. This is because the information is processed by different neural channels and these channels "*differ in their sensitivity to spatial patterns (known as spatial contrast sensitivity)*" [64], with the luminance channel having the highest sensitivity [64]. Thus, chrominance information is (usually) downsampled in JPEG compression. Second, due to the different resolution on spatial frequencies, JPEG removes information from some frequencies that might be less important for human perception. Plataniotis and Venetsanopoulos state in relation to the perceptually lossless compression that "*an appropriate frequency weighting scheme can be introduced during the encoding process*" [76, p. 281], which is very similar to what the lossy JPEG compression algorithm does.

We will now explain the processing steps of JPEG compression in detail in section 2.2.1 before discussing work on adversarial attacks that is related to JPEG compression in section 2.2.2.

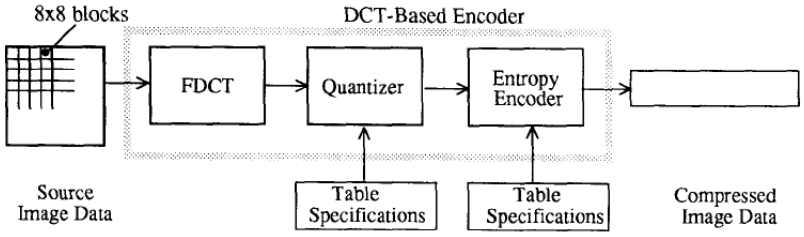
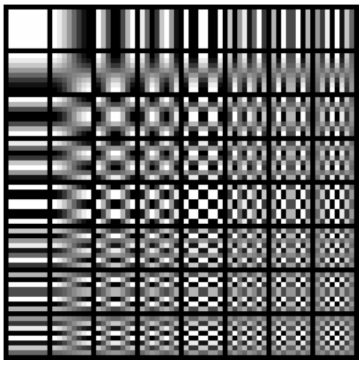
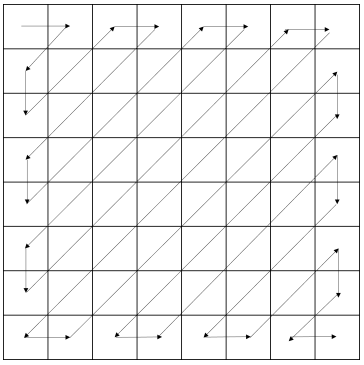


Figure 2.7: Processing steps of a JPEG encoder [95]. These steps are repeated for each channel in a YC_bC_r image.



(a) The DCT's two-dimensional spatial frequencies [34].



(b) Zig-Zag processing of a 8×8 coefficient matrix.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

(c) Indices of Coefficients after the zig-zag ordering.

Figure 2.8: DCT frequencies and coefficients.

2.2.1 Processing Steps

The JPEG compression algorithm consists of 7 steps. From the third step onwards, they are visualized in fig. 2.7. See [95] for details on each step.

1. In general, the JPEG standard does not specify which color space to use. All channels are processed separately and can be recombined after decoding them. However, usually the YC_bC_r color channel is and should be used as its channels are uncorrelated and thus, well suited for separate processing [76]. Therefore, the RGB pixels are first transformed to YC_bC_r pixels.

2. The color channels are usually downsampled by a factor of 2 since the human perception is more sensitive to brightness than to color changes [64, 76], as stated above. The chroma subsampling is the first step of data reduction. There are also versions of JPEG that do not use chroma subsampling.

3. Each channel is then divided into blocks of size 8×8.

4. For each block, the forward discrete cosine transform (DCT) [3] is performed. The DCT transforms the 64 pixel values into 64 coefficients that are amplitudes for cosine functions with 64 unique two-dimensional spatial frequencies [95]. The spatial frequencies are illustrated in fig. 2.8a. The DCT "*lays the foundation for achieving data compression by concentrating most of the signal in the lower spatial frequencies*" [95] as high-frequency coefficients which often have value 0 can be encoded very efficiently later in the compression algorithm.

5. The coefficients are then quantized using some 8×8 quantization matrix. Assuming that $F(u, v)$ with $u, v \in \{1, \dots, 8\}$ are unquantized coefficients, the quantized coefficients are computed by

$$F^Q(u, v) = \left\lfloor \frac{F(u, v)}{Q^{jq}(u, v)} \right\rceil, \tag{2.25}$$

   where Q^{jq} is the quantization matrix for JPEG quality jq [95] and $\lfloor x \rceil$ rounds x to the nearest integer. Usually, the values in the quantization matrix for high-frequency coefficients are much higher than for low-frequency coefficients. This is because human perception is less sensitive to changes on high frequencies and thus, high-frequency information can often be removed. The quantization matrix is dependent of the JPEG quality. A low JPEG quality results in higher values in the matrix and thus, smaller resolution. Usually, one quantization matrix is used for the luma (Y) channel, while another is used for both chroma (C_b, C_r) channels. The JPEG standard does not specify which quantization matrices to use for which quality in general. We use quantization matrices from Tensorflow [1].[10] The tables 2.1 and 2.2 show an example of a quantization matrix for each the luma and the chroma channels. In combination with the entropy encoding (the last step), this irreversible step accounts for the majority of the data reduction [76, 95].

[10] We save images using Tensorflow's *tf.io.encode_jpeg*-function and extract the quantization matrices using torchjpeg [23].

6. The current 8×8 coefficient matrix is then transformed to a sequence of length 64 by performing a zig-zag reordering which is shown in fig. 2.8. This step orders the coefficients by its frequency such that high spatial frequencies are placed towards the end of the sequence. As high frequencies often have an amplitude of 0, especially after quantization, the zig-zag ordering leads to a long sequence of zeros at the end of the sequence.

7. This sequence can then be used by an entropy encoder, which can now save the number of zeros instead of individual numbers and thus save the sequence efficiently. Usually, Huffman coding is used in this step which also just saves the difference between coefficients instead of their actual values, since their difference is represented by much smaller values [76]. However, as we will work on coefficients of step 6, we will not discuss the entropy encoding in detail.

8	6	5	8	12	20	26	31
6	6	7	10	13	29	30	28
7	7	8	12	20	29	35	28
7	9	11	15	26	44	40	31
9	11	19	28	34	55	52	39
12	18	28	32	41	52	57	46
25	32	39	44	52	61	60	51
36	46	48	49	56	50	52	50

Table 2.1: Luminance quantization table: JPEG quality 75, from Tensorflow [1].

9	9	12	24	50	50	50	50
9	11	13	33	50	50	50	50
12	13	28	50	50	50	50	50
24	33	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

Table 2.2: Chrominance quantization table: JPEG quality 75, from Tensorflow [1].

2.2.2 JPEG-resistant Adversarial Attacks

There have been some attacks that already use JPEG or try to bypass JPEG compression, which is a technique to defend against adversarial attacks. More information on JPEG compression as an adversarial defense technique will follow in section 2.4.

Kang et al.'s JPEG attack

Kang et al. [48] were the first to adversarially perturb JPEG coefficients. They were analyzing the robustness of adversarially trained nets against unforeseen threat models

including a JPEG attack in which they perturb unquantized coefficients by the L_∞-constrained BIM, quantize them and then transform the coefficients to RGB pixel values. They find the attack to be very effective against the adversarial training nets that were trained with other threat models, e.g. RGB L_∞ attacks. However, they do not explain how they weight perturbations across frequencies and thus, we assume they perturbed all coefficients by the same, absolute value. Accordingly, they also did not analyze the effect of weighting perturbations differently across frequencies which is one of the major motivations for this work.

Shin & Song's JPEG-resistant adversarial attack

Shin and Song [86] propose a method that still perturbs the image's RGB representation but tries to bypass JPEG compression in defense by including an approximation of JPEG compression in the target model. The perturbation of their FGSM variant is defined by

$$x' = x + \varepsilon \cdot \text{sign}(\nabla_x \, \text{J}(JPEG^{jq}_{\text{approx}}(x), y)), \tag{2.26}$$

where $JPEG^{jq}_{\text{approx}}(x)$ is an approximation of JPEG compression for quality jq that receives and returns RGB data. In fact, only the rounding step to the nearest integer during the quantization step is replaced by the differentiable operation

$$\lfloor x \rceil_{\text{approx}} = \lfloor x \rceil + (x - \lfloor x \rceil)^3. \tag{2.27}$$

Including the approximation of JPEG compression significantly increases the attack's success on models defended with JPEG compression. However, the perturbation itself is still applied on RGB images.

As they find that their attack overfits to the JPEG quality used in attack and thus, is often not transferable when other JPEG qualities are used in defense, they also propose an ensemble attack that combines the gradients for multiple JPEG qualities by

$$g_{\text{ens}} = \sum_i (1 - \frac{\exp(L_i)}{\sum_i \exp(L_i)}) \nabla_x L_i, \tag{2.28}$$

where $L_i = \text{J}(JPEG^{q_i}_{\text{approx}}(x), y)$ is the loss for JPEG quality q_i. Indeed, the ensemble attack leads to better generalization [86].

Shi et al.'s Adversarial Rounding Attacks

An approach that receives RGB images as input but returns JPEG coefficients was recently proposed by Shi et al. [85]. Their attack first executes an RGB attack. They use FGSM or BIM but it can be replaced by any RGB attack. Then, it converts the RGB pixels to intermediate DCT coefficients that are already divided by the quantization matrix, but not rounded. To return rounded coefficients, they propose a rounding scheme that aims at rounding each coefficient in the direction that maximizes the attack's success without leading to large perturbations instead of just rounding to the nearest integer. In untargeted attacks, they only use a fast adversarial rounding scheme in which they first compute the coefficient's gradients on the source model's loss function, and round every coefficient, where the nearest integer is consistent with the gradient's direction, in that direction. In a second step, they measure the importance of the remaining intermediate coefficients F^i for the attack's success using the update gradients g' as

$$\tau = \frac{|g'|}{((f^+ - f^-) \cdot q)^2}, \tag{2.29}$$

where q is the entry of the Quantization matrix Q and $f^+ = \lceil f^i \rceil - f^i, f^- = f^i - \lfloor f^i \rfloor$. Then, they round the $\eta \cdot 100$ percent coefficients with the highest importance value τ in the gradient's direction, while the others are set to the nearest integer. For targeted attacks, they also propose an iterative rounding scheme which will not be detailed here.

Despite showing good performance compared to traditional RGB attacks, the majority of the perturbation is still performed inside the RGB attack and only the quantization step is made straight on JPEG coefficients.

2.3 Perceptual Metrics

With adversarial examples ideally looking indistinguishable from original images, measuring perceived difference between images is an important part of research on adversarial attacks. While for a long time only L_p norms, measured in the images' RGB representations, were used to measure the perturbations and, as explained in section 2.1, minimize the distance made by an attack, research has developed into using metrics that might be more suitable for measuring human perception. In section 2.3.1, we will explain how suitable RGB L_p norms are to measure human perception. Alternatives are explained in section 2.3.2, followed by a summary on how the suitability of a perceptual metric can be quantified in section 2.3.3. Finally, section 2.3.4 will be about adversarial attacks that already use perceptual metrics.

2.3.1 The Suitability of RGB L_p-Norms as Perceptual Metrics

The reason for RGB L_p-norms being used as a distance measure and limit for adversarial perturbations is mainly its simplicity: Perturbations can be easily controlled and projected back into the L_p-ball and. In a real-world scenario, though, the suitability of RGB L_p norms, and L_p norms in general, can be doubted. Adversarial examples are defined as images with ideally imperceptible perturbations from the benign samples. Therefore, an ideal distance metric that is used for attacks, but also for comparing attacks, would be as close as possible to the human perception. While initially the focus has been on the pixel-wise L_p-norms as a distance metric for adversarial attacks, there has been some research on the suitability of L_p-norms and possible alternatives.

Sharif et al. [83] have analysed the suitability of RGB L_p-norms for adversarial images. They conducted user studies in which images from L_0-, L_2- and L_∞-attacks were examined by letting the participants classify the images. For all three norms, they found that a small L_p distance is not necessary for perceptual similarity and does also not imply it such that there can be images with a big L_p distance that are perceptually very similar, but there also are examples of images with a small L_p distance that look strongly disturbed. As a suitable example for a small L_p distance not being a necessary condition for perceptual similarity, they name geometric transformations which tend to have very high L_p distances as they use pixel-wise differences. Figure 2.9 shows an example, where an IMAGENET sample image was transformed in four different ways. The one that was shifted to the left and filled with the nearest pixels on the right, looks the most similar to

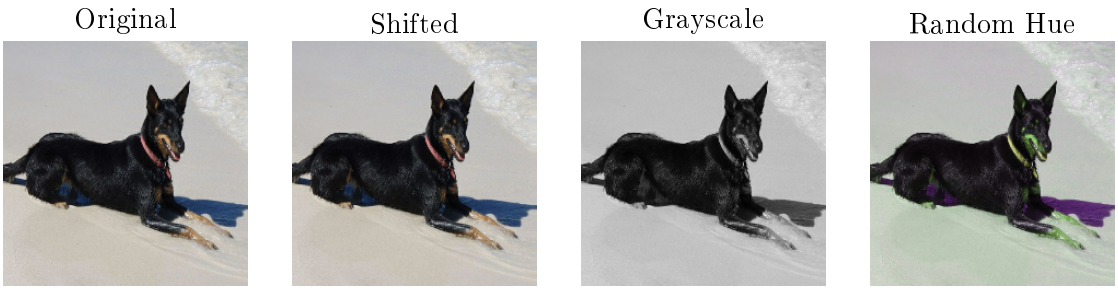


Figure 2.9: An IMAGENET [18] sample is transformed by shifting the image to the left, transforming it to grayscale by setting all three channels to the average of the RGB-channels, and applying a random hue transformation. The numbers below the images are the L_2 distances from the original image, measured in the RGB representation.

the original images, but has by far the highest L_p distance. Geometric transformations are a great example for the unsuitabilty of L_p-norms and it has been shown that they can also be used for successful adversarial attacks [25].

An example for the unsuitability of RGB L_p-norms that is much closer to what is looked at in this work, since it is about noise, is shown in fig. 2.10. The right and middle image have the same L_p-distance from the original image, as both have been perturbed by the same noise but in different channels. The right image looks indistinguishable from the original, while the perturbations in the middle image are very obvious, what illustrates the unsuitability of RGB L_p norms for measuring human perception.

One reason for RGB L_p norms not successfully measuring human perception is that RGB is not based on the human perception, but on physiological properties, namely the three types of cones in the human retina: L (red), M (green) and S (blue) [76]. This makes RGB well-suited for displays, as it is an additive color model and "*the perceived red, green, and blue intensities are approximately related*" [76, p. 20] to the values of the three channels, when using the non-linear RGB space. But RGB

> "*isn't perceptually uniform, meaning that one unit of coordinate distance doesn't correspond to the same perceived color difference in all regions of the color space*" [64].

Thus, differences in RGB pixel values, as used in the L_p-norms, are not always related to perceptual differences.

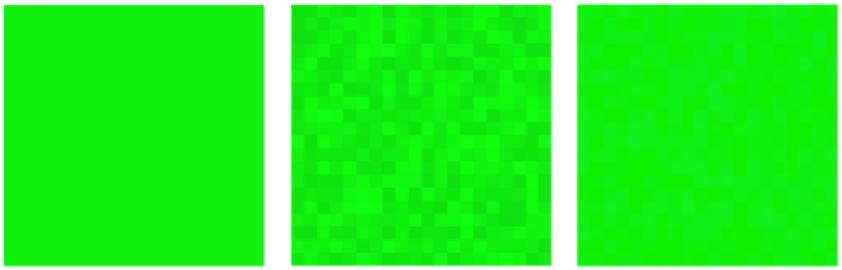


Figure 2.10: Example for the unsuitability of RGB L_p-norms to measure human perception [102]. Left: Original RGB Image. Middle: RGB image perturbed in the G channel by some random noise. Right: RGB image perturbed in the B channel by the same noise. As both are perturbed with the same noise, they have the same RGB L_p distances to the original image.

Additionally, the human visual system uses three neural channels (luminance, red-green, yellow-blue). A corresponding color model would therefore be structured as in fig. 2.11. The model has three pairs of elementary colors and thus, three dimensions: lightness, colorfulness and hue. The RGB color space, though, does not correspond with these three dimensions and thus, with the human perceptual system [64].

Another reason are the L_p-norms itself which use pixel-wise differences leading to a strong concentration on high-frequencies. However, it has been shown that "*the human vision system is not sensitive to high spatial frequencies*" [76, p. 293], such that lower frequencies are more important for human perception. However, the difference on lower frequencies cannot be noted by pixel-wise differences such as the L_p-norms. The same applies to the relationship between neighbouring pixels and regions. For example, a colored foreground looks darker on a white than on a black background [64], as shown in fig. 2.12.

The failure of pixel-based distance metrics for measuring human perception also shows when zooming out of an image. This effect is, for example, visible in hybrid images, proposed by Oliva et al. [70]: Hybrid images are images consisting of two images, where one is located in high- and one in low-frequency space. As high frequencies become less important when increasing the distances, the visible content changes when zooming out. An example of such an image is shown in fig. 2.13.

So, an alternative for RGB L_p-norms should ideally

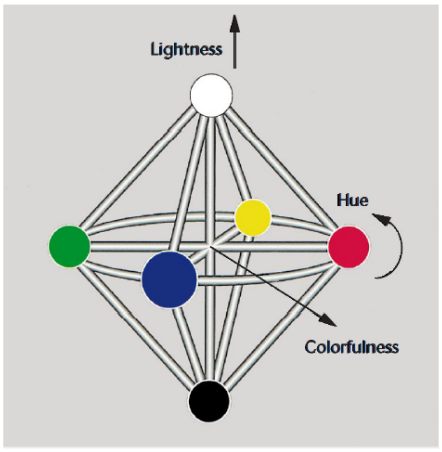


Figure 2.11: A perceptual color model with three axes and six elementary colors. Figure from [64].

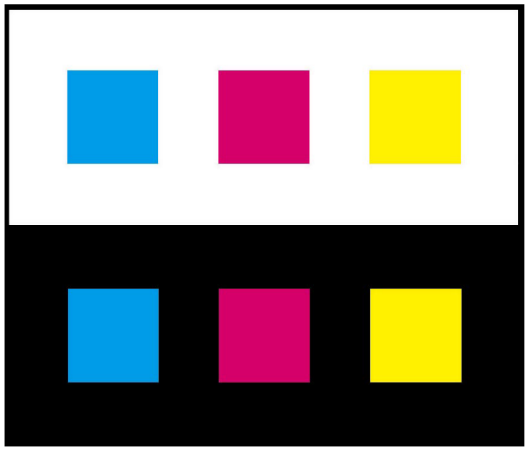


Figure 2.12: A colored foreground looks darker on a white than on a black background. Figure from [64].

1. be able to measure structural and not just pixel-wise differences, since they are focused on high-frequency differences,
2. not use RGB color space, but, if the distance is computed pixel-wise, a more perceptual one.

2.3.2 Alternative Metrics

Due to the limitations of the RGB color model and the pixel-wise L_p-norms in general, research has developed towards evaluating adversarial attacks using more perceptual metrics in recent years [28, 58, 102]. Some alternatives will be discussed in this section.

There are color spaces that are more suitable for measuring the human perception than RGB. One of these color spaces is CIELAB [67], which is a perceptually uniform model. It consists of a lightness (L^*) component and two color axes - a^* that represents the green-red color channel and b^* that represents the yellow-blue color channel from fig. 2.11. As it corresponds with human neural channels, CIELAB is designed and also known to be useful for measuring the human perception. A perceptual color difference formula that is based on CIELAB is CIEDE2000 [62]. CIEDE2000 includes lightness, chroma and hue and is known to be closer to the human perception than the euclidean distance in CIELAB color space [102]. But, both are still pixel-wise and are thus not able to

(a) When zooming out, an elephant gets visible instead of the tiger that is visible from close range.

(b) From close range, the image is perceived as a tiger. When zooming out, a cheetah becomes visible.

Figure 2.13: Two examples of hybrid images. When zooming out, the visible content changes. Both images are from [70].

measure low-frequency differences. Furthermore, similarity metrics such as the structural similarity index measure (SSIM) [97] have been used. SSIM does not correspond to human perception significantly better than the RGB L_2 distance though [101].

Motivated by the fact that pixel-wise distances such as the L_p-norms are not suitable to measure structural differences, perceptual losses have been proposed. They use differences in neural networks' feature spaces to quantify the perceptual distance between images.

The perceptual loss function that will be used in this work was proposed by Zhang et al. [101] and is called Learned Perceptual Image Patch Similarity (LPIPS). An overview of the loss function is shown in fig. 2.14: Two images x, x_0 are passed through a pretrained model F, and the activations $\hat{y}^l, \hat{y_0}^l \in \mathcal{R}^{H_l \times W_l \times C_l}$ of some layers $l \in L$ are unit-normalized in the channel dimension and then used to compute the distance

$$d(x, x_0) = \sum_l \frac{1}{H_l \cdot W_l} \sum_{h,w} ||w_l \odot (\hat{y}^l_{hw} - \hat{y_o}^l_{hw})||_2^2, \tag{2.30}$$

where the weights $w_l \in \mathcal{R}^{C_l}$ determine the importance of each layer and channel for the perceptual loss. They are trained using the right part of fig. 2.14, a model G that takes

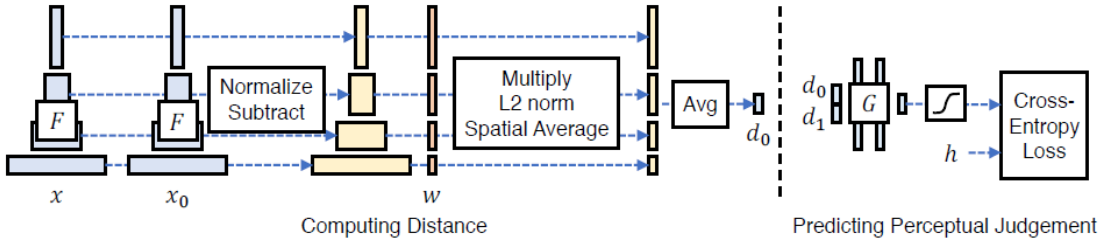


Figure 2.14: Computation of the LPIPS perceptual distance. The part to the left of the dashed line computes the perceptual distance of two images. The part to the right is used to train the weights w.

two distances and consists of two dense layers with 32 neurons, and a dense layer with 1 neuron and sigmoid activation such that it returns some score $\hat{h} \in [0, 1]$. The training itself is then done using their proposed AFC2 dataset, which consists of 3-tuples of images: An original reference image r and two perturbed image patches p_0, p_1. Additionally, each entry contains the percentage of humans that decided that p_1 is closer to the original. By training the model G, they perform a "*perceptual calibration*" [101] of the weights w_l. Zhang et al. experiment with whether the loss network F should be fixed during training. The results vary slightly between tasks. In our experiments, we will freeze F during training.

The use of this perceptual loss is motivated by its ability to better capture structural differences. While perceptual color distances such as CIEDE2000 are limited by their pixel-wise computation, a perceptual loss function can capture structural differences and dependencies such as edges between pixels. Figures 2.15 and 2.16 illustrate the suitability of LPIPS as a perceptual metric. In both figures, the perturbed images contain exactly the same pixel-differences from the reference image but the arrangement varies. As it is a pixel-based distance metric, the CIEDE2000 L_2 values are identical. However, the perceived distance is higher for the images that are perturbed with the structural noise, i.e. the chessboard. In the images perturbed with a chessboard noise the pixel values often vary between neighbouring pixels, especially in fig. 2.16 where the chessboard is applied pixel-wise. Thus, they include many visible edges that are visible to the human observer.

The right image in fig. 2.16 is generated by minimizing the LPIPS distance in 1000 steps of gradient descent. The condition that half of the pixels are perturbed by addition and the other half by subtraction, is enforced every 10 steps. The optimization results in two

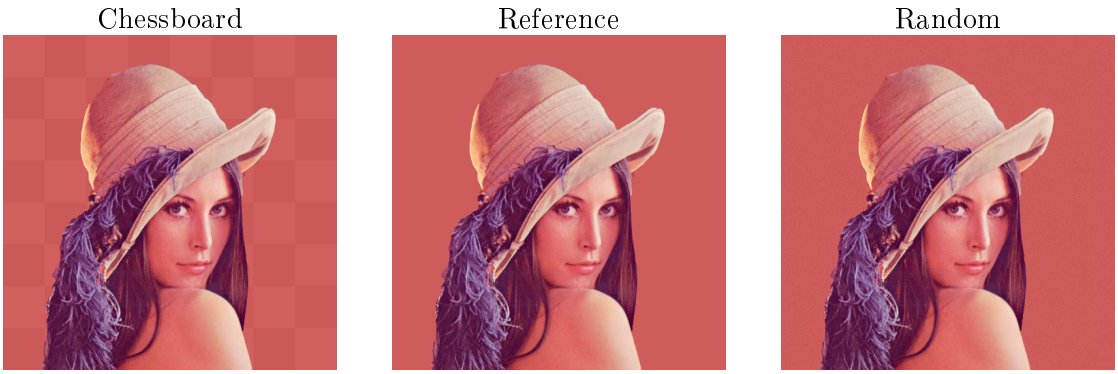


Figure 2.15: The reference image's unicolored background was perturbed with some noise, by adding or subtracting 3 from each RGB channel value. On the left image, the direction of the perturbation is arranged as a chessboard. On the right image, the direction was determined randomly, but half of the pixels are perturbed by subtraction and the other half by addition of 3. Thus, the background contains the same pixels on both perturbed images, but the arrangement varies, which leads to the same CIEDE2000 L_2 distances, but corresponding to human perception, the LPIPS distances vary.

blocks: The lower left of the image is brighter than the upper right. Due to the small number of edges, the image is perceptually very similar to the reference image, which is captured by the LPIPS distance function, but not by the CIEDE2000 L_2 distance. This shows that, unlike pixel-based distance metrics, the perceptual loss (LPIPS) is able to measure such differences and interdependencies between pixels and is thus better suited for measuring human perception.

2.3.3 Evaluating Perceptual Metrics

Perceptual Metrics can also be evaluated quantitatively with regard to their suitability for measuring human perception using the datasets proposed by Zhang et al. [101]: First, they published a two alternative forced choice (2AFC) dataset that was already mentioned in the previous section. Second, they proposed a just noticeable difference (JND) Dataset: The dataset entries consist of two images p_0, p_1, and the study participants decided whether they are the same or not leading to a score, which is the percentage of humans who think p_0 and p_1 are the same. For the evaluation of the datasets, Zhang et al. propose to use the following scores:

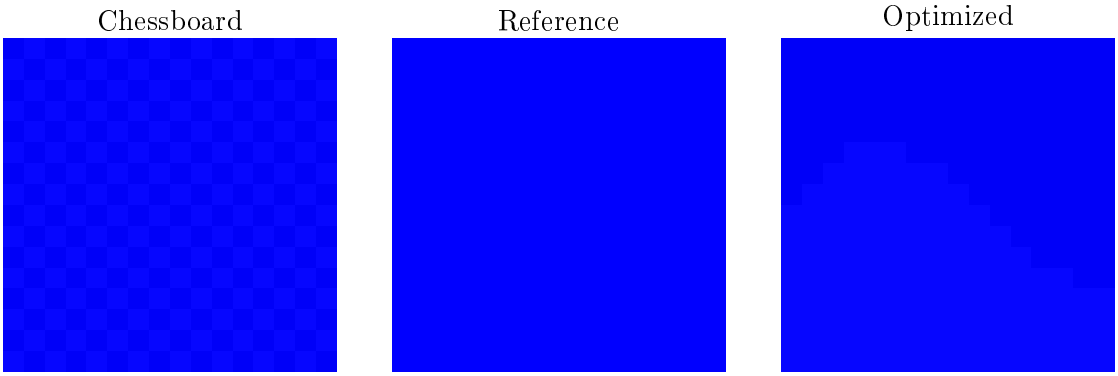


Figure 2.16: A 16x16 unicolored reference image (RGB: 5, 5, 250) was perturbed by adding or subtracting 3 to each channel's value. On the left, the perturbation's direction was determined by a pixel-wise chessboard. On the right, the direction was optimized in a gradient descent that minimized the LPIPS distance, which leads to hardly noticeable uniform areas in which the pixel values are either subtracted or added. Again, the two images contain the exactly same pixels, but in a different order. The CIEDE2000 L_2 distance is thus not able to measure the perceptual distance, but LPIPS is.

For the 2AFC dataset, which is split into a train and a test dataset, a distance metric D can be evaluated by computing the distances $d_0 = D(r, p_0), d_1 = D(r, p_1)$ and scores

$$score = \begin{cases} 1 - s, & \text{if } d_0 < d_1 \\ s, & \text{if } d_0 > d_1 \\ 0.5, & \text{otherwise,} \end{cases} \tag{2.31}$$

where r is the reference image, p_0, p_1 are the perturbed patches, and s is the percentage of participants that decided that p_1 is closer to the original than p_0. The scores can then be averaged across the dataset. For the JND dataset, the evaluation is performed by computing the area under the precision-recall curve.

The perturbed images in Zhang et al.'s dataset are generated by various distortions. Among others, these include traditional distortions such as blur, shifting, uniform white noise, cnn-based distortions from autoencoding or super-resolution networks. However, perturbations from adversarial attacks are not included in the dataset. Thus, although these datasets can be used to measure whether distance metrics are suitable to measure human perception in general, this does not apply to adversarial attacks [101]. A similar

JND dataset for adversarial attacks has been proposed by Laidlaw et al. [58] though. They collect images from various attacks including BIM and JPEG attack from [48].

Zhang et al. [101] analyze the performance of various architectures like VGG16 [90], AlexNet [54] and SqueezeNet [43]. The AlexNet shows the best average performance and clearly outperforms traditional distance metrics such as L_2 or SSIM. However, in our experiments we will use the VGG16 which shows similar performance since an AlexNet pretrained on IMAGENET is not available for Tensorflow [1].

2.3.4 Perceptual Metrics and Adversarial Attacks

In recent years, perceptual color models and perceptual losses have already been used in the context of adversarial attacks. Zhao et al. [102] proposed two attacks that are based on C&W-L_2 and DDN, respectively, but try to minimize the CIEDE2000 L_2 distance instead of the RGB L_2 distance to receive less perceived perturbation. For the reasons already mentioned regarding the C&W-L_2-attack, namely the high runtime due to the binary search, we will focus on their second attack, called PERCEPTUAL COLOR DISTANCE ALTERNATING LOSS (PerC-AL). As in DDN, the step in each iteration is dependent on whether the current image is adversarial or not. In DDN, the allowed perturbation e_t was decreased in case the current image is adversarial. At each iteration's end, the image was projected back into the current L_2-ball. As the image representation and the representation used for the projection was RGB, this could be easily accomplished. Because Zhao et al. [102] want to minimize the CIEDE2000 distance, this step cannot be performed in the same way. Thus, they do not decrease the allowed norm in case the current image is adversarial, but they minimize the CIEDE2000 distance. So, if the current image x'_{t-1} is not adversarial, the perturbation in iteration t is updated by

$$\begin{aligned} g &= \nabla_{x'_{t-1}} \mathrm{J}(x'_{t-1}, y) \\ \delta_t &= \delta_{t-1} + \alpha_l \cdot \frac{g}{||g||_2} \end{aligned} \tag{2.32}$$

and if the image is adversarial, it is updated by

$$\begin{aligned} d &= -||\,\mathrm{ciede2000}(x, x'_{t-1})||_2 \\ g &= \nabla_{x'_{t-1}} d \\ \delta_t &= \delta_{t-1} + \alpha_c \cdot \frac{g}{||g||_2}, \end{aligned} \tag{2.33}$$

where a_l, a_c are step sizes and usually, the step size for minimizing the perceptual color distance should be smaller. They find that their attacks do result in larger RGB L_2 perturbations but are less perceivable.

The LPIPS distance was first used to quantify and evaluate adversarial distortions by Jordan et al. They combine various attacks such as L_∞-attacks and spatial transformations and state that nets defended against a particular style are not robust against unseen threat models [47]. To defend against unseen threat models, Laidlaw et al. proposed a perceptual threat model, which limits the allowed perturbation using the LPIPS distance. We will not explain their attacks in detail as they are not directly related to the idea of attacking JPEG coefficients, but their proposed defenses, which have been found to generalize well to other threat models and thus, got closer to human perception, will be part of our experiments and will be explained in section 2.4.

2.4 Adversarial Defenses

As mentioned before, the existence of adversarial examples prevents the use of machine learning models in safety-critical applications. Thus, research tries to find ways to make models more robust such that the output complies with the human expectation. To achieve this, various defense techniques have been published. Some of these methods try to detect adversarial examples such that the user can be warned and the wrong classification can possibly be prevented [4, 14], which is useful in applications where a human expert can intervene. In this work, we will focus on defenses that try to classify correctly despite a malicious input. The aim of such methods is basically to receive predictions that are as robust and accurate as human predictions. Thus, it would seem obvious that the net should also use similar features/information as humans do. For standard nets this is not the case. It has been shown that they often use non-robust features [45] and rely on high-frequency information that is barely visible to humans [96, 99]. In fact, these two examples are related, as we will show later.

In general, defenses basically try to regularize neural networks such that their predictions aligns with the human perception. In this section, we will discuss some of those approaches. First, we will focus on some simple input transformations in section 2.4.1, then adversarial training will be explained in section 2.4.2. An overview of further defense techniques and an analysis of their robustness can be found in [6].

2.4.1 Input Transformations

Input transformations can alter the input by either removing the adversarial perturbations or making the perturbations less suited for the target model. Das et al. [17] proposed to use JPEG compression to defend against adversarial attacks. As explained in section 2.2, JPEG compression tries to remove information that is imperceptible for human observers. Adversarial attacks aim to make imperceptible perturbations on images. Consequently, it is plausible that at least some of the adversarial perturbation is removed in the quantization step, especially on high frequencies. First, they only carry out the JPEG compression on the input of a pretrained model and find that this does decrease the success rate significantly but can also reduce the accuracy on the benign images. Second, they "*vaccinate*" [17] the neural networks by using JPEG compressed images during training which increases the robustness of their models even more [17]. Similar results have been presented by Dziugaite et al. [22] who have found that small perturbations are often reversed by JPEG compression.

Further input transformations that have been used for defending neural nets include bit-depth reduction that can remove perturbations similarly to JPEG compression, image-cropping, which can change the positioning of perturbations, and image quilting, which is defined as replacing parts of images by patches from benign samples from a dataset [35].

Raff et al. proposed a Barrage of Random Transforms for Adversarially Robust Defense (BaRT) [77] combining multiple weak defense approaches to form a strong one. They state that if the set of transformations is large and they are combined randomly, their method is robust against iterative attacks such as BIM, even for large perturbations. Since it does not require special training, BaRT and other input transformations, are also useful for large datasets, such as IMAGENET.

Out of the input transformation methods, we will only consider JPEG compression in our experiments, since it seems possible that by attacking straight on JPEG coefficients, one might bypass this defense: As JPEG compression mostly removes high-frequency perturbations, concentrating the perturbations on lower frequencies might circumvent the defense.

2.4.2 Adversarial Training

Adversarial training is a very prominent defense for neural networks that adds adversarial examples to the training set. It was first proposed by Goodfellow et al. [33] and can be seen as a regularization technique that

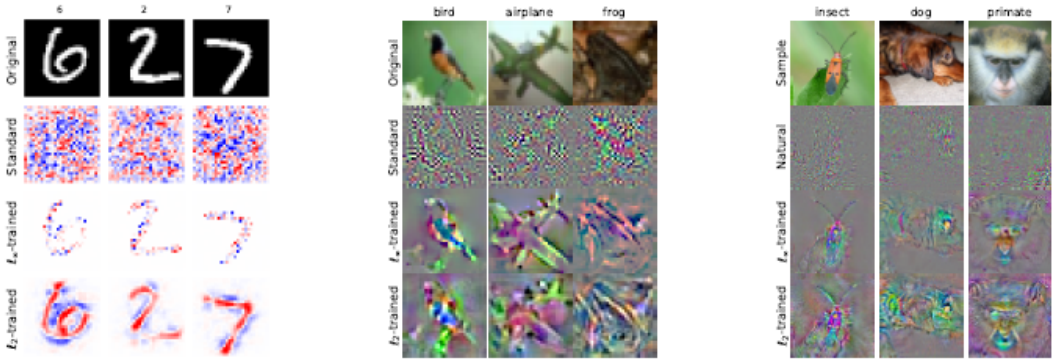
> "*discourages this highly sensitive locally linear behavior by encouraging the network to be locally constant in the neighborhood of the training data.*" [32, p. 262].

In this thesis, we will need two versions of adversarial training.

Madry et al.'s Adversarial Training

Madry et al. [65] proposed to use BIM to create adversarial examples during training. They found that when using random initialization inside the L_∞-ball, the loss often converges to a similar loss value. Thus, they argue that BIM (or PGD) is a universal first-order adversary meaning that "*robustness against the PGD adversary yields robustness against all first-order adversaries, i.e., attacks that rely only on first-order information. As long as the adversary only uses gradients of the loss function with respect to the input, we conjecture that it will not find significantly better local maxima than PGD*" [65]. So, attacking with other first-order attacks like FGSM and MI-FGSM should be little successful if the net is robust against BIM. During the adversarial training, they initialize the adversarial image randomly inside the L_∞-ball and add the adversarial images to the training set.

In some cases, adversarial training can even increase the accuracy on the benign test set. Correspondingly, adversarial training is sometimes seen as a data augmentation method that regularizes the output of neural networks and that prevents overfitting on the training data [33, 32]. However, Tsipras et al. [93] have shown that this effect is mainly apparent when training with few benign samples. When the training data is increased, they show that adversarial training often reduces the accuracy on the benign images. They explain this with the fact that neural networks often classify by using features that are only weakly correlated with the ground-truth label. In a benign setting, using a combination of features that are weakly correlated with the label, which they call a "*meta-feature*" [93], is often sufficient for a correct classification. In an adversarial setting, though, the classifier cannot rely on those weakly correlated or non-robust features,



(a) MNIST (b) CIFAR-10 (c) Restricted ImageNet

Figure 2.17: Visualizations of the loss gradients for various nets trained with standard or adversarial training. The adversarially trained nets tend to use features similar to the human perception. Figure from [93].

as these are usually attacked because they tend to be unrecognizable for humans and have to be perturbed only slightly in contrast to more correlated, robust features. Thus, they argue that "*robustness may be at odds with accuracy*" [93]. This argument is in line with the explanation of linearity from section 2.1.2: A slight perturbation on the input can significantly alter the activations on the weakly correlated features, consequently the meta-feature, and the model's prediction. Note that these weakly correlated features seem to be strongly connected to the non-robust features defined by Ilyas et al. [45]. Additionally, Tsipras et al. show that adversarial training leads to networks that rely on similar features for classifications as humans do. In fig. 2.17, they visualized how adversarial training influences the loss gradients. The figure shows that adversarially trained nets correspond better to human perception than nets trained with original images only. This is in fact a very intuitive observation: Adversarial training tries to make the model robust against perturbations that are not or barely visible for humans. Thus, the net should also rely more on the same features as humans do [93].

Perceptual Adversarial Training

Laidlaw et al. [58] tried to overcome the problem of neural nets that are vulnerable towards unseen threat models by proposing Perceptual Adversarial Training (PAT). They use the LPIPS distance to define the Neural Perceptual Threat Model, where an image

x' is adversarial, if and only if

$$C(x') \neq y \text{ and } \text{lpips}(x,') \leq \varepsilon. \tag{2.34}$$

They propose several attacks for this threat model which will not be covered in detail here. For the PAT, they use the a simplified version of the Lagrangian Perceptual Attack, which is based on the C&W-L_2 and optimizes

$$\max_{x'} \text{J}(x', y) - \lambda \max(0, \text{lpips}(x, x') - \varepsilon). \tag{2.35}$$

The simplified version, FastLPA, which is only used for adversarial training, does not search over λ and does not include a projection step. They find that PAT leads to a better generalization of the model against unseen threat models. As L_p distances can vary a lot between threat models, even if the images look very similar (see fig. 2.9), while the LPIPS distances reflect the peceived distortion much better, the perceptual attacks can lead to perturbations that are similar to those of different threat models and the PAT generalizes better [58].

2.5 Adversarial Attacks and Defenses: A Frequency Perspective

In this section, we will discuss papers that have already looked at attacks and robustness from the perspective of frequencies. They mainly use coefficients from the discrete fourier transform (DFT) or discrete cosine transform (DCT), but no JPEG coefficients itself.

By analyzing how adding noise to each frequency influences the output, Tsuzuku and Sato [94] found that neural networks are sensitive to the directions of fourier basis functions. The sensitivity on each frequency depends on the dataset and the architecture. For Cifar10 most of the tested networks are more sensitive to noise on high and medium frequencies, while for SVHN [69], for example, the low frequencies are most sensitive. They use this observation to propose a universal attack that adds a uniform pattern to the image. Additionally, they analyze where adversarial perturbations are located in the frequency spectrum. When applying the FGSM attack, the perturbations are concentrated in those frequencies that the net is sensitive against. Thus, the perturbations are not necessarily concentrated in high frequencies but it depends on the net's architecture

Figure 2.18: An example for an adversarial attack on low frequencies from [36].

and the dataset. For CIFAR10, it has been observed by Tsuzuku and Sato [94] and others [8, 66] that high-frequency components are usually more important for neural nets. Deng and Karam [20] propose a universal adversarial attack that uses DCT frequency-wise JND thresholds that are based on the sensitivity of the human perception. They find that their attack is more efficient than the baseline method. As it is an universal attack, they do not analyze different distributions of perturbations across frequencies however [20].

Guo et al. [36] have proposed black-box adversarial attacks that are limited to perturbing low spatial DCT frequencies. The perturbation is applied to RGB pixels, but the gradient is converted to the frequency domain using DCT. Then, high frequencies are masked out and the inverse DCT is applied. This leads to colorful blotches in the image, as shown in fig. 2.18. Guo et al. state that their attack can circumvent defenses that build on removing high-frequency components such as JPEG compression. However, they do not do a perceptual evaluation of their attack, as they only use MSE distances.

Sharma et al. [84] analyzed the effectiveness of very similar low-frequency perturbations. Again, the perturbation is applied to RGB pixels. In this case, the δ is transformed to DCT coefficients, then a boolean mask is applied, and the delta is converted to pixel space by the IDCT. This masking is denoted as FreqMask. Then they compute the gradients as $\nabla_\delta J(x + FreqMask(\delta), y)$. They find that perturbing low frequencies is more effective in the defended black-box setting, but not in undefended settings. Again, they do not use a perceptual metric but only the input parameter ε to compare attacks. Especially when attacking different frequencies, it is questionable how meaningful this is as low-frequency attacks yield very different perturbations, as fig. 2.18 and the sample images in [84] show.

In section 2.4, we already mentioned that the explanation of non-robust features [45] and networks relying on high frequencies that are barely recognisable for humans [99] are related. Remember that non-robust features are features that are useful in a benign setting, such that a neural network can and will use them for optimization in such settings. When those features are slightly perturbed, though, they are not positively correlated with the ground-truth label anymore. In the discussion to the paper on non-robust features, Gilmer and Hendrycks [31] state that the main argument in [45], that neural networks rely on non-robust data that is not important for human perception, but useful to maximize the classification accuracy on unperturbed data, is a

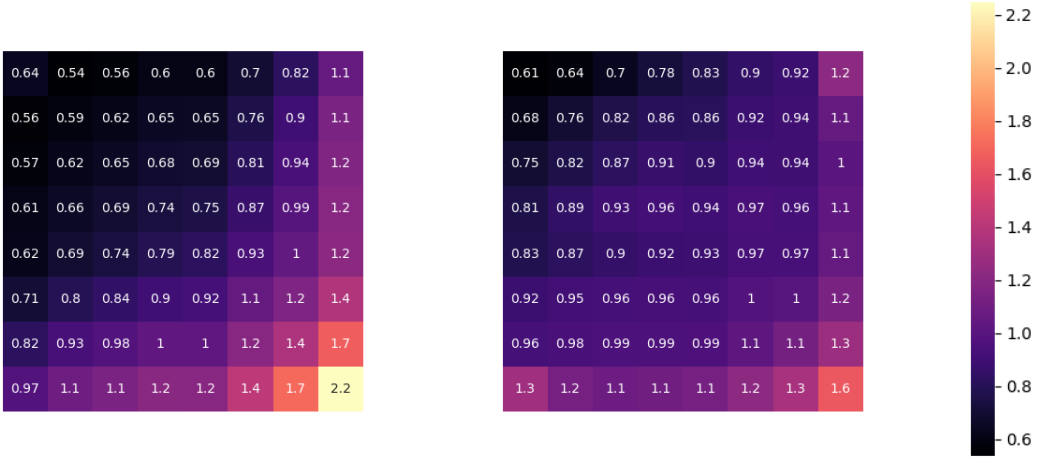
> "*special case of a more general principle that is commonly accepted in the robustness to distributional shift literature: a model's lack of robustness is largely because the model latches onto superficial statistics in the data. In the image domain, these statistics may be unused by and unintuitive to humans, yet they may be useful for generalization in i.i.d. settings*" [31].

Gilmer and Hendrycks name the reliance of neural networks on high frequencies, as shown in [99], as another example. Yin et al. [99] analyzed the robustness of neural networks from a fourier perspective. They make some interesting findings on how adversarial attacks and model robustness are related to spatial frequencies. They train neural networks using only high-frequency information that is barely visible to humans and are able to reach more than 50% accuracy on IMAGENET. This shows that neural nets can use information that is of little importance to humans and that they often rely on high-frequency information that is not important for human perception. Adversarial perturbations that are crafted on undefended nets are usually concentrated in high frequencies[11], as neural networks often rely on high-frequency information that is not visible to humans. The fact that neural networks use such invisible high-frequency information has also been found by Wang et al. [96].

However, there might be a bigger connection between these two examples than just being two examples of neural networks learning from "*superficial statistics*" [31]. Figure 2.19 shows heatmaps of the distribution of normalized coefficients for each frequency for all images in the non-robust and in the robust dataset from Ilyas et al. [45], respectively.

At least for CIFAR10, the non-robust features seem to consist of higher frequency information, as they have much higher coefficients on the high frequencies, while the images

[11] Experimental evidence is given later in chapter 4.



(a) non-robust (b) robust

Figure 2.19: Frequency-wise average absolute normalized coefficients of the robust and non-robust CIFAR10 datasets from [45]. Both datasets are normalized by subtracting the frequency-wise mean and standard deviation of the original CIFAR10 dataset. The non-robust features are more concentrated in high frequencies, while robust features are concentrated in low and medium frequencies, at least for CIFAR10.

from the robust dataset have higher coefficients on the lower frequencies. This confirms that neural networks relying on non-robust features and nets relying on high frequencies are not just two independent examples of nets using information that is not useful to humans, but these two examples can describe the same problem: Neural nets often rely on high-frequency information that is not useful and barely recognisable for humans, as already stated by Yin et al. [99]. In the literature on adversarial attacks from a frequency perspective, this very intuitive relation has been discussed before. For example, Bernhard et al. state that high spatial frequencies "*are predominantly non robust*" [8]. Yin et al. state that "*it seems likely that these invisible high-frequency features are related to the experiments of [45, citation adapted]*" [99]. However, to the best of our knowledge, this relation has not been proofed experimentally before.

Yin et al. have also shown that standard adversarial training makes the net biased towards using low-frequency information and thus more robust to high-frequency perturbation by forcing it to avoid using the non-robust features that are mainly located on high frequencies. Training the net using JPEG compressed images or just using JPEG compressed images at inference time, as proposed by Dziugaite et al. [22], aims at the same thing: Forcing the net to use more low-frequency information, as humans do, and thus, making

the net more robust against high-frequency perturbation which is often unrecognizable to humans.

However, making the net more robust against high-frequency perturbation is not sufficient. Yin et al. [99] state that while standard adversarial training results in the net being more robust against high-frequency perturbations, it also gets more vulnerable towards perturbations in low frequencies. They also state that adversarial perturbations tend to be more concentrated in lower and medium frequencies when they are created on an adversarially trained net, which is a direct consequence of the nets relying on low-frequency information. So, adversarial examples are not necessarily concentrated in high frequencies, but it depends on the source model.

Similar findings have been made by several other researchers: Tsuzuku and Sato state "*that adversarial perturbations do not necessarily lie in a high-frequency area, which denies a common myth that adversarial perturbations tend to be high-frequency*" [94], while Bernhard et al. add that "*adversarial perturbation[s] are not efficient when focused only on HSF [high spatial frequencies]*" [8], meaning that adversarial attacks have to perturb a "*wide part of the spectrum*" [8] to be efficient. The success of low-frequency perturbations [36, 84] also supports the hypothesis that adversarial perturbations are not necessarily a high-frequency phenomenon [66] and that robustness against high-frequency perturbations is not sufficient.

Other papers [8, 66] looked at adversarial robustness from a frequency perspective. Again, both find significant differences between datasets as CIFAR10 models are more sensitive towards high-frequency information than models trained with other datasets. Bernhard et al. [8] use various fourier filters during training. Using low-pass filtered images in training leads to less vulnerability towards high-frequency perturbations. They only evaluate their defenses against standard BIM, though, and do not use a perceptual distance, but the input parameter ε, although the structure of the created adversarial examples should differ significantly as they use different source models (low-, high-pass filtered, unfiltered images in training). In general, they find a two-way transferability between the low-pass filtered and the base classification task. A stronger low-pass filter during training leads to less transferability. They state that "*this indicates that the regular classification task and the LSF task share predominantly robust useful features*" [8] which means that robustness is strongly related to low-frequency features [8]. The intuition behind this is that humans mainly rely on low-frequency information too and, as explained before, a classifier that is more aligned with the human perception should generally be more robust. This

could be accomplished by forcing a classfier to use low-frequency information instead of high-frequency components that are barely visible for humans. However, as stated before, adversarial examples are not necessarily a high-frequency phenomenon such that relying on low-frequency information only is not expected to yield a perfectly robust classifier either. Maiya et al. [66] emphasize that in which frequencies adversarial examples are concentrated depends on the dataset. Again, they use an attack that perturbs on RGB but transforms the gradients using DCT, masks out some frequencies, and applies IDCT. By attacking individual frequencies, they measure how sensitive models are for perturbations on each frequency. For CIFAR10, the undefended models are most sensitive on high frequencies, while for IMAGENET they are most sensitive for lower frequencies. The adversarially trained model for CIFAR10 reverses the sensitivity towards the low frequencies, while the adversarially trained IMAGENET model is still most sensitive towards low-frequency perturbations, but the robustness across the whole spectrum increases. Despite the differences between datasets, this still leaves the question of how to make the model more robust against low-frequency perturbations, where standard adversarial training does not seem to be efficient.

While these works give an overview on the models' sensitivity of different frequencies for different datasets as well as the role of frequencies for adversarial robustness, none of these works [8, 20, 36, 66, 84, 99] is JPEG-related as they mainly perturb images in their RGB pixel representations and just mask the gradients of some DCT/DFT frequencies.

3 Adversarial Perturbations straight on JPEG coefficients

There are four main motivations for us to execute adversarial attacks straight on JPEG coefficients. First, JPEG (usually) uses the YC_bC_r color space instead of RGB. The existing JPEG and frequency attacks that were described in sections 2.2.2 and 2.5 still (mainly) apply the perturbation on the RGB pixel representation. Recently, Pestana et al. [74] analyzed the importance of the luma channel for adversarial attacks. They found that when attacking in the RGB domain, the perturbations are concentrated in the luma channel and when perturbing only the luma channel, the success rate is higher. They explain this by neural nets mainly relying on textures for their predictions and the fact that those shapes and textures are more concentrated in the luma channel. Therefore, attacking the YC_bC_r channels independently is a main motivation for attacking on JPEG coefficients.

The second motivation is the fact that lossy image compression algorithms try to separate perceivable from imperceptible data. JPEG compression does so by the chroma subsampling and, more importantly, separating low frequencies from high frequencies that are less important for human perception to enable an efficient entropy encoding. As explained in section 2.2.2, some attacks related to JPEG compression use RGB pixel and include an approximation of JPEG compression in the target model [86], or only transform to JPEG coefficients after the main perturbation has happened on RGB pixels [85]. While there have been some works that perturbed straight on JPEG coefficients [48, 58], they only use it as one example of many to simulate a threat model that has been unseen during training. Works that analyzed the effect of perturbing spatial frequencies differently for attacks and defenses [8, 20, 36, 66, 84] are often limited by only allowing boolean masking vectors and use the RGB pixel representation which means that some perturbations will be removed during JPEG compression. Additionally, those works do not contain a quantitative evaluation of the perceptual distance. They mainly use the

input parameter ε to compare the attacks' success which is problematic especially when perturbing different frequencies.

Third, one major advantage of perturbing straight on JPEG coefficients is that the transformation of JPEG coefficients to RGB pixels is differentiable, because RGB pixel values do not necessarily have to be rounded: Although some works quantize the RGB output of the attacks, e.g. [81], to simulate saving the images, we assume the data to be saved in JPEG format and thus, the RGB data can be unquantized. Therefore, we can perturb on JPEG coefficients but integrate the JPEG to RGB transformation into the source model, and use a standard RGB net. This could be especially advantageous when perturbing coefficients of lower quality to bypass JPEG compression in defense, as the rounding approximation has a bigger impact here than for quality 100.

At last, we believe that the variability of our attack that follows from using channel-wise perturbation budgets, and the masking vectors that allow to perturb different frequencies, can help making adversarially trained nets more robust against perturbations on all frequencies, and thus, can lead to better generalization.

We will now define the technical details of our proposed attack method. To be able to compute gradients for JPEG coefficients, we design a differentiable conversion of JPEG coefficients to RGB pixels. In Shin and Song [86]'s paper, the reverse approach is applied, which requires the use of a rounding approximation. As the differentiable JPEG to RGB conversion allows us to attack straight on JPEG coefficients, there is no need to include an approximation of JPEG compression into the source model as in [86] or, make the perturbation robust towards JPEG compression by a sophisticated rounding scheme as in [85]. Thus, our method is technically straightforward.

To convert JPEG into RGB, we build a neural network of convolutional layers of fixed weights. First, the Coefficients are reordered using a 1×1 convolution with 64 output channels, reversing the zig-zag. Then, the 64 coefficients are reshaped as an 8×8 matrix. For every channel, the coefficients are then dequantized by multiplying them with the corresponding quantization matrix Q^{jq}. In the next step, the inverse discrete cosine transform computes pixel values from coefficients. The blocks are then put together using a transposed convolution with 1 filter, 8×8 kernel size and 8×8 strides for the luma channel and 16×16 for the chroma channels, if they are downsampled (and 8×8 otherwise). Now, a $\mathrm{YC_bC_r}$ image was reconstructed and can then be converted to RGB. In the following, we will denote this transformation as $\mathrm{rgb}(x)$ for JPEG image x. The

reverse transformation can be implemented correspondingly, but the coefficients have to be rounded after quantization.

We define JPEG versions of both maximum-confidence and minimum-norm attacks that perturb the coefficients of step 6 of the compression procedure explained in section 2.2.1. So, let

$$x^{jq} = (Y, C_b, C_r) \tag{3.1}$$

a quantized JPEG image of quality jq. For an image of shape $h \times w$, the luma channel Y has shape $(h/8, w/8, 64)$, the chroma channel has shape $(h/16, w/16, 64)$ if chroma subsampling is used, and $(h/8, w/8, 64)$ if not. The main difference in implementation between RGB attacks and JPEG attacks is that the gradients have to be computed for all three variables Y, C_b, C_r[1] and separate update steps are made for the three channels using different step sizes.

3.1 Maximum-Confidence JPEG attacks

In maximum-confidence attacks, the perturbation is limited by some L_p-ball. To enable individual control over the perturbation made on each channel, we define three L_p-balls.

In the case of L_∞-attacks, which we will focus on in this section, we thus define six variables, three relative perturbation budgets $\varepsilon_Y^{\text{rel}}, \varepsilon_{C_b}^{\text{rel}}, \varepsilon_{C_r}^{\text{rel}} \in \mathcal{R}_{\geq 0}$ and three masking vectors $\lambda_Y, \lambda_{C_b}, \lambda_{C_r} \in [0, 1]^{64}$ that limit the relative perturbation made on each channel, where the ε values control the amount of perturbation made on each channel and the λ values determine how much perturbation is permitted for every frequency[2].

From these relative budgets and the masking vectors, we then compute absolute limits by

$$\begin{aligned}
\varepsilon_Y^{\text{abs}} &= \varepsilon_Y^{\text{rel}} \cdot \lambda_Y \cdot \max(|Y|, 1) \\
\varepsilon_{C_b}^{\text{abs}} &= \varepsilon_{C_b}^{\text{rel}} \cdot \lambda_{C_b} \cdot \max(|C_b|, 1) \\
\varepsilon_{C_r}^{\text{abs}} &= \varepsilon_{C_r}^{\text{rel}} \cdot \lambda_{C_r} \cdot \max(|C_r|, 1).
\end{aligned} \tag{3.2}$$

[1] The reasons for this are practical, as tensors of different shapes (luma, chroma) cannot be combined to a single tensor. When not using chroma subsampling, it would be possible to implement it using one gradient computation only. However, we use the same implementation for both versions.

[2] In our experiments, we will denote the relative budgets as ε as well.

We use the maximum of the original coefficients and 1 as the coefficients to allow perturbation even if the original coefficient is 0.

Alternatively, the coefficients with value 0 can also be fixed by computing the absolute perturbation budgets by

$$\begin{aligned}\varepsilon_Y^{\text{abs}} &= \varepsilon_Y^{\text{rel}} \cdot \lambda_Y \cdot |Y| \\ \varepsilon_{C_b}^{\text{abs}} &= \varepsilon_{C_b}^{\text{rel}} \cdot \lambda_{C_b} \cdot |C_b| \\ \varepsilon_{C_r}^{\text{abs}} &= \varepsilon_{C_r}^{\text{rel}} \cdot \lambda_{C_r} \cdot |C_r|. \end{aligned} \tag{3.3}$$

We will compare the success of both versions in our experiments. For iterative attacks, the relative step sizes are converted to absolute ones in the same way using αs instead of εs.

These values $\varepsilon_Y^{\text{abs}}, \varepsilon_{C_b}^{\text{abs}}, \varepsilon_{C_r}^{\text{abs}} \in \mathcal{R}_{\geq 0}^{(h/8)\times(w/8)\times 64}$, [3] then have the same shape as the corresponding coefficient vectors and limit the perturbation by

$$\begin{aligned} L_\infty^B(\frac{Y - Y'}{Y}) &\leq \lceil \varepsilon_Y^{\text{rel}} \lambda_Y \rceil \\ L_\infty^B(\frac{C_b - C_b'}{C_b}) &\leq \lceil \varepsilon_{C_b}^{\text{rel}} \lambda_{C_b} \rceil \\ L_\infty^B(\frac{C_r - C_r'}{C_r}) &\leq \lceil \varepsilon_{C_r}^{\text{rel}} \lambda_{C_r} \rceil, \end{aligned} \tag{3.4}$$

where L_∞^B computes the norm across all 8×8 block for each frequency separately.

In a previous work [88], we used an absolute ε only. This led to much more difficult control of the perturbation across frequencies as low-frequency coefficients usually have a much higher absolute value and when the lambda values had higher values on low frequencies this did not lead to the strongest relative perturbation being made on these frequencies, which made argumentation more difficult. Details on this problem can be found in appendix A.

[3] Note that this assumes that chroma subsampling is not used. Otherwise, $\varepsilon_{C_b}^{\text{abs}}, \varepsilon_{C_r}^{\text{abs}} \in \mathcal{R}_{\geq 0}^{(h/16)\times(w/16)\times 64}$.

For BIM, a single perturbation step is defined by

$$\begin{aligned} Y'_t &= Y'_{t-1} + \text{sign}(\nabla_{Y'_{t-1}}(\text{J}(\text{rgb}(x'_t), y))) \cdot \alpha_Y^{\text{abs}} \\ C_{b}'_t &= C_{b}'_{t-1} + \text{sign}(\nabla_{C_{b}'_{t-1}}(\text{J}(\text{rgb}(x'_t), y))) \cdot \alpha_{C_b}^{\text{abs}} \\ C_{r}'_t &= C_{r}'_{t-1} + \text{sign}(\nabla_{C_{r}'_{t-1}}(\text{J}(\text{rgb}(x'_t), y))) \cdot \alpha_{C_r}^{\text{abs}}, \end{aligned} \tag{3.5}$$

where $\text{rgb}(x)$ denotes the transformation from JPEG to unquantized RGB data for JPEG image x.

After the T iterations, the coefficients are projected into each L_p-ball. Then, they are rounded, either to the nearest integer, or using the fast adversarial rounding scheme from [85]. We will analyze the effect of both in the experiments.

3.2 Minimum-Norm JPEG attacks

We convert only one of the minimum-norm attacks mentioned in chapter 2 to JPEG: The PerC-AL by Zhao et al. [102]. The PerC-AL does not require projection into some L_p-ball, which would be difficult to design with the three independent $\text{YC}_\text{b}\text{C}_\text{r}$ channels, and is usually quicker than the C&W-L_2-attack. Remember that the PerC-AL used two step sizes α_l, α_c for the two alternating steps - maximizing the model loss and minimizing the distance (see eqs. (2.32) and (2.33)). As we still want to control the perturbation for each channel and frequency individually, we assume six input step sizes $\alpha_Y^{l,\text{rel}}, \alpha_{C_b}^{l,\text{rel}}, \alpha_{C_r}^{l,\text{rel}}, \alpha_Y^{c,\text{rel}}, \alpha_{C_b}^{c,\text{rel}}, \alpha_{C_r}^{c,\text{rel}}$ and three masking vectors $\lambda_Y, \lambda_{C_b}, \lambda_{C_r}$. The input step sizes are then converted to absolute step sizes in the same way as before (see eq. (3.2). If the image is adversarial, the luma channel is updated by

$$\begin{aligned} g_Y &= \nabla_{Y'_{t-1}} \text{J}(\text{rgb}(x'_{t-1}), y) \\ \delta_{Y,t} &= \delta_{Y,t-1} + \alpha_Y^{l,\text{abs}} \cdot \frac{g_Y}{||g_Y||_2}, \end{aligned} \tag{3.6}$$

and correspondingly for both chroma channels. If the current image x'_{t-1} is not adversarial, though, it is updated by

$$\begin{aligned} d &= -||\,\text{ciede2000}(\text{rgb}(x'_{t-1}), \text{rgb}(x))||_2 \\ g_Y &= \nabla_{Y'_{t-1}} d \\ \delta_{Y,t} &= \delta_{Y,t-1} + \alpha_Y^{c,\text{abs}} \cdot \frac{g_Y}{||g_Y||_2}. \end{aligned} \tag{3.7}$$

The PerC-AL-Attack can be easily adapted such that it minimizes the LPIPS distance instead of CIEDE2000, which is still a pixel-wise distance metric. Thus, we implement a variant of PerC-AL, LPIPS Alternating Loss (LPIPS-AL), for both RGB and JPEG and include it in our experiments.

4 Experiments and Results

In this chapter, we will analyze the efficiency of our method for various parameters, compare it to state-of-the-art attacks and measure its suitability for adversarial training. We start with describing our experimental setup in section 4.1. Then we will evaluate the suitability of some perceptual metrics in section 4.2, followed by the main part of our experiments where we will first address maximum-confidence attacks in section 4.3 and then minimum-norm attacks in section 4.4. Finally, section 4.5 will focus on whether using our JPEG attacks during adversarial training can increase the model's robustness.

4.1 Implementation and Experimental Setup

All our main experiments are generally implemented in Tensorflow [1]. When models are only available for PyTorch [73], such as the Perceptual Adversarial Training model [58], we convert the net's input to PyTorch and convert the output back to Tensorflow such that we can evaluate the results. Our code and notebooks, including all our experiments, are available at https://github.com/KoljaSmn/jpeg-adversarial-attack-masterthesis.

Neural Networks & Adversarial Training For our experiments, we generally use ResNets [38] and DenseNets [42]. For CIFAR10, we use a ResNet56-V2 and a DenseNet100, both implemented by [7]. For IMAGENET, we use the pretrained ResNet152-V2 and DenseNet201 implementations from Tensorflow [1].

For adversarial training, we generally use batches that consist of half original and half adversarial images as proposed by [55]. Other researchers often use batches that only contain adversarial images, which leads to better robustness but worse accuracy on the original images. Tsipras et al. [93] compared both versions.

We use the adversarial training from Madry et al. [65] that uses the Basic Iterative Method to create adversarial examples. For the standard RGB training, we use $\varepsilon = 8$, $\alpha = \frac{\varepsilon}{4}$ and $T = 7$. The nets are pretrained on the original data. The Cifar10 net is trained for 100 epochs, but only the net with the lowest validation loss is saved. The same applies for Imagenet, where the training runs for only 20 epochs though.

A DenseNet defended with Madry et al.'s RGB adversarial training, is denoted as Densenet_M^{RGB}. A DenseNet that is defended by applying JPEG compression to the input at inference time is called Densenet^{jq75}, where 75 would be the jpeg quality in this example.

LPIPS For our LPIPS version, we use a pretrained VGG16 model from Tensorflow[1]. The net expects input of shape $224 \times 224 \times 3$. Thus, the input images (e.g., 32×32 Cifar10 images) are reshaped using bilinear interpolation.

The LPIPS model is then trained by fixing the VGG16's weights and adding the distance computation from eq. (2.30). The distances are then put into the trainer model G that is built exactly as described in section 2.3.2 which enables training on the 2AFC dataset from [101].

As in the original implementation[2], we use an initial learning rate of $1e-4$ that is linearly decayed. The model is trained for 5 epochs.

Attack Implementation and Parameters We generally use our own implementations for the attacks. The DDN and PerC-AL implementations are converted to Tensorflow from the original PyTorch implementations[3]. For C&W-L_2, we use the implementation from Cleverhans [72].

For minimum-norm attacks, we generally use the default parameters. For DDN and PerC-AL we limit the number of iterations to 100.

[1] As in the original implementation, the following layers are used to compute the distances: block1_conv2, block2_conv2, block3_conv3, block4_conv3, block5_conv3

[2] https://github.com/richzhang/PerceptualSimilarity/blob/master/lpips/trainer.py

[3] DDN: https://github.com/jeromerony/adversarial-library/blob/main/adv_lib/attacks/decoupled_direction_norm.py,
PerC-AL: https://github.com/ZhengyuZhao/PerC-Adversarial/blob/master/perc_al.py

For the maximum-confidence attacks, our parameter selection differs from the original one. For BIM, Kurakin et al. [56] recommended to use $\alpha = 1$ and $T = \min\{\varepsilon + 4, 1.25\varepsilon\}$. Such a natural α selection does not exist for our JPEG attacks, though, as it is a relative perturbation budget that also differs across frequencies. Thus, for our experiments, we use a constant number of iterations $T = 10$ for both RGB and JPEG attacks and set $\alpha = \frac{\varepsilon}{T}$.

For the minimum-norm attacks, we use $T = 100$ iterations for all attacks. For the C&W-L_2, 5 binary search steps are executed. All RGB attacks use the original step sizes. The step size selection for the JPEG attacks will be explained later. For CIFAR10, the attacks are performed using four confidence values: 0, 10, 20, 40. For IMAGENET, only one attack with confidence $\kappa = 0$ is performed.

Whenever we do not specify a JPEG quality for our JPEG attacks, quality 100 is used such that every entry in the quantization matrices equals 1.

Attack Evaluation As mentioned before, comparing attacks using the perturbation budget does not fit the actual goal of adversarial attacks which is being perceptually close to some original image. Additionally, as the input parameters for RGB and coefficient attacks are not comparable we are also forced to use perceptual distances to compare the attacks' success. In our experiments, we generally increase some parameter (e.g. the perturbation budget), measure the resulting perceptual distance and the success rate and then plot the success rate in dependence of the perceptual distance. We also call the relation between the attack's success and the perceptual distance the efficiency of the attack.

For CIFAR10, we always use the complete test dataset consisting of 10000 images for our experiments. For IMAGENET, we only use 10000 of 50000 images from the validation dataset. The classes are equally distributed such that each of the 1000 classes occurs ten times in the dataset.

4.2 Perceptual Metrics

In this section, we will evaluate our perceptual metrics' suitability for measuring human perception. We use the same evaluation metrics as used in the original LPIPS paper [101]. They were already described in section 2.3.3. For the 2AFC datasets, where each entry

contains the reference image r, two image patches p_0, p_1, and the percentage of humans p that perceived patch 1 as closer to the reference image than patch 0, the perceptual metric receives score p if $d(r, p_0) < d(r, p_1)$, and $1 - p$ otherwise. For the JND dataset, the score computes the area under the precision-recall curve.

Additionally, we will measure the correlation between the distances and the human judgement scores. Correlation has the advantage that it does not only consider which distance is smaller, but also the differences between the distances. In addition to SSIM, RGB L_2 and our trained LPIPS version, which were already evaluated in [101], we will also include L_2 distances for CIEDE2000 and CIELAB. For the 2AFC dataset, table 4.1 shows the results. The human score is computed as in [101]: If p percent of the humans perceive patch 1 as closer to the reference image than patch 0, a human would be expected to achieve a score of $p^2 + (1-p)^2$. As already mentioned, the RGB L_2 distance is not a good measure of human perception. In terms of both the score and the correlation, it does not perform well. The same applies to SSIM, which has a better correlation than RGB L_2 though. CIELAB L_2 and CIEDE2000 L_2 perform similarly as CIELAB yields the better score, but CIEDE2000 correlates better. Overall, all L_2 distances as well as SSIM achieve very similar scores. The LPIPS distance, which is based on a VGG16 net, outperforms all of them significantly and almost reaches the score expected for humans.

Metric	Score	Correlation
SSIM	0.681	0.467
RGB L_2	0.687	0.417
CIELAB L_2	0.705	0.498
CIEDE2000 L_2	0.697	0.500
LPIPS	0.763	0.684
Human	0.826	1.

Table 4.1: Evaluation of perceptual metrics on the 2AFC dataset from [101].

Metric	Score	Correlation
SSIM	0.565	-0.505
RGB L_2	0.618	-0.465
CIELAB L_2	0.582	-0.438
CIEDE2000 L_2	0.590	-0.482
LPIPS	0.673	-0.615

Table 4.2: Evaluation of perceptual metrics on the JND dataset from [101].

The results for the JND dataset are shown in Table 4.2. In this case, the RGB L_2 distance achieves a better score than CIELAB and CIEDE2000 L_2. However, the CIEDE2000 L_2 distance's correlation is better. Again, LPIPS achieves the best results by far.

For the adversarial JND dataset from [58], we only use those entries where the images are not identical. Table 4.3 shows the corresponding results. Again, LPIPS performs best by far. In this case it is followed by the RGB L_2 distance. Overall, it is surprising how

Metric	Score	Correlation
SSIM	0.877	-0.495
RGB L_2	0.887	-0.616
CIELAB L_2	0.881	-0.570
CIEDE2000 L_2	0.882	-0.573
LPIPS	0.901	-0.688

Table 4.3: Evaluation of perceptual metrics on the adversarial JND dataset from [58].

close the 3 L_2 distances are as we expected RGB L_2 to perform significantly worse. The performance depends on how the dataset is constructed. If every (perturbed) image in the dataset is perturbed in RGB space, the RGB L_p distance could possibly be a decent measure. If however, e.g., $\mathrm{YC_bC_r}$ is used, RGB might not be a well-suited color model for measuring the distance. This is supported by the superior performance of CIEDE2000 on the 2AFC dataset from Zhang et al. [101] which is created by perturbing images in various ways. Thus, collecting a 2AFC dataset of adversarial images that were perturbed on different color channels, might be an interesting topic for future work. Following these results, we will mainly use the LPIPS distance in our experiments. However, when there are significant differences between the results using the LPIPS distance and the CIEDE2000 L_2 distance, we will touch upon that.

4.3 Maximum-Confidence Attacks

Significant parts of this section have already been addressed in our previous work [88]. The most important differences are that we now use relative change budgets instead of absolute ones, distinguish between white-box and black-box attacks, use LPIPS in addition to CIEDE2000 L_2 distance, and have added some further experiments. Although these changes cause significant differences in the results, some arguments and justifications, especially with regard to the success of different weighting vectors and the comparison with $\mathrm{YC_bC_r}$ and RGB attacks, are very similar and some are completely adopted and only supplemented.

Here, we start with comparing the RGB attacks FGSM, BIM and MI-FGSM. Figure 4.1 shows the results. In dependence of the input parameter ε, MI-FGSM is the most successful attack in the black-box setting, which it was designed for as well. However, when perceptual color distance is used for comparison, BIM is slightly more successful. This is because MI-FGSM changes the direction of optimization less often since it uses the cumulated gradients instead of the current ones, and thus, leads to more perceived distortion. So, while MI-FGSM maximizes the success in dependence of ε, it also increases the perceptual distance. This illustrates the unsuitability of the L_∞ distance for comparing attacks. As we try to maximize the relation between success and the perceptual distance, we will use BIM in the following.

4.3.1 Optimizing parameters to find our best attack

Before comparing our best attack with state-of-the-art attacks in section 4.3.2, we will first analyze various parameter selection for our attack regarding their efficiency in several settings. First, we will examine differences between perturbation on the three $\mathrm{YC_bC_r}$ color channels and how chroma subsampling affects the attack's efficiency. Then, we will analyze whether it is advantageous to attack unquantized coefficients instead of quantized JPEG coefficients, followed by an experiment on how fast adversarial rounding influences the efficiency of the attack. Subsequently, we determine in which parts of the frequency spectrum the perturbation should be concentrated in order to optimize the attack's success. This will be split into two parts in which we first use manually defined frequency weighting vectors and then try to learn optimal weighting vectors. Then, we try to remove visible JPEG blocks to improve the JPEG adversarial examples'

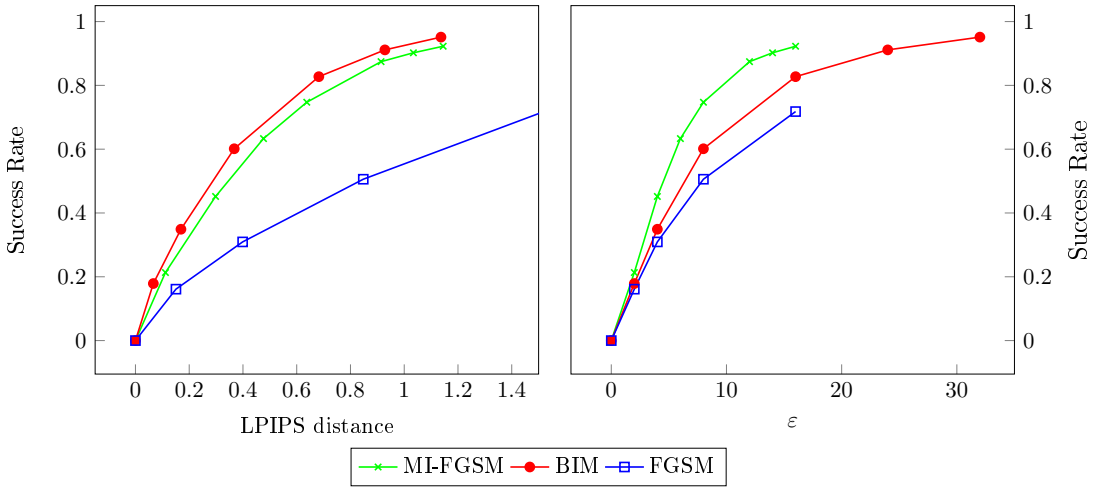


Figure 4.1: CIFAR10 - Comparison of RGB maximum-confidence attacks. Adversarial images are created on an undefended Resnet. Success rates are measured on an undefended Densenet.

perceptual quality. Finally, we analyze how fixing 0-coefficients influences the efficiency of the attack.

Varying Luma and Chroma Perturbations

First, we analyze how successful attacks are across color channels. For this, we perform attacks for which one of $\varepsilon_Y^{\text{rel}}, \varepsilon_{C_b}^{\text{rel}}, \varepsilon_{C_r}^{\text{rel}}$ is gradually increased, while the other two are set to zero. Additionally, an attack with $\varepsilon_{all}^{\text{rel}} = \varepsilon_Y^{\text{rel}} = \varepsilon_{C_b}^{\text{rel}} = \varepsilon_{C_r}^{\text{rel}}$ is performed, where $\varepsilon_{all}^{\text{rel}}$ is gradually increased. The results are illustrated in fig. 4.2. For now, the attacks are unmasked such that $\lambda_{all} = 1$.

The same process is carried out for $\text{YC}_\text{b}\text{C}_\text{r}$ pixel attacks. They are defined in a similar way as our JPEG attacks: We still use three variables for the three channels. However, the perturbation is applied pixel-wise as for RGB attacks. Thus, we do not need the distinction between relative and absolute perturbation budgets. Later, defining these $\text{YC}_\text{b}\text{C}_\text{r}$ attacks will allow us to analyze whether the advantages or disadvantages that JPEG attacks have compared to RGB are reasoned in the use of the $\text{YC}_\text{b}\text{C}_\text{r}$ color model or the use of coefficients.

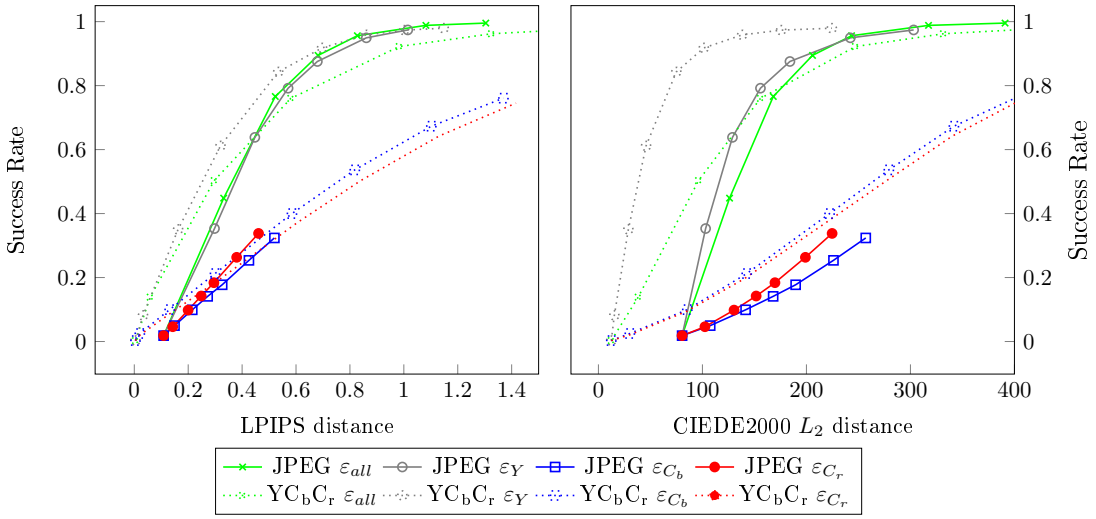


Figure 4.2: CIFAR10 - Success rates for unmasked BIM on an undefended DenseNet. Images are created on a ResNet. The JPEG attacks use chroma subsampling, which shifts the graph on the x-axis. For comparison, attacking images with RGB FGSM and $\varepsilon = 8$ results in an average CIEDE2000 L_2 distance of 187.

In this case, the results differ depending on which perceptual distance is used. In dependence of the CIEDE2000 L_2 distance, perturbing the luma channel is most successful while perturbing only one chroma channel barely influences the net's accuracy, even when it is undefended. This is consisitent with the results presented by Pestana et al. [74] regarding YC_bC_r attacks and implies that the DenseNet mainly uses luma information for classification, including the main information for shapes and textures, which are known to be most relevant for neural nets' classifications [30, 74] and the human perception, which is why the chroma channel is usually subsampled in JPEG compression. An example is shown in fig. 4.3. The three YC_bC_r channels are visualized for two images. The luma channel contains much more detailed information that is much more useful and in these cases sufficient for detecting the object, while the information in the chroma channels is more blurry, especially for the IMAGENET example.

In dependence of the LPIPS distance though, the ε_{all}- and ε_Y-attacks are much closer in terms of success. In case of the JPEG attacks, the ε_{all}-attack is even slightly more successful. So, CIEDE2000 seems to weight chroma perturbations stronger than luma perturbations in comparison to LPIPS. Still, the difference between luma/all perturbations and chroma perturbations remains big.

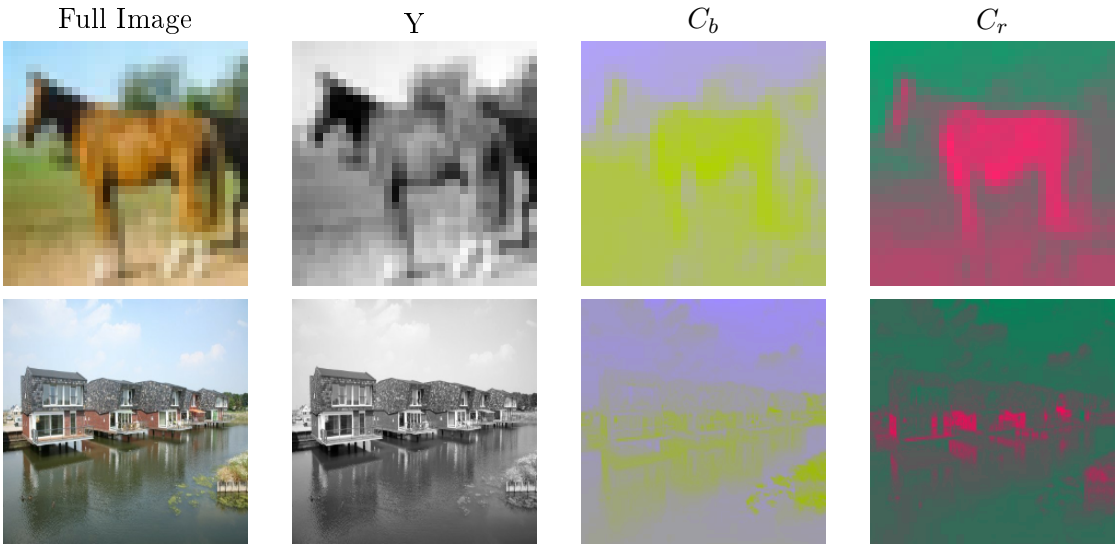


Figure 4.3: Illustration of $\mathrm{YC_bC_r}$ channels for a CIFAR10 (top) and an IMAGENET (bottom) image.

Another interesting finding is that for JPEG attacks the difference between only perturbing the luma channels and perturbing all three channels is much smaller than for $\mathrm{YC_bC_r}$ pixel attacks. We explain this by applying relative perturbations. When applying absolute perturbations for JPEG attacks, the difference was much bigger as well, as shown in fig. A.1 in appendix A. This is only logical as chroma coefficients usually have much smaller absolute values (see fig. C.1) such that perturbing them by the same absolute value as the luma channel is not reasonable. Thus, using a relative $\varepsilon_{all}^{\text{rel}}$ leads to more successful attacks. So, in general adversarial perturbations should be concentrated in the luma channel. When attacking all channels, the chroma perturbations should be smaller than on the luma channel. By using the relative perturbation budgets, our JPEG attacks ensure this implicitly.

Until now, we either set $\varepsilon_Y^{\text{rel}} = \varepsilon_{C_b}^{\text{rel}} = \varepsilon_{C_r}^{\text{rel}}$ or two of them were set to 0. Now, to examine whether the success can be maximized by choosing $\varepsilon_Y^{\text{rel}} > \varepsilon_{C_b}^{\text{rel}}, \varepsilon_{C_r}^{\text{rel}} > 0$, we make an experiment where $\varepsilon_{C_b}^{\text{rel}}, \varepsilon_{C_r}^{\text{rel}} = \gamma\varepsilon_Y^{\text{rel}}$ and γ is continuously increased. The results are visualized for both CIEDE2000 and LPIPS in fig. 4.4. Again, we make the observation that the CIEDE2000 metric puts less weight to luma perturbations: In dependence of the CIEDE2000 L_2 distance, the strongest attack only perturbs the luma channel, as just increasing ε_Y results in a better ratio between success and distance than increasing γ. In

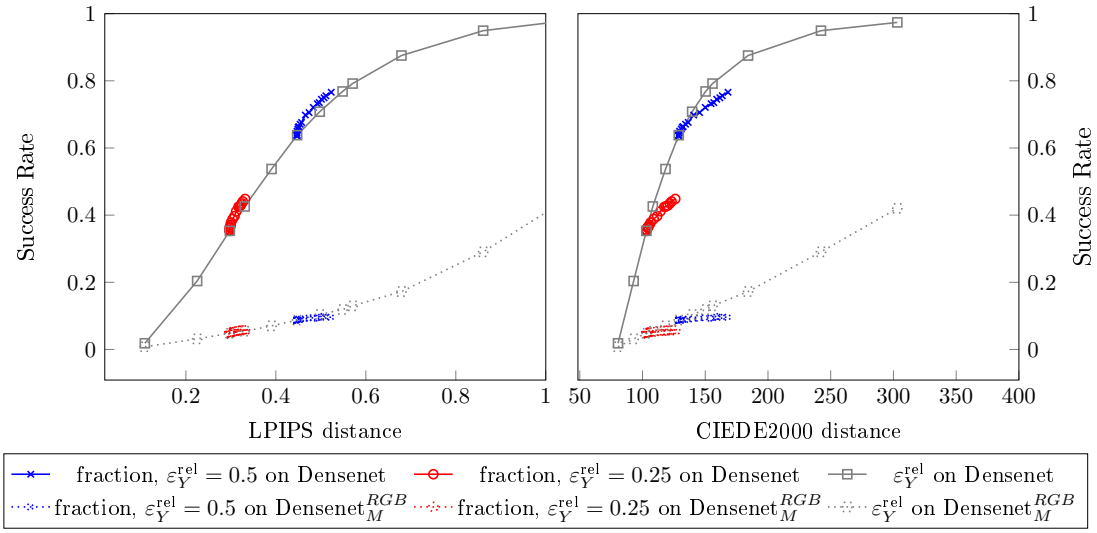
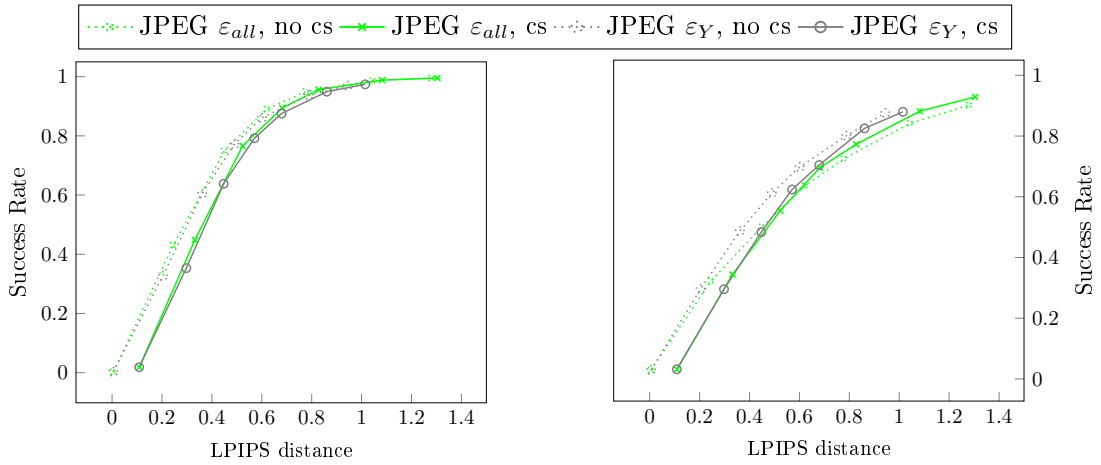


Figure 4.4: CIFAR10 - Success rates for unmasked JPEG BIM in dependence of the perceptual distance. For the fraction attacks, the fraction $\gamma \in [0, 1]$ is increased further and further, and $\varepsilon_{C_b}^{\text{rel}}, \varepsilon_{C_r}^{\text{rel}} = \gamma \varepsilon_Y^{\text{rel}}$. The results differ depending on the distance metric used.

dependence of the LPIPS distance though, increasing γ yields a slightly better relation than increasing ε_Y for luma-only attacks, at least for the undefended net.

Chroma Subsampling

While in the previous experiment we used JPEG coefficients whose chroma channels were subsampled, this led to a significant perceived perturbation even if $\varepsilon_{all}^{\text{rel}} = 0$, which caused the shift on the x-axis. There are versions of JPEG that allow the chroma channel not to be subsampled. When using such a version in attack, the perceived distance is lowered, as fig. 4.5 suggests. For bigger distances, the success rates of both versions converge. When JPEG compression is used as a defense technique and $\varepsilon_{C_b}^{\text{rel}}, \varepsilon_{C_r}^{\text{rel}} \neq 0$, using chroma subsampling in the attack can be useful as well. Otherwise, some perturbations might be removed during the subsampling process. Figure 4.5b illustrates the effect of chroma subsampling on a net defended with JPEG compression. For the attack on all channels, the success of the attack using chroma subsampling exceeds the success when not using chroma subsampling for bigger perturbations. For small perturbations, the distance that results from the chroma subsampling itself leads to the attack being less successful in



(a) Densenet (b) Densenet^{jq50}

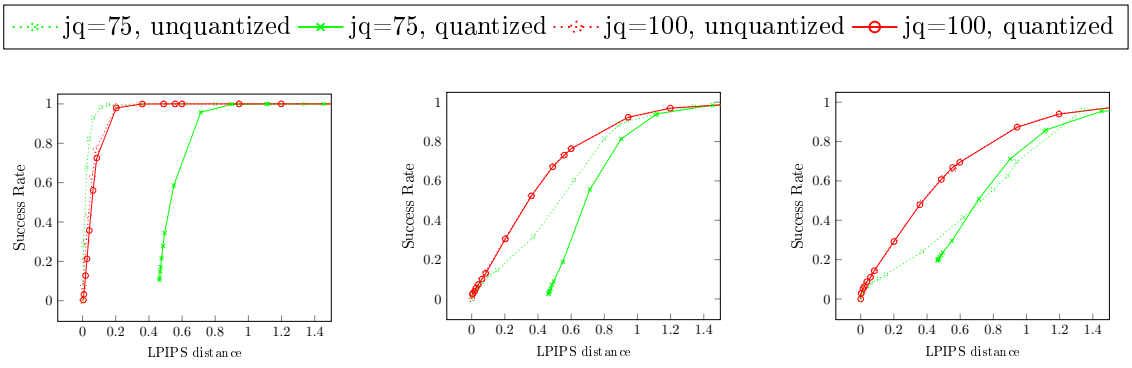
Figure 4.5: CIFAR10 - Success rates for unmasked BIM in dependence of the LPIPS distance on an undefended DenseNet, with and without chroma subsampling. Images are created on a Resnet.

dependence of the perceptual distance. However, one can question whether it makes sense to measure distances between, on the one hand, an original, uncompressed image, and on the other hand, an adversarial, compressed image. In practice, a classifier would probably either expect compressed JPEG images, which implies that the original image would be compressed as well, or uncompressed images. Whether to measure the distance between two compressed images or a compressed and an uncompressed one is thus dependent on how the attack is performed practically.

In the following, we will disable the use of chroma subsampling in the attack since we try to measure the adversarial perturbation only. Using chroma subsampling would only slightly shift the curves though, as shown in fig. 4.5.

Quantized vs. Unquantized Coefficients

As described in section 2.2.1, most of the compression takes place through the quantization of the coefficients. When attacking quantized coefficients, this leads to a significant perceptual distance for lower JPEG qualities even if no adversarial perturbation is applied. Thus, this experiment analyzes whether and when it can be beneficial to attack unquantized coefficients. So, we work on coefficients that are computed in the same way as explained in section 2.2.1, but without the rounding in eq. (2.25).



(a) Resnet (White-Box) (b) Densenet^{jq75} (c) Densenet^{jq50}

Figure 4.6: CIFAR10 - Success rates on various nets for unmasked BIM on the luma channel in dependence of the LPIPS distance. The attacks are performed on quantized or unquantized coefficients, and for two jpeg qualities. Images are created on a Resnet.

Figure 4.6 illustrates the effectiveness of unmasked JPEG attacks on unquantized and quantized coefficients for jpeg qualities 75 and 100, in the white-box setting (fig. 4.6a) and in the black-box settings for two models defended with JPEG compression (figs. 4.6b and 4.6c, respectively). In the white-box setting, the attack on unquantized coefficients of quality 75 is most successful, and surprisingly, more successful than the attack on unquantized coefficients of quality 100 as well. Possibly, this is due to the distribution of perturbations on frequencies: From the computation of the absolute perturbation budgets in eq. (3.2) follows that the absolute budgets on higher frequencies that have amplitude 0 are set to 1. Thus, the absolute perturbation budgets decrease for low frequencies when the JPEG quality is lowered but often remain 1 on high frequencies. Therefore, perturbations should be more concentrated in high frequencies, which might increase the success on the undefended net as we will analyze later. The same effect should apply to quantized coefficients of quality 75 as well, but here, the quantization leads to a significant shift on the x-axis already.

For the black-box setting, we measured success rates on two models defended with JPEG compression, one with quality 75 and one with quality 50. For attacks using quality 100, the difference is only minimal. When using quality 75 in the attack, the perceptual difference that results from the quantization step significantly shifts the curves on the x-axis. Surprisingly, even when using JPEG compression in defense, the use of quantized coefficients does not necessarily imply a higher success rate. Only when the quality used in defense is low enough (50), the attack on quantized coefficients is more successful,

and only for bigger perturbations. As in the previous experiment, we have to mention that it is difficult to compare compressed (quantized) and uncompressed (unquantized) data, as a classifier would expect either compressed or uncompressed data in a practical setting.

If only the relationship between success and distance is to be optimized, these results imply that unquantized coefficients should always be used. However, if a classifier expects compressed data in practice, the distance between the uncompressed original and the compressed adversarial image becomes irrelevant and, quantized coefficients should be used. As this work is about JPEG coefficients, we will focus on quantized coefficients in the following.

Fast Adversarial Rounding

Until now we rounded the coefficients to the nearest integer at the end of the attack. Rounding at least some coefficients in the gradient's direction using Shi et al.'s fast adversarial rounding [85] might lead to bigger success though. Figure fig. 4.7 compares JPEG attacks without fast adversarial rounding to those where fast adversarial rounding is enabled. For FGSM, fast adversarial rounding slightly improves the attack's efficiency. For BIM, however, both versions are equally successful. Possibly, this is because the coefficients are already near to their local optimum using the Basic Iterative Method. Thus, the fast adversarial rounding rounds most of the gradients to the nearest integer anyway. Using the one-step method FGSM, the coefficients are usually not as close to their local optimum. Therefore, the fast adversarial rounding makes more of a difference here.

It should be mentioned that we only experimented using one η value, 0.05, and there might be other choices that slightly improve the attacks' success, especially for FGSM. However, we would not expect a significant improvement from these results. For that reason, we disable the fast adversarial rounding in our JPEG attacks in the following experiments.

Varying perturbations across frequencies

Being able to control how the perturbations are distributed across frequencies was a major motivation for proposing JPEG attacks. Here, we will analyze on which frequencies

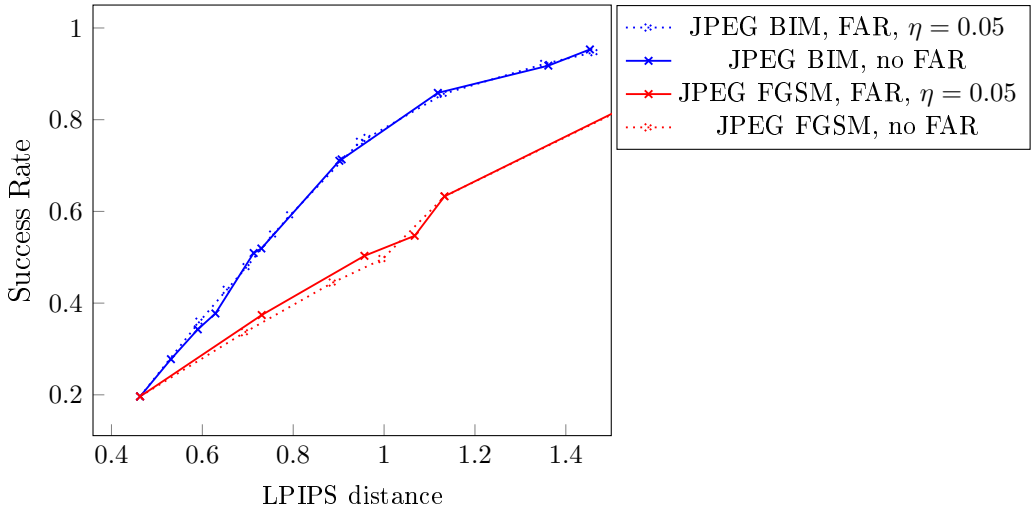
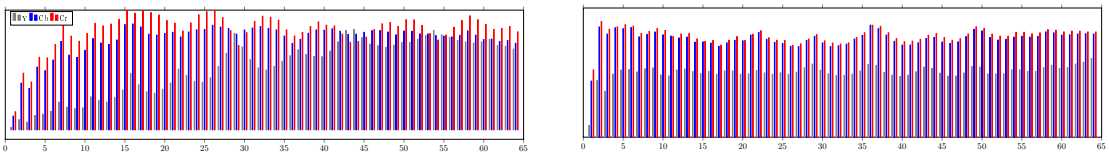


Figure 4.7: CIFAR10 - Success rates for unmasked JPEG attacks in dependence of the LPIPS distance on the Densenet^{jq50}, with and without fast adversarial rounding (FAR). Images are created on a Resnet.

the perturbations must be concentrated to receive the most efficient attack on different models.

Usually, adversarial attacks are assumed to be a high-frequency phenomenon as adversarial perturbations on RGB images are concentrated in high frequencies and, thus, as high frequencies are known to be less important than lower frequencies for human perception, one could argue that adversarial attacks that mainly perturb high frequencies could change a net's classification without being visible to humans as they could be perceived as noise and therefore be more efficient. This does however assume that for neural nets higher frequencies are more important for the classification process than for human perception. Otherwise, small perturbations made on high frequencies might not impact the net's classification significantly. Indeed, Yin et al. have shown that neural nets can rely on high-frequency components that are barely visible for human beings [99], However, which frequencies the net relies on depends on the dataset and how the net was trained. For example, adversarial training can lead to the model relying more on low frequencies [66, 99].

The previous works [8, 36, 66, 84] that analyzed the success of adversarial attacks on different frequencies only used the input parameter ε and L_2 distances to compare the success but did not quantify the perceptual distance of attacks on different frequencies. Thus, it also seems possible that perturbations that are concentrated in lower frequen-



(a) Densenet (b) Densenet$^{RGB}_{M}$

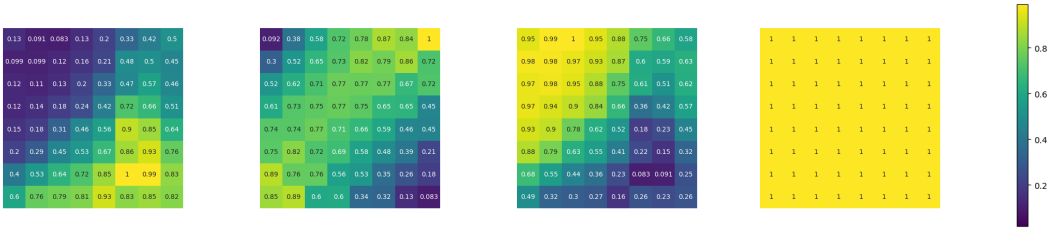
Figure 4.8: CIFAR10 - Average relative perturbations made by RGB BIM ($\varepsilon = 8$) on JPEG frequencies for the CIFAR10 dataset on different source models. The numbers on the x-axis describe the post zig-zag order of frequencies. The attacked images and the original were converted to JPEG images and the relative perturbation is the given by $|\frac{Y'-Y}{Y+1}|$ for luma and correspondingly for chroma channels.

cies are actually more perceptually similar, given that low-frequency components are perceived as less prominent when high-frequency components are visible at the same time, which is the basis of the hybrid images [70] mentioned before. This could lead to higher efficiency when stronger perturbations are made on low frequencies. Thus, we believe the analysis of the effect of controlling the perturbations across frequencies could yield interesting results, especially in combination with using a distance metric that is not pixel-based.

Additionally, we hope that low- or medium-frequency attacks are capable of bypassing JPEG compression used in defense, as JPEG compression predominantly eliminates high-frequency perturbations.

First, we will analyze how perturbations made by RGB BIM are distributed across frequencies. Note that similar experiments have been mentioned in the related work in section 2.5 already.

Figure 4.8 illustrates the relative perturbation on each frequency made by RGB BIM when images are crafted on the undefended or on the the adversarially trained Densenet. The undefended net is sensitive towards high-frequency perturbations for CIFAR10. Thus, the perturbations are concentrated in high frequencies. The defended net, however, is used to high-frequency perturbations and is therefore more reliant on low frequencies which leads to more sensitivity on low frequencies. In the chroma channels, the perturbations are basically uniformly distributed when crafted on the adversarially trained net. In the luma channel, the general structure is unchanged as the most relative perturbations are still concentrated in high frequencies, but the ascent is much smaller. This experiment complements recent findings made by, e.g. Yin et al. [99] and Maiya et al. [66]



(a) qm ascent (b) medium (c) qm descent (d) unmasked

Figure 4.9: Frequency weighting vectors (λ). Unzigzagged.

who state that adversarial training might lead to robustness on high, but vulnerability on low frequencies, at least for CIFAR10.

To examine how these results transfer to our JPEG attack and how they could be used to achieve better robustness with adversarial training, we further analyze the success of attacks using different frequency weighting vectors. We use the following weighting vectors:

- unmasked, $\lambda = (1, \ldots, 1)$,
- qm ascent, which is based on the quantization matrix[4] for quality 50, and computed by dividing each entry by the maximum entry, $\lambda = \text{zigzag}(\frac{Q}{\max(Q)})$,
- qm descent, computed by

$$v = 1 + \min(Q) - Q$$
$$\lambda = \text{zigzag}(\frac{v}{\max(v)}),$$

- medium, which concentrates the perturbations in medium frequencies. When using absolute perturbations, the linear descent masking vector was the most successful one. As absolute coefficients are usually highest for low frequencies, this led to perturbations that were concentrated in medium frequencies. We extracted the resulting average relative perturbations, which resulted in this medium masking vector.

The weighting vectors are illustrated in fig. 4.9. Figure 4.10 shows how the selection of the masking vectors affects the actual relative perturbation on the coefficients. In comparison

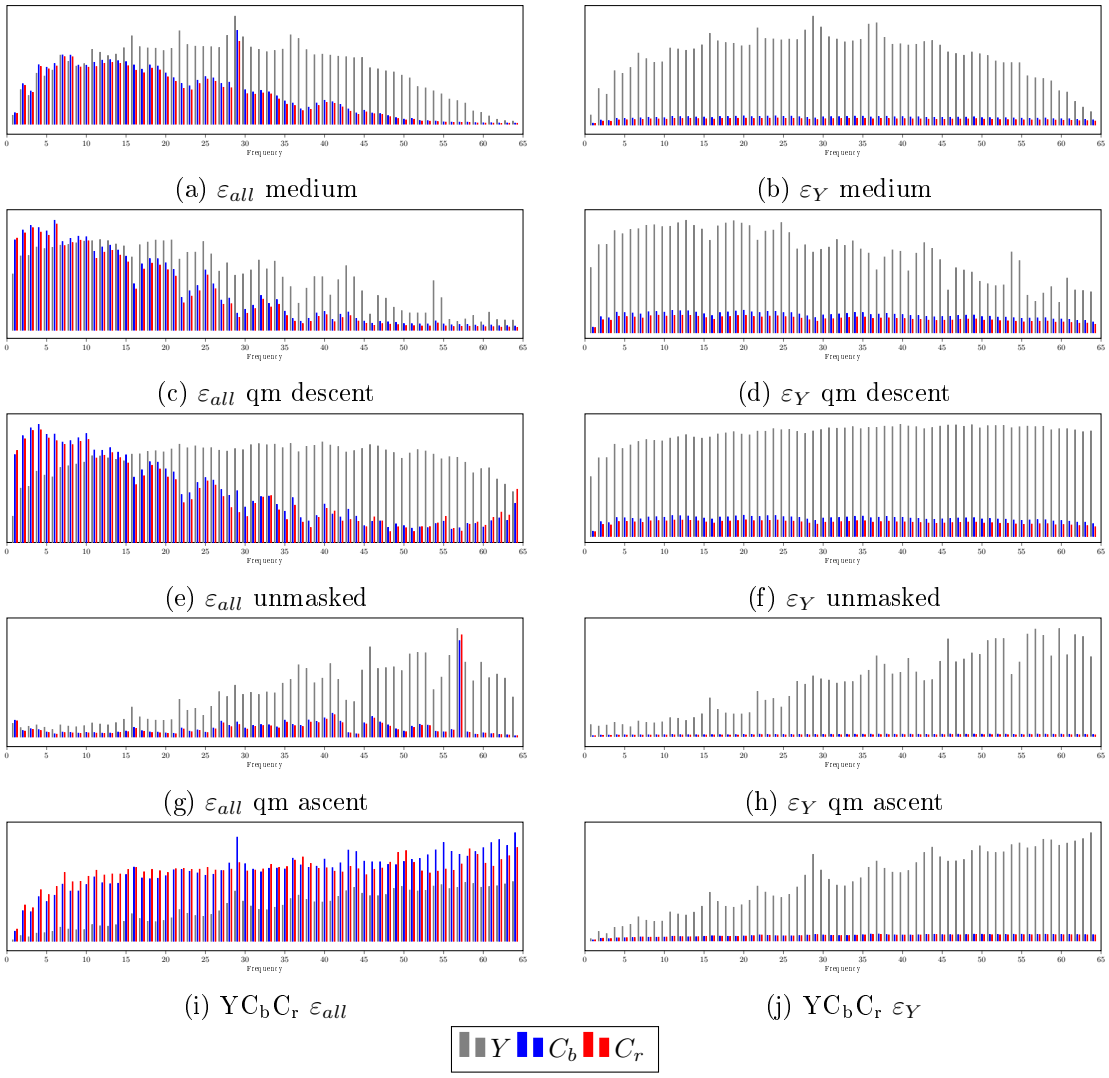
[4] We use the luma quantization matrix for all three channels.

with RGB attacks (see fig. 4.8), where the distribution of perturbations across frequencies depends on the source model, it is now manually controllable.

Figures 4.11 and 4.12 compare the efficiency of each of these masking vectors in the black-box setting, when used for luma attacks[5]. Figure 4.11 shows the efficiency regarding the LPIPS distance, fig. 4.12 uses the CIEDE2000 L_2 distance. Each figure includes the results for CIFAR10 and IMAGENET. We start with analyzing the results for CIFAR10. On the undefended Densenet, applying stronger perturbations to high frequencies results in the highest efficiency for both LPIPS and CIEDE2000 L_2. Stronger perturbations on medium frequencies yield the second-strongest attack. Perturbations that are concentrated in low frequencies are the least effective in this setting. This corresponds to the results from the analysis of the perturbations of the RGB attacks from above. The undefended net is most sensitive towards high-frequency perturbations. Therefore, the greatest success follows from these high-frequency perturbations.

Contrary, when JPEG compression is used in defense, concentrating the perturbations in high frequencies is least efficient in dependence of the LPIPS distance. In dependence of the CIEDE2000 L_2 distance, the ascent vector is still more successful than the descent vector. However, the perturbations on medium frequencies are now the most successful in both cases. This behaviour is as expected since the JPEG compression in defense mainly removes high-frequency perturbations. The differences between the used distance metrics imply that the CIEDE2000 L_2 distance is implicitly more sensitive towards low-frequency perturbations. On the adversarially trained net, the results also slightly differ depending on the distance metric used: Using the LPIPS distance, the success of the weighting vectors is inverted compared to the undefended net: Stronger perturbations on low frequencies now yield the efficiency, while perturbations that are concentrated in high frequencies result in the lowest efficiency. Using the CIEDE2000 L_2 distance, the general trend of the results remains the same: The net is now more robust against high-frequency perturbations, but here the most effective frequency weighting vector is the medium vector. So, again, it seems that the LPIPS distance gives less weight to changes at lower frequencies. Possibly, the low-frequency perturbations result in significant differences in the values of certain pixels which are less noticeable when the picture is viewed as a whole. Thus, the CIEDE2000 L_2 distance is quite high, while the LPIPS distance is not, as has been shown in the example in fig. 2.15.

[5] Figure C.3 in appendix C shows the same plots for attacks on all three channels, but the results do not differ significantly.



(a) ε_{all} medium

(b) ε_Y medium

(c) ε_{all} qm descent

(d) ε_Y qm descent

(e) ε_{all} unmasked

(f) ε_Y unmasked

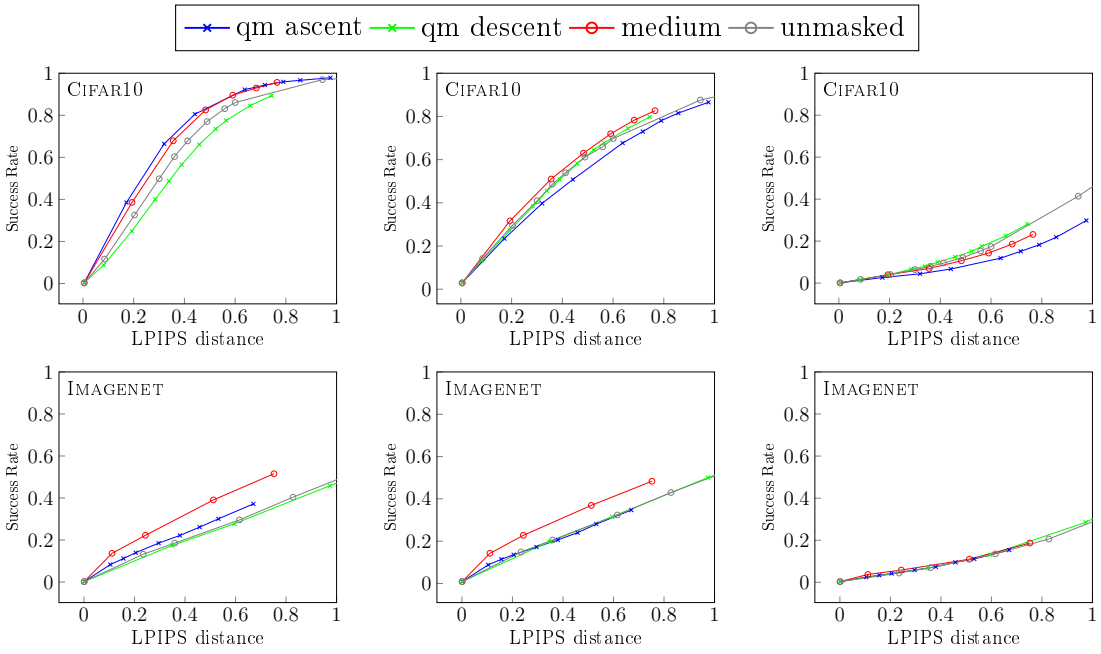
(g) ε_{all} qm ascent

(h) ε_Y qm ascent

(i) $\mathrm{YC_bC_r}$ ε_{all}

(j) $\mathrm{YC_bC_r}$ ε_Y

Figure 4.10: CIFAR10 - Average relative perturbations for JPEG and $\mathrm{YC_bC_r}$ BIM when a relative perturbation budget is used, given by $|\frac{Y'-Y}{Y+1}|$ for luma and correspondingly for chroma channels, images are created on an undefended ResNet. The x-axis represents the frequencies' post-zig-zag order.

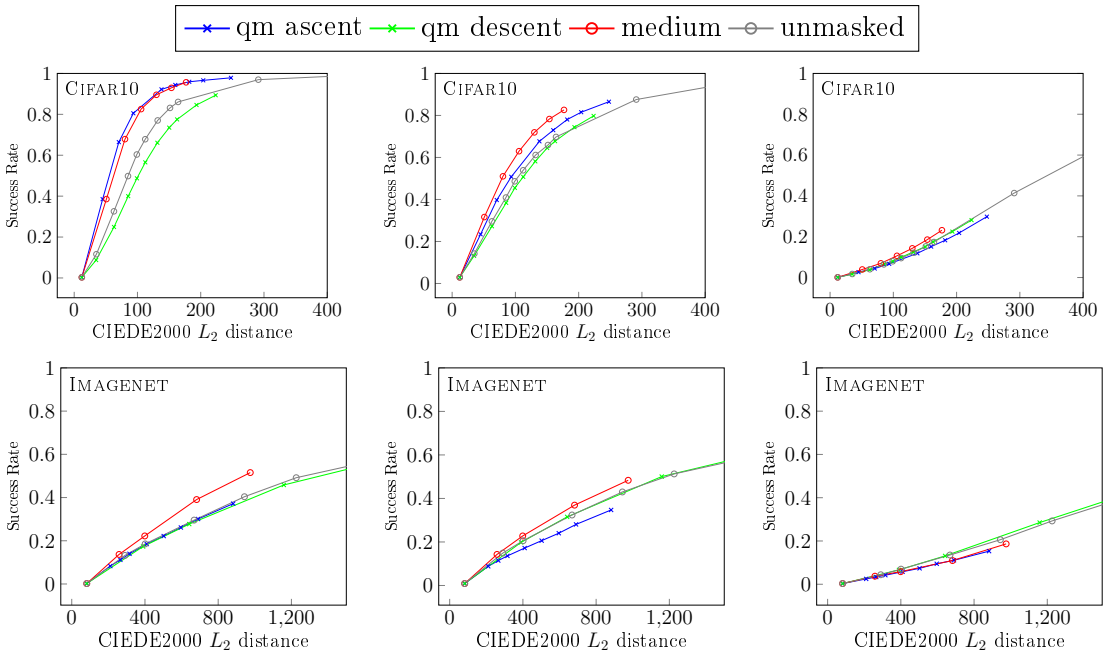


(a) Densenet (b) Densenet^{jq50} (c) Densenet$^{RGB}_{M}$

Figure 4.11: LPIPS efficiency of JPEG BIM luma attacks with different frequency weighting vectors. The weighting vectors are illustrated in fig. 4.9.

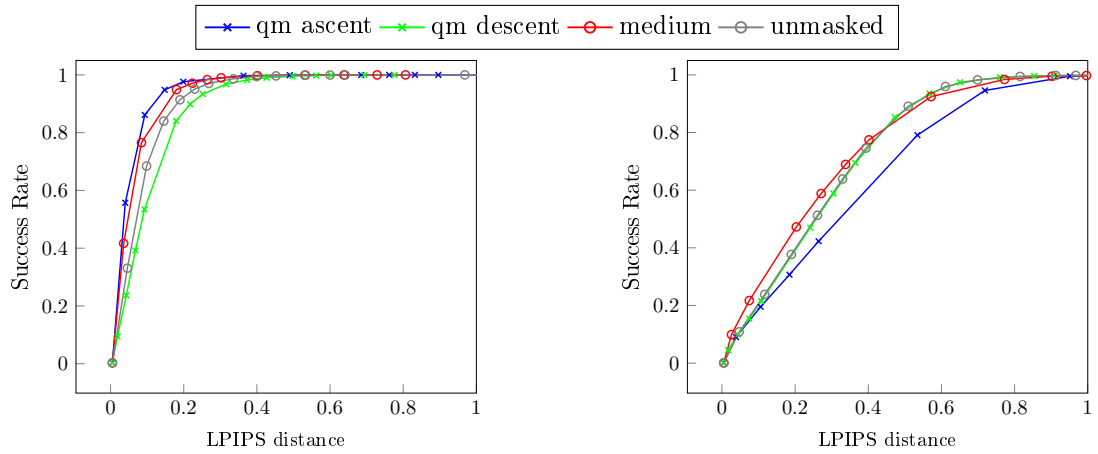
In the white-box setting, we observe equivalent results, as shown in fig. 4.13. Again, the ascent vector is the most successful on the undefended net. On the adversarially trained net, it is the least successful though. In this case, however, the strong perturbations on medium frequencies yield the highest success, even when measuring success in dependence of the LPIPS distance. In general, the relative results should not differ significantly between white-box and black-box settings when the nets are trained on the same original datasets. As explained in section 2.1.2, the existence of adversarial examples is probably due to non-robust, but useful features [45]. Thus, two nets that are trained on the same datasets would be expected to learn similar non-robust features. In the case of CIFAR10, these are located mainly on high frequencies such that classifiers that are trained using the benign CIFAR10 dataset are usually sensitive towards perturbations on high frequencies. So, we can analyze results in the black-box settings and usually draw conclusions for white-box settings as well.

In summary, for CIFAR10, the adversarial training seems to bias the neural net towards using more low-frequency information, such that high-frequency perturbations have less influence on the model's output, but the net is now more vulnerable towards



(a) Densenet (b) Densenet^{jq50} (c) Densenet$^{RGB}_{M}$

Figure 4.12: CIEDE2000 efficiency of JPEG BIM luma attacks with different frequency weighting vectors. The weighting vectors are illustrated in fig. 4.9.



(a) Densenet (b) Densenet$^{RGB}_{M}$

Figure 4.13: CIFAR10 - White-box success rates for JPEG BIM with perturbations on the luma channel only, in dependence of the LPIPS distance.

low-frequency perturbations, as already stated in [66, 99]. It has to be mentioned though that the adversarial training is effective against low-frequency perturbations as well, as the success rate for small and medium perturbations is reduced significantly. This leads us to asking whether a net can be trained to show more robustness on all frequencies, and on which frequencies a net should ideally rely on. In general, a net should rely on the same features as humans do to achieve robustness as a net that relies on exactly the same features as humans would contradict the existence of adversarial examples. As shown in fig. 2.17, the net trained with RGB adversarial training is already close to human perception. Whether controlling perturbations across frequencies when creating the images during adversarial training can lead to robustness across all frequencies and thus, better generalization, will be analyzed in section 4.5.

For IMAGENET, in contrast to CIFAR10, the medium vector is significantly more successful than the ascent vector on the undefended net already (fig. 4.11). This indicates that nets trained with IMAGENET tend to rely more on lower and medium frequencies, and thus they are more sensitive towards perturbations on those frequencies as already pointed out by Maiya et al. [66]. On the net defended with JPEG compression, the ascent attack is slightly less successful, which becomes more visible in dependence of the CIEDE2000 L_2 distance (fig. 4.12). However, when the perturbation is concentrated in medium or low frequencies, JPEG compression is not an effective defense technique at all, at least not when using JPEG quality 50.

On the adversarially trained net, the medium and ascent attacks are similarly unsuccessful. The same applies to the descent attack as well, but in dependence of the CIEDE2000 distance, it is now slightly more successful than the medium and ascent attack. This is surprising as for CIFAR10, the LPIPS distance seemed to put less weight on low-frequency perturbation than the CIEDE2000 distance does implicitly. This could be explained in the sizes of the images. As the LPIPS distance uses a VGG-16 net, trained on IMAGENET, the CIFAR10 images are resized using bilinear interpolation, which could affect the loss network's output.

For both distance metrics we again observe the trend that the adversarial training leads to high robustness on high and medium frequencies, but the net is more vulnerable towards low-frequency perturbations. So, while the Maiya et al.'s [66] argument that sensitivity on each frequency is dataset dependent is correct, adversarial training still seems to bias the net towards using even more low-frequency information, for both CIFAR10 and IMAGENET.

In summary, these experiments show that adversarial perturbations are not necessarily a high-frequency phenomenon, as also stated in by Maiya et al. [66]. In fact, perturbations that are concentrated in medium frequencies are more efficient on all IMAGENET nets considered. The same applies for the CIFAR10 nets defended by JPEG compression and adversarial training, while they are just slightly less efficient than the ascent attack on the undefended net.

Learning Frequency Weighting Vectors

In the previous section, we used very simple frequency weighting vectors. Now, we try to learn an optimal masking vector using a neural network. The net receives a fake input which is just a single 1 and consists of only three dense layers with 64 neurons each, which do not use bias or activations. By using 1 as input for the dense layers, they just return their weights. We denote the weights as $\lambda_Y^{nn}, \lambda_{C_b}^{nn}, \lambda_{C_r}^{nn} \in [0,1]^{64}$ and $\lambda^{nn} = (\lambda_Y^{nn}, \lambda_{C_b}^{nn}, \lambda_{C_r}^{nn})$. Correspondingly, let $\varepsilon = (\varepsilon_Y^{\text{rel}}, \varepsilon_{C_b}^{\text{rel}}, \varepsilon_{C_r}^{\text{rel}})$.

The main target of learning the masking vectors is the maximization of the ratio between the success rate or the model's loss and the perceptual distance, which is in this case measured using the CIEDE2000 L_2 distance. As maximizing the ratio would lead to the distance being reduced by setting all weights to 0, we experiment with two different functions that constraint the weights vectors after each step of gradient descent. Both have in common that they are first clipped to be ≥ 0. Then, the first option is to norm the weights by dividing them by their maximum. The second option is to force them to have a mean higher than some constant, we use 0.5. The model can then be trained in different ways:

The **One-Step Ratio Trainer (OSR)** uses the ratio of the model's loss and the image distance as loss, which is defined as

$$\frac{-\,\mathrm{J}_{\mathrm{TM}}(\mathrm{ADV}_{\mathrm{SM},\lambda^{nn},\varepsilon}(x,y),y)}{d(FGSM_{\lambda^{nn},\varepsilon}(x,y),x)}. \tag{4.1}$$

As for every following trainer, ε is chosen randomly batch-wise and ADV is some adversarial attack that is performed on some source model SM. The loss J can be evaluated on different model TM, but TM = SM is possible too. Then, the lambda model's weights λ^{nn} are updated by descending the losses gradient. Distance function d uses the CIEDE2000

L_2 norm, but divides the CIEDE2000 distances by 120, which is chosen experimentally, before computing the L_2 distance.

The **One-Step Sum Ratio Trainer (OSS)** is a very similar trainer. The only difference is that instead of the ratio, a sum,

$$-\mathrm{J}(\mathrm{ADV}_{\lambda^{nn},\varepsilon}(x,y),y) + d(FGSM_{\lambda^{nn},\varepsilon}(x,y),x). \tag{4.2}$$

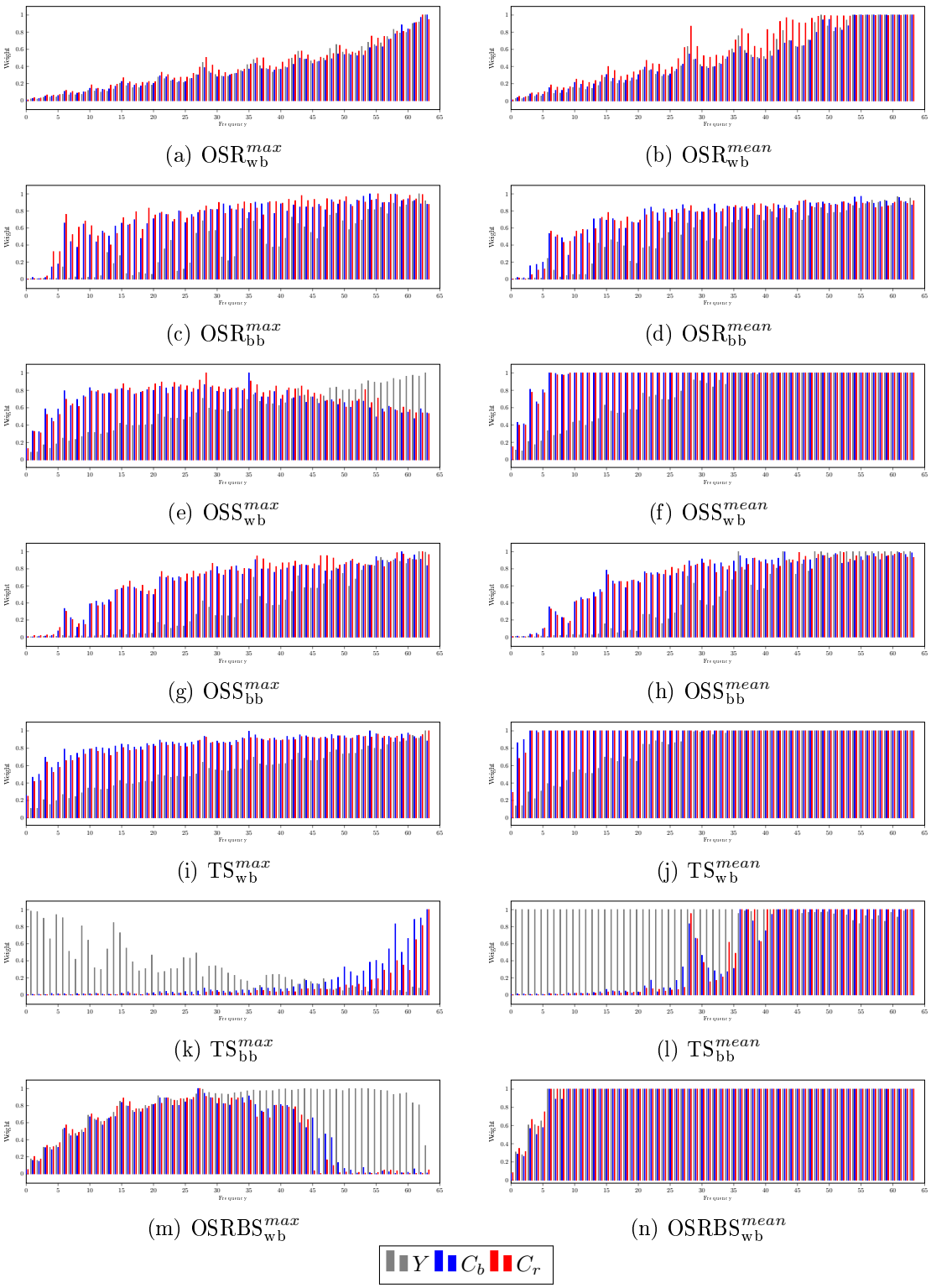
is optimized.

The **Two-Step Trainer (TS)** performs two steps of gradient descent in every iteration, one for ascending the model's loss, and one for descending the image distance. Both steps use the same step size.

We train nets for both an undefended white-box, and a defended black-box setting. For the first case, $TM = SM$ corresponds to an undefended ResNet. In the second case the source model is the undefended ResNet, while the target model is an adversarially defended DenseNet. We mainly use FGSM to create the adversarial images. All FGSM-nets are trained for 10 epochs using batch size 100. Additionally, we use the FGSM binary search for the One-Step Ratio Trainer (**OSRBS**). This could be beneficial as the binary search automatically minimizes the RGB L_∞ distance, but guarantees success. Thus, we can use the One-Step Ratio Trainer without worrying that just the distance is minimized to maximize the relation of success and distance. However, this only works for the white-box settings. The nets using binary search are trained for just 5 epochs with a batch size of 20.

Table 4.4 summarizes how each model was trained and how the models are denoted. First, we will analyze the weights that were learned by each trainer. They are illustrated in fig. 4.14.

The efficiency of each weighting vector is illustrated in fig. 4.15 for all trainers using the max norm, and in fig. 4.16 for all trainers using the mean norm. Because of the high number of trainers and results, we will only discuss the most interesting observations. The success of all white-box trainers is virtually the same on the undefended DenseNet. All of them concentrate the luma perturbation in high frequencies which is where the undefended DenseNet is most vulnerable against, as already discussed. The perturbation on the chroma channels varies between the trainers but is less important for the misclassification and, thus, the success varies only little.



(a) $\mathrm{OSR}_{\mathrm{wb}}^{max}$ (b) $\mathrm{OSR}_{\mathrm{wb}}^{mean}$

(c) $\mathrm{OSR}_{\mathrm{bb}}^{max}$ (d) $\mathrm{OSR}_{\mathrm{bb}}^{mean}$

(e) $\mathrm{OSS}_{\mathrm{wb}}^{max}$ (f) $\mathrm{OSS}_{\mathrm{wb}}^{mean}$

(g) $\mathrm{OSS}_{\mathrm{bb}}^{max}$ (h) $\mathrm{OSS}_{\mathrm{bb}}^{mean}$

(i) $\mathrm{TS}_{\mathrm{wb}}^{max}$ (j) $\mathrm{TS}_{\mathrm{wb}}^{mean}$

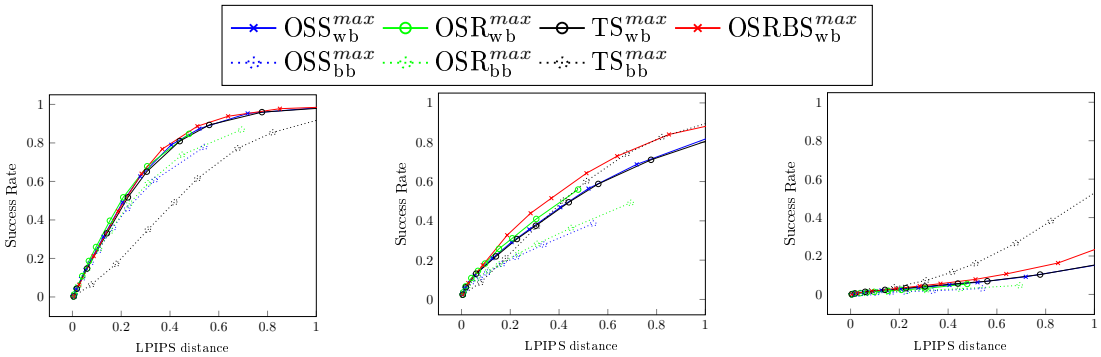
(k) $\mathrm{TS}_{\mathrm{bb}}^{max}$ (l) $\mathrm{TS}_{\mathrm{bb}}^{mean}$

(m) $\mathrm{OSRBS}_{\mathrm{wb}}^{max}$ (n) $\mathrm{OSRBS}_{\mathrm{wb}}^{mean}$

Figure 4.14: Learned frequency weighting vectors.

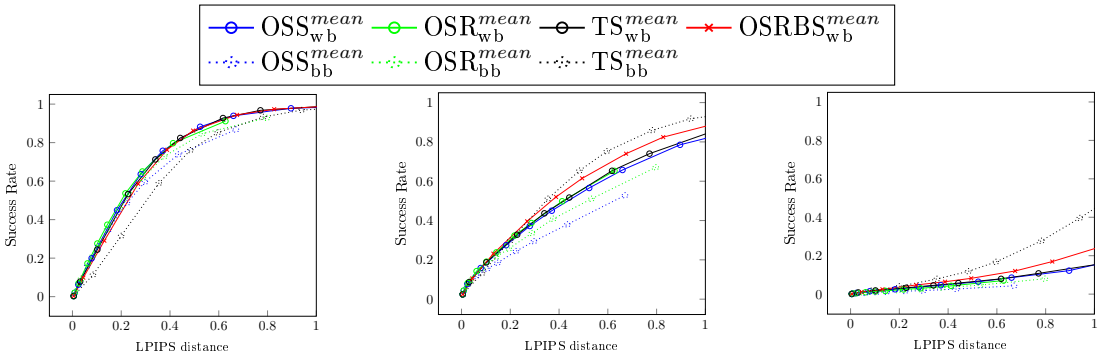
Notation	Attack	Trainer	Norm	Source Model	Target Model
$\text{OSR}_{\text{wb}}^{max}$	FGSM	One-Step Ratio	max	Resnet	Resnet
$\text{OSS}_{\text{wb}}^{max}$	FGSM	One-Step Sum	max	Resnet	Resnet
$\text{TS}_{\text{wb}}^{max}$	FGSM	Two-Step	max	Resnet	Resnet
$\text{OSR}_{\text{wb}}^{mean}$	FGSM	One-Step Ratio	mean	Resnet	Resnet
$\text{OSS}_{\text{wb}}^{mean}$	FGSM	One-Step Sum	mean	Resnet	Resnet
$\text{TS}_{\text{wb}}^{mean}$	FGSM	Two-Step	mean	Resnet	Resnet
$\text{OSR}_{\text{bb}}^{max}$	FGSM	One-Step Ratio	mean	Resnet	$\text{Densenet}_{M}^{RGB}$
$\text{OSS}_{\text{bb}}^{max}$	FGSM	One-Step Sum	mean	Resnet	$\text{Densenet}_{M}^{RGB}$
$\text{TS}_{\text{bb}}^{max}$	FGSM	Two-Step	mean	Resnet	$\text{Densenet}_{M}^{RGB}$
$\text{OSR}_{\text{bb}}^{mean}$	FGSM	One-Step Ratio	mean	Resnet	$\text{Densenet}_{M}^{RGB}$
$\text{OSS}_{\text{bb}}^{mean}$	FGSM	One-Step Sum	mean	Resnet	$\text{Densenet}_{M}^{RGB}$
$\text{TS}_{\text{bb}}^{mean}$	FGSM	Two-Step	mean	Resnet	$\text{Densenet}_{M}^{RGB}$
$\text{OSRBS}_{\text{wb}}^{max}$	FGSM binary search	One-Step Ratio	max	Resnet	Resnet
$\text{OSRBS}_{\text{wb}}^{mean}$	FGSM binary search	One-Step Ratio	mean	Resnet	Resnet

Table 4.4: Notation of learned frequency weighting vectors.



(a) Densenet (b) Densenet^{jq75} (c) $\text{Densenet}_{M}^{RGB}$

Figure 4.15: CIFAR10 - LPIPS efficiency of learned frequency weighting vectors for ε_{all} attacks. All trainers that used the max norm are shown here, all that used the mean norm are shown in fig. 4.16.



(a) Densenet (b) Densenet^{jq75} (c) Densenet$_{M}^{RGB}$

Figure 4.16: CIFAR10 - LPIPS efficiency of learned frequency weighting vectors for ε_{all} attacks. All trainers that used the max mean are shown here, all that used the max norm are shown in fig. 4.15.

From the trainers trained for the white-box setting, the binary search trainers are the most efficient, for all three target models. Here, we do not have the problem that only the distance or the success is optimized, as the binary-search attack implicitly chooses the perturbation that causes minimal L_∞ distortion to find a misclassification. Despite using one of the norms, using the One-Step Sum/Ratio Trainers could still favour one over the other.

The Two-Step Trainer does not have these shortcomings as well. When trained for the black-box setting, its efficiency is the highest on the adversarially trained net for both norms and the highest on the net defended with JPEG compression when the mean norm is used. It stands out that for the black-box two-step trainers the learned weights vary significantly across the channels: While on the luma channel the weights are the highest on the lowest frequencies, they are the highest on the higher frequencies for both chroma channels. The former is in accordance to the results from the previous section that showed that the adversarially trained net is vulnerable on low frequencies. But apparently, this vulnerability is exclusively or predominantly on the luma channel. So, using different weighting vectors for luma and chroma channels, contrary to what we did until now, might lead to even higher success.

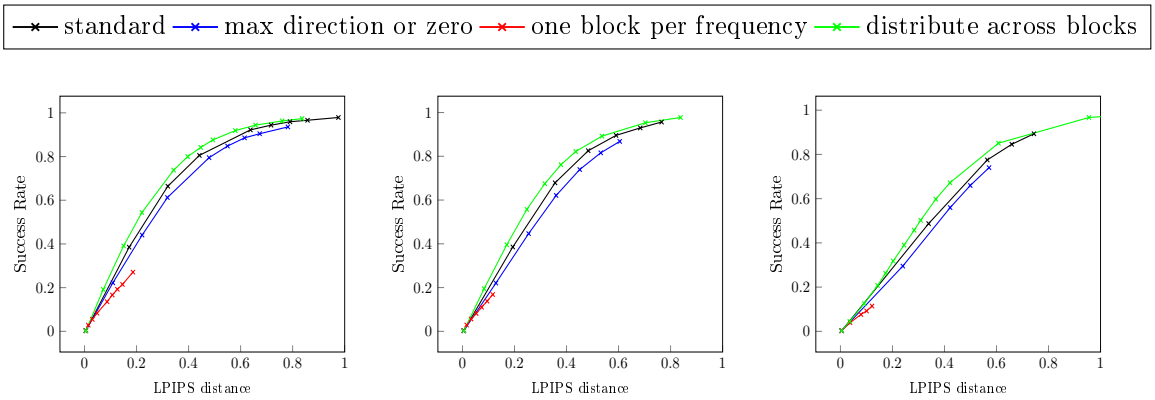
Trying to improve perceptual similarity

A problem of our attacks is that the created images sometimes contain clearly visible JPEG blocks. Usually, this applies to images of a low JPEG quality. But when the coefficients are perturbed, this is also the case when using JPEG quality 100. As fig. 4.17 shows, these blocks mainly occur when lower frequencies are perturbed.

Figure 4.17: Minimum perturbation required for a misclassification by the Densenet^{jq50}. Images are created on a Resnet. Some images show clearly visible 8×8 jpeg blocks.

We make various attempts to overcome this problem. First, we aim to reduce the perceived distortion in regions of the image where it does not significantly affect the prediction. As our attack uses the gradient's sign only and not its absolute value, coefficients in "*irrelevant*" regions of the image are perturbed using the same relative step sizes as in more important regions. This becomes problematic especially when the same frequency is perturbed in different directions for neighbouring blocks. In a first attempt to reduce this distortion, we try to ensure that frequencies are not perturbed in different directions across the blocks. For that, we set every gradient that has a sign that is different from the one of the coefficient with the maximum absolute value for the same frequency to 0. Thus, for every frequency, every gradient is perturbed in the direction of the coefficient with the maximum absoulte value or it is not perturbed at all. We denote this method as *max direction or zero*.

A second attempt (*one block per frequency*) is to only perturb one block per frequency in each iteration. Frequency-wise, the block with the highest absolute value is perturbed, while the other gradients are set to 0 again. Hereby, we try to avoid perturbations in regions of the image where it barely affects the prediction.

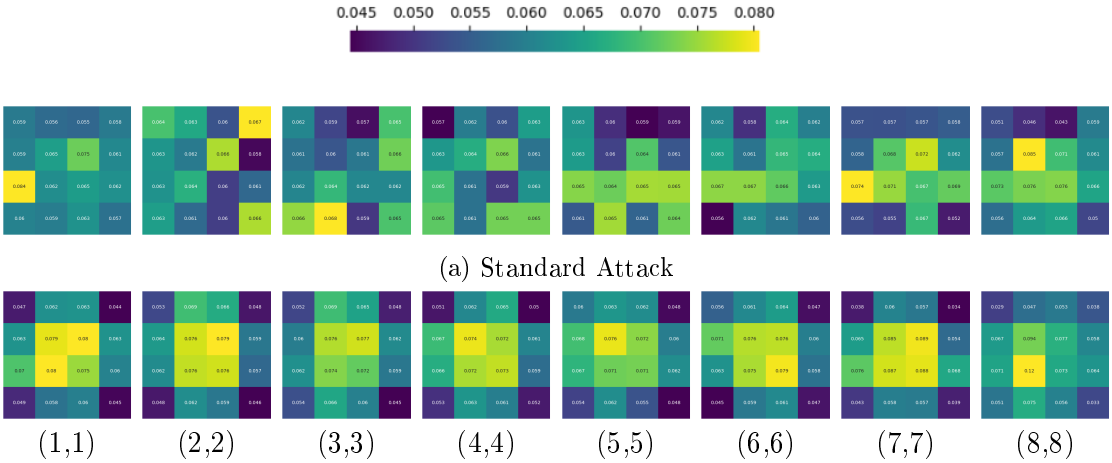


(a) qm ascent (b) medium (c) qm descent

Figure 4.18: CIFAR10 - LPIPS efficiency on an undefended Densenet for attempts to norm gradients to reduce perceived distortion.

In a third attempt (*distribute across blocks*), which is very different from the others, we do not use the sign of the gradient for perturbation, but its normed values. To avoid strong perturbation in "*irrelevant*" regions of the image, we normalize the gradients in the block dimension by dividing by the maximum for each frequency. So, the λ vector is still used to distribute perturbations across frequencies, but now, the perturbation is automatically distributed across the JPEG blocks using the values of the gradient.

Figure 4.18 compares the three attempts to the standard attack. The first two approaches, *max direction or zero* and *one block per frequency* are less efficient than the standard attack. The limitations on the attack's action space seem to be too strong for efficient perturbations. The third approach is slightly more effective than the standard attacks, for all three weighted vectors. Here, the action space is not limited at all. In fact, it is bigger than for the standard attack as the step size is adjusted automatically based on the gradient's absolute value. Thereby, the attack can reduce the distortions in regions of the image that are less relevant for the prediction. Figure 4.19 illustrates in which blocks the standard attack and the *distribute across blocks* attack performs the most relative perturbation. The *distribute across blocks* attack concentrated the perturbation in central regions of the images, while the standard attacks distributes them uniformly across the JPEG blocks. As the relevant object is usually located in the center of the images, the *distribute across blocks* can significantly alter the model's prediction without including irrelevant distortions in the background.

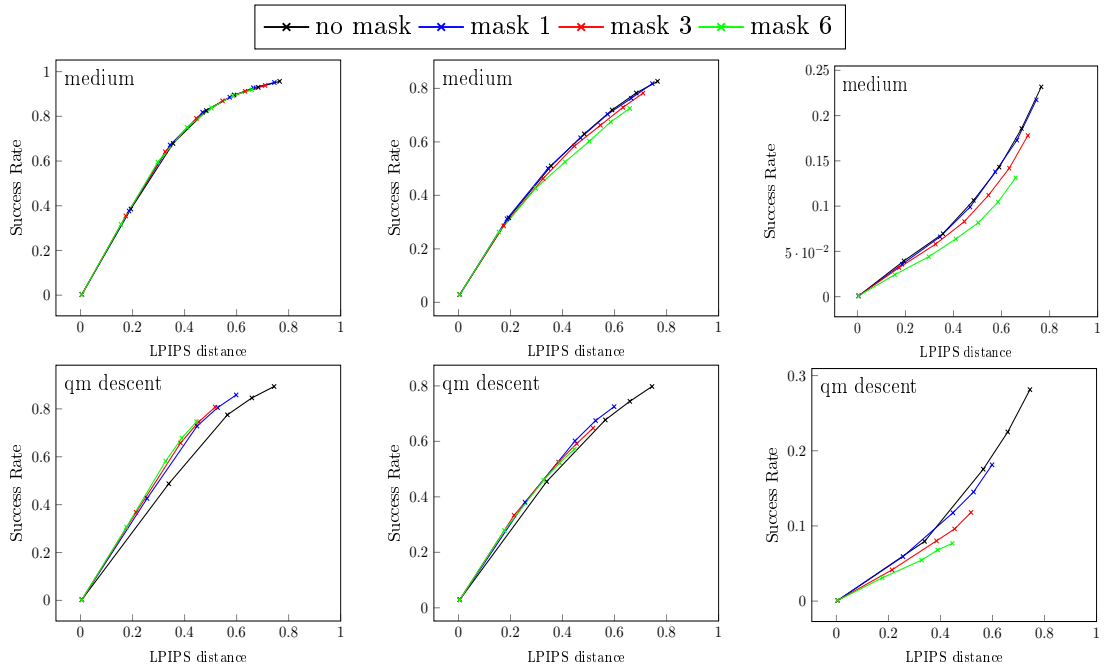


(a) Standard Attack

(b) Distribute Across Blocks

Figure 4.19: CIFAR10 - Distribution of average relative luma perturbation across blocks for every frequency on the main diagonal in fig. 2.8a. Both vertical and horizontal indices are given below the heatmaps. Size images have size 32×32, thus there are 4×4 blocks. The attacks were executed on 1000 CIFAR10 images.

Another reason for the visible JPEG blocks can be that too much perturbation is applied to the lowest frequencies. Thus, fig. 4.20 shows an experiment where the first x frequencies (in zig-zag order) where masked when applying the perturbation. However, we cannot make general observations regarding the success of this attempt. The success seems to depend on the target model and which frequencies it is vulnerable on towards perturbations rather than the masking itself: On the undefended net, which is vulnerable towards high-frequency perturbations, masking the lowest frequencies results in higher efficiency, but on the adversarially trained net it results in less efficiency as the model is vulnerable towards perturbation on exactly those low frequencies. On the net defended with JPEG compression, the results depend on the weighting vector that is used, so we cannot make a general statement.



(a) Densenet (b) Densenet^{jq50} (c) Densenet$^{RGB}_{M}$

Figure 4.20: CIFAR10 - LPIPS efficiency on an undefended Densenet when masking perturbations on the lowest frequencies. Results are shown for the medium (top) and the qm descent (bottom) weighting vector.

Figure 4.21: Minimum perturbation required for a misclassification by the Densenet^{jq50}. Images have been created on a Resnet. The medium weighting vector was used for all of the images. Perturbations are limited to the luma channel. *mdoz* := *max direction or zero*, *obpf* := *one block per frequency*, *dab* := *distribute across blocks*.

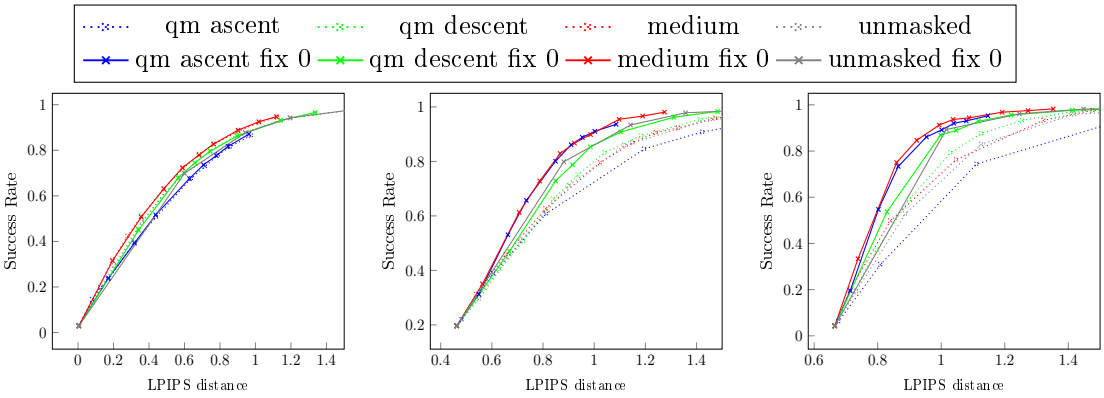
Figure 4.21 shows some sample images for all attemps discussed in this section. Indeed, distributing the perturbation across blocks can reduce the perceived distortion in comparison to the standard attack significantly as it avoids unnecessary perturbation in the background which is especially clear in the top left of the second image where the JPEG block is much less visible. Despite its efficiency, we do not use the *distribute across blocks* attack in the following, but the standard attack. In this example, masking the lowest frequencies does indeed increase the perceptual similarity. However, as we have seen in the quantitative results this is dependent on the target model and not a general fact.

Trying to bypass JPEG compression

In this section, we will try to find ideal parameters for bypassing JPEG compression in defense. As we have already shown that luma attacks are stronger when JPEG compression is used in defense, we will focus on those.

As mentioned in chapter 3, we use two alternative ways of computing absolute from relative perturbation budgets. Here, we will compare these two alternatives. Figure 4.22 illustrates the experiment for three JPEG qualities (100, 75, 50) used in attack. The defense uses JPEG compression with JPEG quality 50. When not fixing coefficients that have value 0 (the dotted lines), the ascent attack is generally the weakest one, especially for lower qualities. For JPEG quality 100, the medium masking vector is still the most successful. The lower the JPEG quality used in the attack is, the more successful the descent vector becomes, though. We think this is because of how we compute absolute perturbation budgets from relative ones, because the perturbation budgets for high frequencies increase in relative terms as the JPEG qualities increase. The same explanation applies to the decrease of the ascent vector's success, as the perturbations concentrate even more in high frequencies, such that they are removed during the JPEG compression in defense.

In comparison to the attacks with fixed 0-coefficients, there is barely any differences in efficiency between the two settings for JPEG quality 100. For lower qualities though, the success rate for all weighting vectors is improved significantly when fixing coefficients of value 0. And the medium and ascent masking vectors remain the most successful ones for lower qualities. Changing coefficient values that were 0 obviously has a big influence on the perceptual distance, especially on low JPEG qualities. Thus, we assume that fixing those leads to less perceptual distance and more efficiency of the attack. Additionally,



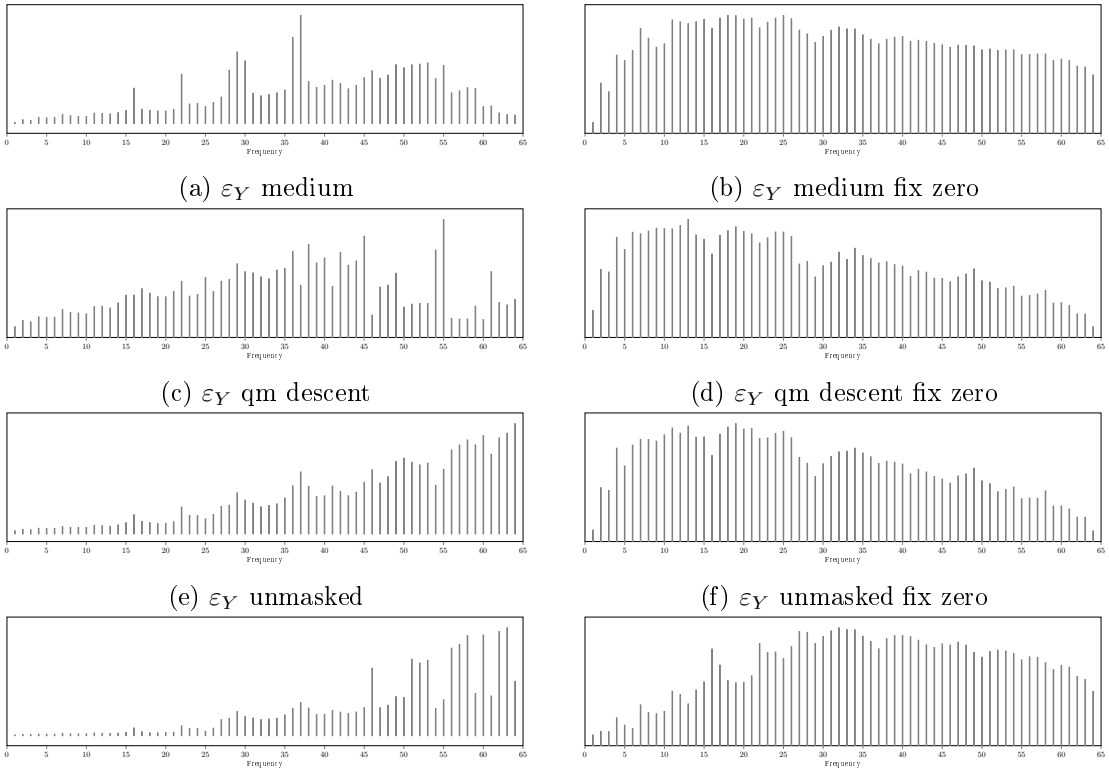
(a) jq=100 (b) jq=75 (c) jq=50

Figure 4.22: CIFAR10 - Black-box success rates on the Densenet^{jq50} for JPEG BIM with perturbations on the luma channel only, in dependence of LPIPS distance, for different JPEG qualities. The attack qualities (100, 75, 50) vary between the subfigures. For the solid lines, zero coefficients are fixed (eq. (3.3) is used), while for the dotted lines, eq. (3.2) is used. Images were created on a Resnet.

the effect that perturbations are implicitly concentrated in high frequencies due to the computation of absolute perturbation budgets with eq. (3.2) is prevented by fixing 0-coefficients. Therefore, the distribution of the perturbations across frequencies is closer to what was intended by the weighting vector λ. And, since the undefended Densenet is sensitive towards high- and medium-frequency perturbation, the ascent and medium weighting vectors are the most successful ones.

When fixing 0-coefficients, it is interesting that the ascent vector is less successful than the descent vector for JPEG quality 100 but more successful for lower qualities. Presumably, this is because the perturbation made on high frequencies for JPEG quality 100 is removed in defense by the JPEG compression anyway. On lower qualities, these perturbations are not made by the attack as high-frequency coefficients often have value 0. Thus, these unnecessary perturbations are not being made.

To verify whether fixing 0-coefficients does lead to a distribution of the perturbation that is closer to the weighting vector, we illustrate the average relative luma perturbation when fixing 0-coefficients compared to when they are not fixed in fig. 4.23. JPEG quality 75 was used in attack in this example. Indeed, not fixing 0-coefficients leads to the perturbation being concentrated in higher frequencies than intended by the weighting vector λ_Y. However, fixing 0-coefficients does not exactly correspond to the weighting



(a) ε_Y medium

(b) ε_Y medium fix zero

(c) ε_Y qm descent

(d) ε_Y qm descent fix zero

(e) ε_Y unmasked

(f) ε_Y unmasked fix zero

(g) ε_Y qm ascent

(h) ε_Y qm ascent fix zero

Figure 4.23: Average relative luma perturbations when a relative perturbation budget is used, given by $|\frac{Y'-Y}{Y+1}|$ for luma and correspondingly for chroma channels, made by JPEG and $\mathrm{YC_bC_r}$ BIM on JPEG frequencies for CIFAR10 on an undefended ResNet. Quality 75 was used in attack. On the left, eq. (3.2) is used, i.e. 0-coefficients are not fixed, while on the right, eq. (3.3) is used such that they are fixed. The x-axis represents the frequencies' post zig-zag order.

vector as well. In fact, it shifts the perturbation towards lower or medium frequencies. Presumably, this is because many of the high-frequency coefficients have value 0 for lower JPEG qualities and since they cannot be changed, the average perturbation decreases on those frequencies. Although the perturbation still does not perfectly correspond with the weighting vectors for lower JPEG qualities, we will use this version in the following experiments, as it is closer to the intended distribution, increases the success and we cannot come up with a better solution since having many 0-coefficients for lower JPEG qualities will always distort the distribution of perturbations.

4.3.2 Comparison with state-of-the-art attacks

In section 4.3.1, we tried to find the best parameters for our attacks. Now, we will compare our most successful attacks to state-of-the-art attacks: First, we will compare with the corresponding RGB and YC_bC_r attacks, and then, with the attacks that try to bypass JPEG compression, proposed by Shin and Song [86] and Shi et al. [85]. As it yielded superior results, we use eq. (3.3) to compute the absolute perturbation budgets, i.e., coefficients that have an amplitude of 0 remain unchanged.

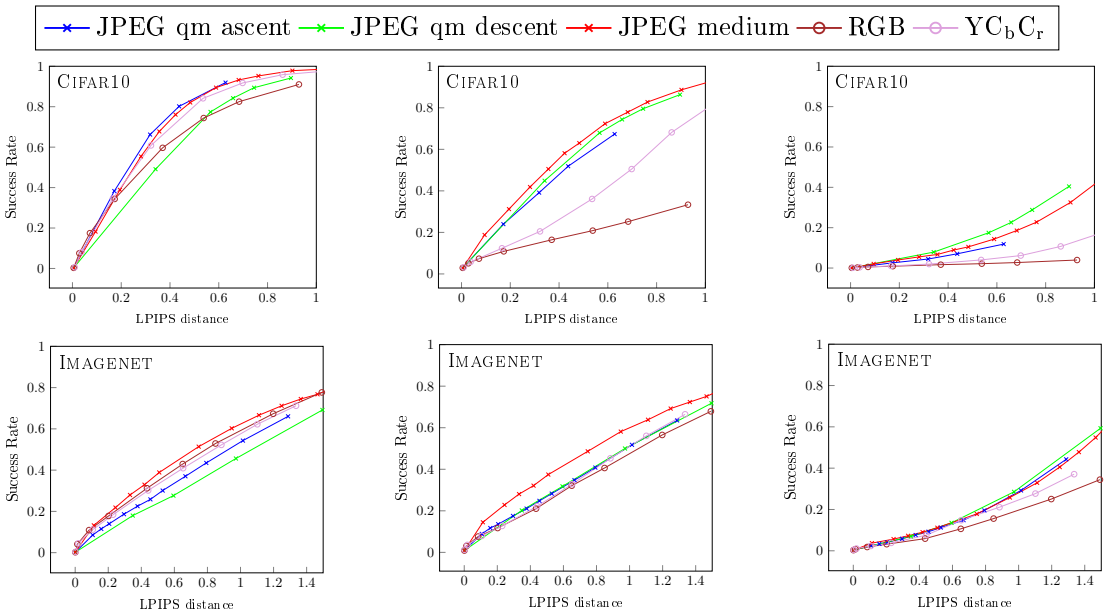
For our most successful attacks, we disable chroma subsampling and fast adversarial rounding. We analyze perturbations on both the luma channel and all three YC_bC_r channels and also use various masking vectors, as their success depends on the model.

As our attacks return compressed JPEG data, while other attacks yield uncompressed RGB data, the distances between the original images and the adversarial images created by our or, respectively those, attacks would not be comparable. In a real-world scenario, either compressed or uncompressed data would be expected. To make the attacks comparable, we compress the data from the standard RGB and YC_bC_r attacks as well as [86]'s attacks to the same JPEG quality. Chroma subsampling is not used either for the purpose of a fair comparison.

Comparison with RGB and YC_bC_r attacks

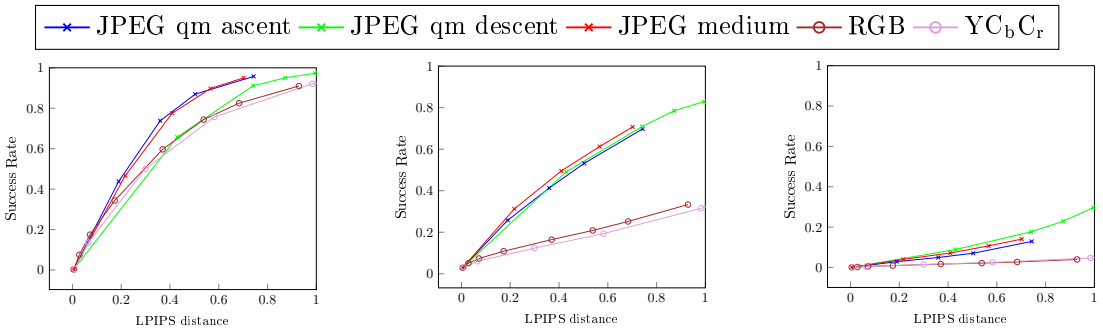
This section will compare the efficiency of our JPEG attacks with RGB and YC_bC_r pixel attacks and thus analyze whether JPEG coefficients are more suited for adversarial attacks than the usually used pixel representations. Figure 4.24 illustrates this comparison for JPEG luma attacks. A comparison with ε_{all}-attacks is shown in fig. 4.25, but only for CIFAR10.

For CIFAR10, we observe that on the undefended nets, both our luma- and all-JPEG attacks are more successful than RGB attacks. Compared to YC_bC_r attacks, our ascent and medium attacks are slightly more successful when attacking luma information only, but significantly more successful when attacking all three channels. As mentioned before, we believe that this is because we use relative perturbation budgets. So, it is possible that YC_bC_r attacks on all three channels are similarly successful when using smaller perturbation budgets for the chroma channels than for the luma channel. Our



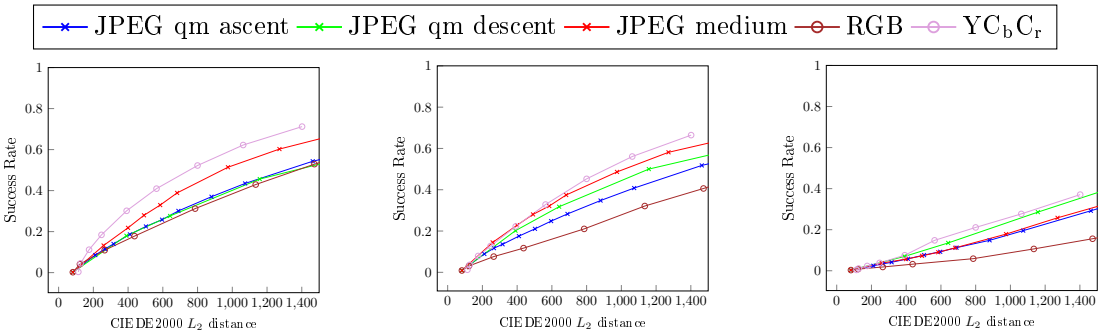
(a) Densenet (b) Densenet^{jq50} (c) Densenet$^{RGB}_{M}$

Figure 4.24: Success rates in dependence of the LPIPS distance for our JPEG BIM attacks (jpeg quality 100), RGB BIM and YC_bC_r BIM. For JPEG and YC_bC_r attacks, only the luma channels are perturbed.



(a) Densenet (b) Densenet^{jq75} (c) Densenet$^{RGB}_{M}$

Figure 4.25: CIFAR10 - Success rates in dependence of the LPIPS distance for our JPEG BIM attacks (jpeg quality 100), RGB BIM and YC_bC_r BIM. For JPEG and YC_bC_r attacks, all three channels are perturbed using the same ε. Images are created on a Resnet.



(a) Densenet (b) Densenet^{jq50} (c) Densenet$^{RGB}_{M}$

Figure 4.26: IMAGENET - Success rates in dependence of the CIEDE2000 L_2 distance for our JPEG BIM attacks (jpeg quality 100), RGB BIM and YC_bC_r BIM. For JPEG and YC_bC_r attacks, only the luma channels are perturbed. Images are created on an undefended Resnet.

relative perturbation budgets implicitly use the fact that color information is less important for human perception and, for neural networks, as the perturbation is relative to the generally smaller amplitudes of chroma coefficients. When the same absolute pixel perturbations are applied to all three YC_bC_r channels, the color information is perturbed disproportionately strong. As mentioned before, neural nets mainly use the information from the luma channel, shapes and textures for classification [30, 74]. So, strongly perturbing all three YC_bC_r channels leads to color distortions that are not necessary for an incorrect classification and, thus, an inefficient attack. The relative JPEG perturbations can overcome this problem without a sophisticated selection of perturbation budgets.

The same reason also explains the superiority of both JPEG and YC_bC_r luma attacks over RGB attacks: When attacking in JPEG or YCbCr representation, attacking only the luma channel can prevent these inefficient distortions. When attacking RGB pixels, though, the shapes and textures are more distributed across the channels such that the perturbation also effects color information that does not need be perturbed. Thus, the perceived distance is higher than needed.

On the net defended with JPEG compression, the difference to YC_bC_r and RGB attacks becomes much more significant. The perturbations created by the RGB and YC_bC_r pixel attacks are now significantly less efficient, while the JPEG attacks still reach similar levels of success as before, meaning that JPEG compression is basically ineffective against them, at least when perturbing medium and low frequencies stronger. There are two reasons why the RGB and YC_bC_r attacks are less effective when JPEG compression

is used in defense: First, for RGB attacks, significant parts of the color perturbations are removed by the chroma subsampling during the JPEG compression anyway, which makes the attack even less efficient. The same effect can be seen with JPEG ε_{all}-attacks in fig. C.3, where the efficiency on the net using JPEG compression in defense is reduced much more from the undefended net than for pure luma attacks (fig. 4.11), and also for YC_bC_r all-attacks: They are much less successful on the net using JPEG compression than luma attacks. The difference is even bigger than for JPEG ε_{all}-attacks, which is due to disproportionately strong perturbations on the chroma channels, which are prevented for JPEG attacks by the relative perturbation budgets, as argued before for the undefended net. But when using JPEG compression, the chroma subsampling removes the perturbation on chroma channels and thus results in the color perturbation being even less efficient. So, independent of which color model is used, color perturbations always seem to be inefficient when JPEG compression, or chroma subsampling, is used in defense. Second, both RGB and YC_bC_r attacks concentrate perturbations in higher frequencies, as already shown in fig. 4.8 and respectively, fig. 4.10. For the most part, those perturbations are removed during JPEG compression as well. The same effect is visible for JPEG ascent attack, for which the decrease of efficiency on the net defended with JPEG compression is much stronger than for descent/medium attacks.

On the adversarially trained net (Densenet_M^{RGB}), we observe that the net has built good robustness against RGB attacks and YC_bC_r pixel attacks on all three channels, but also the luma channel exclusively. Against JPEG attacks though, the robustness is much smaller, and as mentioned before, the net is vulnerable especially against low-frequency perturbations. While this is not a surprising result in general, as adversarially trained networks are known to be vulnerable towards unseen threat models [48, 58], the difference between the efficiency of JPEG and YC_bC_r luma attacks is still interesting as both only perturb the luma information. The reason why the net also builds robustness against YC_bC_r pixel attacks is that they concentrate in high frequencies as well, just as the RGB attacks. Our JPEG attacks can thus circumvent the defense by concentrating the perturbation in medium or lower frequencies, where the adversarially trained net is sensitive towards distortions. Again, that leads to the question whether our JPEG attacks can help to achieve robustness against perturbations on all frequencies and thus, towards RGB and YC_bC_r pixel attacks as well.

For IMAGENET, the LPIPS efficiency is also displayed in fig. 4.24. Additionally, fig. 4.26 shows the CIEDE2000 efficiency. For LPIPS, the JPEG medium attack is the most successful attack in the undefended setting. However, the distance to RGB and YC_bC_r

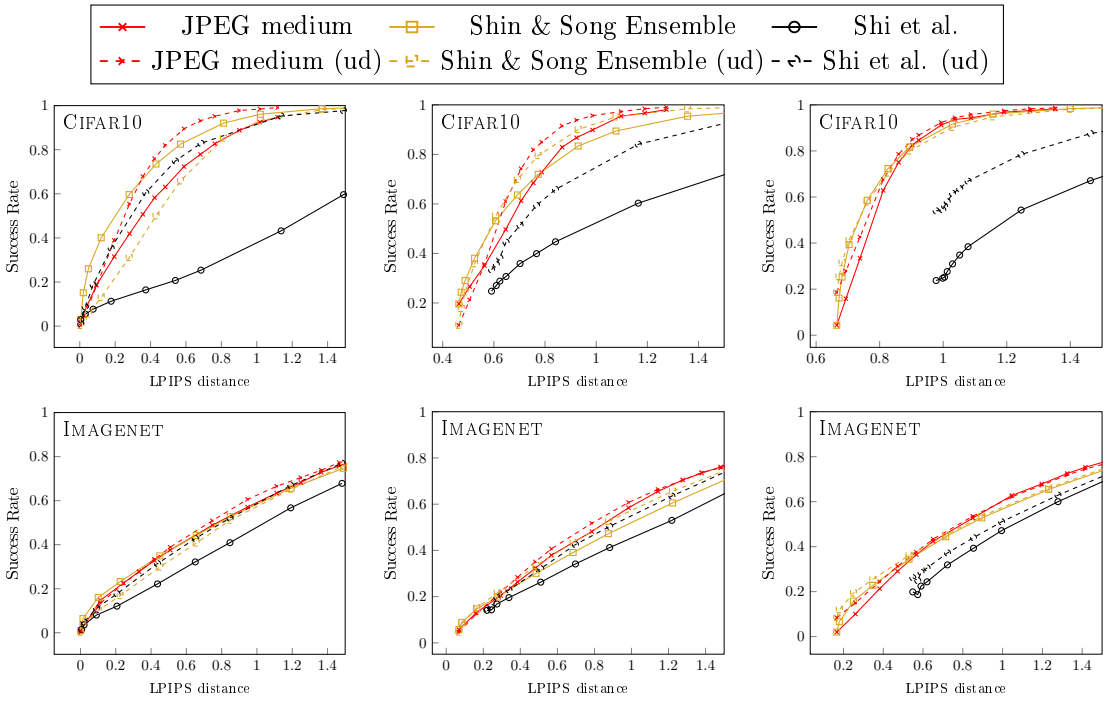
attacks is relatively small. The difference becomes much clearer when the defense uses JPEG compression which partially removes the pixel-wise perturbations, and the RGB attack's color perturbations. On the adversarially trained net, all JPEG attacks are slightly more successful than the pixel-wise RGB and YC_bC_r attacks.

When using the CIEDE2000 L_2 distance for comparison though (fig. 4.26), the YC_bC_r attack is the most efficient, while the RGB attack is even less successful on all three nets. The latter again indicates that the CIEDE2000 L_2 puts more weight to chroma perturbations. The former might be reasoned by the fact that the pixel-based distance is not able to quantify structural changes as illustrated in figs. 2.15 and 2.16. The gray pixel-wise, high-frequency noise might result in only little pixel-based distance but very visible structural changes. While we do not know for sure that LPIPS is better aligned with human perception, the results from section 4.2 and [101] indicate that it is. Thus, we believe that our JPEG medium attack is more successful than the YC_bC_r in dependence of the true perceptual distance for IMAGENET as well.

In summary, due to the superiority regarding the LPIPS efficiency on all three nets, and the results for CIFAR10 as well, it can be stated that our JPEG attacks offer much higher flexibility, as they allow to control perturbations across channels and frequencies. Concentrating the perturbation on medium frequencies seems to be a universal choice and is more efficient than RGB and YC_bC_r pixel attacks in all black-box settings considered. Thus, JPEG coefficients seem to be the superior representation for creating adversarial examples compared to RGB and YC_bC_r pixel representations and it is likely, that that they can also be the basis for more generalising defenses. This will be analyzed later in section 4.5.

Comparison with JPEG-resistant attacks

One of our main motivations for proposing attacks straight on JPEG coefficients was bypassing JPEG compression in defense or when saving the adversarial images. While we already tried to determine how to set up our attacks to bypass JPEG compression in section 4.3.1, we will now compare our attacks to the approaches that specifically try to bypass JPEG compression, proposed by Shin and Song [86] and Shi et al. [85]. Again, one has to remember that some of these attacks yield compressed data (ours, Shi et al.'s) which shifts the plots on the x-axis, while others do not (RGB, YC_bC_r, Shin and Song). Thus, all uncompressed outputs are JPEG compressed to the same quality that



(a) jq=100 (b) jq=75 (c) jq=50

Figure 4.27: LPIPS efficiency on the Densenet^{jq50} and the undefended (ud) DenseNet for BIM attacks, for different JPEG qualities (100, 75, 50) used in the attack.

we use for our attacks (100, 75, 50). Note that for a fair comparison, we also disable chroma subsampling in the output of those attacks. In the internal JPEG compression approximation in Shin and Song's [86] attack, chroma subsampling is enabled however.

As it performs well for all tested JPEG qualities, we will use the medium weighting vector in the following comparison. This comparison is illustrated for both CIFAR10 and IMAGENET in fig. 4.27.

In internal experiments, it stood out that Shin and Song's standard attack is only successful when applied on a net that uses the same quality in defense. Their ensemble attack does indeed circumvent this problem, as already stated by Shin and Song [86]. We thus only include the ensemble attack in the figure to keep things clear. When JPEG compression is used in defense, we observe that their attack is more successful for JPEG quality 100 for CIFAR10. For lower qualities, the efficiency of their attack is higher for small perturbations but for bigger perturbations our attack is advantageous. For IMAGENET, the general trend remains: For bigger perturbations, our attack is more

efficient, but for small perturbations, the ensemble attack from Shin and Song [86] is more successful. For IMAGENET, the gaps are generally less clear and the efficiency on JPEG quality 100 barely differs between the two attacks. When using CIEDE2000 as a distance metric (fig. C.5), our attacks performs better, relatively speaking. Again, this is because CIEDE2000 weights color perturbations higher than LPIPS and the attack from Shin and Song is applied in the RGB representation, which results in color perturbation that can be prevented using our JPEG attacks.

Despite the generally balanced performance between the attacks on nets defended with JPEG compression, there are several advantages of our method: First, our attack significantly outperforms Shin and Song's ensemble attack when the net is undefended. Possibly, the ensemble attack induces color perturbations that are ideal to fool nets that use chroma subsampling in defense, as in the net defended with JPEG compression, but not in the undefended net. In a black-box setting though, it would be unknown whether and how the target model is defended. Therefore, generalization across multiple models is an important measure of a black-box attack's success. In fact, our JPEG attack seems to generalize very well, as it performs well on both undefended and defended nets, and efficiency barely differs between the attack qualities used in attack. Second, their ensemble attack is much more time-consuming as it requires multiple gradient computations in every iteration, one for each JPEG quality. On an NVIDIA P6000, attacking the whole CIFAR10 test dataset took 218 seconds for Shin and Song ensemble attack, but only 72 seconds for our JPEG luma attack. Both were executed for 10 iterations.

For Shi et al.'s [85] attacks, we already observe a significant perceptual distance even if $\varepsilon = 0$. So, the fast adversarial rounding alone already disturbs the images heavily, as the perceptual distance is much bigger than for our JPEG attacks which also return compressed data, especially for low JPEG qualities. As Shi et al. only used the input ε and Peak signal-to-noise ratio (PSNR) to compare the attacks' success, they did not measure the actual perceptual distance created by the fast adversarial rounding. Their attack only improves the success in dependence of the ε, which again shows the unsuitability of the L_∞ norm as a perceptual metric. Proof that the attack improves the success in dependence of the ε is given in fig. C.4 in appendix C. The success of their attacks rises quickly in dependence of the perceptual distance though, which implies that although it induces a significant perceptual distortion, the fast adversarial rounding can indeed make the perturbations more robust against JPEG compression, and experimenting with more η's might actually improve the efficiency as well. When experimenting with fast adversarial rounding for our JPEG attacks, we mentioned that we believe that the coefficients

are already close to their local optima, such that the fast adversarial rounding does not have much influence on the attack's output. Since the main perturbation is applied in RGB pixel representations for Shi et al.'s attacks, and the images are then converted to JPEG coefficients, they might not be that close to local optima, such that the fast adversarial rounding is more effective here. In comparison with Shi et al.'s attacks, ours always show superior efficiency for CIFAR10 though, on both the undefended net the one defended with JPEG compression. Apart from only attacking the luma channel, this is reasoned by the fact that the perturbation was applied straight on coefficients and not on pixel values, which allows us to concentrate perturbations in medium frequencies. This is advantageous as high-frequency perturbations tend to be removed during the JPEG compression in defense, as argued before.

Thus, our attack provides many options to control the perturbation and can be more efficient than state-of-the-art attacks when JPEG compressed data is required, while being technically much more straightforward as we do not have to include an approximation of JPEG compression into the target model, or design a sophisticated rounding scheme as the coefficients are already close to the local optima before rounding to the nearest integer.

4.3.3 Sample Images

Now we compare our attacks with standard RGB and $\mathrm{YC_bC_r}$ pixel attacks as well as Shi et al.'s and Shin and Song's methods using sample images.

The former comparison is illustrated in fig. 4.28 and fig. 4.29 for IMAGENET and CIFAR10, respectively. The images generally support the quantitative results from the previous sections. Attacks that perturb color information (RGB, $\mathrm{YC_bC_r}$ ε_{all}, JPEG ε_{all}) result in a colored, and often very visible noise. The luma perturbations, though, are generally less visible, especially when they are concentrated in medium frequencies. All JPEG attacks sometimes create images that contain visible JPEG blocks. Generally, this becomes especially clear when the lowest frequencies are perturbed the most. However, the ascent attack (where the highest frequencies are perturbed the most) often shows these artefacts as well. The images are the minimum perturbation for each attack that results in a misclassification on the Densenet^{jq50}. Thus, the ascent vector is very ineffective as the high-frequency perturbation is removed during the JPEG compression. This results in the attack requiring very high ε values to be successful, which leads to strong perturbations on

low frequencies as well despite concentrating the perturbation in high frequencies. Note that distributing the perturbations across blocks could reduce the visibility of JPEG blocks in the background, as shown before.

The latter comparison is illustrated in fig. 4.30 for ImageNet and fig. 4.31 for Cifar10. Again, we observe that Shi et al.'s and Shin and Song's attacks result in a colored noise. However, the noise created by Shin and Song's attacks is less obvious. Presumably, this is due to the inclusion of JPEG compression (and chroma subsampling) in the target model. Thus, Shin and Song's images do often look similarly close to the original as the images created by our JPEG medium attack, which again corresponds to the quantitative results.

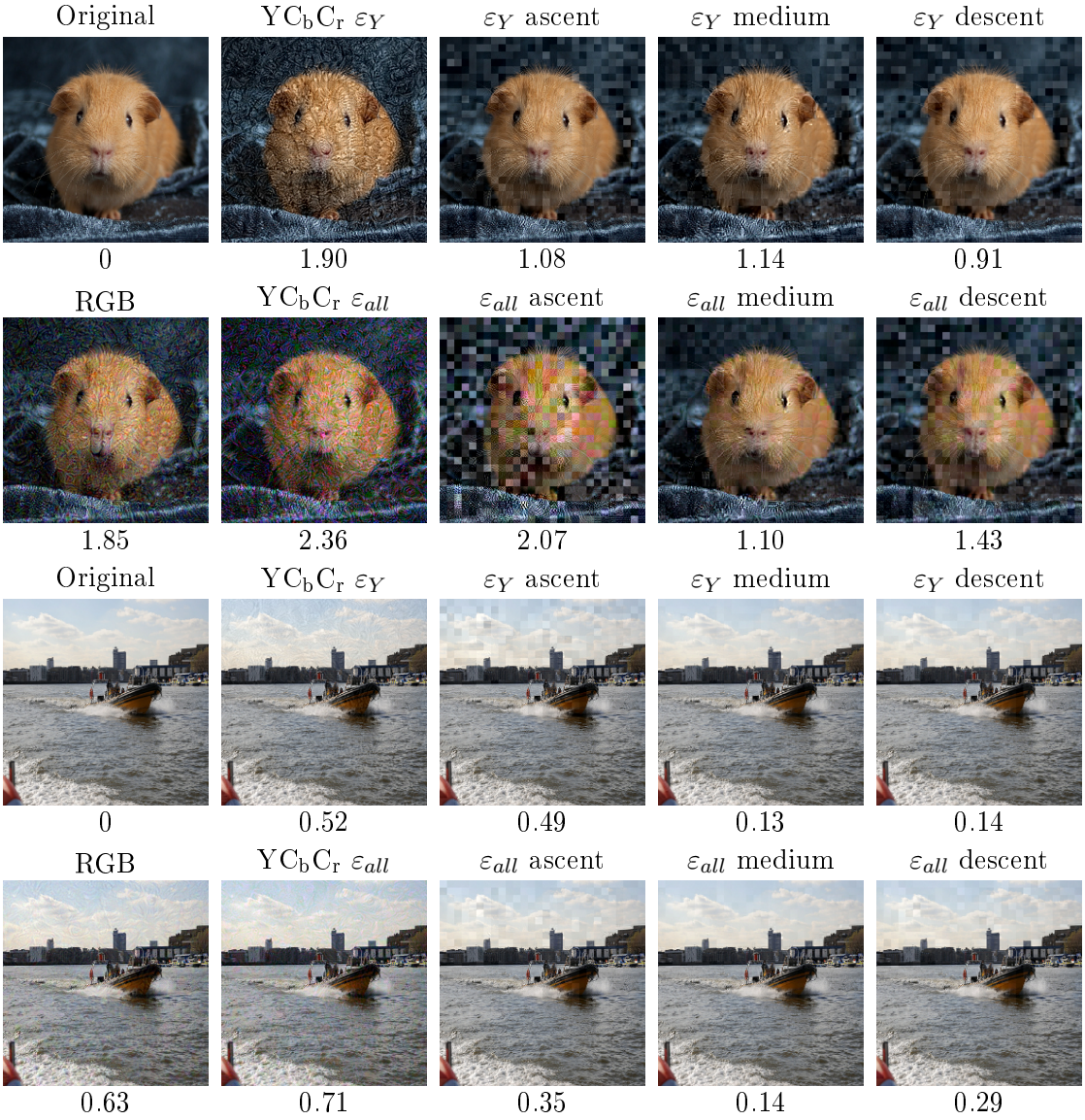


Figure 4.28: IMAGENET - Minimum BIM perturbation required for a misclassification by the Densenet^{jq50}, for our JPEG attacks and RGB and YC_bC_r pixel attacks. Images are created on a Resnet. The LPIPS distance is given below the images.

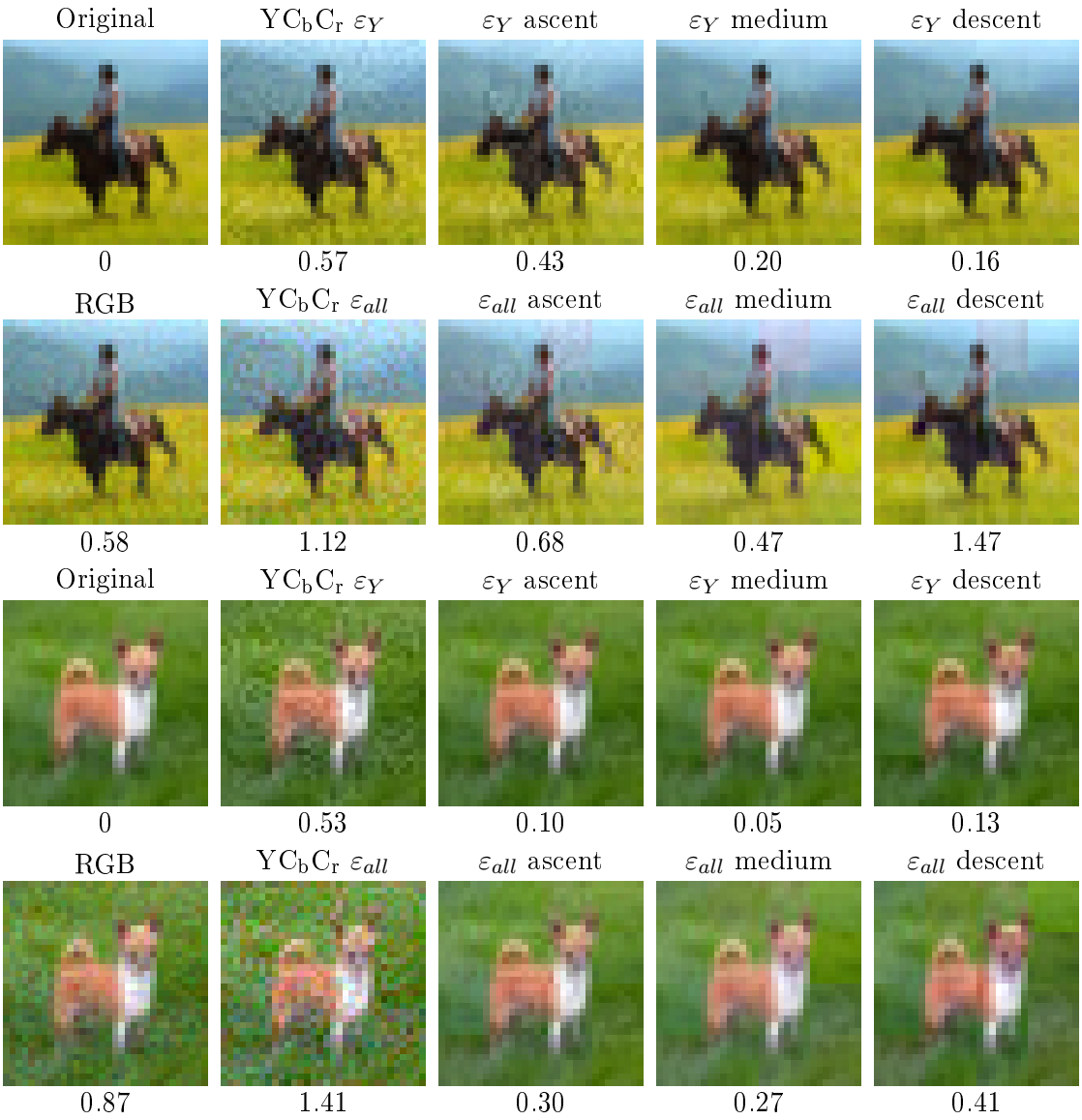


Figure 4.29: CIFAR10 - Minimum BIM perturbation required for a misclassification by the Densenet^{jq50}, for our JPEG attacks and RGB and YC_bC_r pixel attacks. Images are created on a Resnet. The LPIPS distance is given below the images.

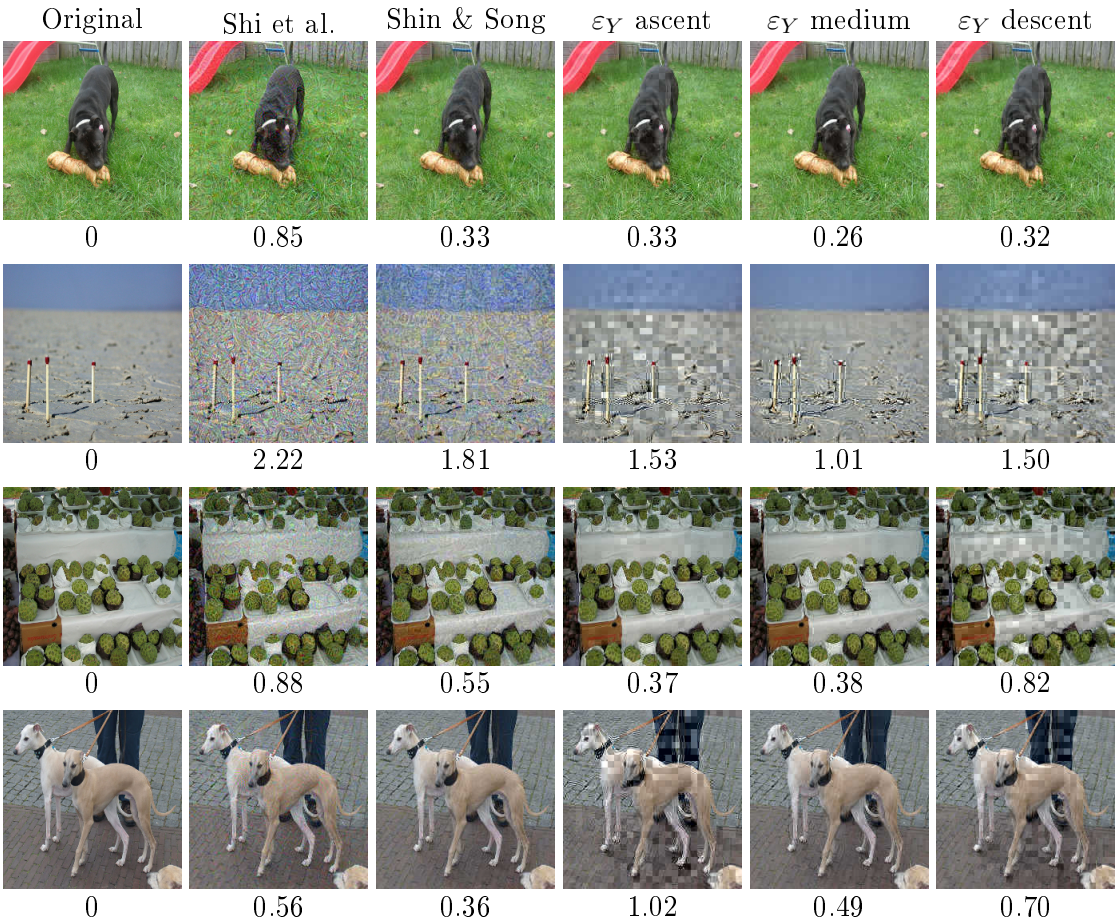


Figure 4.30: IMAGENET - Minimum BIM perturbation required for a misclassification by the Densenet^{jq50}, for our JPEG attacks and Shi et al. [85]'s and Shin and Song's attacks. Images are created on a Resnet. The LPIPS distance is given below the images. JPEG quality 50 is used in the attacks.

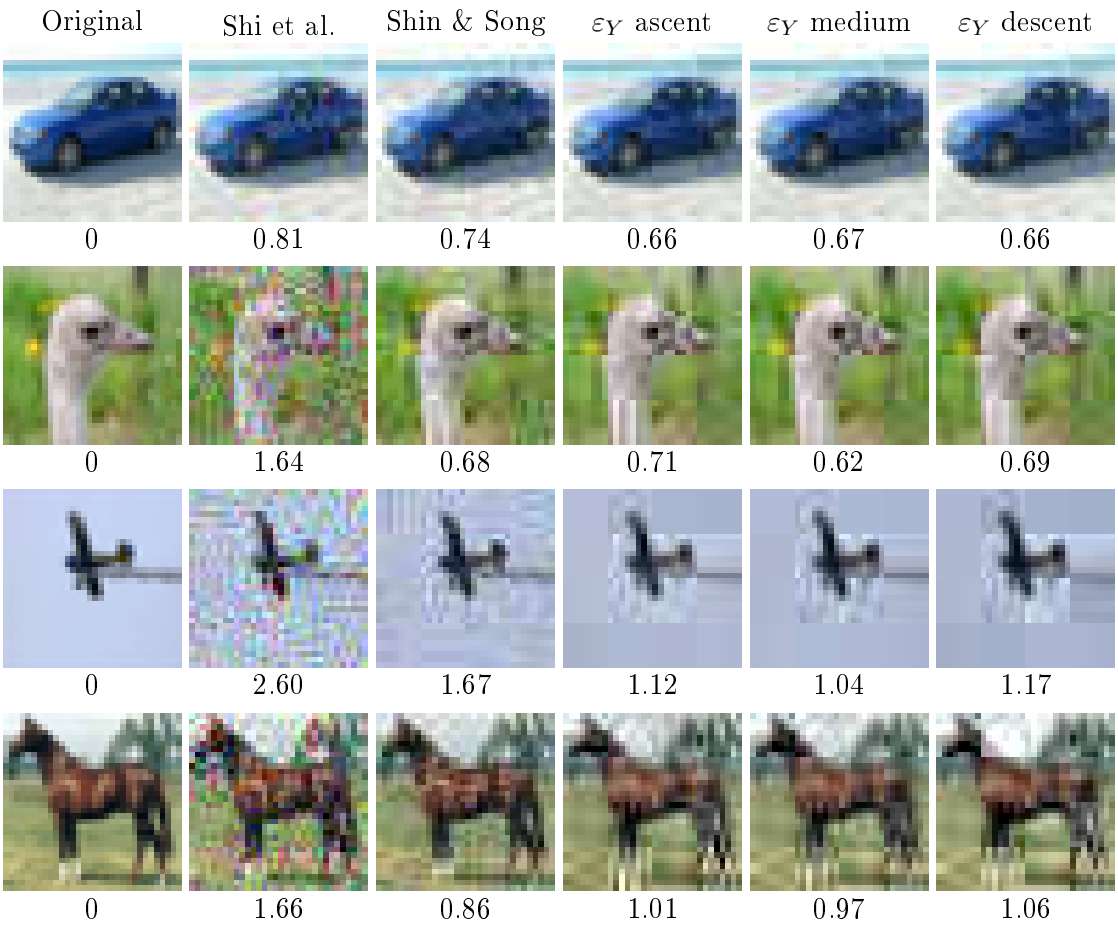


Figure 4.31: CIFAR10 - Minimum BIM perturbation required for a misclassification by the Densenet^{jq50}, for our JPEG attacks and Shi et al. [85]'s and Shin and Song's attacks. Images are created on a Resnet. The LPIPS distance is given below the images. JPEG quality 50 is used in the attacks.

4.4 Minimum-Norm Attacks

This section will be about our experiments on minimum-norm attacks. Until now, the attacks always tried to maximize the confidence of the wrong prediction but we still measured the efficiency of attacks as the ratio of success and perceived distortion. Now, we analyze whether JPEG is also a superior representation for attacks that try to minimize the distortion. As the attacks search for the minimum distortion, it could be possible that they automatically distribute the perturbations differently across frequencies as maximum-confidence attacks, which is our main motivation for also considering these minimum-norm attacks.

As explained in chapter 3, we design two minimum-norm attacks that perturb straight on JPEG coefficients: The PerC-AL, which minimizes the CIEDE2000 distance, and the LPIPS-AL, which minimizes the LPIPS distance. Both attacks are also included as RGB versions. Only the PerC-AL attack has been proposed by Zhao et al. [102], but the extension to use the LPIPS distance is straightforward. In the original paper on the PerC-AL, Zhao et al. used $\alpha_l = 1$ as a step size for maximizing the model's loss and $\alpha_c = 0.1$ for minimizing the distortion. Following an experiment on the step sizes, we use $\alpha_Y^{l,\text{rel}} = 2, \alpha_Y^{c,\text{rel}} = 0.2$ as step sizes for our JPEG attacks. However, it has to be mentioned that the results of this experiment are limited to the undefended Resnet, such that we cannot state that our selection is optimal in other settings. Due to the high computation time, we only consider JPEG luma minimum-norm attacks.

As mentioned in chapter 2, minimum-norm attacks are most relevant for white-box settings as they try to find the minimum perturbation that leads to a misclassification and this minimum perturbation is often not sufficient to fool the unknown black-box model. However, the undefended white-box setting is difficult to analyze as all the attacks are able to fool the net with very little perturbation and there are barely any differences that can be observed. The fact that the RGB attacks return uncompressed data while our attacks return JPEG coefficients makes the comparison even more difficult. For the maximum-confidence attacks we were able to simply compress the output. In this case though, as the minimum required perturbation is computed, compressing the output would usually lead to the attack not being successful anymore. Thus, we have to compare uncompressed RGB data to compressed JPEG coefficients (of quality 100). This induces a slight increase of the perceptual distance for our JPEG attacks that has to be kept in mind. When the images are not perturbed, the JPEG compression to quality

100 (no chroma subsampling) results in an average LPIPS distance of 0.00497 on the CIFAR10 test dataset.

We will analyze both white- and black-box settings where the images are created on the undefended Resnet. White-box attacks on adversarially trained nets will then be considered later in the section on adversarial training (4.5). Generally, in difference to the maximum-confidence attacks, where we incrementally increased the input ε's, we now increase the confidence parameter κ, and measure success rate and perceptual distance. As the white-box success-rate almost always reaches 100 %, we will use tables to visualize the corresponding white-box results.

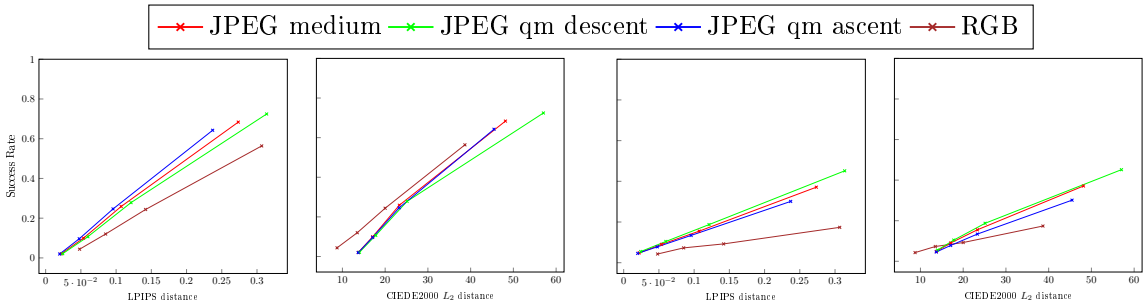
To first determine which RGB attack is most suited to minimize the perceptual distance, we start with comparing the RGB attacks. The white-box results on the undefended Resnet are presented in table 4.5.

	Success Rate	CIEDE2000 L_2	RGB L_2	LPIPS
PerC-AL	100%	8.74	48.03	0.048
LPIPS-AL	100%	20.81	72.94	0.0057
DDN	100%	8.31	15.96	0.0068
C&W-L_2	100%	40.54	75.52	0.094

Table 4.5: CIFAR10 - White-box results for RGB minimum-norm attacks on the undefended Resnet. $\kappa = 0$ was used for all attacks.

While PerC-AL shows a low CIEDE2000 L_2 distance, the LPIPS distance is much higher than for LPIPS-AL and DDN. Surprisingly, and contrary to the results from the original paper [102], DDN also outperforms PerC-AL in terms of the CIEDE2000 L_2 distance, but only slightly. The C&W-L_2 achieves the worst results. Presumably, this is due to the reduced number of iterations as the original paper used 1000 internal iterations instead of 100.

The LPIPS-AL attack results in the lowest LPIPS distance, but very high L_2 distances for both CIEDE2000 and RGB. Presumably, this can be explained by effects similar to those illustrated in figs. 2.15 and 2.16 which result in big pixel-wise, but only little perceptual distances. As they try to reduce the perceived distortion, this section will focus on PerC-AL and LPIPS-AL.



(a) Densenet (b) Densenet^{jq50}

Figure 4.32: CIFAR10 - Black-box efficiency for JPEG and RGB PerC-AL. The JPEG attacks only perturbed the luma channel. Images were created on a ResNet.

4.4.1 Varying Perturbations across frequencies

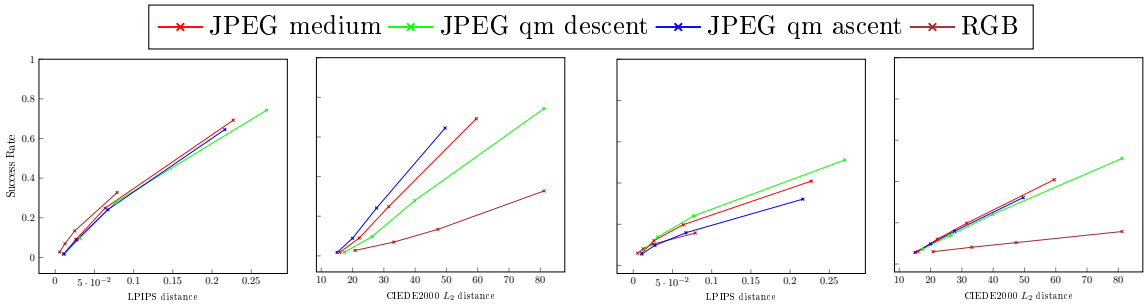
We will now compare minimum-norm attacks when they distribute their perturbations differently across the frequency spectrum. For the undefended white-box settings, table 4.6 shows the results for CIFAR10. As mentioned before, the differences between the attacks are only minimal in the undefended black-box setting. However, the descent attacks, as expected, perform worst, since the undefended net is least vulnerable towards low-frequency perturbations, while the ascent attacks perform best.

	Success Rate	CIEDE2000 L_2	LPIPS
LPIPS-AL medium	100%	15.93	0.011
LPIPS-AL ascent	100%	15.065	0.011
LPIPS-AL descent	100%	17.39	0.012
PerC-AL medium	100%	13.72	0.022
PerC-AL ascent	100%	13.70	0.020
PerC-AL descent	100%	13.99	0.024

Table 4.6: CIFAR10 - White-box results for JPEG minimum-norm attacks on the undefended Resnet. $\kappa = 0$ was used for all attacks.

The black-box setting allows us to determine the differences between the weighting vectors much better. The corresponding results are shown in figs. 4.32 and 4.33 for PerC-AL and, respectively, LPIPS-AL. The figure also includes the RGB curves for the comparison in the following subsection. For now, we will focus on the comparison between our JPEG attacks, though.

The general observations are the same as for the maximum-confidence attacks. On the undefended net, the ascent vector performs best as it concentrates the perturbation in



(a) Densenet (b) Densenet^{jq50}

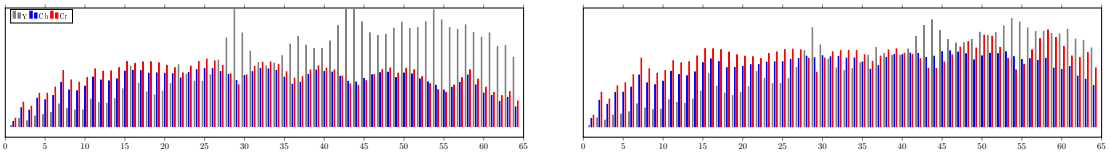
Figure 4.33: CIFAR10 - Black-box efficiency for JPEG and RGB LPIPS-AL. The JPEG attacks only perturbed the luma channel. Images were created on a ResNet.

high frequencies. On the Densenet^{jq50} though, the medium and descent vectors are more efficient as the high-frequency perturbations are removed during the JPEG compression. So, there are no significant differences to the maximum-confidence attacks.

4.4.2 Comparison with RGB attacks

Now we compare our JPEG attacks to the RGB attacks. Again, see figs. 4.32 and 4.33 for the CIFAR10 results for PerC-AL and LPIPS-AL, respectively. On the undefended net, both RGB attacks seem to be more successful than the corresponding JPEG attacks regarding the distance that is minimized by the attack, which is a significant difference to the results we obtained for the maximum-confidence attacks. This is also supported by the white-box results presented in tables 4.5 and 4.6. Because of the amount of the difference between the RGB and JPEG attacks, this can not exclusively be due to comparing uncompressed RGB data with compressed JPEG data.

Instead, it seems that by minimizing the distortion, the RGB attack is able to reduce the amount of inefficient perturbation, while for maximum-confidence attacks only the loss was maximized which also led to perturbation that influenced the prediction only slightly and ineffectively. Here, however, this ineffective distortion is avoided by the minimization. For example, this could be achieved by automatically avoiding color perturbation which we showed to be ineffective. To determine whether this is indeed the case, we again plot the average relative perturbation for every channel and frequency in fig. 4.34. When comparing this with the results for the RGB BIM attack in fig. 4.8, one can conclude that



(a) PerC-AL (b) LPIPS-AL

Figure 4.34: CIFAR10 - Average relative perturbations made by RGB PerC-AL and LPIPS-AL on JPEG frequencies for the CIFAR10 dataset on the undefended Densenet.

the color perturbation is indeed much smaller than the luma perturbation for minimum-norm attacks, which proves the assumption. So, small color perturbations seem to be effective. As the JPEG attacks only perturb the luma channel, their scope is more limited.

The JPEG attacks are also limited by the quantization but also the weighting vectors which manually distribute perturbations across frequencies. The RGB attack, though, automatically distributes the perturbation where it is most efficient.

This can be a big advantage in settings where the source model relies on the same features and frequencies as the target model but it has been shown that it is a disadvantage otherwise. The same applies here: On the Densenet^{jq50}, the RGB attacks are much less efficient than the JPEG attacks as the high-frequency perturbation that was very efficient on the undefended model is not effective anymore. This is the same effect that also applies to the JPEG ascent attack, but less strongly, which could be reasoned in the RGB attack applying chroma perturbations that are removed during the chroma subsampling in the JPEG compression.

An interesting observation from this comparison is that the RGB attacks are only efficient regarding the distance metric that is minimized during the attack: The RGB LPIPS-AL attack is very inefficient regarding the CIEDE2000 L_2 distance and the RGB PerC-AL attack is very inefficient regarding LPIPS. This does not apply to the JPEG attacks at all though. A possible explanation could be that the RGB attacks are able to use the properties of each distance metric better as they are less limited than the JPEG attacks and can, e.g., optimally distribute the perturbations across frequencies, which can lead to more efficiency regarding the distance metric used in the attack. One could describe this phenomenon as overfitting. The JPEG attack, on the other hand, is regularized by the masking vector that limits the scope of the attack and thus, the efficiency on the distance

metric used in the attack, but avoids overfitting and leads to better generalization which has been a major advantage of our JPEG maximum-confidence attacks as well.

As mentioned earlier, we only perform white-box experiments using confidence $\kappa = 0$ for ImageNet. For the undefended Resnet, the results are presented in table 4.7 for LPIPS-AL. None of the attacks reaches 100% success rate which indicates that the parameter selection was not optimal. Presumably, the number of iterations has to be increased. The implementation of both RGB and JPEG attacks return the image with the best confidence value in case the target confidence, which is 0 in this case, is not reached. Thus, the distances are still more or less comparable even if not every image is adversarial. We observe that for ImageNet, the JPEG LPIPS-AL is far more efficient than on RGB pixels, regarding both CIEDE2000 and LPIPS. Especially the medium weighting vector results in only little distortion. As for Cifar10, there will be experiments on the adversarially defended white-box setting later in section 4.5.

In summary, the results for minimum-norm attacks are less promising for Cifar10 as the JPEG representation seems to limit the attack's scope too much. For ImageNet, however, minimum-norm attacks on JPEG coefficients are more efficient than on RGB pixels. Possibly, the differences between Cifar10 and ImageNet results are due to the relative size of each JPEG block in each image. For Cifar10, there are only 16 JPEG blocks and, as we have seen before, they can be clearly visible if perturbed enough. For ImageNet, specific blocks are much less visible. Thus, the JPEG representation itself does not account for as much perceptual distance as for Cifar10.

	Success Rate	CIEDE2000 L_2	LPIPS
LPIPS-AL RGB	99.83%	265.11	0.088
LPIPS-AL medium	96.46%	88.78	0.0022
LPIPS-AL ascent	95.35%	90.35	0.0041
LPIPS-AL descent	97.05%	96.35	0.0049

Table 4.7: ImageNet - White-box results for LPIPS-AL on the undefended Resnet. $\kappa = 0$ was used for all attacks.

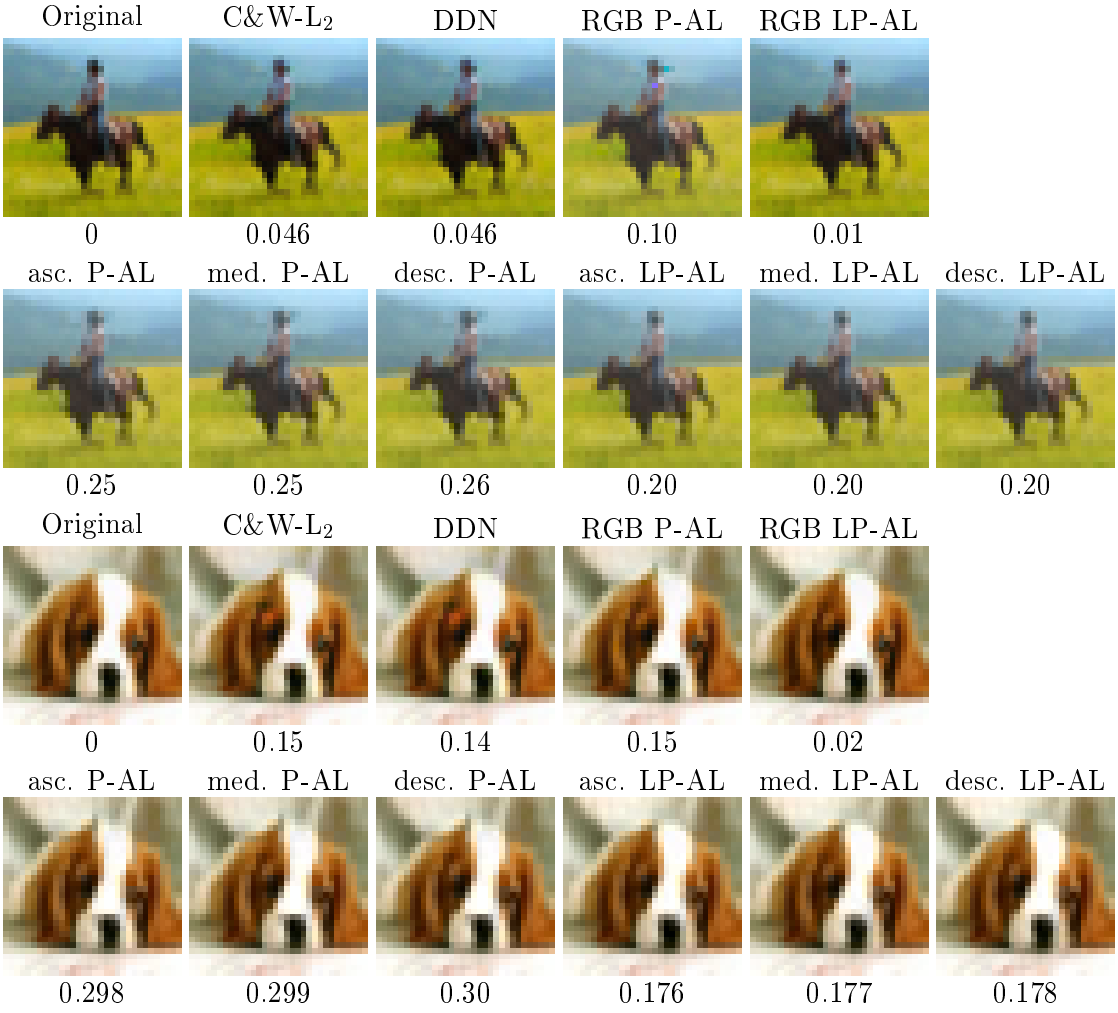


Figure 4.35: CIFAR10 - Minimum-norm attacks applied on the Densenet$_M^{RGB}$. The LPIPS distance is given below each image. LP-AL = LPIPS-AL. P-AL = PerC-AL.

4.4.3 Sample Images

Figures 4.35 and 4.36 show sample images for minimum-norm attacks created on the Densenet$_M^{RGB}$. Even though the net is adversarially defended, minimum-norm attacks in the white-box setting are able to find perturbations that are barely visible. There are barely any differences between the attacks that can be seen.

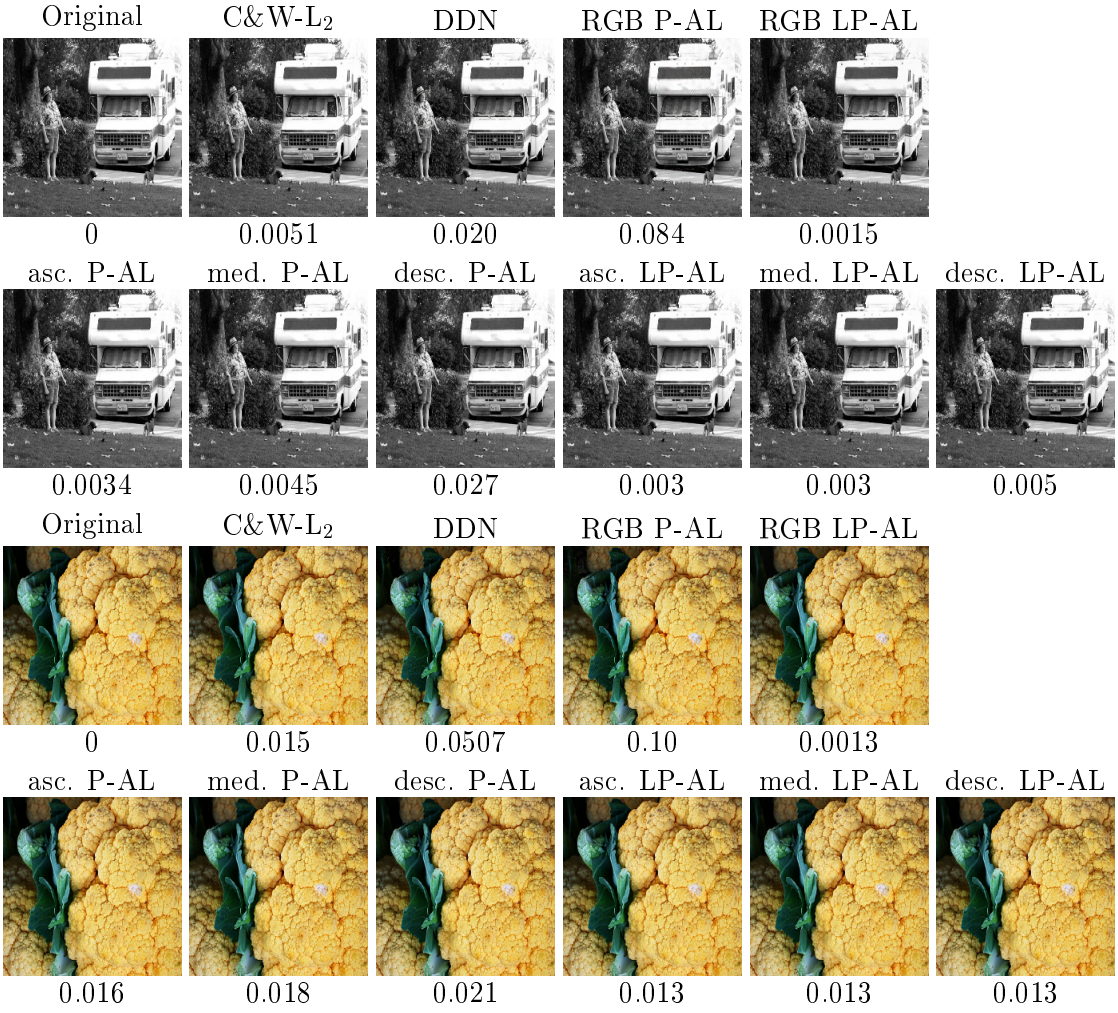


Figure 4.36: IMAGENET - Minimum-norm attacks applied on the Densenet$_M^{RGB}$. The LPIPS distance is given below each image. LP-AL = LPIPS-AL. P-AL = PerC-AL.

For the CIFAR10 image, the example reflects the general result that the RGB attacks are more efficient than JPEG attacks due to the reasons named in the previous subsections. For one of the IMAGENET samples, the LPIPS distance of the perturbation created by the JPEG LPIPS-AL is slightly smaller than for the RGB attack. Again, no actual distortion can be observed, though.

4.5 Adversarial Training

As explained in chapter 2, adversarial training has shown to be a strong defense against adversarial attacks [55, 65, 68] and increases the similarity to human perception [93]. However, it is also known to be vulnerable against unseen threat models [48, 58]. In the previous sections, we have shown that nets defended with RGB adversarial training tend to be vulnerable against perturbations on low frequencies. In this section, we will analyze whether the variability of our attacks, which follows from using channel-wise perturbation budgets and manually distributing the perturbations across DCT frequencies, can increase the net's robustness across the whole frequency spectrum, the ability to generalize and thus, the robustness against unforeseen threat models.

4.5.1 JPEG Adversarial Training

The idea that our attacks could form the basis for an adversarial training method that overcomes the problem of adversarial training leading to more vulnerability on lower frequencies and leads to nets being more robust against perturbations on all frequencies and thus, generalize better, is mainly reasoned by controlling the perturbations applied across frequencies. Thereby, we believe the net can be forced to use similar features as the human perception does, such that it is better aligned with human perception and generalizes better.

We use both ε_{all}- and ε_Y-attacks during the adversarial training. The relative εs are selected to lead to a similar average LPIPS distance as the RGB BIM with $\varepsilon = 8$, which is ~ 0.6 for CIFAR10 and ~ 0.4 for IMAGENET. They are shown in tables 4.8 and 4.9 for each dataset. The corresponding weights are presented in table 4.10. They were chosen more experimentally than analytically as they led to strong robustness on all frequencies for CIFAR10. To also achieve robustness against attacks that perturb color information (such as RGB attacks), the ε_{all}-attacks are weighted twice as much as ε_Y-attacks. Using the percentages from table 4.10, one attack is randomly chosen for each batch during the adversarial training. The medium vector is weighted the most, followed by the ascent and descent attacks. The resulting net is denoted as Densenet_M^{JPEG}.

λ	ε_Y	ε_{all}
medium	1.2	0.9
qm descent	0.6	0.45
qm ascent	2.7	2.0
unmasked	0.5	0.4

Table 4.8: JPEG adversarial training: εs - CIFAR10.

λ	ε_Y	ε_{all}
medium	1.0	0.8
qm descent	0.2	0.15
qm ascent	1.5	1.0
unmasked	0.2	0.15

Table 4.9: JPEG adversarial training: εs - IMAGENET.

λ	ε_Y	ε_{all}
medium	14.81%	29.63%
qm descent	7.41%	14.81%
qm ascent	9.26%	18.52%
unmasked	1.85%	3.70%

Table 4.10: JPEG adversarial training: Weights.

4.5.2 Evaluation

As explained in section 2.4 and by Tsipras et al. [93], RGB adversarial training leads to the net using features that are more similar to those used by humans. They proved this by visualizing the loss gradients of different nets, as shown in fig. 2.17. To examine differences between standard RGB and our JPEG adversarial training, we will now complement this comparison by our Densenet$_M^{JPEG}$ for CIFAR10. Figure 4.37 visualizes the loss gradients in the same way as in [93].

As in [93], the loss gradients of the undefended net are noisy and barely show any structure. Both adversarially trained nets yield loss gradients that are much better aligned with human perception. There are significant differences between the Densenet$_M^{RGB}$ and the Densenet$_M^{JPEG}$, though. The loss gradients of the Densenet$_M^{RGB}$ are much more fine-grain and contain regions of uniform color, while the gradients of the Densenet$_M^{JPEG}$ show much coarser structures and the colors change where it would not necessarily be expected.

Another way of illustrating the features that a net associates with each label is to learn images that reduce the model's loss for each label, which has been done for fig. 4.38 for the undefended Resnet and both adversarially trained nets. Corresponding to the results from above, the images created to minimize the Densenet$_M^{JPEG}$'s crossentropy loss can be described as much more abstract and contain less sharp edges compared to those created for the Densenet$_M^{RGB}$. This is especially visible for the deer and the horse.

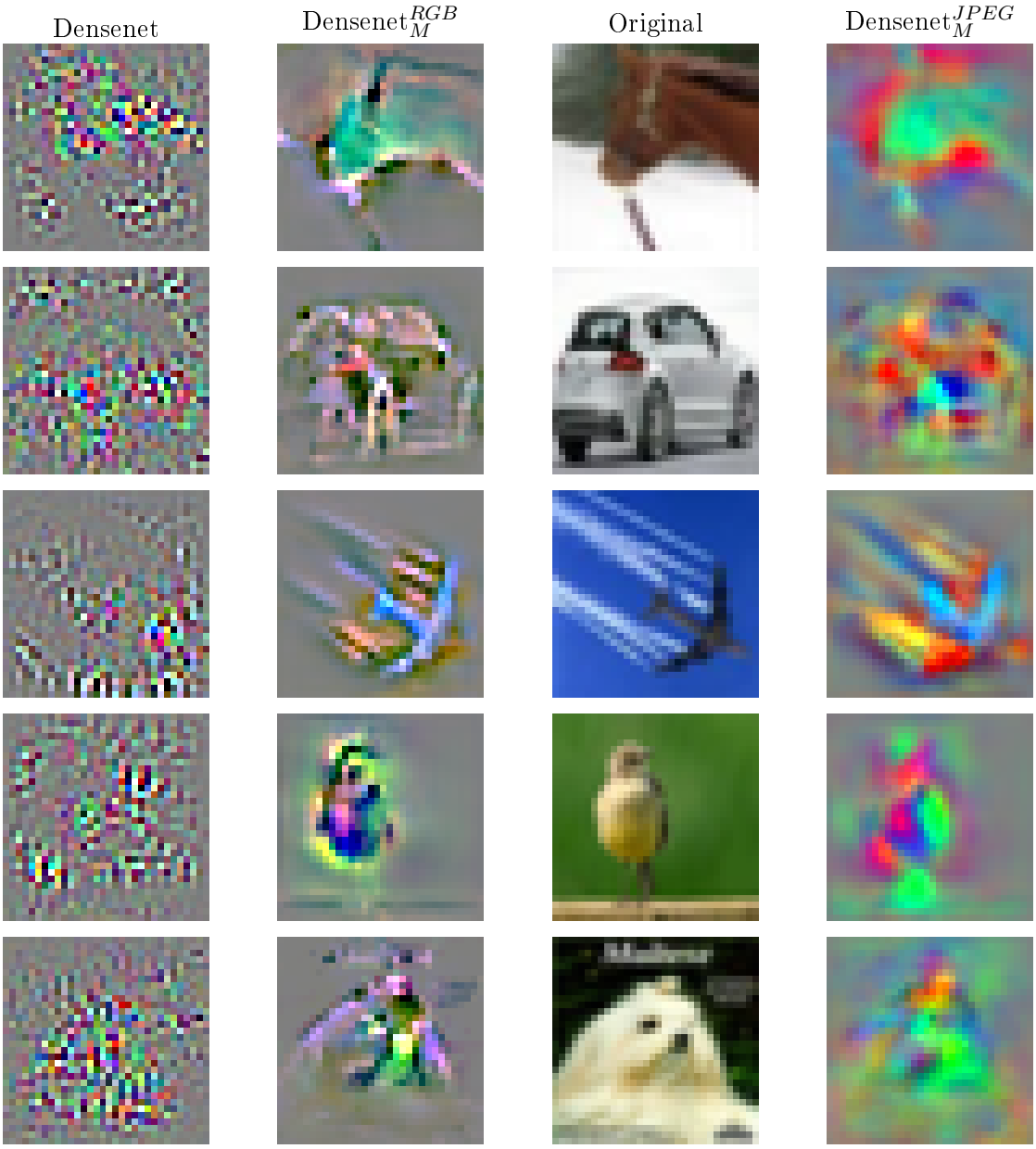
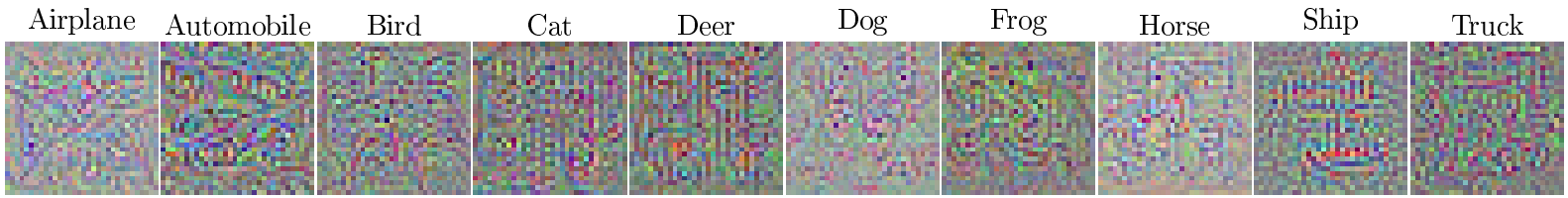


Figure 4.37: Loss gradients for some CIFAR10 images for the undefended Densenet and the adversarially trained Densenet$_M^{RGB}$ and Densenet$_M^{JPEG}$. The gradients are normalized as in [93].



(a) Resnet

(b) Densenet$_M^{RGB}$

(c) Densenet$_M^{JPEG}$

Figure 4.38: Starting from unicolored images (R, G, B = 128, 128, 128), each model's loss was minimized in 100 steps of gradient descent for each of CIFAR10's classes.

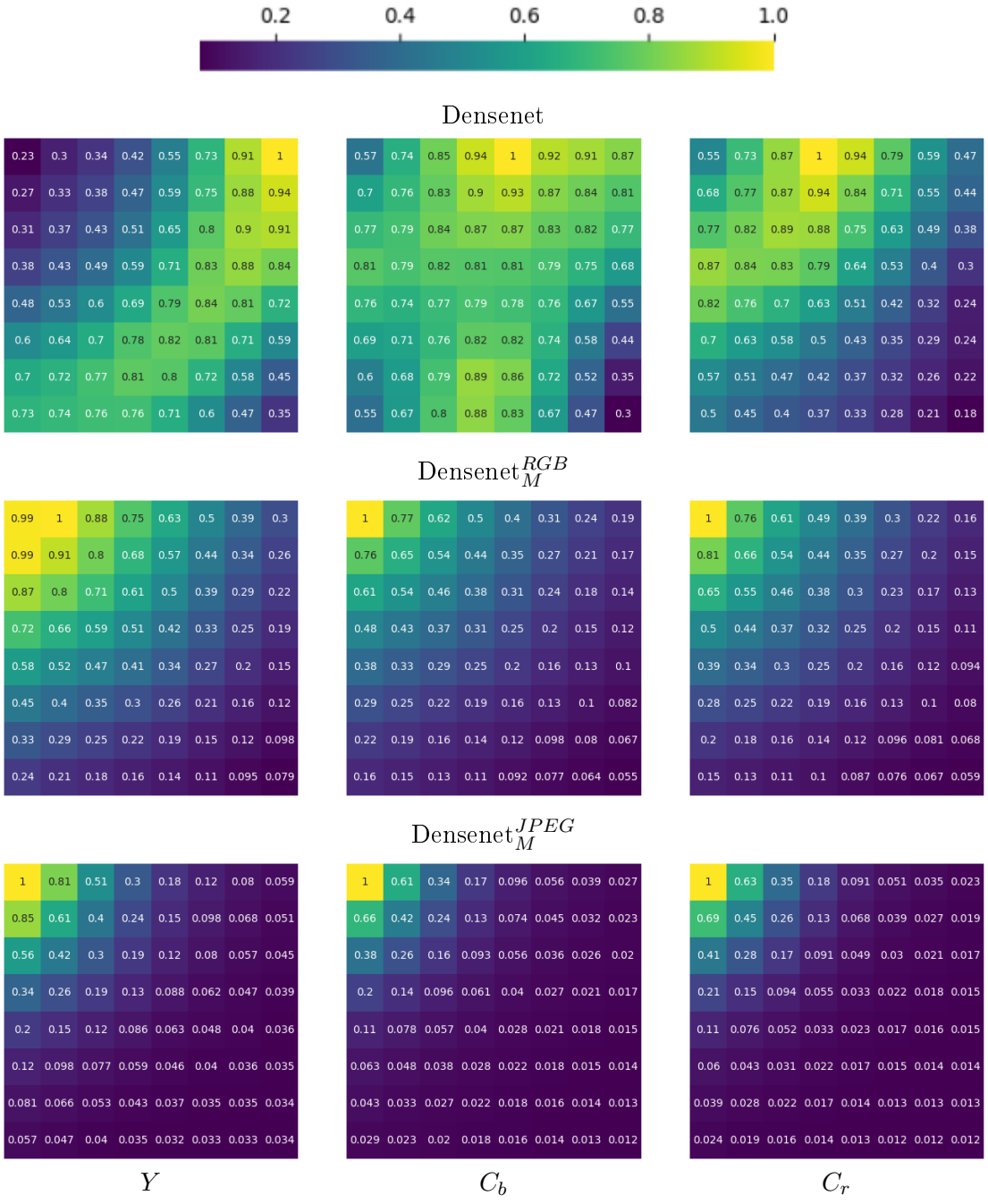


Figure 4.39: Loss gradient distribution for different nets. The frequency-wise mean absolute gradients are divided by their maximum for normalization.

This is also supported by fig. 4.39, where normalized loss gradients are visualized for all three nets. On the undefended net, the gradients are highest on medium and, for luma, higher frequencies. On the defended nets though, the sensitivity shifts towards lower frequencies as explained in section 2.5. The JPEG net relies on the lowest frequencies even more than the RGB defended net, which results in the low-frequency structures visible in fig. 4.37. Our observations from figs. 4.37 to 4.39 imply that the Densenet$_M^{JPEG}$ relies on robust features more than the Densenet$_M^{RGB}$. Actually, the non-robust features in fig. C.2 show a similar, abstract structure as the images created by minimizing the Densenet$_M^{JPEG}$'s loss in fig. 4.38. Whether this translates to robustness towards perturbations on different frequencies will be analyzed further in the following subsections.

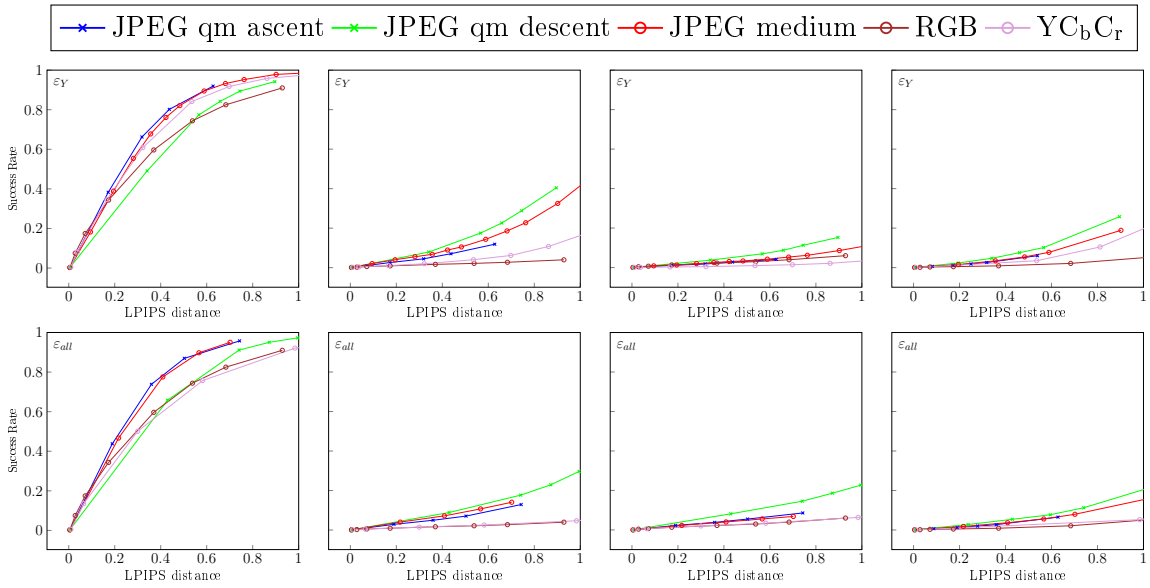
Maximum-Confidence Attacks

For maximum-confidence JPEG attacks on CIFAR10, fig. 4.40 shows that our Densenet$_M^{JPEG}$ is now much more robust against perturbations on all frequencies, compared to the Densenet$_M^{RGB}$. For luma attacks, the maximum success rate does not surpass 20% for a LPIPS distance < 1, whereas it was more than 40% for the RGB adversarially trained net. The highest vulnerability is still towards the descent weighting vector, i.e. low-frequency perturbations. This corresponds to the findings from figs. 4.37 to 4.39 which showed that the net relies on low frequencies even more than the Densenet$_M^{RGB}$. The strong robustness against ε_Y is a consequence of including pure luma attacks in the adversarial training and explains the coarse, colored structures that were visible in the loss gradients in fig. 4.37 as the net is robust against luma and, comparatively, sensitive against chroma perturbation.

The figure also includes black-box results on the Perceptual Adversarial Training (PAT) net from Laidlaw et al. [58][6]. The model shows similar robustness as our Densenet$_M^{JPEG}$. The highest vulnerability is against low-frequency perturbation. Against ε_{all}-attacks, the PAT model is slightly more robust, while our model performs better against ε_Y-attacks. But the results for ε_{all}-attacks barely differ between all three adversarially trained nets as they show good robustness here.

Figure 4.40 also includes the results of the RGB and $\mathrm{YC_bC_r}$ pixel attacks for CIFAR10. The robustness against the RGB attack is just slightly reduced in comparison to the

[6] We use the version that computes the LPIPS distance on the model itself. The attacks are limited by a LPIPS distance of 0.5, which is very similar to the LPIPS distance that the attacks in our JPEG adversarial training resulted in for CIFAR10 (~ 0.6).



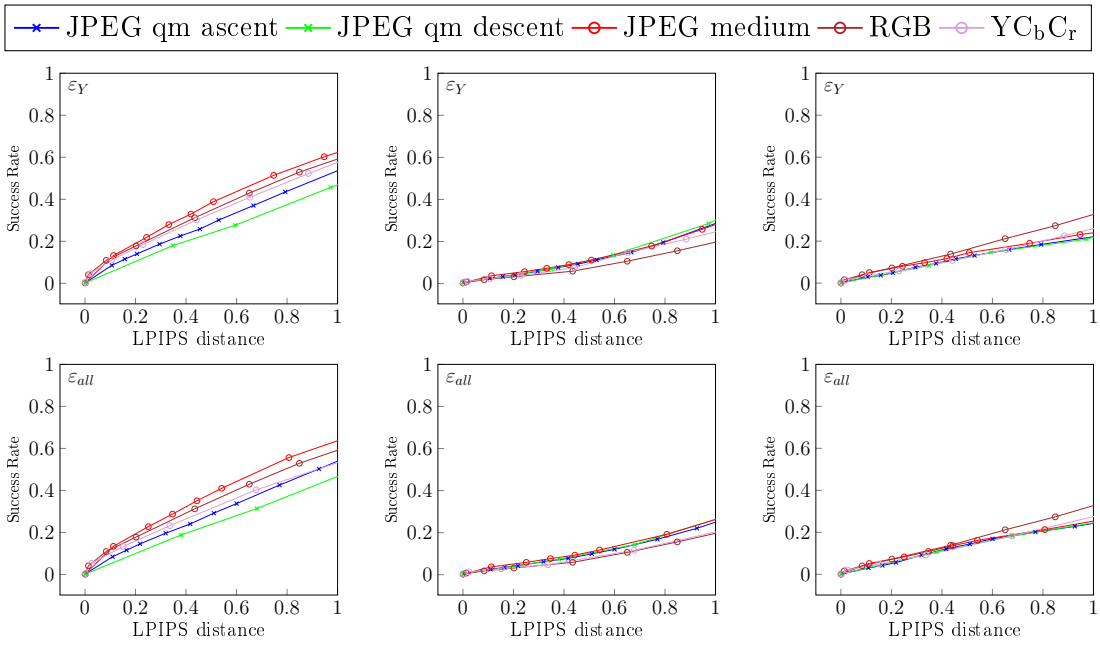
(a) Densenet (b) Densenet$_M^{RGB}$ (c) Densenet$_M^{JPEG}$ (d) PAT

Figure 4.40: CIFAR10 - Black-box LPIPS efficiency of JPEG BIM luma (top) and all (bottom) attacks in comparison with RGB and YC_bC_r attacks on adversarially trained nets. The RGB attack is the same across both rows.

Densenet$_M^{RGB}$. Despite our Densenet$_M^{JPEG}$ being trained using JPEG adversarial examples, it is also very robust against both pixel attacks. In fact, the biggest vulnerability is still towards our JPEG descent attack.

Comparing our JPEG net and the PAT, it stands out that both are basically robust against RGB attacks as well, at least in the black-box setting. They are also robust against YC_bC_r pixel attacks when they are executed on all three channels. As stated above, the PAT is slightly less robust against pure luma attacks, including the YC_bC_r luma attack. Together with the results from above, this shows that our JPEG adversarial training achieves very similar black-box robustness and generalization as the PAT net, despite our attack method being much more straightforward than the sophisticated perceptual attacks from Laidlaw et al. [58]. Additionally, it shows that our JPEG attacks are more efficient at circumventing the PAT, too, in comparison to pixel attacks, which again underlines the importance of frequencies for both adversarial attacks and defenses.

For IMAGENET, fig. 4.41 shows the corresponding comparison. Here, as mentioned before, the Densenet$_M^{RGB}$ does not show significant vulnerabilities on any part of the frequency



(a) Densenet (b) Densenet$_M^{RGB}$ (c) Densenet$_M^{JPEG}$

Figure 4.41: IMAGENET - Black-box LPIPS efficiency of JPEG BIM luma (top) and all (bottom) attacks in comparison with RGB and YC_bC_r attacks on adversarially trained nets. The RGB attack is the same across both rows.

spectrum, for both ε_{all}- and ε_Y-attacks. Still, the JPEG adversarial training improved the robustness against JPEG attacks, as the success of big perturbations with an LPIPS distance close to 1 is slightly lowered.

Here, we observe that the Densenet$_M^{JPEG}$ is significantly more vulnerable towards RGB attacks than the Densenet$_M^{RGB}$. So, while the Densenet$_M^{JPEG}$ for IMAGENET improves the robustness against JPEG attacks, it reduces the robustness against RGB. To some extent, this is expected as the RGB threat model is unseen during training [48, 58], but the results are significantly worse than for CIFAR10, where the robustness against RGB decreases only slightly.[7]

Figure 4.41 also shows that the Densenet$_M^{JPEG}$ generally builds robustness against pixel attacks, as it is very robust against YC_bC_r attacks on both luma and all three channels as well. The robustness against YC_bC_r pixel attacks indicates that the choice of weights for each attack was not optimal for IMAGENET. As it was optimized experimentally for

[7] As Laidlaw et al. only used a subset of IMAGENET with 100 classes, there is no pretrained model available and we cannot include experiments for PAT.

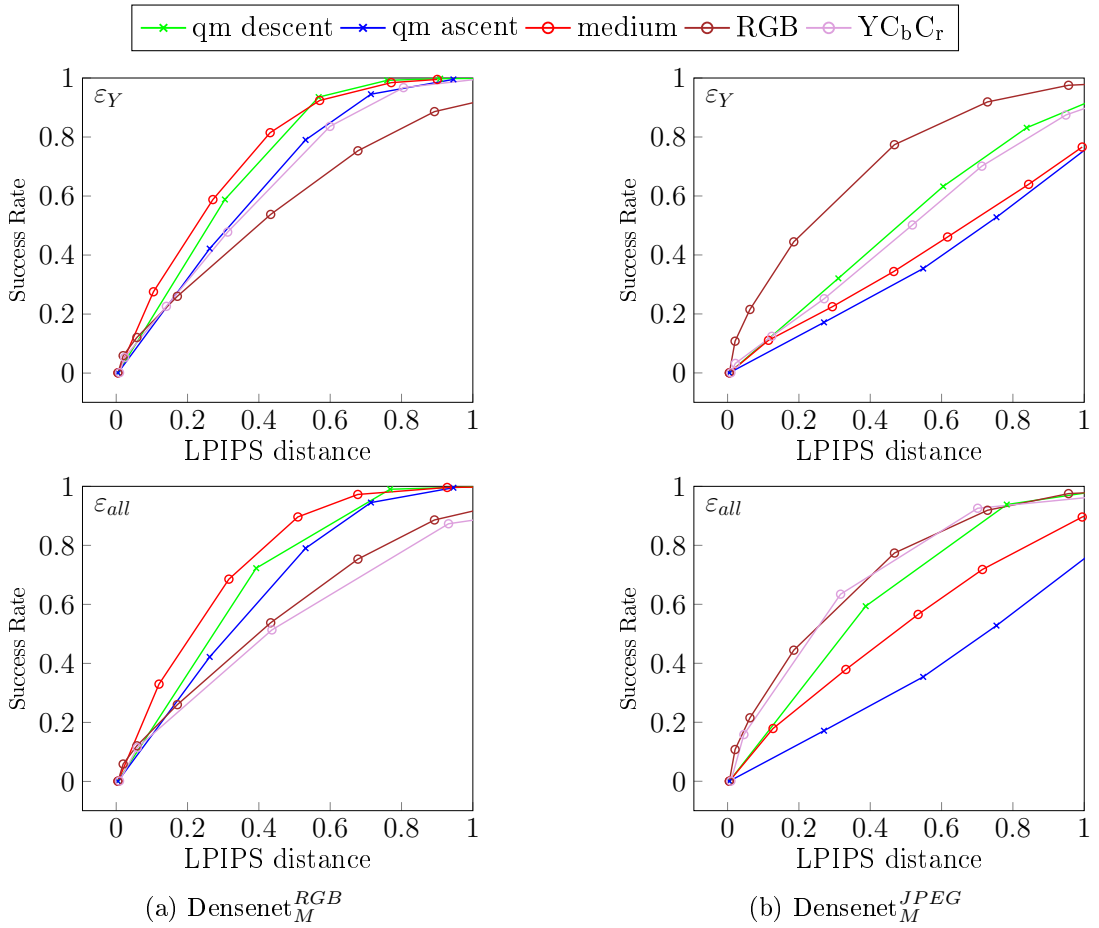
CIFAR10, this is not surprising because classifiers trained on the standard IMAGENET data tend to use different parts of the frequency spectrum, as already shown. Such differences between datasets can also be a consequence of the classes of a dataset. Compared to CIFAR10, which uses very general classes like dog or cat, IMAGENET's classes are much more detailed, containing various breeds of dogs for example. Thus, it could be that IMAGENET classifiers need to rely on wider parts of the spectrum to be accurate as it is insufficient to rely on the lowest frequencies as our Densenet$_M^{JPEG}$ does for CIFAR10.

Therefore it is only logical that different weights are needed for IMAGENET. Presumably, putting more weight to the ε_{all}-attacks would decrease the vulnerability towards RGB attacks and lead to better generalization for IMAGENET as well. However, as adversarial training on IMAGENET is computationally expensive, we cannot experiment with the weights in detail.

For CIFAR10, we also experiment with a defended white-box scenario. The results for the corresponding comparison of RGB and JPEG attacks are shown in fig. 4.42.[8] Here, the Densenet$_M^{JPEG}$ is still quite vulnerable towards RGB and $\mathrm{YC_bC_r}$ pixel attacks. That is because those attacks automatically distribute the perturbation across frequencies depending on the gradients offered by the white-box model. Thus, their perturbation is already concentrated in low frequencies which is implied by the fact that the descent attack achieves a performance that is closest to the pixel attacks' efficiencies and the previous result that our net is most sensitive towards perturbations on lower frequencies. We also expect that by manually choosing a weighting vector we could further increase our attack's success on the Densenet$_M^{JPEG}$. Presumably, it would put even more perturbation on the lowest frequencies. Using the weights from fig. 4.39 might be a sensible choice.

The fact that the RGB and $\mathrm{YC_bC_r}$ attack perform better in the white-box setting on the Densenet$_M^{JPEG}$ actually underlines the advantage of our method: Using the pixel attacks, one has to choose a source model that is sensitive towards changes on a similar part of the frequency spectrum as the target model. When the source model is identical to the target model, as in the white-box setting, this is obviously the case. But the white-box setting is not usually assumed to be realistic in a real-world scenario. Using our method, though, the perturbation can be manually distributed across frequencies. And, as our previous results have shown, the medium weighting vector seemed to be a universally successful choice.

[8] As we use the pretrained PAT model which is only available as a PyTorch model, we cannot include white-box attacks on the PAT net.



(a) Densenet$_M^{RGB}$ (b) Densenet$_M^{JPEG}$

Figure 4.42: CIFAR10 - White-box LPIPS efficiency of JPEG BIM luma (top) and all (bottom) attacks in comparison with RGB and YC_bC_r attacks on adversarially trained nets. The RGB attack is the same across both rows.

Here, we also observe that the JPEG medium attack is most efficient in the white-box setting on the Densenet$_M^{RGB}$. Applying the attack on the Densenet$_M^{RGB}$ is also slightly more successful than the RGB attack on the Densenet$_M^{JPEG}$. And, while our net does not achieve white-box robustness against RGB attacks, the average robustness across all considered attacks is significantly reduced in comparison to the Densenet$_M^{RGB}$.

So, in all maximum-confidence scenarios considered for CIFAR10, we observe that our JPEG adversarial training results in much better generalization. This proves that our JPEG attacks can be the basis of a defense that shows more generalizing robustness, against both luma- and all-attacks in both pixel and frequency representations. Possibly, further optimization of the weights of each attack could yield a defense that generalizes

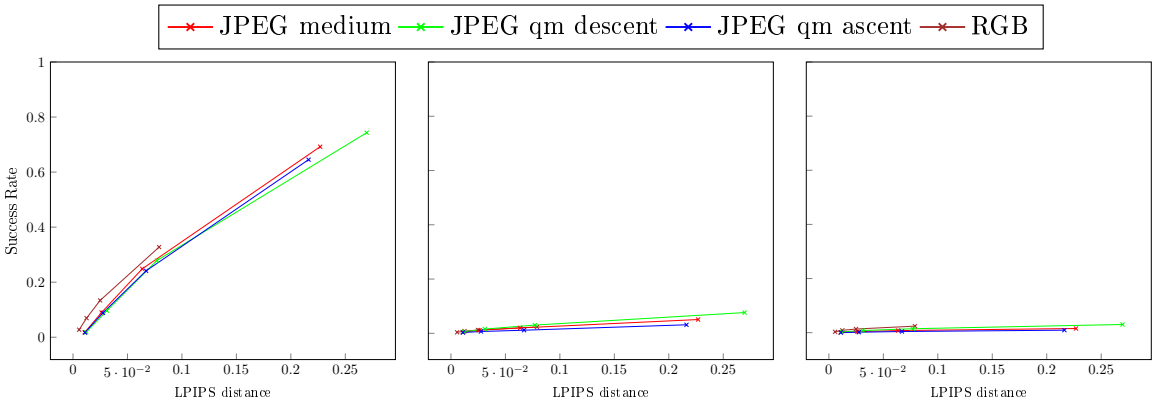
even better and decreases the vulnerability towards low-frequency perturbations for CIFAR10 as well as RGB attacks for IMAGENET. However, it is questionable whether a net can be trained so that it does not show any slight vulnerability on some parts of the frequency spectrum. As our net relies on low-frequency information, it is sensitive and, thus, vulnerable towards low-frequency perturbations. And, as our experiments show, this leads to strong robustness, at least for CIFAR10. Whether the net is more aligned with the human perception or whether it is over-relying on low-frequency information is difficult to determine as the loss gradients do not necessarily reflect the features that reason the net's output but only how the output can be changed. However, the low-frequency structures in the loss gradients imply that the perturbation, which is necessary for a misclassification, would be of a similar and thus visible structure which would tendencially be visible for humans too.

A major disadvantage is that the accuracy of our Densenet$_M^{JPEG}$ on the benign CIFAR10 test dataset is reduced from 91.95% for the undefended Densenet and 82.09% for the Densenet$_M^{RGB}$ to 74.74%. As "*robustness may be at odds with accuracy*" [93], this is an expectable outcome as the net shows better robustness.[9] The reduced clean accuracy is reasoned in the net avoiding to use non-robust features [45] which are mainly located on high frequencies for CIFAR10 as shown before. Thus, the net cannot use these invisible, superficial statistics in the data [31] which reduces the ability to optimize the loss in the benign setting, but increases the robustness and the similarity to the human perception. For CIFAR10, this is a significant finding since it achieves state-of-the-art robustness against unforeseen threat models as well while relying on methods that are more straightforward than the perceptual attacks from [58].

Minimum-Norm Attacks

Having shown the strong robustness and generalization of our Densenet$_M^{JPEG}$ against maximum-confidence attacks, we will now analyze whether the same applies for minimum-norm attacks. Figure 4.43 shows the comparison of both adversarially trained nets, Densenet$_M^{RGB}$ and Densenet$_M^{JPEG}$, in the black-box setting. Both nets are basically robust against all attacks, independent of whether they are applied on RGB pixels or JPEG coefficients.

[9] The PAT net also achieved a similar accuracy of 74.51%.

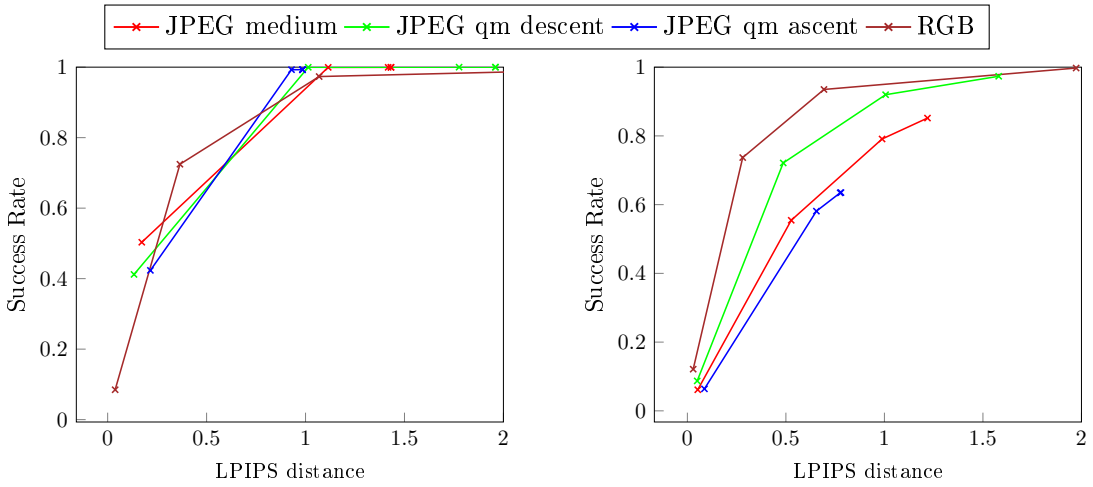


(a) Densenet (b) Densenet$_M^{RGB}$ (c) Densenet$_M^{JPEG}$

Figure 4.43: CIFAR10 - Black-box efficiency for LPIPS-AL on adversarially trained nets. The JPEG attacks only perturbed the luma channel. Images were created on a ResNet.

The Densenet$_M^{JPEG}$ is slightly more vulnerable against RGB perturbations, while the Densenet$_M^{RGB}$ is slightly less robust against JPEG descent perturbations. The minimum-norm attacks are generally not suitable for the adversarially defended black-box setting. Remember the explanations for the existence of adversarial examples named in chapter 2. The explanation of non-robust features states that "*adversarial vulnerability can arise from flipping features in the data that are useful for classification of correct inputs*" [24]. The explanation of linearity states that the vulnerability arises from the linearity and high dimensionality of neural networks that force a heavily changed output when the input is slightly changed in many dimensions. While maximum-confidence attacks try to maximize the difference between the predictions and, thus, the perturbations are often sufficient for transferability even to adversarially trained nets, the minimum-norm attacks only apply the minimal perturbation required to force a misclassification, and as the correctly-classified space around the original image becomes larger, the perturbation is not sufficient anymore. For minimum-norm attacks to be efficient in black-box settings, the source and target model should be relying on similar features and thus offer similar spaces of correctly classified images.

We therefore execute an experiment where an adversarially trained net is used as a source model and the transferability on the other adversarially trained net is measured. The results for CIFAR10 are illustrated in fig. 4.44. For the JPEG attacks, the transferability from the Densenet$_M^{JPEG}$ to the Densenet$_M^{RGB}$ is significantly higher than the other way around. Even for confidence $\kappa = 0$, which is the first point for each attack, about $40-50\%$



(a) Densenet$_M^{JPEG} \rightarrow$ Densenet$_M^{RGB}$ (b) Densenet$_M^{RGB} \rightarrow$ Densenet$_M^{JPEG}$

Figure 4.44: CIFAR10 - Black-box efficiency for LPIPS-AL. The JPEG attacks only perturbed the luma channel. In subfigure a, an additional confidence of $\kappa = 5$ was used.

of the images created on the Densenet$_M^{JPEG}$ are also able to fool the Densenet$_M^{RGB}$. One could now assume that this is only due to the JPEG attack requiring much stronger perturbation on the Densenet$_M^{JPEG}$ as the net is trained with exactly those attacks, which would mean that such strong perturbations are likely to cause a misclassifcation on transfer models as well. However, for the RGB attack we do not see a similar effect. Here, the success when creating the images on the Densenet$_M^{RGB}$ and transferring them to the Densenet$_M^{JPEG}$ is only about $10-15\%$ for $\kappa = 0$, which implies that this observation comes from the general properties of the network. Presumably, large parts of the correctly classified space of the Densenet$_M^{RGB}$ are also included in the Densenet$_M^{JPEG}$'s correctly classified space, as illustrated in fig. C.6, and the minimum-norm adversarial example on the Densenet$_M^{JPEG}$ is likely still adversarial on the Densenet$_M^{RGB}$.

The white-box results for LPIPS-AL on the adversarially trained nets are shown in tables 4.11 and 4.12. In general, we observe that the attacks require much more perceived distortion to be successful on the Densenet$_M^{JPEG}$ than on the Densenet$_M^{RGB}$. This applies to both the CIEDE2000 L_2 and the LPIPS distance. Significantly, the required distance is not just bigger for the JPEG attack, which the net is trained against, but also the RGB attack. Interestingly, the difference becomes even bigger when aiming at a misclassification with a high confidence. Here, the average disortion created for confidence $\kappa = 10$ is 1.07 compared to $\kappa = 0.28$ for the Densenet$_M^{RGB}$, which indicates that the correctly

	Success Rate	CIEDE2000 L_2	LPIPS
LPIPS-AL RGB	100%	60.37	0.029
LPIPS-AL medium	99.98%	57.28	0.053
LPIPS-AL ascent	99.96%	63.45	0.087
LPIPS-AL descent	100%	67.18	0.050

Table 4.11: CIFAR10 - White-box results for JPEG minimum-norm attacks on Densenet_M^{RGB}. $\kappa = 0$ was used for all attacks.

	Success Rate	CIEDE2000 L_2	LPIPS
LPIPS-AL RGB	100%	69.46	0.038
LPIPS-AL medium	99.89%	118.89	0.172
LPIPS-AL ascent	98.07%	135.15	0.22
LPIPS-AL descent	100%	132.99	0.13

Table 4.12: CIFAR10 - White-box results for JPEG minimum-norm attacks on the Densenet_M^{JPEG}. $\kappa = 0$ was used for all attacks.

classified space is much bigger for the Densenet_M^{JPEG} and that the JPEG adversarial training is indeed more successful at "*encouraging the network to be locally constant in the neighborhood of the training data*" [32, p. 262]. We conclude from these results that our net is closer to a "*true human-level understanding*" [32, p.261] as it also generalizes towards the RGB attack that was unseen during training. On both nets, the medium weighting vector is the most efficient regarding the pixel-based CIEDE2000 L_2 distance, while the descent vector is the most efficient regarding the LPIPS distance which is again an example of the LPIPS distance being able to measure structural differences. Similar to the example in fig. 2.16, the ascent and, to a lower extent, the medium weighting vector result in high-frequency noise that might show many edges compared to the descent vector, where there might be less edges and smoother color transitions.

	Success Rate	CIEDE2000 L_2	LPIPS
LPIPS-AL RGB	99.82%	160.34	0.0058
LPIPS-AL medium	92.73%	101.34	0.0056
LPIPS-AL ascent	83.92%	97.01	0.0088
LPIPS-AL descent	99.26%	125.63	0.0090

Table 4.13: IMAGENET - White-box results for JPEG minimum-norm attacks on Densenet_M^{RGB}. $\kappa = 0$ was used for all attacks.

	Success Rate	CIEDE2000 L_2	LPIPS
LPIPS-AL RGB	100%	287.03	0.0060
LPIPS-AL medium	95.93%	103.38	0.0086
LPIPS-AL ascent	80.09%	103.07	0.016
LPIPS-AL descent	97.37%	122.23	0.017

Table 4.14: IMAGENET - White-box results for JPEG minimum-norm attacks on the Densenet$_M^{JPEG}$. $\kappa = 0$ was used for all attacks.

The corresponding IMAGENET results are shown in tables 4.13 and 4.14. Again, we observe that the attack require more perceived distortion when applied to the Densenet$_M^{JPEG}$. For LPIPS, this applies to all attacks. For CIEDE2000, it applies to all attacks but the descent attack. For the RGB attack, the difference between both models is very clear for the CIEDE2000 L_2 distance, but only minimal for the LPIPS distance. Because the RGB attack should concentrate the perturbation in lower frequencies as well since the source model is vulnerable towards low-frequency perturbation, we believe that this can be explained by the same reason we just explained for the disparity between the CIEDE2000 and LPIPS distance for the descent vector.

Overall, these results show that the better generalization of the net defended with our JPEG attacks also applies to minimum-norm attacks as the required distortion is usually bigger than on the net defended with RGB attacks. Again, this underlines the importance of the frequency perspective as it helps achieving better robustness and better alignment with the human perception. In the case of CIFAR10, relying on the lowest frequencies seems to lead to the highest robustness. From the visualization in fig. 4.38, it can be concluded that small perturbations are less likely to be successful as the Densenet$_M^{JPEG}$ relies on the general composition of the image rather than small details.

5 Conclusion

Motivated by the fact that JPEG compression separates perceptible from imperceptible information, this thesis experimented with adversarial perturbations that are applied straight on JPEG coefficients. We found that perturbations on JPEG coefficients are significantly more efficient than RGB or YC_bC_r pixel perturbations in numerous settings:

First, our attacks allow to manually control the perturbation applied on each YC_bC_r channel. Since adversarial perturbations are most effective in the luma channel, as already found by Pestana et al. [74], this enabled us to avoid inefficient color perturbations that are often the result of RGB maximum-confidence attacks.

Second, by weighting the perturbations applied across frequencies we found that against the general assumption that adversarial examples are mainly a high-frequency phenomenon, perturbations on medium frequencies are often most efficient, or achieve at least similar efficiency as the best attack, as they generalize much better than, e.g., high-frequency perturbations where the success is much more dependent on the target model.

Third, our JPEG attacks are able to bypass JPEG compression used in defense more efficiently than state-of-the-art attacks, which is a result of perturbing straight on JPEG coefficients and thus avoiding that the perturbation is removed during quantization anyway. Considering that our attack is technically more straightforward compared to those [85, 86], as it does not have to include an approximation of JPEG compression in the source model or propose a sophisticated rounding scheme to make the perturbations robust against JPEG compression, its effectiveness is even more impressive. Our attacks are especially effective against JPEG compression when 0-coefficients are fixed, which avoids perturbations on those coefficients being removed in the defense. This is especially effective as it transfers towards lower JPEG qualities used in the defense: When a coefficient is 0 for JPEG quality 100, it is 0 for lower qualities as well. As JPEG

compression predominantly removes high-frequency information and perturbations, concentrating perturbation in medium frequencies has been shown to be especially efficient in this setting. Note that these results also imply that JPEG adversarial examples can be saved efficiently as JPEG files without removing perturbation.

Fourth, as found by others [8, 66, 99], we have shown adversarial training using RGB images can result in nets being vulnerable towards low-frequency perturbation. To avoid vulnerability on some parts of the frequency spectrum, we proposed adversarial training with using multiple, weighted JPEG attacks that distribute their perturbation differently across the frequency spectrum. Indeed, this led to a net that generalizes well against perturbations on all frequencies and, importantly, against RGB and $\mathrm{YC_bC_r}$ pixel attacks as well. For now, these results could only be achieved for CIFAR10 though. For IMAGENET, we were unable to experiment with the frequency vectors' weights during training due to the high computational costs. Large parts of these results regarding maximum-confidence attacks have already been submitted as a conference paper which is currently under review [89].

Fifth, this thesis also considered minimum-norm attacks. As these attacks try to find the minimum distortion and should thus optimize efficiency, considering them allowed us to make some interesting observations. Here, the results differ between the datasets. For CIFAR10, the RGB attack is usually stronger in the white-box setting as it is less limited in its scope. While the JPEG attacks only perturbed the luma channel, the RGB attacks also applied color perturbations. While we generally found that color perturbations are often inefficient, applying small color perturbation can make the attack more successful, as also illustrated for JPEG attacks in fig. 4.4. In maximum-confidence attacks, the color perturbations made by the RGB attack were often too strong as they only maximized the prediction loss. In minimum-norm attacks though, the distortion is minimized and thus, only the efficient color perturbations are applied. The quantization step further limits the JPEG attack's scope. For IMAGENET, where the JPEG minimum-norm attacks were more efficient than RGB attacks, these advantages might be less decisive as each JPEG block makes up a much smaller part of the image. Thus, the limitations might be less important for the image's composition and, for example, attacking only luma information might become more important for the attack's efficiency again.

Finally, we also analyzed which frequencies the adversarially trained nets rely on by visualizing the loss gradients in fig. 4.37, creating images by optimizing the models' losses in fig. 4.38 and illustrating the distribution of the gradients across frequencies

in fig. 4.39. We found that, the net trained with our JPEG adversarial training with the specified weights relies on the lowest JPEG frequencies even more than the RGB net. The visualization of loss gradients and features associated with each label implies that the net uses more abstract structures than the RGB net. Consequently, small perturbations have less impact on the net's prediction. From the net's generalization we can conclude, though, that it uses features that are better aligned with human perception than the one trained with RGB adversarial examples as the RGB net can still be fooled by barely visible perturbations when they are concentrated in lower frequencies and the luma channel. Thus, our JPEG attacks can indeed help achieving better robustness and alignment with the human perception by forcing the net to rely on frequencies and features that are relevant for humans as well, and preventing it from using non-robust features, which shows the importance of looking at adversarial attacks and robustness from a frequency perspective.

Presumably, there are even better ways of encouraging the net to use robust features. During JPEG compression, each 8×8 block is transformed to the frequency space using the DCT. As some works mentioned in section 2.5 , e.g. [8, 36, 66, 84], already did, the DCT could also be performed image-wise instead of block-wise. In our attack, the frequency weighting vector corresponds to one block. Thus, perturbations as illustrated in fig. 2.18 can not be achieved. While each 8×8 block makes up a significant part of the image for CIFAR10, the perturbations are often perceived block-wise anyway for IMAGENET such that the perceived difference between low- and high-frequency perturbations is quite small. While there were still significant differences between each weighting vector's success, this might influence the adversarial training's effectiveness as the images often look quite similar. Thus, applying the DCT on the whole image and then weighting the perturbation for each of the $h \times w$ frequencies might increase the variety of the created images and, thus, improve the net's generalization. The existing studies on such attacks apply their perturbation in RGB space though, whereas we have shown that $\mathrm{YC_bC_r}$ is the more suitable color space, and does not measure the true perceptual distance, but L_2 distances. This is problematic especially when perturbing different parts of the frequency spectrum this is problematic, as they do not measure structural differences.

Another approach to achieve more robustness is the perceptual adversarial training by Laidlaw et al. [58] which limits the perceived distortion of the adversarial examples using the LPIPS distance. As shown in the original paper and our experiments on adversarial training, the approach shows much better generalization than the standard RGB adversarial training, and similar performance as our JPEG net, although it was

slightly more vulnerable towards medium- and low-frequency perturbations on the luma channel. The perceptual threat model introduced by Laidlaw et al. [58] can be seen as a big advance in achieving strong robustness and generalization. As it limits the distortion using the LPIPS distance, the perturbation should vary much more between images than for standard pixel-based RGB attacks and especially maximum-confidence attacks like BIM, which do not yield the most efficient perturbation but the one that maximizes the loss. Thus, the PAT implicitly leads to stronger robustness on all frequencies and against many attacks. Additionally, it is also effective against common corruptions [51, 58].

Comparing the effectiveness of our approach and PAT is difficult as our experiments did not include the same attacks as [58]. On the attacks included in our work, both performed very similarly with the PAT net being slightly more robust against ε_{all}-attacks while our Densenet$_M^{JPEG}$ was slightly more robsut against luma attacks. However, our method requires manual weighting of the attacks for each dataset. Therefore, we generally believe that an approach that includes the LPIPS distance and automatically finds the most efficient perturbation in the attack might be the most promising approach to achieve a "*true human-level understanding*" [32, p.261].

But what does this actually mean and how could it be achieved? And how could the results from our work help improving the current state-of-the-art methods, e.g., the perceptual adversarial training?

A "*true human-level understanding*" [32, p.261] would imply that the net uses similar features as humans, shows similar robustness towards various types of perturbations and also achieves a similar accuracy on the clean, benign dataset. The adversarially trained nets considered in this thesis, the Densenet$_M^{JPEG}$ and PAT, achieved a clean accuracy on CIFAR10 of only about 74%. According to Ho-Phuoc [40], a human achieves about 94% accuracy. While standard nets achieve and surpass that number, adversarially trained nets do not yet, as "*robustness may be at odds with accuracy*" [93]. In [40], it is also stated that standard nets achieve better accuracy for images that are difficult to classify for humans which is an example for them using "*superficial statistics in the data*" [31]. While adversarial training reduces the usage of such information and increases robustness, it also reduces the clean accuracy which shows that a "*true human-level understanding*" [32, p.261] is not reached.

So, how could the approach be further improved? The perceptual adversarial training [58] uses adversarial images that minimize the perceptual distance. Aiming at a net that uses similar features as humans, using a distance metric that is well aligned with human

perception in order to limit the perturbation seems like a sensible choice. And while the created images should vary much more than for standard maximum-confidence attacks like BIM due to the use of the perceptual distance, the scope of the LPA attack used to create adversarial examples is still very limited as it only applies noise to the image. This noise, though, is also influenced by how the LPIPS distance weights different parts of the frequency spectrum which itself depends on how the LPIPS network was trained. It might be that LPIPS overvalues low- or high-frequency perturbation and when using it for creating adversarial examples, one is preferred over the other. So, manually determining the frequencies in which the perturbation is concentrated could also be helpful even for perceptual adversarial training, as it increases the variety of adversarial images.

From the result that JPEG adversarial training leads to stronger robustness than RGB adversarial training, we draw the conclusion that this variety is very important for achieving robustness against unforeseen threat models as well. The effectivity of the JPEG adversarial training is not necessarily or exclusively due to the use of JPEG coefficients but the variety of the created images as well. Other methods that perturb in the frequency spectrum (see section 2.5) might thus be as effective as ours. But using JPEG coefficients has the advantage that non-robust, high-frequency information is removed during the quantization and their usage is therefore prevented.

Another way of achieving more variety of adversarial examples would be to design an attack that does not just use noise but also geometric transformations such as rotations, zooming or cropping. Unlike standard data augmentation, these transformations would be applied such that the loss is maximized. A difficulty when designing such an attack could be that LPIPS might overvalue the distortion created by the geometric transformations. Figure 5.1 shows examples where the original images have been rotated or disturbed by some random noise. The noisy images tend to have smaller LPIPS distances. It is unclear, though, which of the altered images is, for human perception, more similar to the original because all three show the same objects. Thus, quantifying the perceptual distance is not an easy task and while the LPIPS distance is much more suited than L_2 distance, it still is not a perfect measure.

However, Laidlaw et al. [58] already adversarially trained a net where the LPIPS distance is computed using the net itself. When more variety is used from the training's beginning by also including adversarial geometric transformations, the LPIPS distance might thus be even better at approximating the true perceptual distance and, simultaneously, the robustness could be increased. We experiment with an attack that enables us to distribute

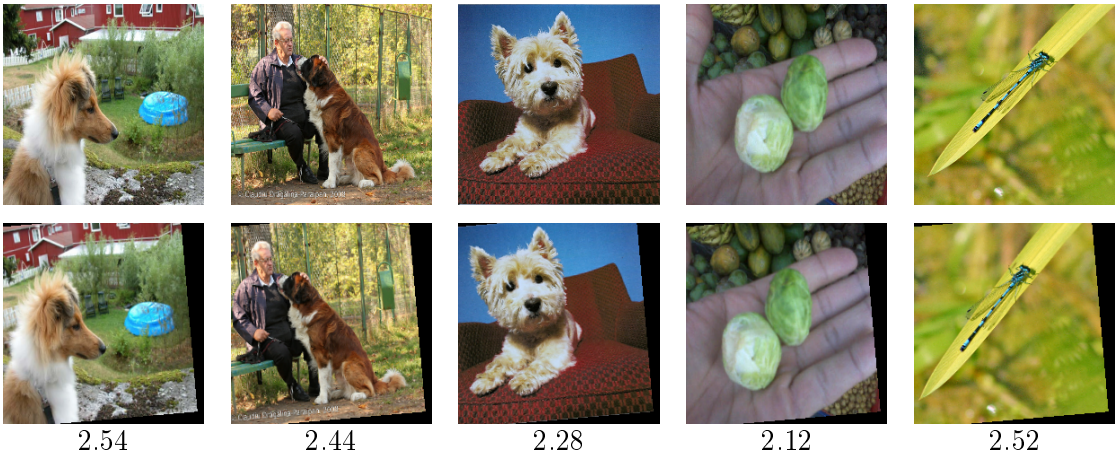


Figure 5.1: Original (top), rotated and shifted (middle) and noisy (bottom) IMAGENET images. The images in the middle row were created using a Tensorflow implementation [100] of Jaderberg et al.'s Spatial Transformer Networks [46]. The LPIPS distances are given below the images. Rotated images tend to have higher LPIPS distances.

the perturbations across DCT frequencies, where the DCT is performed image-wise, though, as described above, as well as to rotate and shift the image using the Spatial Transformer Network [46]. The application of noise is similar to the previous works named in section 2.5 as the DCT and frequency weighting is applied on the gradients of the pixel image. However, it uses YC_bC_r and weights the DCT frequencies instead of masking them. The attack uses the same loss function (see eq. (2.35)) as the FLPA attack which uses the LPIPS distance and is used in the PAT [58]. Details on the attack can be found in appendix B. Figure 5.2 shows adversarial examples that were created using the attack. Due to the frequency weighting vectors, the distinction between ε_{all}- and ε_Y-attacks, and the geometric transformations, the images show much more variety which might increase the nets generalization and robustness as well as the alignment with human perception when this or a similar attack is used during adversarial training.

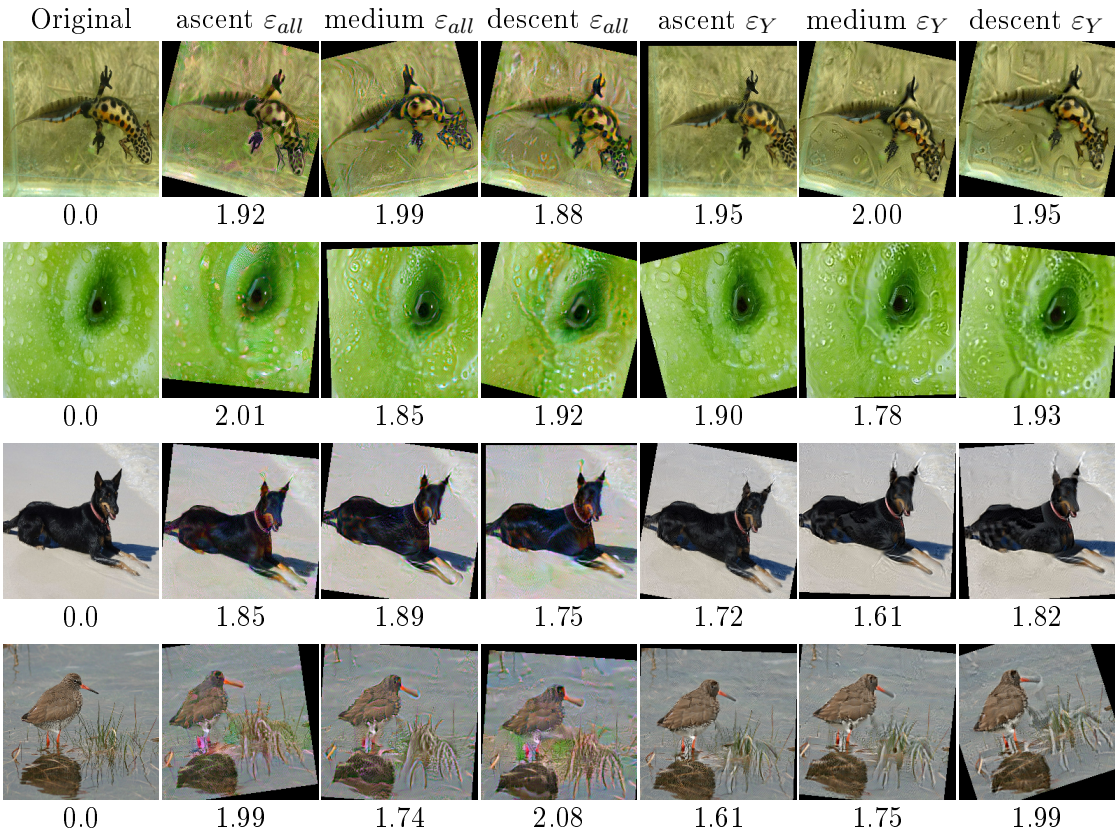


Figure 5.2: Adversarial examples created by perturbing $\mathrm{YC_bC_r}$ images with adversarial noise, where the gradients are multiplied by the frequency weighting vectors, but also rotation and spatial shifts. To visualize differences, the perturbation was chosen to be bigger than necessary for a misclassifcation; the images were on the $\mathrm{Densenet}_M^{RGB}$. Details on the attack can be found in appendix B.

It is difficult to predict whether one of the approaches discussed here would result in more robustness, high clean accuracy and alignment with human perception. While adversarial training significantly increases the robustness, the sample images in section 4.4.3 have also shown that there is still no visible distortion necessary when attacking in the white-box setting. Using a perceptual distance, like PAT does, might be the most promising approach to overcome this. However, the variety of the created images should be increased to also include perturbations on different parts of the frequency spectrum, geometric transformations etc.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[2] Antonio A. Abello, Roberto Hirata, and Zhangyang Wang. Dissecting the high-frequency bias in convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 863–871, June 2021.

[3] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. doi: 10.1109/T-C.1974.223784.

[4] Ahmed Aldahdooh, Wassim Hamidouche, Sid Ahmed Fezza, and Olivier Déforges. Adversarial example detection for DNN models: a review and experimental comparison. *Artif. Intell. Rev.*, 55(6):4403–4462, 2022. doi: 10.1007/s10462-021-10125-w.

[5] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018.

[6] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of

Proceedings of Machine Learning Research, pages 284–293. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/athalye18b.html.

[7] Rowel Atienza. *Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more*. Packt Publishing Ltd, 2020.

[8] Rémi Bernhard, Pierre-Alain Moëllic, Martial Mermillod, Yannick Bourrier, Romain Cohendet, Miguel Solinas, and Marina Reyboz. Impact of spatial frequency based constraints on adversarial robustness. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. doi: 10.1109/IJCNN52387.2021.9534307.

[9] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017. URL http://arxiv.org/abs/1712.09665.

[10] Adrian Bussone, Simone Stumpf, and Dympna O'Sullivan. The Role of Explanations on Trust and Reliance in Clinical Decision Support Systems. In *2015 International Conference on Healthcare Informatics*, pages 160–169, October 2015. doi: 10.1109/ICHI.2015.26.

[11] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, CCS '19, pages 2267–2281, New York, NY, USA, 11 2019. ACM. ISBN 9781450367479. doi: 10.1145/3319535.3339815. URL https://doi.org/10.1145/3319535.3339815.

[12] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. *CoRR*, abs/1907.05418, 2019. URL http://arxiv.org/abs/1907.05418.

[13] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, may 2017. doi: 10.1109/SP.2017.49.

[14] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, page 3–14, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450352024. doi: 10.1145/3128572.3140444. URL https://doi.org/10.1145/3128572.3140444.

[15] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7, 2018. doi: 10.1109/SPW.2018.00009.

[16] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35(1):53–65, January 2018. ISSN 1558-0792. doi: 10.1109/MSP.2017.2765202.

[17] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E. Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression. *CoRR*, 2017. URL http://arxiv.org/abs/1705.02900.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[19] Yao Deng, James Xi Zheng, Tianyi Zhang, Chen Chen, Guannan Lou, and Miryung Kim. An analysis of adversarial attacks and defenses on autonomous driving models. In *2020 IEEE International Conference on Pervasive Computing and Communications, PerCom 2020, Austin, TX, USA, March 23-27, 2020*, pages 1–10. IEEE, 2020. doi: 10.1109/PerCom45495.2020.9127389.

[20] Yingpeng Deng and Lina J Karam. Frequency-tuned universal adversarial perturbations. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 494–510. Springer, 2020.

[21] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9185–9193. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00957.

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

[23] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Quantization guided jpeg artifact correction. *Proceedings of the European Conference on Computer Vision*, 2020.

[24] Logan Engstrom, Andrew Ilyas, Aleksander Madry, Shibani Santurkar, Brandon Tran, and Dimitris Tsipras. A discussion of 'adversarial examples are not bugs, they are features': Discussion and author responses. *Distill*, 2019. doi: 10.23915/distill.00019.7. URL https://distill.pub/2019/advex-bugs-discussion/original-authors.

[25] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling CNNs with simple transformations, 2019. URL https://openreview.net/forum?id=BJfvknCqFQ.

[26] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. *CoRR*, abs/1807.07769, 2018. URL http://arxiv.org/abs/1807.07769.

[27] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] Sid Ahmed Fezza, Yassine Bakhti, Wassim Hamidouche, and Olivier Déforges. Perceptual evaluation of adversarial attacks for cnn-based image classification. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, 2019. doi: 10.1109/QoMEX.2019.8743213.

[29] Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

[30] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

[31] Justin Gilmer and Dan Hendrycks. A discussion of 'adversarial examples are not bugs, they are features': Adversarial example researchers need to expand what is meant by 'robustness'. *Distill*, 2019. doi: 10.23915/distill.00019.1. URL https://distill.pub/2019/advex-bugs-discussion/response-1.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, Cambridge, Massachusetts, 2016. ISBN 9780262035613.

[33] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[34] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, Canada.*, 2018. URL https://papers.neurips.cc/paper/2018/file/7af6266cc52234b5aa339b16695f7fc4-Paper.pdf.

[35] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

[36] Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 1127–1137. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/guo20a.html.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90.

[39] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[40] Tien Ho-Phuoc. CIFAR10 to compare visual recognition performance between deep neural networks and humans. *CoRR*, abs/1811.07270, 2018. URL http://arxiv.org/abs/1811.07270.

[41] Andreas Holzinger, Georg Langs, Helmut Denk, Kurt Zatloukal, and Heimo Müller. Causability and explainability of artificial intelligence in medicine. *WIREs Data Mining and Knowledge Discovery*, 9(4):e1312, 2019. ISSN 1942-4795. doi: https://doi.org/10.1002/widm.1312.

[42] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.243.

[43] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016. URL http://arxiv.org/abs/1602.07360.

[44] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*,

volume 80 of *Proceedings of Machine Learning Research*, pages 2142–2151. PMLR, 2018. URL http://proceedings.mlr.press/v80/ilyas18a.html.

[45] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 125–136, May 2019. URL https://proceedings.neurips.cc/paper/2019/hash/e2c420d928d4bf8ce0ff2ec19b371514-Abstract.html.

[46] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/33ceb07bf4eeb3da587e268d663aba1a-Paper.pdf.

[47] Matt Jordan, Naren Manoj, Surbhi Goel, and Alexandros G. Dimakis. Quantifying perceptual distortion of adversarial examples, 2019. URL https://arxiv.org/abs/1902.08265.

[48] Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. *CoRR*, abs/1908.08016, 2019. URL http://arxiv.org/abs/1908.08016.

[49] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[50] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[51] Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*,

pages 1012–1021. PMLR, 01–05 Aug 2022. URL https://proceedings.mlr.press/v180/kireev22a.html.

[52] Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[53] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[55] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016. URL http://arxiv.org/abs/1611.01236.

[56] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HJGU3Rodl.

[57] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10408–10418, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/6e923226e43cd6fac7cfe1e13ad000ac-Abstract.html.

[58] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *International Conference on*

Learning Representations, 2021. URL https://openreview.net/forum?id=dFwBosAcJkN.

[59] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, jul 2018. doi: 10.24963/ijcai.2018/585. URL https://doi.org/10.24963%2Fijcai.2018%2F585.

[60] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128:261–318, 2020.

[61] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles, 2017. URL https://arxiv.org/abs/1707.03501.

[62] M. R. Luo, G. Cui, and B. Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application*, 26(5):340–350, 2001. doi: 10.1002/col.1049.

[63] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 110:107332, February 2021. ISSN 0031-3203. doi: 10.1016/j.patcog.2020.107332. URL https://www.sciencedirect.com/science/article/pii/S0031320320301357.

[64] L.W. MacDonald. Using color effectively in computer graphics. *IEEE Computer Graphics and Applications*, 19(4):20–35, 1999. doi: 10.1109/38.773961.

[65] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

[66] Shishira R. Maiya, Max Ehrlich, Vatsal Agarwal, Ser-Nam Lim, Tom Goldstein, and Abhinav Shrivastava. A frequency perspective of adversarial robustness. *CoRR*, abs/2111.00861, 2021. URL https://arxiv.org/abs/2111.00861.

[67] K. McLAREN. Xiii—the development of the cie 1976 (l* a* b*) uniform colour space and colour-difference formula. *Journal of the Society of Dyers and Colourists*, 92(9):338–341, 1976. doi: https://doi.org/10.1111/j.1478-4408.1976.tb03301.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1478-4408.1976.tb03301.x.

[68] Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. Cascade adversarial machine learning regularized with a unified embedding. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=HyRVBzap-.

[69] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[70] Aude Oliva, Antonio Torralba, and Philippe G. Schyns. Hybrid images. *ACM Trans. Graph.*, 25(3):527–532, jul 2006. ISSN 0730-0301. doi: 10.1145/1141911.1141919.

[71] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, page 506–519, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349444. doi: 10.1145/3052973.3053009. URL https://doi.org/10.1145/3052973.3053009.

[72] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

[73] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[74] Camilo Pestana, Naveed Akhtar, Wei Liu, David Glance, and Ajmal Mian. Adversarial attacks and defense on deep learning classification models using ycbcr color images. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2021. doi: 10.1109/IJCNN52387.2021.9533495.

[75] Maura Pintor, Fabio Roli, Wieland Brendel, and Battista Biggio. Fast minimum-norm adversarial attacks through adaptive norm constraints. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20052–20062. Curran Associates, Inc., 2021.

[76] Konstantinos Plataniotis and Anastasios N Venetsanopoulos. *Color image processing and applications*. Springer Science & Business Media, 2000.

[77] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. Barrage of random transforms for adversarially robust defense. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[78] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. doi: 10.1162/neco_a_00990.

[79] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.91.

[80] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346 – 360, mar 2020. ISSN 2095-8099. doi: 10.1016/j.eng.2019.12.012.

[81] Jérôme Rony, Luiz G. Hafemann, Luiz S. Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long*

Beach, CA, USA, June 16-20, 2019, pages 4322–4330. Computer Vision Foundation / IEEE, November 2019. doi: 10.1109/CVPR.2019.00445. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Rony_Decoupling_Direction_and_Norm_for_Efficient_Gradient-Based_L2_Adversarial_Attacks_CVPR_2019_paper.html.

[82] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 1528–1540, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978392. URL https://doi.org/10.1145/2976749.2978392.

[83] Mahmood Sharif, Lujo Bauer, and Michael K. Reiter. On the suitability of lp-norms for creating and preventing adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[84] Yash Sharma, Gavin Weiguang Ding, and Marcus A. Brubaker. On the effectiveness of low frequency perturbations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 3389–3396, Macao, China, 2019. AAAI Press. ISBN 9780999241141.

[85] Mengte Shi, Sheng Li, Zhaoxia Yin, Xinpeng Zhang, and Zhenxing Qian. On generating jpeg adversarial images. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021. doi: 10.1109/ICME51207.2021.9428243.

[86] Richard Shin and Dawn Song. Jpeg-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*, volume 1, page 8, 2017.

[87] Bruno Siciliano. *Handbook of Robotics*. Springer, Berlin, 2008. ISBN 978-3-540-23957-4.

[88] Kolja Sielmann. Adversarial Perturbations straight on JPEG Coefficients. *Student Term Paper, Hauptprojekt, Hamburg University of Applied Sciences.*, November 2022.

[89] Kolja Sielmann and Peer Stelldinger. Adversarial Perturbations Straight on JPEG Coefficients. *Submitted to GCPR 2023. Currently Under Review.*, 2023.

[90] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

[91] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[92] Erico Tjoa and Cuntai Guan. A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2020. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2020.3027314. URL http://arxiv.org/abs/1907.07374. arXiv: 1907.07374.

[93] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, May 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

[94] Yusuke Tsuzuku and Issei Sato. On the structural sensitivity of deep convolutional networks to the directions of fourier basis functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[95] Gregory K. Wallace. The JPEG still picture compression standard. *Commun. ACM*, 34(4):30–44, 1991. doi: 10.1145/103085.103089.

[96] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[97] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

[98] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, volume abs/1801.02612. OpenReview.net, 2018. URL https://openreview.net/forum?id=HyydRMZC-.

[99] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/b05b57f6add810d3b7490866d74c0053-Paper.pdf.

[100] Kevin Zakka. Stn: Spatial transformer networks. https://github.com/kevinzakka/spatial-transformer-network, June 2018. Tensorflow Implementation.

[101] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[102] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[103] Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019. doi: 10.1109/TNNLS.2018.2876865.

[104] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, dec 1997. ISSN 0098-3500. doi: 10.1145/279232.279236. URL https://doi.org/10.1145/279232.279236.

[105] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD*

International Conference on Knowledge Discovery & Data Mining, KDD '18, page 2847–2856, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220078. URL https://doi.org/10.1145/3219819.3220078.

A Absolute vs. Relative Perturbations

As mentioned in chapter 3, we used absolute perturbation budgets in a previous work [88]. The FGSM perturbation was then computed by

$$\begin{aligned} Y' &= Y + \text{sign}(\nabla_Y(\text{J}(\text{rgb}(x), y))) \cdot \varepsilon_Y^{\text{abs}} \cdot \lambda_Y \\ C_b' &= C_b + \text{sign}(\nabla_{C_b}(\text{J}(\text{rgb}(x), y))) \cdot \varepsilon_{C_b}^{\text{abs}} \cdot \lambda_{C_b} \\ C_r' &= C_r + \text{sign}(\nabla_{C_r}(\text{J}(\text{rgb}(x), y))) \cdot \varepsilon_{C_r}^{\text{abs}} \cdot \lambda_{C_r}, \end{aligned} \tag{A.1}$$

where unlike in the current version, $\varepsilon_Y^{\text{abs}}, \varepsilon_{C_b}^{\text{abs}}, \varepsilon_{C_r}^{\text{abs}}$ were scalars.

This resulted in some difficulties in controlling the perturbation and differences in results that will be shortly explained here.

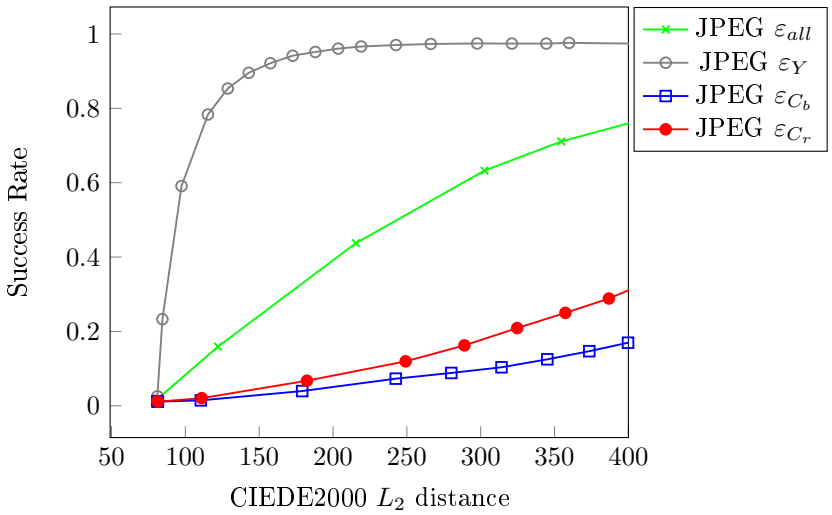
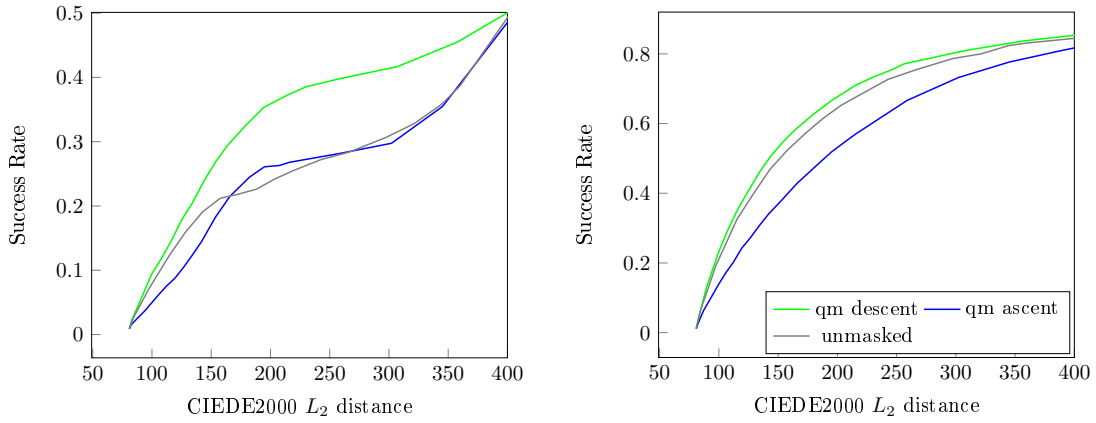


Figure A.1: CIFAR10 - Success rates for unmasked MI-FGSM in dependence of CIEDE2000 L_2 Distance on an undefended DenseNet. Here, absolute epsilon values were used.



(a) Densenet$^{\varepsilon 16}_{RGB}$ (b) Densenet^{jq95}

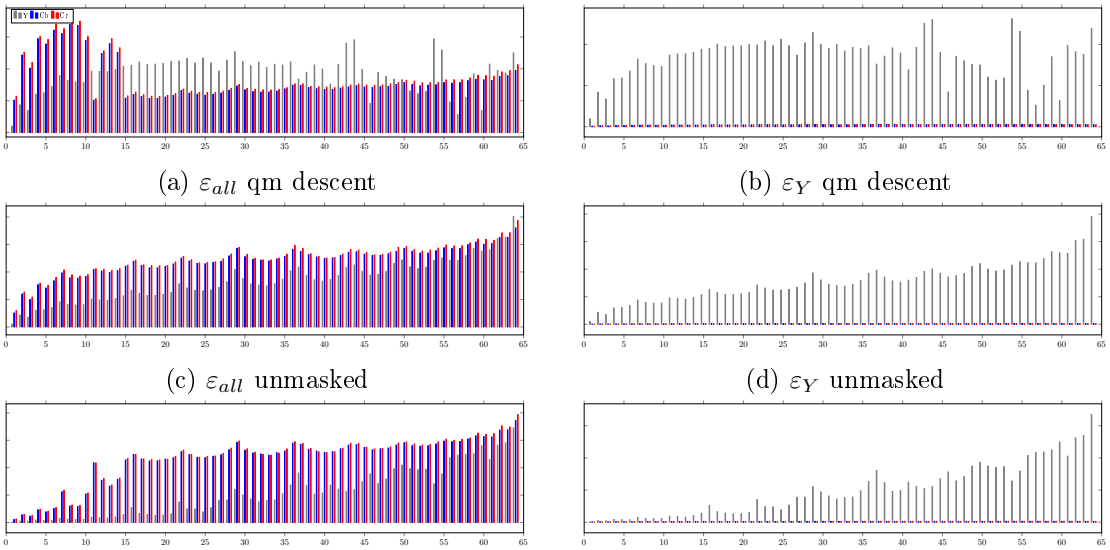
Figure A.2: CIFAR10 - MI-FGSM with perturbations on the luma channel only. The weighting vectors are illustrated in fig. 4.9. Absolute epsilon values were used. The Densenet$^{\varepsilon 16}_{RGB}$ was trained with the cascade adversarial training from [68].

Besides the fact that the difference in attack efficiency between luma and all attacks is much bigger when using an absolute perturbation budget that has been explained in section 4.3.1 and is visualized in fig. A.1, there are some important observations on the success of the different weighting vectors.

Figure A.2 illustrates this comparison for an absolute perturbation budget. As we did for relative perturbations, we observed also for absoulte perturbations that the qm descent weighting vector is much more successful than the ascent and unmasked vector on the adversarially trained net. However, due to the fact that low frequency coefficients usually have much higher amplitudes, the qm descent weighting vector led to the perturbation not really being concentrated on low frequencies, but on medium frequencies, as shown in fig. A.3.

Similarly, the unmasked vector leads to perturbations that are concentrated on high frequencies. This made the argumentation more difficult, as a high value in the weighting vector λ did not necessarily imply a high relative perturbation on the corresponding frequency. For comparison, fig. 4.10 illustrates this for relative perturbation budgets. There, the resulting perturbation is closer to the weighting vectors.

Generally, we expect that both versions can yield the same results when the weighting vectors are chosen accordingly, but using relative perturbation budgets simplifies the



(a) ε_{all} qm descent (b) ε_Y qm descent

(c) ε_{all} unmasked (d) ε_Y unmasked

(e) ε_{all} qm ascent (f) ε_Y qm ascent

Figure A.3: Average relative perturbations when an absolute perturbation budget is used, given by $|\frac{Y'-Y}{Y+1}|$ for luma and correspondingly for chroma channels, made by JPEG and $\mathrm{YC_bC_r}$ BIM on JPEG frequencies for CIFAR10 on an undefended ResNet. The x-axis represents the frequencies' post zig-zag order.

selection of a weighting vector and the argumentation. A positive side effect is the increased success of attacks on all channels.

B Image-DCT and Geometric Transformations Attack

To achieve more variety during adversarial training, we design an attack that is able to weight perturbations across DCT frequencies and also apply geometric transformations. The geometric transformation is applied using the Spatial Transformer Network [46].[1] The DCT is performed on the gradients of the full image, instead of on blocks of pixels as for our JPEG attacks, which makes the attack similar to those mentioned in section 2.5. However, the perturbation and frequency weighting is applied in $\mathrm{YC_bC_r}$ color space and again, the perturbation can be controlled for each channel separately. Additionally, as for our JPEG attacks, the frequency vectors are used to weight rather than to mask DCT frequencies.

It is important to mention that this attack is designed for adversarial training and not necessarily for evaluation. As geometric transformations usually result in more perceived distortion, at least when measured using a perceptual distance, they might be less efficient than attacks that do not geometrically transform the image.

The attack uses the same loss function as LPA (see eq. (2.35)), where, for now, we use a fixed $\lambda = 2$. The loss function results in a soft LPIPS bound. In the original LPA attack, the image is projected back into the bound at the attack's end. As in the FLPA attack which is used for adversarial training, we do not perform this step. In fact, projecting it back into the bound would be difficult due to the geometric transformations. As the attack is designed for adversarial training, this is not problematic though. The attack's pseudocode is presented in fig. B.1.

[1] We use the Tensorflow implementation [100].

Algorithm 1 Image DCT and Geometric Transformation Attack.

Input: $\mathrm{YC_bC_r}$ pixel images $x = (Y, C_b, C_r)$, where $Y, C_b, C_r \in [0, 255]^{h \times w}$,
noise step-sizes for each channel $\alpha_Y, \alpha_{C_b}, \alpha_{C_r}$,
rotation degree step size α_d, shift step sizes $\alpha_{x_s}, \alpha_{y_s}$, number of steps T,
LPIPS bound ε, frequency weighting vector $\lambda \in [0, 1]^{64}$.
Output: adversarial image x'.

procedure ATTACK
  Unzigzag λ and resize from shape 8×8 to $h \times w$ using bilinear interpolation
  ▷ Since we apply the DCT on the whole image.
  $d = 0, x_s = 0, y_s = 0$ ▷ Variables for rotation degree and x,y-shift.
  Initialize $x' = (Y', C_b', C_r') = (Y, C_b, C_r)$

  for $t = 1, \ldots, T$ **do**
    $L = \max_{x'} \mathrm{J}(\mathrm{stn}(\mathrm{rgb}(x')), y) - 2.\max(0, \mathrm{lpips}(\mathrm{rgb}(x), \mathrm{stn}(\mathrm{rgb}(x'))) - \varepsilon)$
    ▷ rgb() converts the $\mathrm{YC_bC_r}$ image to RGB.
    ▷ stn() performs the geometric transformation using d, x_s, y_s.

    $g_{noise}^{Y}, g_{noise}^{C_b}, g_{noise}^{C_r} = g_{noise} = \nabla_{x'} L$
    ▷ Compute the gradients for applying the adversarial noise
    $g_{noise}^{Y} = idct(\lambda \cdot dct(g_{noise}^{Y}))$
    ▷ Apply frequency-wise weighting. Correspondingly for C_b, C_r.

    $g_d = \nabla_d L$ ▷ Compute the gradient for the rotation degree
    $g_{x_s} = \nabla_{x_s} L, g_{y_s} = \nabla_{y_s} L$ ▷ Compute the gradient for the x,y shifts

    $Y' = Y' + \mathrm{sign}(g_{noise}^{Y}) \cdot \alpha_Y$ ▷ Apply the noise perturbation.
    $C_b' = C_b' + \mathrm{sign}(g_{noise}^{C_b}) \cdot \alpha_{C_b}$
    $C_r' = C_r' + \mathrm{sign}(g_{noise}^{C_r}) \cdot \alpha_{C_r}$

    $d = d + \mathrm{sign}(g_d) \cdot \alpha_d$ ▷ Perturb the rotation degree.
    $x_s = x_s + \mathrm{sign}(g_{x_s}) \cdot \alpha_{x_s}, y_s = y_s + \mathrm{sign}(g_{y_s}) \cdot \alpha_{y_s}$ ▷ Perturb x,y shift.

    $x' = (Y', C_b', C_r')$
    ▷ Note that the geometric transformations are not included here but they are included when computing the loss and in the return statement.

  Return $\mathrm{stn}(x')$

Figure B.1: Algorithm - Image DCT and Geometric Transformation Attack.

C Additional Figures

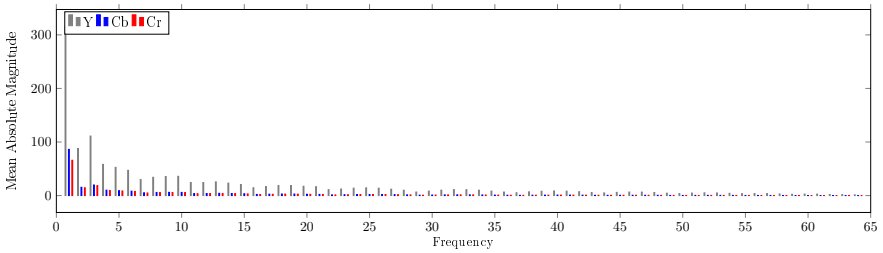
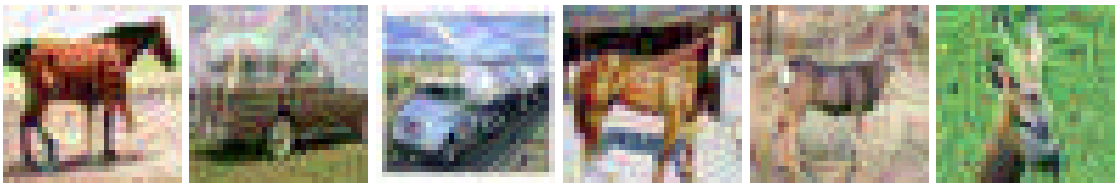


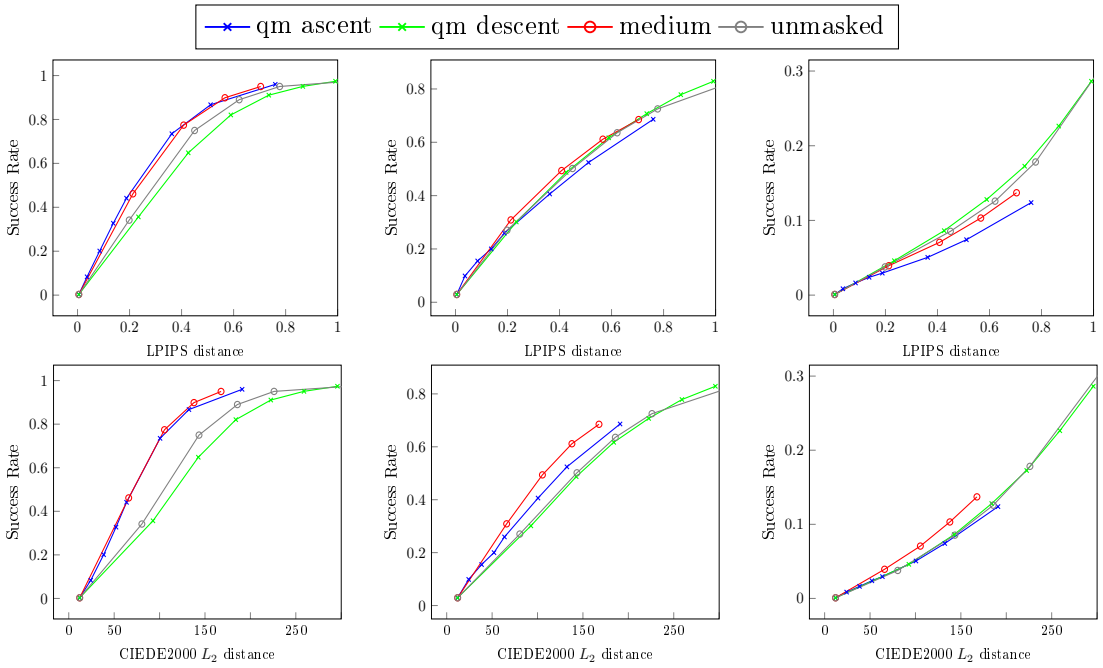
Figure C.1: CIFAR10 - Mean absolute value for JPEG coefficients. Chroma coefficients tend to have smaller absolute values than luma coefficients.



(a) non-robust

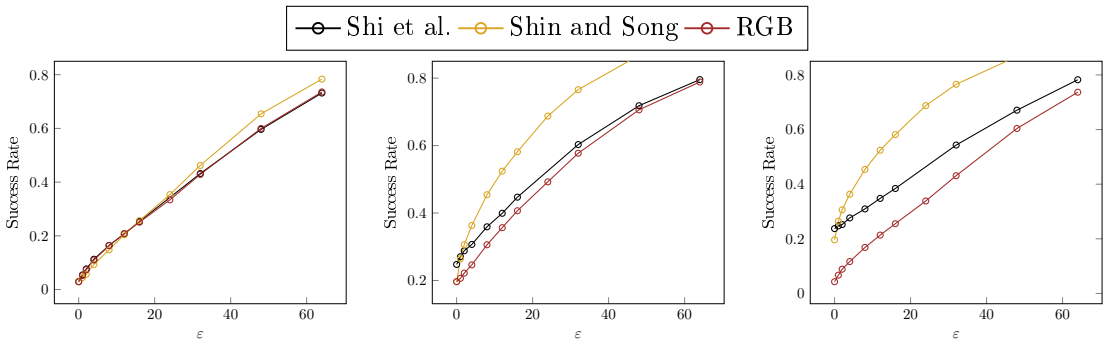
(b) robust

Figure C.2: Sample images from the CIFAR10 robust and non-robust images from the datasets created in [45].



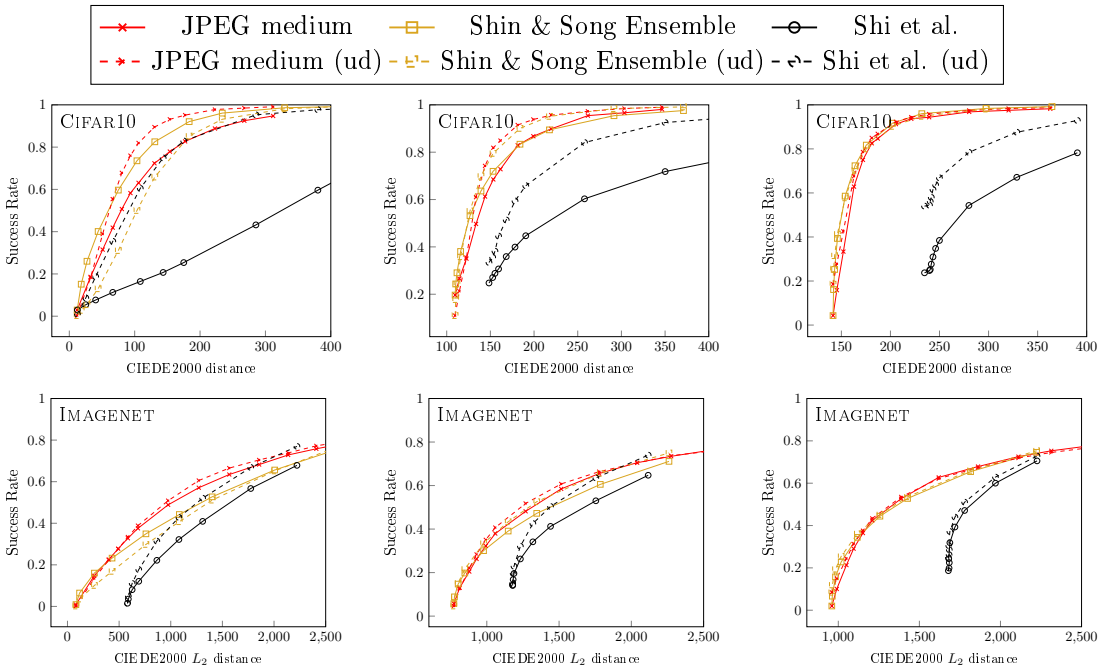
(a) Densenet (b) Densenet^{jq50} (c) Densenet$^{RGB}_{M}$

Figure C.3: CIFAR10 - Black-box efficiency for JPEG BIM with perturbations on all three channels.



(a) jq=100 (b) jq=75 (c) jq=50

Figure C.4: CIFAR10 - Black-box success rates on the Densenet^{jq50} for BIM attacks. For the JPEG and YC_bC_r attacks, only luma was perturbed. The JPEG quality used in attack varies between the subfigures. Images were created on a Resnet.



(a) jq=100 (b) jq=75 (c) jq=50

Figure C.5: CIEDE2000 efficiency on the Densenet^{jq50} and the undefended (ud) DenseNet for BIM attacks, for different JPEG qualities (100, 75, 50) used in attack.

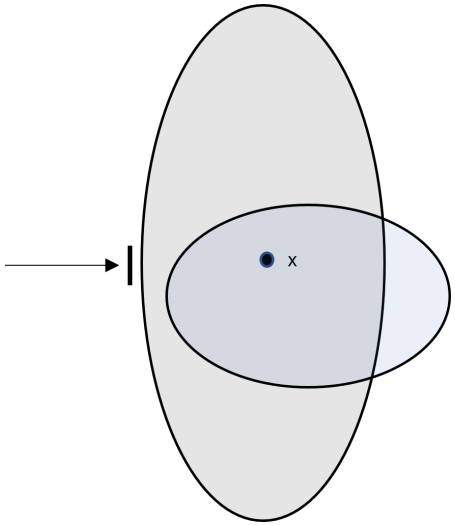


Figure C.6: Venn diagram of correctly classified spaces for an original image x. The gray area is correctly classified by the Densenet$_M^{JPEG}$, while the blue area is the Densenet$_M^{RGB}$'s correctly classified input space.

Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original