University of Münster

Department of Information Systems

Application of Transferable Adversarial Attacks on Convolutional Neuronal Networks: An Evaluation of Existing Attack and Defense Mechanisms

Bachelor Thesis

submitted by

Linus Stach

Chair of Data Science:
Machine Learning and Data Engineering

Principal Supervisor	Professor Fabian Gieseke
Supervisor	Moritz Seiler, M.Sc. Chair for Data Science: Machine Learning and Data Engineering
Matriculation Number	505109
Field of Study	Information Systems
Contact Details	linus.stach@uni-muenster.de
Submission Date	28.07.2022

Contents

1 Introduction

Following the last twenty years of substantial improvements in computational capabilities, it is now possible to efficiently train large machine learning models with many parameters, which has brought artificial neural networks (ANN) into the focus of major research efforts since the 2010s (Krizhevsky, Sutskever, & Hinton, 2012). The increase in computational efficiency in regards to training ANNs has led not only to increased research activity but also to these methods being deployed in various domains, such as satellite data analysis, medical data analysis, and speech recognition. In the future, it is expected that more and more systems will be replaced, extended, or defined by deep learning components.

Early advances in deep learning date back to the 1940s with McCulloch and Pitts (1943), who investigated biologically inspired methods of training models. Subsequent milestones include the perceptron (Rosenblatt, 1958, 1961) and back-propagation (Rumelhart, Hinton, & Williams, 1986). A particularly promising field that relies on deep learning is computer vision, in which large amounts of spatial and image data are processed and analyzed. In the annual ImageNet Large Scale Visual Recognition Challenge, where objects have to be classified from 1,000 possible classes, Convolutional Neural Networks (CNN) now achieve close to human results (Dodge & Karam, 2017). A CNN is a special ANN architecture, which can extract features from multidimensional data and is therefore suitable for spatial data and primarily used in computer vision. CNNs in their modern form are based on the results of LeCun et al. (1989). Advances by Krizhevsky, Sutskever, and Hinton (2012) showed the potential of CNNs and triggered the rapid developments in computer vision and other deep learning topics of the last decade.

Adversarial properties. Szegedy et al. (2014) discovered that they can cause CNNs to misclassify by adding seemingly negligible noise designed to maximize the prediction loss for the true class. Due to their adversarial property, these perturbed images are called adversarial examples (AE). At the same time, they observed that these AEs can generalize to other networks and thus they can become transferable. If an adversarial example generalizes over several individual CNNs, it is a transferable adversarial example (TAE) and poses a particular threat. Goodfellow, Shlens, and Szegedy (2015) proposed with the fast gradient sign method (FGSM) a simple and effective method to generate AEs that are also likely to transfer and investigated why neural networks are so highly prone to adversarial examples. They identified their linearity with respect to the input as the underlying cause of the existence of AEs.

When we want to increasingly implement CNNs in our physical environment, for example for self-driving cars, authentication by facial recognition or the control of military drones, safety-critical aspects of deep learning systems are one of the most important challenges (Pereira & Thomas, 2020). The field of adversarial machine learning connects machine learning topics with IT security. In many cases, we are in an area with practical zero error margin. Nevertheless, it is often possible to significantly fool a CNN by adding faintly perceptible, yet contrived engineered perturbation to images (Goodfellow, Shlens, & Szegedy, 2015). Adversarial machine learning is about the generation of adversarial examples and the defense against it. A variety of different methods to attack and defend exist in the literature.

Research objective. This work empirically investigates the application of TAEs to CNNs in a black-box scenario. The selection of defenses and attacks is based on existing research. Drawing on different results, characteristics of of attacks and defenses can be derived. In particular, the property of gradient masking should be mentioned. Thematically, this work is situated in the broader context of machine learning at the interface to IT security. The underlying models and problems come from computer vision.

Besides presenting these intriguing properties of CNNs, the objective is to practically investigate the impact of AEs and thus the consequences for robust training of models. We investigate the training time, the validation accuracy, the effect of random noise, possible gradient masking effects and the general robustness. Defenses studied include adversarial training (Goodfellow, Shlens, & Szegedy, 2015), the Madry defense (Madry et al., 2018), superimposing (Seiler, Trautmann, & Kerschke, 2020) and defensive distillation (Papernot et al., 2015). The MI-FGSM (Dong et al., 2018) attack is used for generating TAEs.

Structure of thesis. This work is divided into six chapters. For orientation, the chapters cover the following aspects:

Chapter 1 introduces this work. It recapitulates the several accomplishments in the field under study and identifies the need for further research. Thus, this work and its objectives are presented in the larger context. Furthermore, formal aspects are addressed.

Chapter 2 and chapter 3 give the formal background to understand the objectives of the evaluation together with the underlying models and methods. The former chapter introduces image classification from the perspective of deep learning and how it can be successfully accomplished with CNNs. It builds on fundamental concepts such as the perceptron, ANN, and gradient descent. Chapter 3 introduces the field of

adversarial machine learning and covers the fundamental aspects, again from a deep learning perspective. The problem is introduced and different threat models, attacks and defenses are presented.

Chapter 4 presents the methodology used for the evaluation. It discusses the general setup, the concrete implementation and execution and highlights how the results can be interpreted.

Chapter 5 covers the obtained results of the evaluation in detail. At the same time, the results are discussed based on the current state of research. Possible characteristics of the defenses and attacks are derived.

Chapter 6 concludes this thesis and the results are summarized. Future research opportunities and the limitations of this work are discussed.

1.1 Notation

The used notation is based on that of Carlini (Athalye, Carlini, & Wagner, 2018; Carlini et al., 2019; Carlini & Wagner, 2016). It has been suitably adapted and extended. For the purpose of better readability, all elements are presented in more detail when used. The major elements are:

- (x, y) are data x and associated label y from elements of a dataset $(x, y) \in (X, Y)$.
- $d(x, x^{Adv})$ is the distance according to the ℓ_∞ norm between two elements x and x^{Adv}.
- $F(\theta; x)$ is a machine learning model with parameters θ.
- $C(\theta; x)$ is the predicted class of a model F.
- $L(F(\theta; x), y)$ is a loss function for a model F.

2 Deep Learning

This chapter is a brief introduction to convolutional neural networks and their application to image classification in computer vision. It provides the architecture that adversarial attacks and defenses target.

2.1 Introduction to Deep Learning

Deep learning. Deep Learning is a subfield of machine learning and part of the broader context of AI (Goodfellow, Bengio, & Courville, 2016). In contrast to other machine learning methods, such as logistic regression or support vector machines, deep learning methods are characterized by their distinctive depth. As an example, the EfficientNetV2 CNN in the B3 form, a rather small and efficient architecture, has in total about 12 million parameters (Tan & Le, 2021). Utilizing this depth, complex relationships can be learned as a combination of the variety of small structures (Goodfellow, Bengio, & Courville, 2016). With this flexibility and abstraction ability, deep learning methods are particularly applicable to the often complicated real domain of computer vision. A model may identify a human on a picture based on his face comprising eyes, nose, and mouth, which again are made up of corners and contours.

Representation learning and predictive modeling. With deep learning the costly and time-consuming manual extraction of features is eliminated. In contrast, the deep learning model learns the representation intrinsic to the data. Goodfellow, Bengio, and Courville, 2016 therefore calls this type of learning representation learning. With the learned representation, predictions can then be made for new data, also referred to as algorithmic or predictive modeling (Breiman, 2001). Classification is one of these predictive tasks and the focus of this thesis.

2.2 Image Classification

According to Murphy, 2012, independently of the specific characteristics of a problem, the approach in deep learning is the same from an abstract perspective.

Regression and classification. A model learns a mapping to output $y \in Y$ based on input data $x \in X$. Depending on the nature of the application, the model performs either regression or classification. Regression assumes that the variable to be predicted is real-valued with $y \in \mathbb{R}$. In classification, on the other hand, y is categor-

ical and only takes values from a finite set $y \in \{1, .., K\}$. In this context, y is also called the class label, since it labels the observed data x.

$F(\theta; x)$ denotes a general machine learning model with parameters θ and input x, where $F(\theta; x)_i$ is the i-th element of the output vector. The classification of such a model with parameters θ is then the class with the greatest magnitude (Athalye, Carlini, & Wagner, 2018)

$$C(\theta; x) = \arg\max_i F(\theta; x)_i.$$

Image representation. The input can take a variety of different types. In the context of this work, we are interested in the processing of images. Color images can be represented by computers as a three-dimensional pixel array with the dimensions $width \times height \times channel$ where the channels represent the RGB color spectrum (Elgendy, 2020). For each of the color channels, a pixel takes an 8 bit intensity value between $[0, 255]$. In this form, images can be efficiently processed by computer and algorithms.

Image classification. In object or image classification, an image x is to be categorized into a class y based on its content. Depending on the classification task, possible instances could be $\{day, night\}$, $\{dog, cat\}$ or $\{America, Antarctica, Europe, Asia, Australia\}$. Image classification is one of the key tasks in computer vision. The quality of the classification can be measured by different metrics. One of the simplest and most descriptive is the top-1 accuracy, which is the proportion of elements that are correctly classified in the evaluated dataset (Goodfellow, Bengio, & Courville, 2016).

Efforts to classify images using deep learning methods date back to the late 1980s with LeNet (LeCun et al., 1989), but the lack of computing power at that time limited efficient training of large-scale networks. Momentum in the development of image classification was brought by AlexNet in 2012 (Krizhevsky, Sutskever, & Hinton, 2012), which outscored existing models by absolute 10.8% accuracy in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

The ILSVRC challenge. The ILSVRC highlights developments in the classification of large image datasets using deep learning methods (Russakovsky et al., 2014). In this benchmark, the classification accuracy of different approaches and architectures has been compared since 2010. The evaluation is performed on a dataset with a total of over 1 million images from 1,000 possible classes. Well adapted models like EfficientNetV2B3 achieve a top-1 accuracy of 81.5 % for this complex task (Tan &

Le, 2021). The evaluation of this thesis is performed on the smaller Tiny ImageNet subset with 100,000 images downsized to 64x64x3, each belonging to one of 200 possible classes (Le & Yang, 2015).

2.3 Perceptron

The perceptron is a simple machine learning model inspired by the human neuron (Rosenblatt, 1958, 1961). The model iteratively learns a binary classification $y \in \{-1, +1\}$ based on the model parameters θ and the input x (Murphy, 2012):

$$F(\theta; x) = sign(\theta^T x).$$

Since the latter is considered fixed, the objective of the training is to optimize the parameters θ so that the accuracy is maximized. This is accomplished by using the perceptron algorithm, which keeps the parameters unchanged if the single classification is correct and updates them using the learning rate η in iteration i if the classification is incorrect:

$$\theta_i = \begin{cases} \theta_{i-1} + \eta y_i x_i & F(\theta; x_i) \neq y_i \\ \theta_{i-1} & \text{else} \end{cases}.$$

The main limitation of the perceptron model is that it only learns a linear decision boundary and it is only capable of performing a binary classification (Murphy, 2012). This makes it insufficient for more complex problems like image classification with non-linear relationships and where the number of classes often exceeds two by many magnitudes.

2.4 Artificial Neuronal Network

Building on the shortcomings of the perceptron, two main requirements arise for models used to classify images:

1. the models must be able to capture non-linear relationships
2. the models must be able to distinguish between more than two classes

Architecture of ANNs. The deep learning approach of ANNs can efficiently address these challenges. According to Goodfellow, Bengio, and Courville, 2016, they are typically organized in consecutive layers of artificial neurons. Each of these neurons is structurally very similar to the perceptron. The input layer takes x as input

and the last layer returns the prediction of the model $F(\theta; x)$. All layers between input and output are referred to as hidden layers. In the most basic deep feedforward network or multilayer perceptron, all layers are connected only to the immediate, if existing, predecessor and successor layers (Goodfellow, Bengio, & Courville, 2016). In addition, the input is only propagated in one direction when a prediction is made. Usually all neurons of connected layers are also connected pairwise, which is a densely connected or dense layer.

$$F = softmax(z^n) = F^n \circ F^{n-1} \circ ... \circ F^1$$

Dense layer. In a dense layer n, the following non-linear transformation from the output of the previous layer $n-1$ is performed:

$$F^n = \sigma(z^n) = \sigma(\theta^n F^{n-1}) = \sigma(\theta^n_k F^{n-1}_k + \theta^n_{k-1} F^{n-1}_{k-1} + ... + \theta^n_1 F^{n-1}_1 + \theta^n_0).$$

Here the subscript stands for the k-th element of the output from a layer. In the first layer, let $x = F^1$ hold and θ^n is the parameter matrix to be trained for the respective layer, where θ^n_0 is referred to as the bias. $\sigma(\cdot)$ is a non-linear activation function to introduce non-linearity to the model. Otherwise, the model would be just an affine transformation of the input (Goodfellow, Bengio, & Courville, 2016). Common choices are the hyperbolic tangent, sigmoid, softmax, rectified linear unit (ReLU) and exponential linear unit (ELU). The frequently used ReLU activation function is defined as:

$$ReLU(z) = max(0, z).$$

In deep learning the pre-activation values z^n of an n-layer network is often referred to as the logits. The softmax activation function normalizes the logits to a probability distribution and is used in ANNs as the final activation function for classification (Bishop, 2007). By the definition, the classification of an image x is the class with the highest probability:

$$softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}.$$

In contrast to the perceptron, ANNs now fulfill the desired properties:

1. by the universal approximation theorem, ANNs with at least one hidden layer, a sufficient number of neurons and linear output can approximate any continuous function, linear and non-linear (Hornik, Stinchcombe, & White, 1989).

2. it is possible to distinguish between any number of classes by setting the number of neurons in the last layer to the number of classes to be classified.

Shortcomings of ANNs. However, ANNs have further shortcomings that make an application in image classification impractical. A problem is the processing of the input. An image with its three dimensions is processed in by an ANN as a flattened vector. A 64x64x3 image produces a vector with 12288 elements, which for a first dense layer with 128 neurons already leads to about $12288 \cdot 128$ parameters and therefore a very high model complexity (LeCun et al., 1999). Furthermore, this makes them not invariant to transformations of the input, e.g. scaling, translation or geometric distortion (LeCun et al., 1999). Another problem is that the flattened vector coding leads to the complete loss of spatial information of the image (LeCun et al., 1999). The information in the two-dimensional space of highly correlated adjacent pixels is thus ignored.

2.5 Convolutional Neuronal Network

CNNs are a special form of ANNs and the preferred technology for image classification and related computer vision tasks. By extending the ANN architecture with two further types of layers, namely convolution and pooling layers, they can solve the deficiencies of ANNs in spatial data processing(LeCun et al., 1999). In doing so, these two layers enable three different concepts: local receptive fields, shared weights, and sub-sampling (LeCun et al., 1999).

Architecture of CNNs. CNNs consist in their basic architecture of two parts (Figure 1). A more comprehensive insight gives LeCun et al., 1999. First, feature extraction takes place in the convolutional and pooling layers, alternating between them. This multiple non-linear combinations of the input to feature maps are then the input for one or more dense layers, which learn the actual classification task based on the available features.

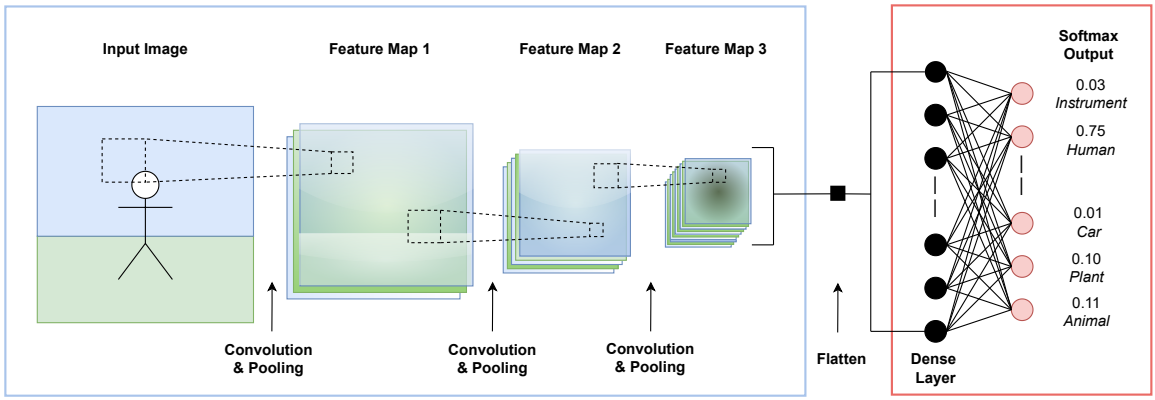

Figure 1 Architecture of a CNN with the two parts for feature extraction and classification.

Convolution. Individual pixels of an image are not very representative for the entire object, but this changes when they are considered jointly with the neighboring pixels (shared-weights). This is achieved in the context of image processing by having neighboring pixels share weights in the model. The operation, which is performed on two real-valued functions, is to be considered here in its discrete form (Elgendy, 2020). A two dimensional, but smaller kernel K with parameters (local receptive fields) convolves over the input I and yields a two-dimensional feature map S. The individual elements of the map can be considered as the by the kernel weighted sum of the input part:

$$S(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n).$$

The values of the kernel K, which in contrast to the input is not fixed, determines the output features map and like all other parameters θ has to be learned in training. Subsequently, just as with the dense layers, a non-linear activation function is applied to the pre-activation feature map (Goodfellow, Bengio, & Courville, 2016).

Following the representation learning approach, convolution enables the learning of complex representations.

Pooling. Pooling (sub-samping) is an operation that subscales the feature map by creating locale summaries and takes place after one or multiple convolutions (Goodfellow, Bengio, & Courville, 2016). Thus, the learned representation not only becomes more invariant to smaller translations but also requires fewer parameters.

There are different ways to implement pooling. Particularly popular is max pooling developed by Zhou and Chellappa, 1988, which only takes the largest value from each patch of the feature map.

2.6 Training Neuronal Networks

Previously, ANNs and CNNs were only considered structurally, however, the parameters θ are initially not optimal and must be trained similarly to the perceptron. One way to initialize the parameters is randomly (Goodfellow, Bengio, & Courville, 2016).

Loss function. The objective of training is to maximize the prediction accuracy for the images. Similarly, to train a network, the loss function L, i.e. the deviation from the desired output, can be minimized for a training dataset (Murphy, 2012). The predominately used loss function for classification problems is the (categorical) cross-entropy:

$$L_{CE}(F(\theta; x), y) = -\sum_{k=1}^{K} \mathbb{1}(y = k) log(F(\theta; x)_k).$$

Gradient descent. To minimize the loss, the method of gradient descent is applied (Bishop, 2007). Iteratively the loss is estimated, the gradient of the loss function is calculated, the parameters are updated and the loss is re-evaluated. For large datasets it is not efficient to consider all elements in a single iteration, so in practice the gradient is estimated based on mini-batches or single points (stochastic gradient descent) (Murphy, 2012):

$$\theta_{i+1} = \theta_i - \eta \nabla L_{CE}(F(\theta_i; x_i), y_i).$$

Backpropagation. The direct analytical calculation of the gradient for a loss function L is complex and costly (Goodfellow, Bengio, & Courville, 2016). The backpropagation algorithm from Rumelhart, Hinton, and Williams, 1986 provides an efficient step-by-step method to obtain the gradient and thus enables the minimization of the loss by means of gradient descent. More details can be found in chapter 6.5 of Goodfellow, Bengio, and Courville, 2016. The algorithm consists of a forward propagation and a backward propagation. In the first, the loss is determined for a subset $x \in X$, $L_{CE}(F(\theta_i; x_i), y_i)$. Then, in backward propagation, the part of the gradient required in a layer for gradient descent is determined and passed backward by applying the chain rule.

3 Adversarial Machine Learning

Adversarial machine learning, a subfield of Machine Learning, deals with the properties of models facing adversarial examples. Models can be attacked with adversarial attacks, which necessitates the existence of sophisticated defenses. Based on the kind of defense, different types of threats can be identified and scenarios characterized. In the context of this thesis, we will focus on the methods that can be applied to deep learning models in the form of CNNs.

3.1 Adversarial Example

An AE is an image perturbed with noise x^{Adv}. It is generated from an image x that is originally correctly classified by a CNN, whereas the AE itself is classified wrong (Goodfellow, Shlens, & Szegedy, 2015):

$$C^*(\theta; x) \neq C(\theta; x^{Adv}).$$

C^* denotes the true classification for an image. The defining property of AEs is that the distance $d(x^{Adv}, x)$ is small and the image is perceived by the human as hardly distinguishable from the original input (Figure 2) (Goodfellow, Shlens, & Szegedy, 2015). Often a small perturbation is enough to reduce the accuracy from almost 100% to 0% for a dataset like MNIST (Carlini & Wagner, 2017). To measure the distance d of two images, different norms can be used. Even if the real-world appropriateness is to be questioned, primarily the ℓ_p norms are utilized for distance measuring (Carlini et al., 2019):

$$d(x^{Adv}, x) = ||x^{Adv} - x||_p$$

Creating AEs. AEs can be created by performing a simple linear transformation of the input image:

$$x^{Adv} = x + \epsilon \cdot \eta.$$

The objective of applying adversarial perturbation is to maximize the loss for the true class prediction while always staying close to the original input. Thus, the perturbation η can be generated using a wide variety of methods, the attacks, with the parameter ϵ controlling the magnitude of distortion. In its structure, the perturba-

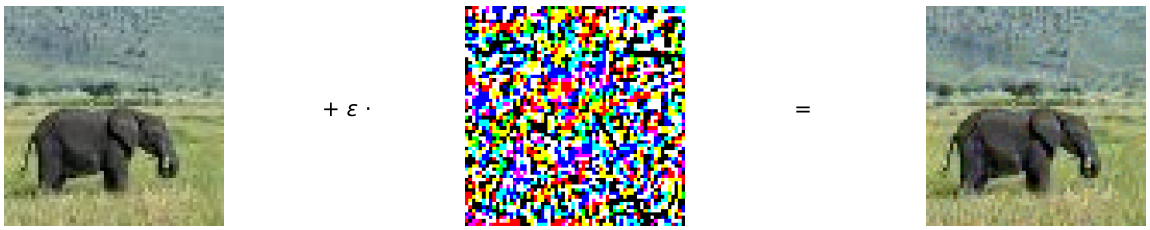

Figure 2 Creation of a TAE with MI-FGSM ($\epsilon = 8$). From left to right: original image, total perturbation and TAE. The first is correctly classified with 99.91% confidence as african elephant. However, the adversarial right image is with 99.99% confidence classified as a king penguin. Figure inspired by Goodfellow, Shlens, and Szegedy, 2015.

tion is not randomly following some distribution, like Gaussian noise, but explicitly exploits the properties of the model to be attacked (Szegedy et al., 2014). Depending on the scenario, the attacker can make use of existing knowledge about the model. In addition to this single-step AE, there are numerous methods that iterative add smaller amounts of perturbation (Madry et al., 2018). These multi-step AEs are generally stronger and result in better attacks in many scenarios.

CNN resilience to AE. Though CNNs are to some extent resistant to random Gaussian noise (Szegedy et al., 2014), they can be significantly fooled by AEs (Goodfellow, Shlens, & Szegedy, 2015; Szegedy et al., 2014). This makes AE a threat especially for the application of computer vision methods in safety-critical systems. In experiments, it has already been shown that CNNs in real domains, such as those used in self-driving cars, can be fooled with AEs so that a physical stop sign is no longer recognized as such (Eykholt et al., 2018).

3.2 Threat Model

If not otherwise referenced, this section is based on Carlini et al., 2019.

The threat model specifies the conditions a defense claims to be resilient against. It parameterizes the attack and defense scenario. Both should be evaluated primary under the defined circumstances. The following depict the key elements of the different threat models.

Targeted and non-targeted attacks. This aspect defines the general objective of the attack, where in the basic non-targeted scenario an arbitrary misclassification is an AE. In the targeted scenario, an adversarial example is only considered as one if a specific misclassification $y^{target} \in Y$ is achieved. Since the set of target AEs is a subset of the non-target examples, it is generally more difficult to perform a targeted attack.

White-box and black-box attacks. The constraint is focused on the attackers knowledge of the attacked model. In the white-box scenario, the attacker has access to all information, such as the exact architecture, training data, and parameters. The attacker can access the gradient of the loss function, but cannot modify the target model. The black-box scenario is more limited and assumes that the attacker does not have access to this information. He can only access the predicted probabilities or classes in a restricted way, often with a limited number of available queries.

ℓ_p-norm constraints. Frequently, when performing adversarial attacks, the attacker is forced to keep the perturbation small (according to some distance measure), so that a human-perceivable similarity is preserved. Therefore, attacks and defenses are often constrained as proxy by one of the standard in machine learning used norms. The ℓ_{inf} norm or max norm is used to measure the maximum difference between the pixels of two images (Goodfellow, Bengio, & Courville, 2016). The ℓ_1 and ℓ_2 norm measure the average absolute and euclidean distance of two images (Goodfellow, Bengio, & Courville, 2016). In addition, there is the distance measure denoted as ℓ_0, which formally does not satisfy the requirement of a norm. This measure gives the number of non-zero elements, which is the number of different pixels in two pictures (Goodfellow, Bengio, & Courville, 2016). The latter is relevant for attacks that aim to change as few pixels as possible.

3.3 Transferable Adversarial Example

Already Szegedy et al., 2014 noticed that AEs, generated for a special model, may also generalize and transfer to other CNNs. Such models are trained for a similar task. These special types of AEs are called transferable adversarial examples (TAEs) and are especially interesting for black-box scenarios without access to the gradient. Instead of generating the AEs directly on the target model, they are generated on one (Papernot, McDaniel, & Goodfellow, 2016) or an ensemble (Liu et al., 2017) of similar substitutes and then subsequently applied to the target model.

For TAEs, the thread model can be further extended (Papernot, McDaniel, & Goodfellow, 2016). First, whether the AEs transfer between two instances of one architecture, e.g. EfficientNetV2 to EfficientNetV2, or also between two instances of different architectures, e.g. EfficientNetV2 to InceptionV3. In addition, it is distinguished whether training was performed on the same dataset or perhaps on different subsets or with different initialization.

Interpretation of TAEs. The presence of TAEs must be explainable by a similarity in the decision space of the different classifiers. Images are very high dimensional and if the perturbation leads to a misclassification, then x^{Adv} crosses the decision boundary at some point. For a transfer to occur, this must hold for both the substitute and target model. The models are thus similar in their decision boundary (Papernot et al., 2017) and the better a substitute can mimic this, the more likely is a transfer.

TAEs for evaluation. The property of TAEs to evade a target model without access to the parameters is also of particular valuable for the evaluation of defenses (Carlini et al., 2019). Many defenses are consciously or unconsciously based on the aspect of gradient masking and are therefore resilient to AEs, but not to TAEs (Papernot et al., 2017). If an adversarial dataset trained on a substitute with similar properties successfully transfers to a target model with defense as well, protection by the defense can be rejected (Carlini et al., 2019). More on this in section 3.6.

3.4 Adversarial Attacks

There are a variety of ways to generate AEs. The gradient-based methods are particularly well studied and offer strong attack capabilities for both white-box and black-box scenarios (Dong et al., 2018). In this context, strong means that the dataset of AEs fools the target network as often and with as much confidence as possible under limited perturbation.

FGSM. The fast gradient sign method (FGSM) of Goodfellow, Shlens, and Szegedy, 2015 offers a fast possibility to find AEs with an additional backwards propagation. It is a single-step attack and, like all other introduced attacks, requires access to the target or a substitute model. The method maximizes the loss, usually the cross-entropy, for the original class by taking the gradient with respect to the input:

$$x^{Adv} = x + \epsilon \cdot sign(\nabla_x L(F(\theta; x), y)).$$

The generated AEs are bounded under the ℓ_{inf} norm by

$$||x^{Adv} - x||_{\text{inf}} \leq \epsilon.$$

Single-steps AEs have the property that they transfer with a higher probability than multi-step attacks, but their success in white-box scenarios is more limited (Dong et al., 2018). To create stronger TAEs, the following attacks also introduce iteration and momentum.

BIM and PGD. The basic iterative method (BIM) (Kurakin, Goodfellow, & Bengio, 2017) and projected gradient descent attack (PGD) (Madry et al., 2018) are two very similar and strong iterative versions of FGSM. Instead of applying the gradient to the input image once, this is repeated in multiple forward and backward passes. Thus, the attack likely overfits to the model (Seiler, Trautmann, & Kerschke, 2020), but also leads to stronger AEs. Iterative attacks are more successful in white-box scenarios, but due to their overfitting property, they transfer worse than single-step attacks in black-box scenarios:

$$x_t^{Adv} = x_{t-1}^{Adv} + \alpha \cdot sign(\nabla_x L(F(\theta; x^{Adv}), y)),$$
$$x_0^{Adv} = x.$$

PGD is essentially BIM, except that the attack is started multiple times from random points within the ℓ_{inf} environment of the image to further explore the loss (Madry et al., 2018).

One way to limit perturbation in both attacks is to set $\alpha = \frac{\epsilon}{number\,iterations}$ or use pixel-wise clipping (Kurakin, Goodfellow, & Bengio, 2017).

MI-FGSM. The momentum iterative fast gradient sign method (MI-FGSM) (Dong et al., 2018) additionally accumulates the gradient over several iterations and thus stabilizes the optimization. This can prevent the property of local optimization of BIM and PGD. To further generalize, the attack can additionally be performed on an ensemble. MI-FGSM was the most successful attack in the 2017 NIPS Adversarial Attack challenge and the generated examples transfer with high probability. The AEs are generated according to

$$g_t = \mu \cdot g_{t-1} + \frac{\nabla_x L(F(\theta; x_t^{Adv}), y)}{||\nabla_x L(F(\theta; x_t^{Adv}), y)||_1},$$
$$x_t^{Adv} = x_{t-1}^{Adv} + \alpha \cdot sign(g_t),$$
$$x_0^{Adv} = x, g_0 = 0.$$

The choice of the suitable decay factor μ must be experimented for the application. The authors use for their evaluation the factor 1.0, which implies that the gradients of all previous iterations are summed up.

3.5 Adversarial Defenses

Possible approaches to make models more robust against AEs are to modify the training or to augment the data. The following defenses are related to these approaches.

Adversarial Training. Goodfellow, Shlens, and Szegedy, 2015, besides introducing the FGSM, also proposed a way to increase a model's robustness to AEs through data augmentation. Inspired by Szegedy et al., 2014 they trained their model on a mixture of original images and AEs. From this method stem many similar defenses. Adversarial training can be implemented by an additional regularization term in the loss function:

$$\tilde{L}(F(\theta; x), y) = \alpha \cdot L(F(\theta; x), y) + (1 - \alpha) \cdot L(F(\theta; x^{Adv}), y)$$

Here the factor $\alpha \in [0, 1]$ determines the strength of the regularization and x^{Adv} is generated with the FGSM.

Madry defense. Madry et al., 2018 discovered that as the strength of the AEs on which they trained their network increased, so did the model's resistance to AEs. In their defense, they suggest training with stronger multi-step AEs instead of with limited single-step AEs. Instead of generating x^{Adv} with the FGSM, methods such as PGD can be used.

Superimposing. Seiler, Trautmann, and Kerschke, 2020 proposed a form of superimposing as a defense, which does not require the calculation of the gradient. The method is based on placing a randomly chosen image x^r from the dataset over the training image x. Thus, the training images are closer to the decision boundary, harder to distinguish and regularize the boundary. Furthermore, this method can augment the training data:

$$\tilde{x} = (1 - \alpha) \cdot x + \alpha \cdot x^r.$$

In addition, they extend the loss function with the Kullback-Leibler divergence, which is expected to further locally smooths the decision boundary. Thereby $\hat{y}$ is the prediction of the model due to x and $\tilde{y}$ is based on $\tilde{x}$:

$$\tilde{L}(F(\theta; x), y) = L(F(\theta; \tilde{x}), y) + \lambda \cdot D_{KL}(\hat{y} || \tilde{y}).$$

Defensive Distillation. Defensive distillation was proposed as a defense by Papernot et al., 2015. It consists of three steps (Carlini & Wagner, 2016):

1. Train the teacher CNN $F_T(\theta; x) = softmax(z(\theta; x)/\tau)$ with standard methods at a training temperature τ. It has the same structure as the following distilled.

2. Evaluate the complete training dataset on F_T. The output $F_T(\theta; x)$ is called the soft labels and contains additional information for the subsequent training step.

3. Train an identical student CNN $F_S(\theta; x) = softmax(z(\theta; x)/\tau)$ using the soft labels, again at the same temperature. The distilled CNN should be regularized by the soft labels during training.

The temperature τ affects the softmax activation and thus the confidence of the prediction. For $\tau \to \text{inf}$, the prediction approximates the uniform distribution. To make predictions with the model subsequently $\tau = 1$ is set.

3.6 Gradient Masking

Gradient masking was discovered by Papernot, McDaniel, and Goodfellow, 2016. It causes a gradient to be unusable for creating AEs, since it points in the wrong direction or have other unpredictable behavior. Defenses that, often unconsciously, lead to gradient masking appear to be robust in white-box scenarios against strong iterative attacks (Athalye, Carlini, & Wagner, 2018). However, the models are still sensitive when the AEs are generated on a substitute with similar decision boundary (Figure 3).

One form of gradient masking is the obfuscated gradient (Athalye, Carlini, & Wagner, 2018). Possible manifestations of obfuscated gradients are shattered, stochastic and exploding or vanishing gradient. In a case study, 7 out of 9 ICLR 2018 defenses could be reduced to the property of obfuscated gradient and broken by modified attacks. Thus they lead to gradient masking. The Madry defense was one of the two defenses in the study that does not or only slightly seem to have this property.

Discovering gradient masking. To test defenses on gradient masking or obfuscated gradient, TAEs are especially helpful (Carlini et al., 2019; Papernot, McDaniel, & Goodfellow, 2016). Since the black-box scenario is a subset of the white-box scenario, these attacks should generally perform worse than white-box attacks. However, if the defense is robust against white-box attacks, but fails against transfer attacks generated from a substitute, this is a good indicator that some form of gradient

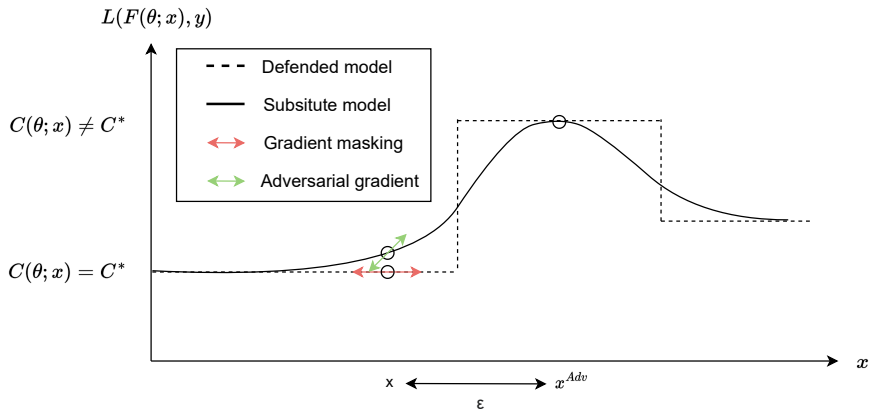

Figure 3 Visualization of the effect of gradient masking. The defended model breaks gradient-based attack and following the gradient does not increase the loss. For a substitute, an adversarial direction can be found which is globally identical to the defended model. Graphic based on the one of Papernot et al., 2016.

masking is present. Substitute and target then share approximately the same decision boundary, only that the first is more accessible.

3.7 Current Trends

The discoveries of Szegedy et al., 2014 demonstrated the intriguing properties of neural networks and established the field of adversarial machine learning. Since then, there has been an 'arms race' in this field. Application of CNNs to many real-world problems requires the models to be robust against AEs. A CNN that detects stop signs for a self-driving car should achieve close to 100% accuracy, since any error could be a potential accident. An attack that reduces the accuracy by even by 1% represents an unjustifiable risk in these cases, since it increases the potential number of accidents by a multiple. However, many of the attacks reach nearly 100% success (Carlini & Wagner, 2017). This trend is also reflected in the number of papers published (Figure 4).

Arms race. The existence of AEs quickly made suitable defenses necessary. Following this, the defenses were broken by stronger iterative attacks. The discovery of stronger attacks also necessitated the need for new defenses. This spiral has characterized adversarial machine learning since then. There are a large number of defenses and attacks in the literature. In order to call a new proposed defense safe, everything must first be tried to reject it. Carlini et al., 2019 provides a suitable starting point for this endeavor.

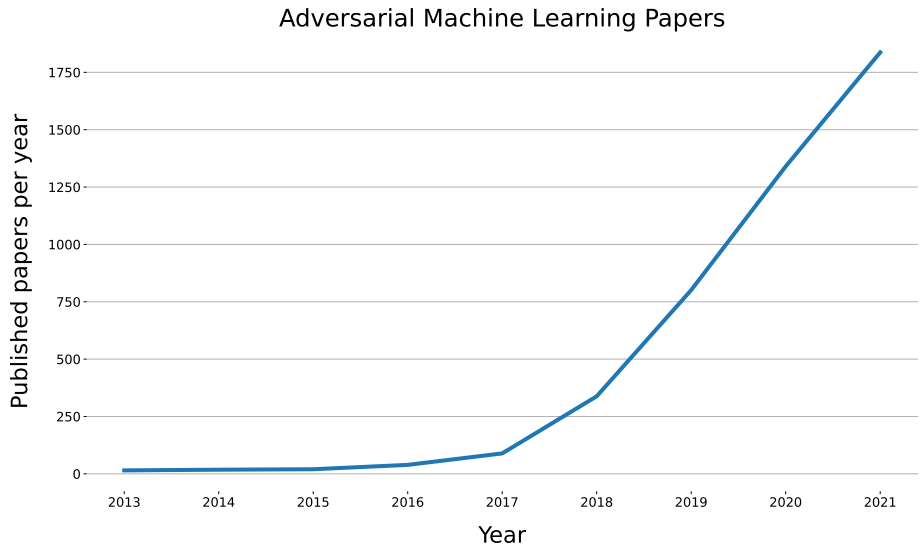

Figure 4 Number of papers published on Scopus with context adversarial example, adversarial attack or adversarial defense.

Motivation. There are three main motivations for desiring to develop effective defenses to AEs (Carlini et al., 2019):

1. As already discussed, one goal of the defense is to make CNNs and other machine learning models more secure against attacks, so that the risk for real scenarios is reduced as much as possible.

2. AEs pose a great challenge to models and thus allow for test them in scenarios that cannot be reproduced by repeated randomness. Therefore, they offer a possibility to test the worst-case robustness of a model.

3. Understanding why models fail in the face of AEs helps us further understand the gap between human cognition and machine learning. AEs can fool machine learning models yet are not even detected by human observers, understanding what causes this disparity could lead to better models in the future.

4 Evaluation of Adversarial Defenses

In the practical part of this thesis, several promising defense mechanisms are investigated with respect to their resilience against TAEs. Unlike in cryptography, where there are methods that are assumed to be truly secure, this is not the case in the area of adversarial machine learning. Thus, existing defenses need to be investigated and their robustness empirically validated. This chapter discusses the methodological approach of the evaluation and includes the overall setup, implementation and execution, lastly possible interpretations of the obtained data.

The design of this evaluation is inspired in particular by Carlini et al., 2019, which covers equally specific and general guidance, common flaws and pitfalls.

4.1 Evaluation Setup

The evaluation setup includes three phases: training the different models, generating adversarial datasets and finally evaluating the datasets on the trained models (Figure 5).

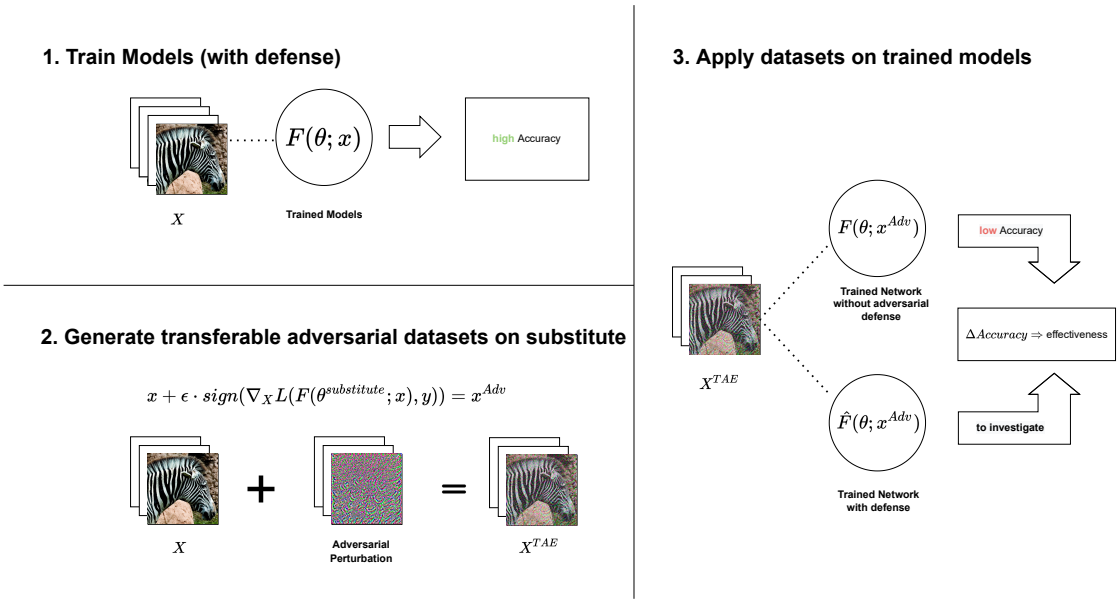

Figure 5 A simplified perspective on all three phases of the evaluation. The FSGM represents any arbitrary adversarial attack that has high transferability.

Defense Models. In the first phase, the models are trained based on different defenses. All the presented defenses adversarial training on single-step AEs, adversarial training on PGD AEs (Madry defense), superimposing and defensive distillation are investigated. The hyperparametrization of the different instances of a defense can be derived from Table 1. For all defenses, the proposed hyperparametrization of the paper was initially tested and, if beneficial, optimized. For the Madry defense, besides the amount of perturbation, the number of iterations are particularly relevant, so both are assessed in different configurations. For the superimposing defense, the protection solely provided by training on superimposed images and independently of the regularization by the Kullback-Leibler divergence is tested as well.

Defense	
Model	Parametrization
Base Model	None
Adversarial Training	$\alpha = 0.5, \epsilon = 16$
	$\alpha = 0.5, \epsilon \sim U(0, 32)$
Madry Defense	$\alpha = 0.5, \epsilon = 8, \epsilon$ iteration $= 0.8$, number iterations $= 10$
	$\alpha = 0.5, \epsilon = 16, \epsilon$ iteration $= 0.8$, number iterations $= 20$
Superimposing	$\alpha \sim B(0, 10), \lambda = 0$
	$\alpha \sim B(0, 10), \lambda = 10$
Defensive Distillation	$\tau = 50$, one epoch of pre-training

Table 1 Hyperparametrization of the evaluated defense models.

CNN architecture. A modern CNN architecture is the backbone for the evaluation. There is a large number of possible architectures, such as Inception, VGG, ResNet and DenseNet. Each of these comes with slightly different qualities. Ideally, a good defense should work equally well for all types of models. The evaluation aims to be as independent of the specifics associated with individual architectures as possible. However, for this evaluation, a single concrete architecture has to be chosen. The considered criteria for the selection of an architecture are:

- a high accuracy for established evaluation datasets
- strong popularity and acceptance
- an efficient design
- no too strong of a deviations from the general CNN model

EfficientNetV2 is state of the art and provides these desired properties. It not only has a fraction of the parameters and training time of other models, but achieves top-level accuracy for the ImageNet dataset (Tan & Le, 2021). Furthermore, it comes in different sizes, while for evaluation B3 is selected.

Training. All models are trained on a Quadro RTX 6000 GPU, which is also used to generate the adversarial datasets. The used optimizer is stochastic gradient descent (SGD). It achieves very good results in the configuration of Carlini[1]. Up to 100 epochs are trained, with accuracy and loss determined for the validation set after each epoch. If the validation loss does not improve significantly over two epochs, the training ends early (early stopping).

Datasets. The Tiny ImageNet dataset[2] is used for the evaluation. This contains 100,000 images, 500 for each of the 200 classes. The training dataset is split into a training set of 80,000 images and a validation set of 20,000 images. In a first layer of each model, the images are additionally scaled up from 64x64x3 to 128x128x3 to meet the requirements of EfficientNetV2. All models are trained and evaluated on the same datasets. However, all defenses modify or augment them in the training process.

Subsequently to the training, adversarial datasets are generated on a substitute structurally identical to the target without defense. The TAEs are created from the separate test set of 10,000 images, which are images that were not trained on. This testset is also used to estimate the validation accuracy in the evaluation. Only images correctly classified by the substitute are selected for creating the perturbed image since no AE is possible for a clean and naturally misclassified image. The size of each evaluation dataset is fixed to 2,500 images. This enables an efficient generation and evaluation while still allowing a differentiated observation.

TAEs. The major method used to generate strong TAEs is MI-FGSM, which at a perturbation of $\epsilon = 8$ reduces the accuracy for the substitute from 100% to about 1% and for the base model without defense from 87.4% to about 6.8%. The ℓ_{inf} perturbation is changed in discrete steps to create different datasets. The decay factor for MI-FGSM is set to 1.0 and $\alpha = \frac{\epsilon}{number\,iterations}$. For all attacks, pixel-wise clipping is used to ensure that the perturbation does not result in pixels outside the displayable range of [0,255]. An overview of created datasets can be seen in Table 2.

[1] https://github.com/carlini/nn_robust_attacks
[2] http://cs231n.stanford.edu/tiny-imagenet-200.zip

Evaluation Datasets			
Type	Epsilon	Number iterations	Size
Validation Dataset	0	0	10,000
Clean Dataset	0	0	2,500
Random Noise	8	1	2,500
MI-FGSM	[4,8,16,32]	20	2,500

Table 2 All used or created dataset in the evaluation. Epsilon refers to the ℓ_{inf} bound of the perturbation, number iteration to the number of steps performed, for single-step AEs correspondingly one. Size is number of images in the dataset.

Application. With increasing perturbation the loss increases and misclassification becomes more likely (Carlini et al., 2019). A property of effective defenses is that they are robust and undercut accuracy thresholds only as perturbation increases to a significant level. The more perturbation is required to cause misclassification, the more distant x^{Adv} is from the original input and the more likely it is to be identified as adversarial by the human. This is the desired property to be observed in a good defense.

Threat model. The examined scenario is to be categorized in the black-box domain. It does not provide access to the parameters and gradient of the different trained target models. However, the substitute is trained on exactly the same data and is structurally identical to the target model without defense. This is very unlikely in a realistic black-box scenario, but permissible for the evaluation since the inherent properties of the defenses should be investigated in the first place. To test the gradient masking property, we also use white-box attacks

For the evaluation, the more general setting of non-targeted attacks is considered. This means that any misclassification leads to an AE. Again, the primary objective of the evaluation is to examine the defenses for their general resilience, not a too narrow setting. A restriction to targeted TAEs would weaken the attacks and make it difficult to capture minor variations in the resilience.

For all attacks and, if required, also for defense, the ℓ_{inf} norm is used to limit the perturbation of the AEs. On the one hand, this results in an intuitive interpretation by limiting the maximum change of each pixel of the image by ϵ. At the same time, it fits well into the context of existing evaluations, which often also examine the ℓ_{inf} norm (Carlini et al., 2019).

4.2 Implementation and Execution

The evaluation is implemented in Python in a modular design. Attacks and defenses are implemented and integrated as individual classes and can thus be flexibly exchanged and the evaluation modified. All code is available at Github[3], which allows reproducible research. In addition, the pre-trained models and datasets are made public as well.

The dataset is initially downloaded from the source, processed as already described, and then saved as a Tensorflow dataset. EfficientNetV2 can be downloaded via Tensorflow, where the pre-trained parameters are not used, but randomly initialized by Tensorflow.

In order to implement and train CNNs efficiently, the open-source machine learning library Tensorflow[4] is used. Besides its high performance, Tensorflow also offers the integration of Keras, a high-level neural network API.

When implementing the defenses, the method proposed by the authors is followed as closely as possible. If there is a reference implementation of the defenses, possibly by the authors themselves, this is used and linked in the code. This is the case for defensive distillation (Carlini & Wagner, 2016).

The Cleverhans[5] library is utilized for the generation of AEs. Cleverhans was co-developed by some of the leading researchers in this field and provides standardized implementations for different adversarial attacks. This is a desirable property for evaluations, especially with comparability in mind.

4.3 Interpretation

The black-box and white-box robustness of adversarial defenses can be evaluated according to the following scheme:

Figure 6 illustrates possible ways of interpretation of the evaluation on an adversarial dataset when comparing the accuracy of a model with defense and the base model without. If the TAEs do not succeed in deceiving the base model, the attack is too weak or transfers poorly. Another attack must be tested. In the unlikely case that the accuracy of the defense model is low and that of the base model is high, then it is probable the defense may even be harmful. If the dataset fools the base model successfully and the defense model significantly less, then the protection by the

[3] https://github.com/stach/adversarial-evaluation
[4] https://github.com/tensorflow/tensorflow
[5] https://github.com/cleverhans-lab/cleverhans

defense cannot be rejected in this scenario. In the last case, the defense is similarly fooled as the model without defense. The defense then offers no additional value in this scenario and should not be used.

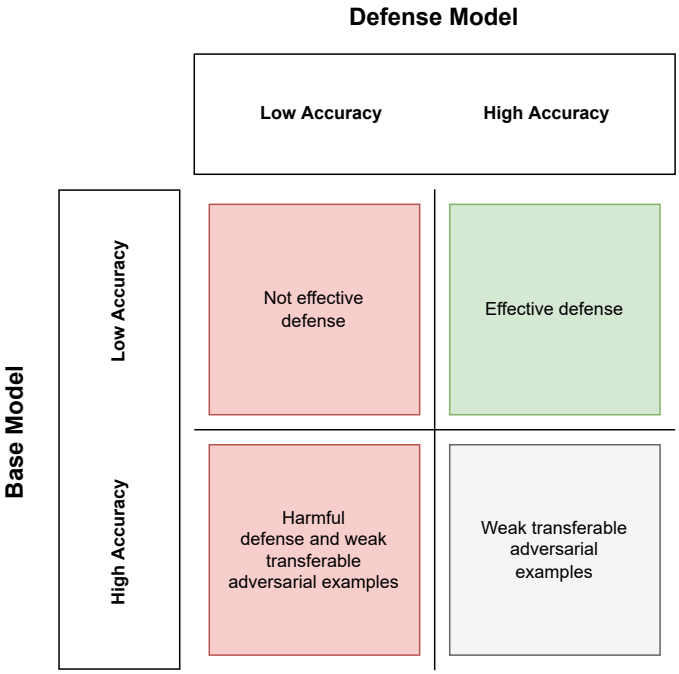

Figure 6 Four possible outcomes when comparing the evaluation results of a defense model with the undefended model.

Similarly, the gradient masking property can be studied and interpreted. Gradient masking is most likely present when a white-box attack fails on a defense model, but the black-box attack succeeds on it. If the black-box accuracy is lower than in the white-box scenario, gradient masking cannot be presumed. The black-box scenario is a subset of the white-box scenario and assumes less information, only with gradient masking this additional information is adverse.

5 Results and Discussion

The following presents and interprets the different results of the evaluation. In particular, the following is addressed: training time, validation performance, effect of random noise, transferability, perceptibility, the robustness of the different defenses, gradient masking and the decision space.

Training time. Table 3 illustrates the training time of the different models. If the validation loss has not improved over two epochs, the training ends prematurely. The implementation of the defense can affect both the number of epochs trained and the average training time per epoch. Assuming that further training does not lead to any improvement after a certain point, it is desirable to converge fast. A dynamic environment may require more frequent (re)training of models, or a model may need to be trained with limited computational resources. For adversarial defenses, this may imply that some are not suitable for specific scenarios because they are simply too costly.

Training time			
Model	Trained epochs	Mean time per epoch	Training cost
Base Model	5 epochs	135.0 seconds	1.0x
Adversarial Training	4 epochs	380.3 seconds	2.3x
	4 epochs	384.5 seconds	2.3x
Madry Defense	5 epochs	1170.2 seconds	8.7x
	7 epochs	2030.3 seconds	21.1x
Superimposing	3 epochs	715.7 seconds	3.2x
	7 epochs	714.7 seconds	7.4x
Defensive Distillation	1+4+5 epochs	133.0 seconds	2.0x

Table 3 Training time for each model. The training cost was estimated by the total training time in relation to the total training time of the base model.

As the name promises, the training with EfficientNetV2 is very efficient. It took Seiler, Trautmann, and Kerschke, 2020 more than 3 hours on an identical GPU and a very similar dataset with VGGNet to reach the threshold of 70% validation accuracy. EfficientNetV2 achieves this in just over 11 minutes. However, the values only give a ranking for this evaluation and a comparison is generally not very meaningful, since the training time depends a lot on the concrete implementation, the parametrization of the defense and the computational resources. Adversarial training can be implemented by regularizing the loss function, but likewise by training directly on an adversarial perturbed dataset. For example, Seiler, Trautmann, and Kerschke, 2020

converges with the superimposing defense even faster than the base model. In this implementation, superimposing is more costly than adversarial training. This difference can be attributed to the mentioned differences. Comparing Adversarial training with the two Madry defense models, especially the repeated iterative determination of the gradient is costly. For the latter, the average training time per epoch scales approximately linearly with the number of iterations performed: 380.3 seconds for one, 1170.2 seconds for ten iterations and 2030.3 seconds for twenty iterations.

Validation performance. In addition to the fact that EfficientNetV2 converges quickly, it achieves a decent accuracy for the Tiny ImageNet dataset with just under 73% for the classification task (Table 4). In comparison, the developers of the dataset achieved about 60% in 2015 (Le & Yang, 2015). Seiler, Trautmann, and Kerschke, 2020 also reached similar values with 74%, but on a 128x128 Tiny ImageNet dataset. Through further adaptation and augmentation, an improvement of this is possible, however, this is not the focus of this work.

The defenses have effects on the validation accuracy. Overall, all models can overcome the threshold of 50% and except for the Madry defense and superimposing (λ = 10), even the 70%. As can be seen for the models with Madry defense, robustness to black-box or white-box attacks is often accompanied by reduced accuracy for the original task (Table 4). This general trade-off has already been identified and discussed by others (Yang et al., 2020). At the same time, all the presented defenses augment the training data in some form: adversarial training and Madry defense add adversarial perturbation to the images, superimposing combines images and defensive distillation trains on the soft labels. More data to train on generally improves both the generalization and validation accuracy of the models (Goodfellow, Bengio, & Courville, 2016). This leads to defensive distillation and adversarial training performing even better than the base model. Defensive distillation reaches with 76.2% even the highest validation accuracy of all models. Here, as in the underlying implementation, an epoch was trained normally to get a good initial start. For the implementation of defenses this possibility should be considered.

In practice, decisions often have to be made and weighed up for the particular application: How important is the accuracy for the original task and how high is the need for robustness against AEs. What are the implications of AEs for the system? The Madry defense has lower accuracy than the simpler adversarial training, but is also more robust. Increasing perturbation and especially iterations further seems to have a negative impact on accuracy. The accuracy will approach adversarial training for a smaller number of iterations.

MI-FGSM. The results of the evaluation (Table 4) show that all models can be, in some cases completely, fooled by MI-FGSM, both in the black-box and white-box scenarios. For a larger $\epsilon = 16$, the substitute models accuracy drops from 100% for the evaluation dataset to 0.1%. To illustrate, out of 2500 images that were originally correctly classified, only two to three are correctly classified after the perturbation. For 200 classes, this is even worse than random guessing. Furthermore, the AEs generated by MI-FGSM transfer sufficiently well to the attacked target models, which allows a suitable differentiation between the defenses. A look at the base model shows that the accuracy can be reduced arbitrarily by increasing the perturbation. This only requires a bit more than for the substitute.

Because of both aspects, MI-FGSM is an appropriate attack for this and other evaluations. Part of the strength in the black-box setting may stem from the fact that the black-box scenario under investigation is not a realistic. In typical black-box scenarios, the attacker does not know the training dataset and the architecture to be attacked.

Random Noise. In order to investigate whether random distortion of the input could also produce AEs, all models were also tested for noisy images. However, the effect of random noise with a ℓ_{inf} perturbation of $\epsilon = 8$ has little effect on the prediction accuracy overall (Table 4). Comparing this with the black-box datasets of $\epsilon = 0$ and $\epsilon = 8$, we can see that noise reduces the accuracy by only a few percent for the majority of models. For the base model, this means a concrete reduction from 87.4% to 85.6%. Only the effect on the adversarial trained models is slightly out of line, with the accuracy being reduced by significantly more. However, this is still in no relation to the adversarial dataset generated with MI-FGSM, which brings with the same amount of perturbation the base model down to only 6.8%. Augmenting the images of the training data set with random noise, which was not used for this evaluation, could eliminate this effect.

Although it is possible to reduce the accuracy with random noise, much more of it is needed. With enough perturbation, however, we can make every image completely black or white, which then cannot be correctly classified by humans either. Thus, random noise is not suitable to structurally enforce misclassifications in the analyzed scenario. It can be rejected that the investigated effects can also be generated by random shifts of the input.

	Validation	Noise	Black-box					
Model		$\epsilon = 8$	clean	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$	Avg.
Base Model								
	72.8%	85.6%	87.4%	20.8%	6.8%	2.9%	1.6%	8.0%
Adversarial Training Model								
$\epsilon = 16$	77.2%	79.4%	92.4%	35.8%	23.0%	50.4%	50.2%	39.9%
$\epsilon \sim U(0, 32)$	74.7%	73.7%	91.2%	66.6%	65.2%	59.7%	38.9%	57.6%
Madry Defense Model								
$\epsilon = 8, n = 10$	59.6%	73.7%	73.7%	71.3%	69.2%	63.9%	50.7%	63.8%
$\epsilon = 16, n = 20$	56.1%	66.6%	70.9%	68.1%	66.1%	58.7%	46.4%	59.8%
Superimposing Model								
$\lambda = 0$	72.8%	86.6%	89.4%	15.4%	4.5%	1.8%	1.1%	5.7%
$\lambda = 10$	65.6%	79.6%	80.4%	57.1%	39.9%	22.0%	11.1%	32.5%
Defensive Distillation Model								
$\tau = 50$	76.2%	88.9%	90.7%	19.7%	6.0%	2.6%	1.5%	8.6%

Table 4 Evaluation of the accuracy from different models under black-box adversarial attacks. The accuracy for the validation dataset and a dataset with random noise ($\epsilon = 8$) give a benchmark. All black-box datasets were generated on the substitute using MI-FGSM. The average is calculated based on the four adversarial perturbed datasets with $\epsilon > 0$.

Black-box robustness. Each of the defenses shows a characteristic behavior in the evaluation. Descending by total robustness, following the points can be made about the defenses:

the base model does not implement any defense and therefore performs among the worst (8.0% average accuracy). Only the superimposing defense with $\lambda = 0$ (5.7% average accuracy) and defensive distillation (8.6% average accuracy) are similar or even slightly worse. However, these differences are not significant. Both defensive distillation and superimposing without regularization by the Kullback-Leibler divergence thus seem to add no protection to the base model. In a practical scenario, the only resilience of the three models arises from the challenge of generating TAEs.

The superimposing defense with $\lambda = 10$ provides more protection (32.5% average accuracy), but fails for larger amounts of perturbation. However, the protection provided by this defense has proven to be rather discontinuous during evaluation. In particular, when training multiple instances of this defense, the protection varied more widely, with some on average barely improving the base model. Further research is needed here. However, what distinguishes this defense from adversarial training and the Madry defense is that it works without the costly generation of AEs.

Adversarial training with $\epsilon = 16$ provides higher protection (39.9% average accuracy) compared to the previous models for its low training and validation cost compared to the Madry defense. Noticeably, though, the protection is lower for small ℓ_{inf} distances ($\epsilon = 4$ and $\epsilon = 8$) than for the larger perturbation values ($\epsilon = 16$ and $\epsilon = 32$). This effect does not occur for any other defense. Presumably, this is due to the fact that a fixed and high perturbation was used for training with $\epsilon = 16$ and that the model still has many blind adversarial spots close to the original input. Seiler, Trautmann, and Kerschke, 2020 observed the same effect in the analysis of the virtual adversarial training defense. To investigate this further, training was also performed with a variable perturbation of $\epsilon \sim U(0, 32)$.

Adversarial training on variable perturbation sampling $\epsilon \sim U(0, 32)$ has a very positive effect on the robustness of the model and performs significantly better than with constant values (on average 57.6% and 39.9%). Similar to pre-training for the validation accuracy, a variation of perturbation seems to be beneficial for the general robustness.

The Madry defense provides the best resilience of all defenses and deviates little from the clean evaluation dataset even with a very high perturbation of $\epsilon = 32$. Small and average amounts of perturbation ($\epsilon = 4$ and $\epsilon = 8$) have slightly less impact. This protection comes at the cost of both training time and validation accuracy. Increasing the number of iterations and the perturbation used in training did not improve the robustness. It led overall to slightly worse results, even for high perturbation. Inversely, the Madry defense should approach adversarial training for iterations $\to 1$.

Visual comparison of AEs. Figure 7 illustrates AEs for some images of the datasets and allows the comparison under the different ℓ_{inf} bounds. Due to the fact that the largely very detailed images of the ImageNet dataset have been downscaled to 64x63x3 for Tiny ImageNet, there is a loss of information and the original input already exhibits some fuzziness (Le & Yang, 2015). It is important to be aware of this factor in the following.

Especially for the larger values of $\epsilon = 16$ and $\epsilon = 32$ a distortion of the images is clearly visible. At the same time, the images can probably still be recognized correctly by a significant number of humans even at the highest perturbation, which we know many CNNs fail to do. For lower values of $\epsilon = 4$ it is difficult to impossible to distinguish the original from the contaminated image with the unaided view. A good choice for evaluation is around $\epsilon = 8$ exactly in between. The distinction is still difficult, however, a base model is already fooled very noticeably. Making AEs as indistinguishable as possible from the original input can be achieved by suitable adjustments (Zou et al., 2020).

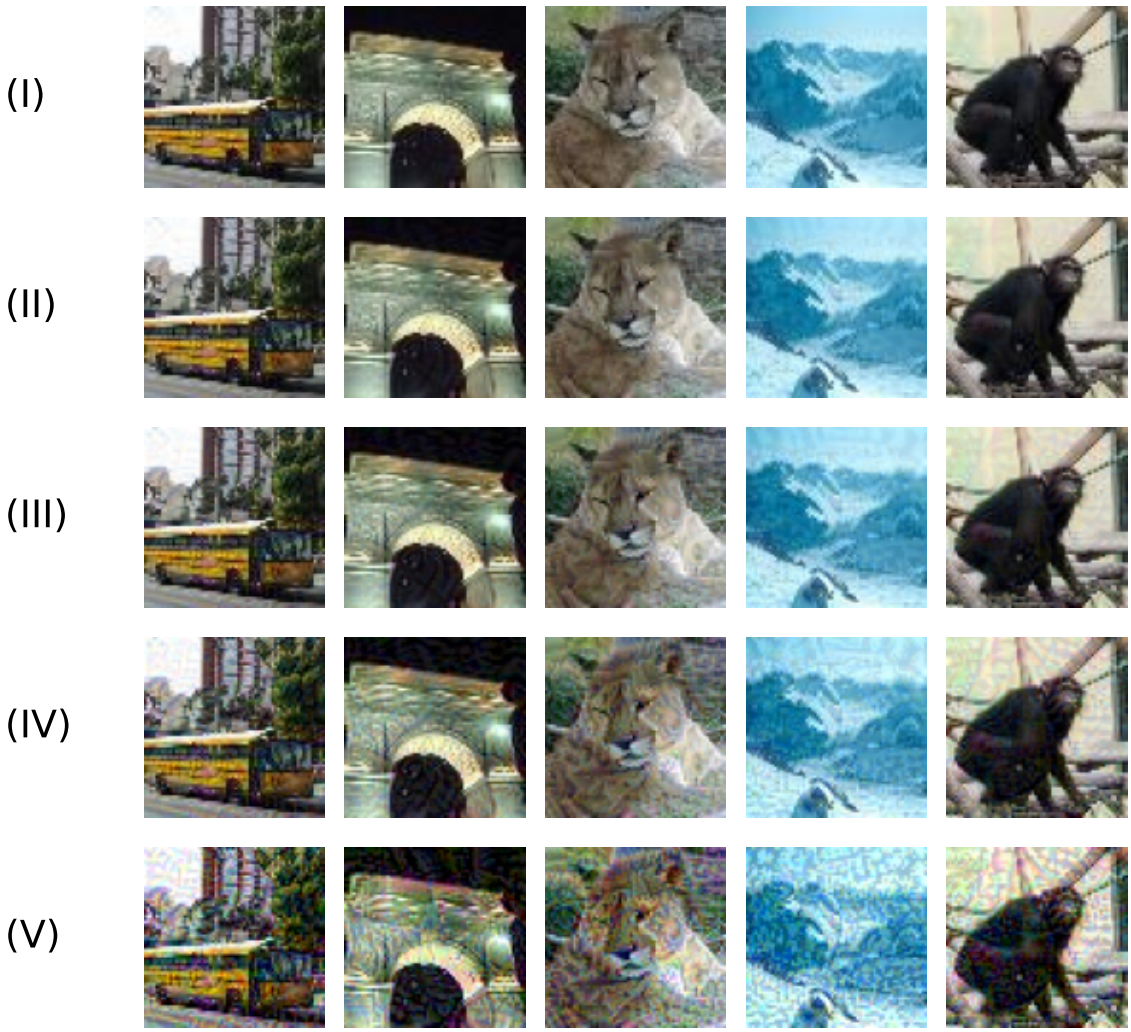

Figure 7 The effects of adversarial perturbation on five images of the Tiny ImageNet dataset. (I) shows the original images. For (II-V) adversarial perturbation is generated with MI-FGSM using $\epsilon = [4, 8, 16, 32]$.

Gradient masking. The gradient masking property can be evaluated by comparing the black-box and white-box attack accuracy. Comparing the different models (Table 5), for adversarial training, Madry defense and superimposing ($\lambda = 10$) the white-box accuracy is significantly lower due to the additional information provided by the target gradient. For the base model and superimposing ($\lambda = 10$) this is not the case, but insignificant, since the base model does not implement any defense and superimposing ($\lambda = 0$) in general performs worse than the base model. Therefore, gradient masking cannot be assumed for any of these models.

Defensive distillation exhibits a different behavior. In the white-box scenario, the defense shows an accuracy of 87.9%, which is very close to the non-corrupted dataset with 90.7%. One could argue that it offers a suitable defense against adversarial attacks. However, looking at black-box accuracy (6.0%) it can be seen that this

	White-box	Black-box
Model	$\epsilon = 8$	$\epsilon = 8$
Base Model		
	7.1%	6.8%
Adversarial Training Model		
$\epsilon = 16$	4.9%	23.0%
$\epsilon \sim U(0, 32)$	6.9%	65.2%
Madry Defense Model		
$\epsilon = 8, n = 10$	26.9%	69.2%
$\epsilon = 16, n = 20$	30.2%	66.1%
Superimposing Model		
$\lambda = 0$	6.0%	4.5%
$\lambda = 10$	8.8%	39.9%
Defensive Distillation Model		
$\tau = 50$	87.9%	6.0%

Table 5 For both scenarios, adversarial perturbation was generated using the MI-FGSM ($\epsilon = 8$). The AEs for the black-box evaluation were generated on the substitute and for white-box directly on the attacked model itself.

robustness against AEs disappears completely when facing TAEs. This indicates that gradient masking causes the white-box attack to fail. Defensive distillation does not provide any improvement over the base model. Protection by defensive distillation must therefore be rejected. These results were also confirmed by Papernot et al., 2017.

Analysis of decision space. The high dimensionality of the input images and the aggregated view of whole datasets by accuracy does not allow evaluation of the decisions of the models on a single image level. At the same time, this is very important to understand how a defense changes the decision space and thus why defenses fail or work. Ideally, the observation at the individual level is consistent with the aggregated perspective of accuracy. From the theory, AEs occur by maximizing the loss for the original class. Figure 8 shows for some of the models how the loss changes when attacking the target model black-box or white-box.

The observed robustness of the models is reflected in the average loss level. The more robust a defense in both scenarios, the smaller the red and white subspaces. If gradient masking does not occur, a stronger attack is possible with the additional information of the target. The loss increases for less perturbation when followed the target direction than for the substitute direction. Except for defensive distillation, all models show this behavior.

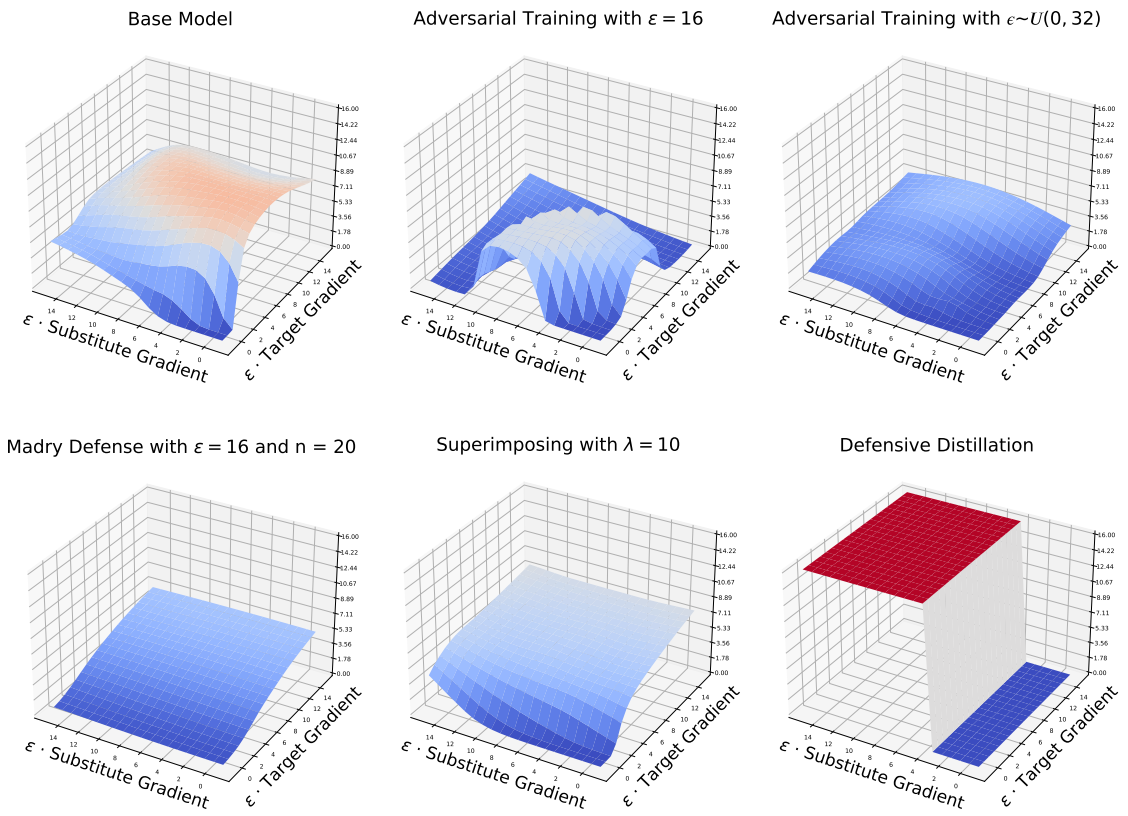

Figure 8 Loss landscape of six different models on the same original input image. On the x-axis is the direction of the gradient for the substitute and on the y-axis the direction of the gradient for the model itself depicted. Both were determined using the single-step FGSM method. The former is thus a black-box and the latter a white-box attack. The maximum perturbation $\epsilon \in [-1, 16]$ was varied for both directions and the loss was determined using cross entropy. High function values in the warmer regions correspond to high loss values and thus very likely result in AEs.

Since the base model does not implement any defense, the adversarial subspace is respectively large. We can also match and verify the identified surprisingly low robustness of adversarial training to smaller perturbations (e.g. $\epsilon = 2$ to $\epsilon = 10$) in the landscape. Sampling the epsilon in the training further flattens the loss landscape and this subspaces disappeared.

A defense leading to gradient masking the gradient is not improving the loss in white-box attacks, so the loss landscape should have a characteristic shape: it increases only in one direction, which is the one of the substitute gradient. Exactly this behavior can be observed for defensive distillation and confirms the hypothesis from the whole datasets. An attack on defensive distillation itself fails to maximize the loss, but succeeds via a substitute.

6 Conclusion

The field of adversarial machine learning takes an important role in the overall context of machine learning. It is possible to fool even the best trained and tuned models by the use of special perturbation. Unlike in cryptography, there are as yet no defenses that we can assume to be secure. Since the first results, this field has seen an arms race between attacks and defenses, thus allowing weaknesses of one or the other to be identified and improved. Whether this converges against safe defenses in the long run, or one day a safe defense is discovered, or whether there is no safe defense due to the inherent properties of the models, has not yet been resolved.

The aim of this thesis was not only to present the theoretical aspects of scenarios, attacks and defenses, but to empirically investigate selected scenarios. The data collected about the models include the training time, the evaluation accuracy of black-box and white-box attacks, the effect of random noise, and information about the decision space. Many failed defenses unconsciously lead to gradient masking, which makes the model robust only against white-box attacks using the gradient. Investigating this property was one of the goals. For this purpose, particular use was made of the black-box property of TAEs.

Contributions. The following contributions were made by this evaluation:

- in the scenario, MI-FGSM generates TAEs that both transfer frequently and strongly deceive the model.
- robustness to AEs comes with increased training time and reduced validation accuracy.
- for defensive distillation, gradient masking could be identified by comparing white-box and black-box attacks, as well as by analyzing the decision space. The effect of gradient masking could not be identified and assumed for the other defenses.
- superimposing without regularization by the Kullback-Leibler divergence offers no improvement over the base model.
- among the methods evaluated, adversarial training and the Madry defense have been shown to be the most robust, with random sampling of perturbation and training on multi-step AEs being particularly beneficial. However, increasing perturbation and iterations did not lead to any improvement.
- random noise is not able to reproduce the effect of an adversarial attack.

Limitations of evaluation. Restricted by time and scope boundaries, this work only examines a very specific scenario and can therefore only make limited general claims. However, the following aspects, in particular, are possible extensions of this work:

only the untargeted scenario was investigated, whereas there is also a targeted scenario for which the evaluation may come to different results. Furthermore, ensembling was not considered in the evaluation. However, it can enrich both the defenses and the attacks (Tramèr et al., 2017). In general, less effort was put into the optimal configuration of hyperparameters, since arbitrary configurations can be justified and often quickly bypassed. Pre-training was only tested for defensive distillation and random sampling only for adversarial training, though perhaps other defenses could benefit from these as well. Moreover, different CNN architectures can be tested (ResNet, Inception, ...) to assess topological factors. The evaluation was based solely on the ℓ_{inf} distance, although other ℓ_p norms can be used to measure distance, or the distance can be judged in a fundamentally different way.

Implications for research and practice. The implications for practice arise from the existence of AEs. Even if it is intended to keep the model secret to prevent white-box attacks, it must be assumed that black-box attacks can and will be used. This applies especially to safety-critical areas such as autonomous driving, but appropriate considerations should also be made for areas where this is not the case. For the selection of suitable defenses, the properties such as training time, validation accuracy, robustness have to be carefully aligned with the need for security and available resources such as computational power. If the gradient masking property is assumed for a defense, it should not be used, as this protection can be circumvented particularly easily.

For the research, this work implies in particular which aspects are to be emphasized in the development of an attack. When proposing a new defense, it should be extensively examined for gradient masking. Among other things, this can and should be done with the methods used. The costs, both in terms of validation accuracy and training time, should also always be considered. For reproducible research, the code should also be made public to validate results and test them with other methods.

Future research. While the number of publications in this area is steadily increasing and adversarial machine learning is now an established niche, there is still much future research to be done. The following directions could be interesting for future research:

how can existing defenses be extended in a practical way, for example through ensembling, pre-training, or perturbation sampling, to maximize the performance? Further-

more, it could be analyzed how humans actually perceive adversarial examples and under which circumstances they recognize them and not. Adversarial perturbation is usually measured by one of the ℓ_p norms, but this is only a proxy for how people see differences in pictures and does not describe what actually happens. There are many domains for which adversarial examples can be studied further, for example in remote sensing. It can be studied to what extent adversarial attacks are only a theoretical threat or whether these vulnerabilities are already being exploited today. Furthermore, the development of a library to perform and analyze large-scale evaluations would be a great benefit. Although the libraries Cleverhans, Foolbox[6] and Robustness[7] already provide great features for creating AEs, none of them really offers the functions needed for executing an extensive evaluation.

Further research in this area is needed. The existence of adversarial examples prevents the safe use of machine learning in many domains. As a consequence, existing safe methods must be used, which limits the further success of machine learning.

[6] https://github.com/bethgelab/foolbox
[7] https://github.com/MadryLab/robustness

Bibliography

Athalye, A., Carlini, N., & Wagner, D. A. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In J. G. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning, ICML 2018, stockholmsmässan, stockholm, sweden, july 10-15, 2018* (pp. 274–283). PMLR. http://proceedings.mlr.press/v80/athalye18a.html

Bishop, C. M. (2007). *Pattern recognition and machine learning, 5th edition*. Springer.

Breiman, L. (2001). Statistical Modeling: The Two Cultures. *Statistical Science*, *16*(3), 199–231. https://doi.org/10.1214/ss/1009213726

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I. J., Madry, A., & Kurakin, A. (2019). On evaluating adversarial robustness. *CoRR*, *abs/1902.06705*. http://arxiv.org/abs/1902.06705

Carlini, N., & Wagner, D. A. (2016). Defensive distillation is not robust to adversarial examples. *CoRR*, *abs/1607.04311*. http://arxiv.org/abs/1607.04311

Carlini, N., & Wagner, D. A. (2017). Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, 39–57. https://doi.org/10.1109/SP.2017.49

Dodge, S. F., & Karam, L. J. (2017). A study and comparison of human and deep learning recognition performance under visual distortions. *26th International Conference on Computer Communication and Networks, ICCCN 2017, Vancouver, BC, Canada, July 31 - Aug. 3, 2017*, 1–7. https://doi.org/10.1109/ICCCN.2017.8038465

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., & Li, J. (2018). Boosting adversarial attacks with momentum. *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 9185–9193. https://doi.org/10.1109/CVPR.2018.00957

Elgendy, M. (2020). *Deep learning for vision systems*. Manning Publications.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., & Song, D. (2018). Robust physical-world attacks on deep learning visual classification. *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 1625–1634. https://doi.org/10.1109/CVPR.2018.00175

Goodfellow, I. J., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. http://www.deeplearningbook.org

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference*

on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings. http://arxiv.org/abs/1412.6572

Hornik, K., Stinchcombe, M. B., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *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* (pp. 1106–1114). https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

Kurakin, A., Goodfellow, I. J., & Bengio, S. (2017). Adversarial examples in the physical world. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. https://openreview.net/forum?id=HJGU3Rodl

Le, Y., & Yang, X. S. (2015). Tiny imagenet visual recognition challenge. http://vision.stanford.edu/teaching/cs231n/reports/2015/pdfs/yle_project.pdf

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object recognition with gradient-based learning. *Shape, contour and grouping in computer vision* (pp. 319–345). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-46805-6_19

Liu, Y., Chen, X., Liu, C., & Song, D. (2017). Delving into transferable adversarial examples and black-box attacks. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. https://openreview.net/forum?id=Sys6GJqxl

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. https://openreview.net/forum?id=rJzIBfZAb

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Murphy, K. P. (2012). *Machine learning - a probabilistic perspective*. MIT Press.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. Practical black-box attacks against machine learning. In: 2017, 506–519. https://doi.org/10.1145/3052973.3053009.

Papernot, N., McDaniel, P. D., & Goodfellow, I. J. (2016). Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. *CoRR*, *abs/1605.07277*. http://arxiv.org/abs/1605.07277

Papernot, N., McDaniel, P. D., Sinha, A., & Wellman, M. P. (2016). Towards the science of security and privacy in machine learning. *CoRR*, *abs/1611.03814*. http://arxiv.org/abs/1611.03814

Papernot, N., McDaniel, P. D., Wu, X., Jha, S., & Swami, A. (2015). Distillation as a defense to adversarial perturbations against deep neural networks. *CoRR*, *abs/1511.04508*. http://arxiv.org/abs/1511.04508

Pereira, A., & Thomas, C. (2020). Challenges of machine learning applied to safety-critical cyber-physical systems. *Machine Learning and Knowledge Extraction*, *2*(4), 579–602. https://doi.org/10.3390/make2040031

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rosenblatt, F. (1961). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms* (tech. rep.). Cornell Aeronautical Lab Inc Buffalo NY.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., & Fei-Fei, L. (2014). Imagenet large scale visual recognition challenge. *CoRR*, *abs/1409.0575*. http://arxiv.org/abs/1409.0575

Seiler, M. V., Trautmann, H., & Kerschke, P. (2020). Enhancing resilience of deep learning networks by means of transferable adversaries. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN48605.2020.9207338

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. In Y. Bengio & Y. LeCun (Eds.), *2nd international conference on learning representations, ICLR 2014, banff, ab, canada, april 14-16, 2014, conference track proceedings*. http://arxiv.org/abs/1312.6199

Tan, M., & Le, Q. V. (2021). Efficientnetv2: Smaller models and faster training. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning, ICML 2021, 18-24 july 2021, virtual event* (pp. 10096–10106). PMLR. http://proceedings.mlr.press/v139/tan21a.html

Tramèr, F., Kurakin, A., Papernot, N., Boneh, D., & McDaniel, P. D. (2017). Ensemble adversarial training: Attacks and defenses. *CoRR*, *abs/1705.07204*. http://arxiv.org/abs/1705.07204

Yang, Y., Rashtchian, C., Zhang, H., Salakhutdinov, R., & Chaudhuri, K. (2020). A closer look at accuracy vs. robustness. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, neurips 2020, december 6-12, 2020, virtual*. https://proceedings.neurips.cc/paper/2020/hash/61d77652c97ef636343742fc3dcf3ba9-Abstract.html

Zhou, & Chellappa. (1988). Computation of optical flow using a neural network. *IEEE 1988 International Conference on Neural Networks*, 71–78 vol.2. https://doi.org/10.1109/ICNN.1988.23914

Zou, J., Pan, Z., Qiu, J., Duan, Y., Liu, X., & Pan, Y. (2020). Making adversarial examples more transferable and indistinguishable. *CoRR*, *abs/2007.03838*. https://arxiv.org/abs/2007.03838

Declaration of Authorship

I hereby declare that, to the best of my knowledge and belief, this thesis titled *Application of Transferable Adversarial Attacks on Convolutional Neuronal Networks: An Evaluation of Existing Attack and Defense Mechanisms* is my own, independent work. I confirm that each significant contribution to and quotation in this thesis that originates from the work or works of others is indicated by proper use of citation and references; this also holds for tables and graphical works.

Münster, 28.07.2022

Linus Stach

Consent Form

Name: Linus Stach
Title of Thesis: Application of Transferable Adversarial Attacks on Convolutional Neuronal Networks: An Evaluation of Existing Attack and Defense Mechanisms

What is plagiarism? Plagiarism is defined as submitting someone else's work or ideas as your own without a complete indication of the source. It is hereby irrelevant whether the work of others is copied word by word without acknowledgment of the source, text structures (e.g. line of argumentation or outline) are borrowed or texts are translated from a foreign language.

Use of plagiarism detection software. The examination office uses plagiarism software to check each submitted bachelor and master thesis for plagiarism. For that purpose the thesis is electronically forwarded to a software service provider where the software checks for potential matches between the submitted work and work from other sources. For future comparisons with other theses, your thesis will be permanently stored in a database. Only the School of Business and Economics of the University of Münster is allowed to access your stored thesis. The student agrees that his or her thesis may be stored and reproduced only for the purpose of plagiarism assessment. The first examiner of the thesis will be advised on the outcome of the plagiarism assessment.

Sanctions Each case of plagiarism constitutes an attempt to deceive in terms of the examination regulations and will lead to the thesis being graded as "failed". This will be communicated to the examination office where your case will be documented. In the event of a serious case of deception the examinee can be generally excluded from any further examination. This can lead to the exmatriculation of the student. Even after completion of the examination procedure and graduation from university, plagiarism can result in a withdrawal of the awarded academic degree.

I confirm that I have read and understood the information in this document. I agree to the outlined procedure for plagiarism assessment and potential sanctioning.

Münster, 28.07.2022

Linus Stach