Adversarial Preprocessing

Markus Karrenbauer
markus.karrenbauer@student.kit.edu
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany

Figure 1: Demonstrating an attack image A and the resulting output image D after being down-scaled [27].

ABSTRACT

Especially in the domain of computer vision, machine learning based applications have achieved remarkable success. Hence such applications are widely used and also present in security critical applications such as self-driving cars or biometric authentication systems. However, the fact that the decision-making process of machine learning models often is not well analyzed, makes them vulnerable to new attacks that exploit properties that haven't been investigated yet. An example for such attacks are adversarial examples which were reported by [23]. In order to increase the security of machine learning based applications these adversarial examples have received lots of attention in current research. Recently a new kind of attack has been demonstrated by Xiao et al.[27] which is called image-scaling attack. In order to provide a better understanding of this attack Quiring et al. further analyzed the attack [18, 19]. This report summarizes the new phenomenon of image-scaling attacks and recaps the key results from Quiring et al. and Xiao et al. That especially includes the presentation of the key concepts of the attack as well as the suggested defenses and detection strategies from Quiring et al. and Xiao et al.

KEYWORDS

image-scaling attacks, adversarial machine learning, preprocessing, evasion attacks, poisoning attacks

1 INTRODUCTION

Within the last years, machine learning based applications have achieved remarkable success in lots of different application domains. One very popular domain is computer vision, where especially the usage of convolutional neural networks results in applications that, in some benchmarks, even can beat human performance. Since convolutional neural networks provide state-of-the-art prediction results in lots of different scenarios, also security critical applications may rely on them. Examples for such applications are self-driving cars, biometric authentication systems or surveillance systems.

Besides the obvious benefits of machine learning based applications, these technologies also offer a new attack surface for adversaries. The fact that machine learning algorithms often are treated as black-boxes makes them susceptible for new kinds of attacks. That's why the security of such systems is a hot topic in current research. Two very popular attacks on machine learning models that already have received lots of attention are evasion attacks, which commonly are applied by using adversarial examples, and poisoning attacks. While evasion attacks aim to exploit the test stage of a machine learning pipeline, poison attacks try to manipulate the attacked model at the training stage by inserting malicious training data.

Recently a new kind of attack has been demonstrated by Xiao et al.[27] which can be used to attack potentially any kind of application that involves image processing, i.e. image scaling, including image based machine learning models. The new attack is called image-scaling attack and is applied at the preprocessing stage of the application's processing pipeline. The key idea of this attack is to manipulate an image in a way that its visual appearance stays similar to the original image, but is completely changed after being down-scaled. After Xiao et al.[27] presented the attack, Quiring et al.[18] further analyzed its root-cause and investigated the scaling algorithm implementations of the commonly used machine learning frameworks TensorFlow, Caffe and PyTorch. Furthermore they showed that the attack can either be used in a similar way to evasion attacks or to support poisoning attacks [19]. In addition they presented a proper defense that can sanitize an attacked image.

This report gives an overview about current research in the field of adversarial preprocessing with respect to the domain of

Seminar "Adversarial Machine Learning", Karlsruhe Institute of Technology (KIT), 2020

computer vision. Since we couldn't find any works about other attacks (within the domain of computer vision) that explicitly attack the preprocessing stage of a machine learning pipeline, we mainly focus on image-scaling attacks. Thus, the report summarizes the new phenomenon of image-scaling attacks and recaps the key results from Quiring et al. and Xiao et al.[18, 19, 27]. Therefore an introduction into image preprocessing in general as well as a background on evasion attacks and poisoning attacks is presented. Furthermore the concept of the image-scaling attack is explained. The methodology section of this work covers the concepts of creating an attack-image for an image-scaling attack as well as the root-cause that enables these attacks. In addition possible mitigation strategies, i.e. detection methods [27] and defensive methods [18], are described. Following on that, it is briefly explained, how evasion-similar attacks and poisoning attacks can be applied, using an image-scaling attack. The experiments section briefly covers some of the experimental results of Quiring et al. and presents a few own experiments, based on their provided open source algorithms. Finally a conclusion on image-scaling attacks is drawn.

2 BACKGROUND & RELATED WORK

This section introduces the different domains which are related to the scope of image-scaling attacks. At first, a brief overview on different preprocessing steps in deep learning pipelines, focusing on image scaling, is given. To give a further understanding of adversarial machine learning and how image-scaling attacks can be applied, evasion attacks as well as poisoning attacks are explained.

2.1 Preprocessing and Image Scaling in Machine Learning

In order to optimize and clean up the data that is put into a machine learning model, it is mandatory to preprocess the captured sensor signal that should be processed. In the field of image processing such steps may be scaling, cropping, filtering and applying different affine transformations. For example Taigman et al.[24] use a complex alignment pipeline, including cropping and affine transformations to extract and frontalize a face from an image which then is used as input for their convolutional neural network.

As the state of the art for image processing applications is based on convolutional neural networks (CNN) which generally require a fixed sized input, image scaling has become mandatory for such applications. This is even more amplified by the fact that it is a common practice to involve pretrained models in the training process of a CNN. Xiao et al.[27] reported input image sizes between 32×32 pixel and 299×299 for nine popular deep learning models. Since images mostly are captured in a higher resolution, they are often down scaled to fit the input size of the used model.

This omnipresent usage of down-scaling for image processing applications results in a perfect base for the image-scaling attack which exploits the implementation of downscaling algorithms.

There exist multiple different algorithms to down-scale an image which share the same goal and principles. In general such an algorithm aims to reduce the original size $m \times n$ of an image to the new size $m' \times n'$ while preserving the visual features of the image so that it looks similar to a human before and after the scaling process. This is achieved by interpolating the pixels of the source image. Each pixel of the down-scaled image is calculated by a weighted sum of its surrounding pixels. The exact choice and weighting of the considered pixels is determined by the underlying algorithm. Common algorithms for example are nearest-neighbor or bilinear downscaling. The concept of downscaling is illustrated in figure 2.

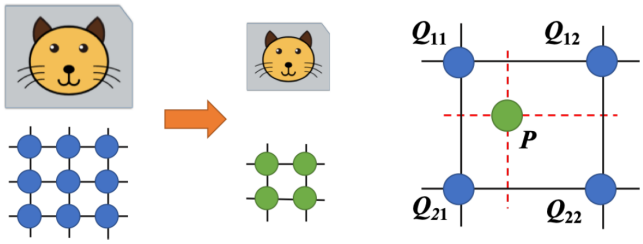

Figure 2: Demonstrating the concept of image downscaling. The pixels Q_{11}, Q_{12}, Q_{21}, Q_{22} of the original image are interpolated to gain the new pixel value P of the down-scaled image [27].

Since downscaling is a mandatory step for image preprocessing, common deep learning frameworks like TensorFlow, Caffe and PyTorch support different downscaling algorithms such as nearest-neighbor-, bilinear-, bicubic- and area-downscaling. According to Quiring et al.[18] other deep learning libraries are either based on the mentioned frameworks or on OpenCV and Pillow which also is the base for Caffe and PyTorch. Thus the availability of different downscaling algorithms is limited to a few implementations which makes it easy for an attacker to guess the image scaling algorithm in a scenario where no explicit knowledge about the used algorithm is given.

2.2 Evasion Attacks

The term evasion attacks commonly describes attacks against machine learning models that are performed at test time. The goal of such an attack is to produce false prediction results of the attacked model for specific inputs. In general these specific inputs are adversarial examples. An adversarial example is a precisely manipulated model input that can be created, by applying small perturbations to the original input. Therefore the perturbations are calculated in respect to the attacked model, a surrogate model or a specific data set. While the attacked model creates wrong predictions, if an adversarial example is given, its intended behavior under normal circumstances is not influenced.

In contrast to adversarial preprocessing, adversarial examples have seen lots of attention within the last years. Firstly the concept of adversarial examples was reported by Szegedy et al.[23]. In their work, they called them intriguing properties and showed that these properties are non-random and can be applied to multiple different deep learning models. In a following work, Goodfellow et al.[6] presented a more detailed analyze of adversarial examples and explained that the root-cause of adversarial examples is that most deep learning models are too linear. Furthermore they presented a simple and fast method to generate such examples called *fast gradient sign method*.

Since this method directly calculates the required gradients and assumes the knowledge of the used cost function of the attacked

model, this method can be seen as a white-box attack (an attack with full knowledge about the used model, algorithms and other relevant details). One early example of an attack with adversarial examples was presented by Kurakin et al.[10]. In 2016 Papernot et al.[16] showed that it is also possible to perform black-box attacks (attacks with limited knowledge about the used model, algorithms etc.) with adversarial examples. This could be achieved by creating a surrogate model which mimics the attacked model and is used to generate the adversarial examples. [1]

Besides developing different evasion attacks, there also has been research towards defenses against such attacks. One potential defense which involves a second model, that is trained to predict the input of another model and thus learns soft labels instead of hard labels, was presented by Papernot et al.[17]. However even though they showed, that this so called defensive distillation could reduce the effectiveness of an adversarial attack by a lot for a specific deep learning model, it turned out, that the approach does not provide proper protection against advanced evasion attack strategies. This was proven by Papernot et al.[16] as well as Carlini & Wagner[4]. [2]

2.3 Poisoning Attacks

In general poisoning attacks aim to manipulate the training data or their labels of a machine learning model to ultimately change the model's behavior at test time. There are two types of poisoning attacks that can be distinguished: non-targeted attacks and targeted attacks. While non-targeted attacks simply try to force the attacked model to misbehave, targeted attacks aim to make the model sensible for specific input patterns.

An early approach on poisoning a machine learning model was reported by Biggio et al.[2]. In their work they investigated the model's behavior under the attack of an adversary that could flip a subset of the labels of the training data and proposed a method to make the SVM more robust against this kind of attack.

One of the first poisoning attack against deep learning models was proposed by Muñoz-González et al.[14]. Until then, the computational power that was needed to optimize the poisoning points (instead of arbitrarily changing the data or labels) was too high to effectively be applied in attack scenarios. Their proposed poisoning algorithm, called back-gradient optimization, greatly reduced the computational complexity and thus was feasible to be applied to deep learning models such as CNNs. Due to the usage of a surrogate learner, they also showed, that a poisoning attack can be applied with limited knowledge and thus be applicable in a black-box setting. Another interesting work to mention, which applied an poisoning attack on deep learning models is [8].

An approach of a targeted poisoning attack is given by Liu et al.[12]. They introduced the trojaning attack on neural networks which is used to install a trojan trigger into the network. Such a trigger can be a small, specifically engineered patch of pixels on an image. The trigger installation is not directly applied at the train time of the model. Moreover, Liu et al. propose a retrain mechanism that can be applied after the model was trained and thus requires much less computational effort than an attack that happens at train time. When retraining the model, it learns a specified malicious behavior (i.e. targeted misclassifications) which finally will be triggered on test time, as soon as an image with the trigger on it is used as input image.

Another targeted poisoning attack on deep learning models that can be triggered on specific patterns or inputs was shown by Gu et al.[7]. They called the networks that are affected by such an attack BadNets and described a realistic scenario of attacking a street sign classifier to ultimately predict stop signs as speed limits, whenever a special sticker is added to a stop sign. Note that BadNets obtain the capability to do correct predictions in normal circumstances, i.e. if no trigger pattern is present.

Shafahi et al.[21] introduced a new kind of targeted poisoning attacks which they call clean-label poisoning attack. Such an attack can be performed without actually having control over the labeling or training data. [3]

3 IMAGE-SCALING ATTACKS

This section explains the principles of image-scaling attacks. After formally introducing the attack, it is described, how an attack image automatically can be created solving an optimization problem. To simplify things, only the strong attack (see section 3.1), which is the more realistic one, is explained. Following on that, the root-cause that enables image-scaling attacks is presented. Furthermore the possibilities of using an image-scaling attack for attacks at test and train time as well as defense strategies against these attacks are explained.

3.1 Image-Scaling Attack Introduction

The goal of an image-scaling attack is to manipulate a source image, which might serve as input for an image processing model, in a way, that its appearance and thus its semantic changes when it is processed by a downscaling algorithm. This means that the attacker tries to break the property of a downscaling algorithm that the visual features of the down-scaled image are preserved. One major constraint on such an image-scaling attack is, that the manipulated image has to be similar to the original source image in order to make the attack hardly visible for a human observer.

Symbol	Size	Description
S	$m \times n$	The source image that is used to create the attack image.
T	$m' \times n'$	The target image that the adversary wants to obtain after scaling.
A	$m \times n$	The attack image, a slightly perturbed version of S
D	$m' \times n'$	The output image of the scaling function scale.

Figure 3: Showing the major objects of an image-scaling attack [18].

[1] Other interesting works in the field of adversarial examples are [1, 3, 5]. However, they are not required to understand this report and thus are not further explained.
[2] Other works that observe defense strategies against adversarial examples for example are [11, 13].
[3] Potential defenses against poisoning attacks for example are handled in [25, 26].

Formally there are four major objects to consider for an image-scaling attack, shown in figure 3. In order to perform the attack, the attacker first crafts the attack image A by carefully manipulating key pixels of the source image S in respect to the target image T. Then S is replaced by A so that the scaling procedure $scale()$ of the image processing pipeline is applied to A. In the case of a successful attack, the output image D looks similar to the target image T while the attack image A looks similar to the source image S. These two conditions can be seen as the main objectives of an image-scaling attack and can be formalized as:

$$\boldsymbol{O_1}) \; scale(A) \sim T \tag{1}$$

$$\boldsymbol{O_2}) \; A \sim S \tag{2}$$

Here $\sim$ expresses similarity between the two operands.

In figure 4 the described concept of an image-scaling attack is demonstrated. Furthermore an example of a potential attack image A and its corresponding output image D is shown in figure 1. The left image of the figure, which is the attack image A, clearly shows sheep, while the right image, the output image D, perfectly shows a wolf.

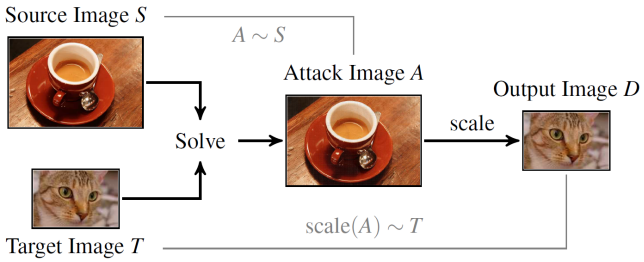

Figure 4: Demonstrating the concept of an image-scaling attack [18].

According to Xiao et al.[27] an image-scaling attack can be performed in two attack modes which are called weak attack mode and strong attack mode. While the attacker is allowed to specify the source image S as well as the target image T in the strong attack mode, the weak attack mode only allows to choose the target image T, while S is fixed.

Since an image-scaling attack is performed within the preprocessing stage of an image processing pipeline, which is located before the feature extraction stage, it can effect all subsequent stages. In the case of a deep learning model this especially includes the training as well as the testing stage. This makes it possible for an attacker to either execute cloaked poisoning attacks by adding malicious data to the training data set or executing attacks at test time by manipulating the tested input image. Furthermore an image-scaling attack is independent from the attacked model itself and thus can be applied without any knowledge of the attacked model. Even image processing applications that do not use machine learning models might suffer from such an attack.

Considering the required knowledge of an attacker, Xiao et al. showed that it is possible to perform white-box as well as black-box scaling attacks. There are only two things an attacker must know: **a)** the used scaling algorithm **b)** the size of the scaled image. In a white-box scenario these information are present per definition. In a black-box scenario Xiao et al. showed that the attacker is able to infer the required knowledge using an exhaustive search where they send a series of probing images, "crafted by the scaling method with various scaling parameters" (see [27]) to the attacked model and observe the classification result. This especially is possible due to the limited potential options for the scaling algorithms, as mentioned before. In their work they additionally describe an optimized approach that can retrieve the desired information more efficiently.

As mentioned above, image scaling is only one of many possible preprocessing steps that might be performed within an image processing pipeline. Depending on the kind of preprocessing steps and their ordering within the pipeline, they can interfere with the image scaling attack. If an image for example is first rotated and then scaled, the image scaling attack probably won't work anymore, since the carefully manipulated pixels changed their location and thus their effect on the scaling algorithm. However such interference can be bypassed, if the attacker knows about them, by considering them in the attack.

3.2 Creating the Attack Image

In order to perform an effective image-scaling attack it is mandatory to automatically and efficiently create a strong attack image A. Mathematically the attack image A can be created by adding a perturbation matrix Δ_1 to the source image S: $A = S + \Delta_1$. As described in section 3.1 the output image D results by scaling the attack image A with a certain scaling function $scale()$ which can be formalized as: $D = scale(A)$. Note that there exist multiple different possible A_i that result in the same D. Thus the attacker wants to find the A that satisfies $\boldsymbol{O_1}$ and $\boldsymbol{O_2}$ (equations 1, 2) the most. The relation between D and T, which is relevant for objective $\boldsymbol{O_1}$ is noted as follows: $D = T + \Delta_2$, where Δ_2 is another perturbation matrix. To measure the similarity between A and S as well as the similarity between D and T, Xiao et al. have chosen the L-norm. Putting all together, Xiao et al.[27] concluded that the optimal attack image A can be created solving the following objective function:

$$\min(\|\Delta_1\|^2), \; s.t. \; \|\Delta_2\|_\infty \leq \epsilon * \mathbb{N}_{max},$$

where $\mathbb{N}_{max}$ is the maximum pixel value in the current image format (e.g. 255 for 8-bit RGB images).

To solve this function in an efficient way, Xiao et al. make use of the fact, that all analyzed scaling algorithm performed the scaling in two steps: first reducing the horizontal dimension and then reducing the vertical dimension (or vice versa). This results in the following equation for an arbitrary image I: $scale(I) = L * I * R$. where L and R are two fixed coefficient matrices with dimension $m' \times m$ and $n \times n'$ assuming dimension $m \times n$ of the scaled image I. Note that L and R are determined by the used scaling algorithm and have to be calculated by the attacker. Overall the relation between source and target image that is required to create the attack image is described by Xiao et al. by:

$$L * (S + \Delta_1) * R = D$$

$$D = T + \Delta_2$$

Finally Xiao et al. show that the attack image ultimately can be generated solving a convex optimization problem based on these

equations. For more details and information on the slight differences between the weak and the strong attack see [27].

3.3 Image-Scaling Attack Analysis

Quiring et al.[18] picked up the work of Xiao et al. and performed a deeper analysis of image-scaling attacks to understand the root-cause that enables such attacks. At first Quiring et al. compared image-scaling to signal processing and argued that the information loss that happens on down-scaling can be regarded as the loss of certain frequencies of the image signal. Thus the Nyquist-Shannon theorem [15] can be applied which says that the original signal, i.e. the original image I, can not unambiguously be reconstructed if the sampling rate is too low. This effect, which is exploited by the attacker, is called aliasing effect.

In the case of image scaling, the sampling of the original image I is done by a convolution between I and a kernel function. Each pixel of the down-scaled image results from a weighted sum of a pixel subset of the original image. Therefore the kernel function, which defines the weighting of the regarded pixels, is moved as a window over the image with a specific step size. This procedure is done in order to minimize the aliasing effect when down-scaling an image. However not every pixel regarded by the kernel function contributes equally to the down-scaling process and the step size of the kernel movement highly depends on the scaling ratio between I and the scaled image I'. Thus an attacker only has to manipulate the subset of pixels that have a big contribution to the scaling process and can disregard the other pixels. Quiring et al. presented figure 5 to demonstrate such a convolution for horizontal scaling within a single row. Note that this concept is transferable to the scaling of a whole image.

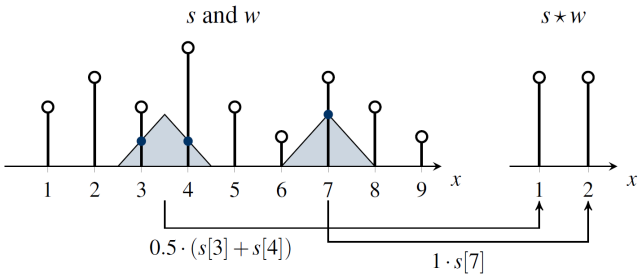

Figure 5: Illustration of a convolution of the signal s with the kernel w. Only 3 out of 9 values of s contribute to the resulting signal $s \star w$ [18].

Based on these insights, Quiring et al. identified two critical parameters that define the sparsity of the sampled pixels with high contribution, i.e. high weights, of the source image and thus affect the potential success of an attacker. These are the scaling ratio β which describes the size ratio between the original image I and the scaled image I' and the kernel width σ which defines the size of the kernel window that is moved over the source image I. Note that β and σ each can be separated in horizontal and a vertical component in case of a non-quadratic image: β_h, β_v, σ_h, σ_v. While increasing the scaling ratio β when σ is fixed, increases the sparsity of pixels, the increase of σ on a fixed β works vice versa. Thus a high β and a small σ result in a perfect attack surface for an attacker.

Bringing all of this to a conclusion, the fact that potentially only a few subset of pixels of the original image I are crucial for the down-scaling process, enables an attacker to perform an image-scaling attack that is hard to detect and thus satisfies both of the required objectives $\boldsymbol{O_1}$ and $\boldsymbol{O_2}$ (equations 1, 2).

3.4 Detecting Image-Scaling Attacks

Since one major benefit of image-scaling attacks is that they are hard to notice by a human observer, one major avoidance mechanism might be the automatic detection of such an attack. While an attack detection can not recover the model or prevent the attack, it can make the human aware of a security issue so that the human can prevent a malicious use of the affected image processing system.

Xiao et al.[27] proposed two detection methods that aim to uncover big visual dissimilarities between the original image I (which is equals S or A in an attack scenario, depending whether there is an attack or not) and the scaled image I' (which is equals D in an attack scenario).

The first proposed method, which is a very simple and fast method, is the color-histogram-based detection. A color histogram simply counts how often each possible color occurs within an image and represents the result as a n-dimensional vector, where n represents the amount of all possible colors (e.g. $n = 256$ for a 8-bit gray scale image). Thus it can be used to roughly compare the color distribution of two images. To apply the color-histogram based detection, the color-histogram of both images I and D, converted to gray-scale, is taken and compared using the cosine similarity.

The second proposed method, that considers the spatial distribution of the pixel's color values, is the color-scattering-based detection. Similar to the color-histogram, the scattering-histogram of a 8-bit gray scale image can be represented by a 256-dimensional vector. Instead of the amount of the occurring color, each element of the vector represents the average distance of all pixels with the corresponding color to the image center. Again the similarity between two scattering-histogram vectors can be measured using the cosine similarity.

Xiao et al. recommend to use the color-scattering-based detection in addition with the color-histogram-based detection. However an attacker that has the ability to choose the source image as well as the target image, might choose them in a way, that their color distribution is quite similar, which makes the attack harder to detect.

3.5 Defending Image-Scaling Attacks

Another, probably more powerful, mitigating method might be to defend an image-scaling attack at all. This especially is relevant and applicable in attack scenarios where the attacker tries to feed a manipulated image into the model at test time.

Quiring et al.[18] proposed a method that can be applied before the input image is fed into the model and thus is not dependent on the attacked model. That means that their method can be applied universal without interfering with the other processing steps of a machine learning pipeline.

The idea of this method is to reconstruct an image S' that is similar to the original source image S from the potential attack image A

by recovering the manipulated pixels. To do so, the defender needs to know about the scaling algorithm as well as the target image size. With these information the defender first can identify the pixels that mostly contribute to the down scaled image D and thus are the targeted pixels of the attacker. On a second step the defender tries to recover the attacked pixels as good as possible. Quiring et al. suggest two different strategies to recover them: a median filter based strategy and a random filter based strategy.

Selective Median Filter: This approach makes use of a selective median filter that works similar to a convolution kernel: For each identified pixel of the potential attack image A, the filter captures the surrounding area and calculates the average pixel value within the area, disregarding the currently recovered pixel as well as other potentially attacked pixels. The calculated value then is assigned to the corresponding pixel. If the window size of the recovering window is chosen high enough (i.e. $2\beta_h * 2\beta_v$ according to Quiring et al.) it can be assured that the computation can be applied robust. Quiring et al. showed that this mechanism can recover the original source image close enough so that the image scaling attack fails. However, one major drawback is its high computational cost.

Selective Random Filter: For applications that require a fast processing speed and can afford some loss in visual quality, Quiring et al. suggest to replace the selective median filter by a selective random filter: Instead of computing the average of all pixels within the recovering window, here only one pixel is randomly chosen.

Besides providing this defense method, Quiring et al. analyzed existing scaling algorithms on their ability to defend image-scaling attacks. As reported in section 3.3 there are two critical parameters that determine if an image-scaling attack can successfully be applied without being detected: the scaling ratio β as well as the kernel window σ. In other words: if σ is chosen properly in dependence on β by the scaling algorithm, the chances of a successful attack can be reduced. In fact Quiring et al. noted, that the area scaling algorithm already mitigates image-scaling attacks sufficiently for all evaluated frameworks (due to its dynamic σ), while other scaling algorithm implementations don't. One exception are the implementations of the bilinear, bicubic and lanczos algorithms of the Pillow library since they also use dynamic σ that depends on β.

3.6 Image-Scaling Attacks at Test Time

In these kind of attacks, the adversary aims to manipulate the prediction result of a machine learning algorithm at test time. Therefore a white-box attack is assumed where the attacker has full knowledge of the used scaling algorithm and the size of the down-scaled input image. Since it is shown by Xiao et al.[27] that this information also can be gathered with black-box access to the model, this attack scenario is the most realistic one.

An image-scaling attack at test time can be applied similar to common attacks using adversarial examples (e.g. [3, 10, 16]). In fact an adversarial example X can be expressed with the same equation as the attack image A of an image-scaling attack: $X = S + \Delta$, where Δ is a perturbation matrix and S the original input image. Thus it is possible to replace the adversarial X within a specific attack scenario with a corresponding attack image A. Ultimately that means, that image-scaling attacks can be used as evasion attacks and lead to the same misclassification effect on an attacked model.

However there are a two key differences to adversarial examples that have to be highlighted. The first one is, that the generation of the attack image A is completely independent from any model or data set, while adversarial examples are crafted using either a specific (surrogate) model or a specific data set [4]. Thus no information about the attacked model or its training data is needed. The attack image generation furthermore might be less complex and can easily be achieved, using the assumed knowledge about the scaling algorithm. Secondly, the attack can create an output image D that represents a perfect class instance of the attacked model, instead of an instance that is just looking harmless to a human observer. That means, that the attack also will be successful against models that have defensive mechanisms against adversarial examples.

3.7 Image-Scaling Attacks at Train Time

One major kind of attacks that aim to manipulate a machine learning model at train time, are poisoning attacks. Quiring et al.[19] analyzed how image-scaling attacks can be used to enforce poisoning attacks by hiding the visible traces, they often leave within the training data. These visible traces especially are problematic (from an attacker's view) in cases where one or even several human observers control the training data before using it (which for example might be the case for public training data sets).

In section 2.3 the work of Gu et al.[7] is mentioned as one realistic poisoning attack on street signs. This work also was examined by Quiring et al. and chosen as base for a image-scaling enforced poisoning attack. Note that Quiring et al. also investigated the synergy of image-scaling attacks and the trojan attack of Liu et al.[12]. Since the concepts are quite similar only the attack based on BadNets is presented.

The key contribution of the image-scaling attack is to hide the backdoor pattern that is applied to the training images. Instead of simply adding that pattern to a subset of training images, which would lead to visible traces, the pattern is applied to the target images of the scaling attack. For one specific training image S the target image T is chosen in a way, that it is the down-scaled version of S with the present backdoor pattern. Based on S and T an attack image A can be created and used as new training image. In that case the backdoor pattern won't be visible on the attack image and thus it also won't be visible for a human observer, unless the the down-scaled training data is observed. Once the model is trained with the compromised training data, the backdoor pattern easily can be triggered at test time.

Quiring et al. showed that this attack is as successful as the original attack, considering the original attack goals, and furthermore provides stealthiness.

4 EVALUATION

In this section a subset of the experimental results of Quiring et al.[18, 19] is presented. This covers attacks at test time as well as poisoning attack at training time. Furthermore some additional experiments based on their open source code that support their conclusions are shown.

[4] Note that adversarial examples also can be transferable among different models.

4.1 Image-Scaling Attacks at Test Time

This section covers the kind of attack that is described in section 3.6.

4.1.1 Setup & Evaluation. The setup, chosen by Quiring et al.[18] is as follows: For their evaluation, they use a pre-trained VGG19 model [22] along with the ImageNet data set [20]. By randomly choosing images from the data set, they create two subsets D_r and D_a, each consisting of 600 images. While the D_r stays unmodified and serves as reference, the second subset D_a is used for the attack. For each image of D_a a randomly selected (from the ImageNet data set) target image is assigned, so that the class of the source image and the target image differ. Furthermore D_a is divided into further subsets to evaluate different scaling ratios. As attack strategy, the strong strategy is applied. The considered downscaling algorithms are nearest-neighbor, bilinear, bicubic and area scaling of all frameworks mentioned in section 2.1.

According to Quiring et al. an image-scaling attack is successful, if the objectives $\boldsymbol{O_1}$ and $\boldsymbol{O_2}$ (equations 1, 2) both are satisfied. To evaluate $\boldsymbol{O_1}$, the predictions of the VGG19 of the scaled source images $scale(S)$ are compared to the predictions of their corresponding target images T. If the top-5 prediction of a $scale(S)$ - T pair matches, $\boldsymbol{O_1}$ is satisfied. $\boldsymbol{O_2}$ is qualitatively evaluated by a user study as well as quantitatively by the Peak Signal to Noise Ratio (PSNR) where a PSNR score $\geq$ 15dB is considered to indicate an achievement of $\boldsymbol{O_2}$. In both evaluation methods, the similarity of the source image S and the attack image A is compared.

In the following, the term robustness is used to describe the capability of an algorithm or a defensive strategy to resist or mitigate an image-scaling attack.

4.1.2 Robustness of Existing Algorithms. In order to verify if the attack can be launched successfully without any defensive mechanisms, Quiring et al. evaluated the attack on the mentioned scaling algorithms and frameworks. In their experiments they could prove the correctness of their assumptions on the critical parameters β and σ, explained in section 3.3. In fact all attacks against algorithm implementations with a fixed kernel width σ succeeded, if the scaling ratio β was chosen high enough. Only the implementation of the bilinear downscaling algorithm of the framework Pillow didn't achieve a success rate of 100%. However, algorithms that are implemented using a dynamic σ made the attack unsuccessful in terms of $\boldsymbol{O_2}$. Especially the area-scaling algorithm proved to be resistant against image-scaling attacks for all evaluated framework implementations. Depending on the scaling ratio, PSNR scores $\geq$ 25dB are achieved (for detailed results on that, we want to refer to Quiring et al. [18]). The evaluation results in terms of $\boldsymbol{O_1}$ are shown in figure 6. The results of the user study of Quiring et al. that evaluates the image-scaling attack in terms of $\boldsymbol{O_2}$ are shown in figure 7. Note that the Pillow framework is the only framework that uses a dynamic σ for its linear and bicubic scaling algorithm implementation. Furthermore the area-scaling algorithm is not listed here, because Quiring et al. spent an extra section about it in their work. For more insights on the robustness of the area-scaling algorithm we want to refer on their paper [18].

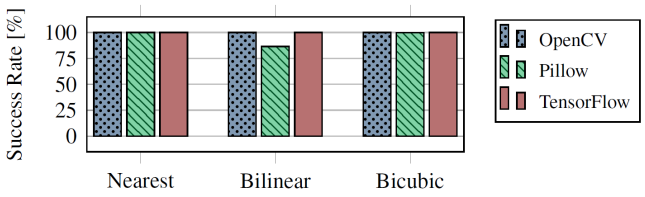

Figure 6: Evaluation results in terms of O_1. The diagram shows the success rate for the different algorithms nearest-neighbor, bilinear and bicubic of the frameworks OpenCV, Pillow and Tensorflow [18].

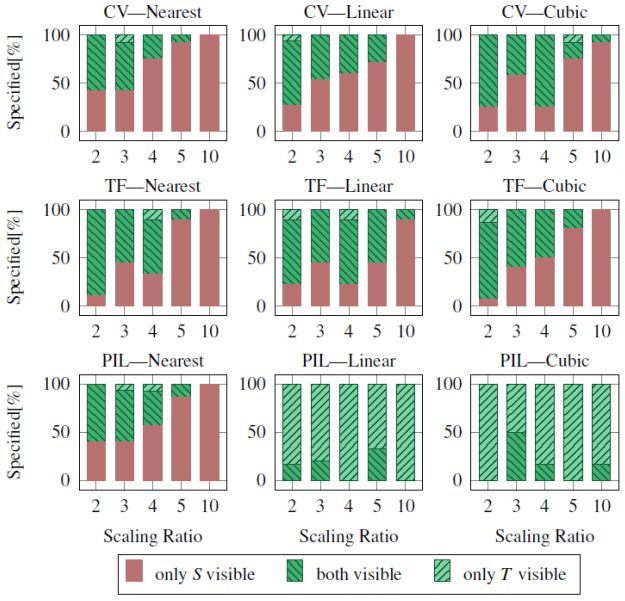

Figure 7: Evaluation results in terms of O_2. The diagram shows the user study results of Quiring et al. in respect to the algorithms nearest-neighbor, bilinear and bicubic of the frameworks OpenCV, Pillow and Tensorflow [18].

4.1.3 Robustness of Defensive Strategies. To analyze if their suggested defensive strategies, namely the median based and the random filter based recovering strategy (3.5), effectively can prevent the image-scaling attack, Quiring et al. evaluated them for all scenarios in which the attacks succeeded without any defenses. Their results show, that the suggested defense successfully can mitigate the achievement of $\boldsymbol{O_1}$ for most cases. Therefore, they evaluated the success rate of reconstructing the source image S by predicting the reconstructed images with the VGG19 model. An image is considered to be correctly reconstructed, if the the model prediction matches the prediction of the original source image S. Besides evaluating the reconstructed images that were affected by an attack, Quiring et al. investigated the effect of their defense on images that were not corrupted by an attacker. The results of their evaluation are shown in figure 8. For the median based recovering strategy a success rate of nearly 100% for all evaluated algorithms could be achieved. However, the results for the random filter based strategy

are worse. The worst reported result is a success rate of only 88.1% for the nearest-neighbor algorithm of the Pillow framework.

Library	Algorithm	Median		Random	
		Attacks	Unmod.	Attacks	Unmod.
OpenCV	Nearest	99.6%	99.0%	89.3%	89.1%
	Bilinear	100.0%	99.4%	97.7%	98.0%
	Bicubic	100.0%	99.2%	91.4%	93.4%
TF	Nearest	99.6%	99.0%	88.9%	89.1%
	Bilinear	100.0%	98.9%	97.7%	97.7%
	Bicubic	100.0%	99.4%	91.7%	92.0%
Pillow	Nearest	100.0%	99.6%	88.1%	90.4%

Figure 8: Evaluation results of the defensive strategies. The table shows the rate of correctly reconstructed images in respect to the algorithms nearest-neighbor, bilinear and bicubic of the frameworks OpenCV, Pillow and Tensorflow. The column "Unmod." shows the prediction accuracy of the evaluated model of images that were not attacked, but still modified by the defensive strategy [18].

4.2 Image-Scaling Attacks at Train Time

This section covers the kind of attack that is described in section 3.7.

4.2.1 Setup & Evaluation. The setup, chosen by Quiring et al.[19] is as follows: For their evaluation, they use the model of Carlini & Wagner [4] along with the CIFAR-10 data set [9]. In order to enable an image-scaling attack, they upscale the images from the CIFAR-10 data set to a size of 256×256. Note that the used model requires inputs of size 32×32. As attack strategy, the strong strategy is applied. The used downscaling algorithm is the bilinear algorithm with the implementation of TensorFlow.

4.2.2 Enforced Backdoor Attack. To perform the attack, Quiring et al. embed a filled black square in the lower left corner of the training images, following the concept explained in section 3.7. In order to assess their results, they compared them to a baseline that represents the results of the backdoor attack without using the image-scaling attack. It could be shown that their attack succeeded in case that at least 5% of the training data were changed while the visible traces on the manipulated data hardly could be detected by a human observer. Furthermore no drawbacks in comparison to the basic backdoor attack could be observed (see figure 9). Thus Quiring et al. proved that a backdoor attack might benefit from an image-scaling attack in a way, that it hides the backdoor pattern. For more detailed information we would like to refer to the paper of Quiring et al. [19]

4.2.3 Detection. To evaluate the visible traces of their image-scaling enforced backdoor attack on the attack image A, they used the detection methods, explained in section 3.4. Since the backdoor pattern is widely spread on the attack image A and only covers a small region on the down-scaled image $scale(A)$ the resulting histograms of a A - $scale(A)$ do not differ enough to detect any obvious dissimilarities. Thus the detection mechanisms fail for this kind of attack. Note that Quiring et al. did not evaluate any defensive strategies for this scenario.

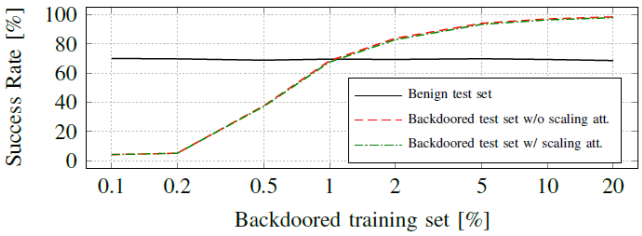

Figure 9: Evaluation results of the image-scaling enforced backdoor attack of Quiring et al. The diagram shows the success rate of the poisoning attacks with and without the combination with the image-scaling attack [19]

4.3 Further Experiments

In order to evaluate some of the results from Quiring et al. this section presents a few, rudimentary results of further experiments that are based on the open source code, provided by Quiring et al.

Since image-scaling attacks as well as their detection and defenses are meant to be independent from any potentially attacked model, the presented experiments rather focus on the image processing than on the relation to any model. Thus an attack is considered to be successful in terms of $\boldsymbol{O_1}$, if the down-scaled image visually is perceptible as the intended class instance by a human (instead of a machine learning model). $\boldsymbol{O_2}$ can be evaluated using the PSNR score.

The followings experiment scenarios were done for each four image pairs from private photos:

- **a)** The image-scaling attack is applied on the images without any defensive mechanisms.
- **b)** The image-scaling attack is applied using the median filter defense.
- **c)** The image-scaling attack is applied without any defenses to hide a backdoor pattern that can be used for a poisoning attack.

The used downscaling algorithm is the bilinear algorithm with the implementation of TensorFlow.

One example of the evaluation is shown in figure 10. Additional results are shown in the appendix A. In the following the results of the different scenarios are presented:

Image-Scaling Attack without defense: Objective $\boldsymbol{O_1}$ is achieved for every image pair. Even without any model predictions, it is clearly visible that every output image D is similar to its corresponding target image T. Considering the chosen threshold of a PSNR score $\geq$ 15dB, the objective $\boldsymbol{O_2}$ is achieved for 3 out of 4 image pairs by a lot. Even though the remaining image pair achieves $\boldsymbol{O_2}$ with a PSNR score of 19.72dB, the corresponding attack image clearly can be detected by human observation (see figure 11).

Image-Scaling Attack with defense: For all image pairs, the image-scaling attack fails in terms of objective $\boldsymbol{O_1}$. As shown in figure 10

and 11, every resulting output image D is clearly similar to its corresponding source image S.

Enforced Backdoor Attack: As the qualitative results in figure 10 and 11 show, the applied backdoor pattern can be hidden within the attack image A, while it is still visible on the resulting output image D. Note that the reported PSNR scores are even higher than the scores reported for scenario **a)**. Thus the scenario can be considered to be successful in terms of O_1 and O_2 .

Overall the experimental results showed to be consistent with the insights of Quiring et al. so that their conclusions can be confirmed for the evaluated scenarios. However, the aforementioned experiments only cover a small scope of potential evaluation scenarios and are based on a very small set of images. Thus they are not empirically representative.

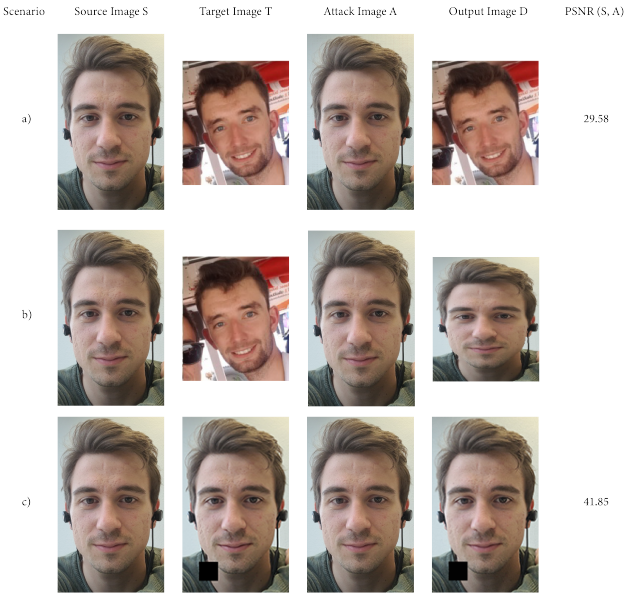

Figure 10: Evaluation results of one image pair of the scenarios a), b) and c). For each scenario, the images S, T, A and D, as well as the PSNR score are shown. Note that the attack image A of scenario b) is the attack image after being reconstructed by the median filter based recovering strategy. Furthermore no PSNR score is reported for this scenario, since the attack fails in terms of O_1.

5 CONCLUSION

In this work the new image-scaling attack, its basic concept and the root-cause that enables the attack were presented. Furthermore several ways to use such an attack to manipulate machine learning based image processing applications as well as proper detection and defense strategies were shown. The efficacy of the suggested defenses as well as the successful realization of an image-scaling attack enforced poisoning attack were proven by the presented experiments of Quiring et al. and further supported by additional experiments.

First of all it can be seen, that the new image-scaling attack enables new ways for attackers to manipulate machine learning based applications. Two major benefits of this new attack strategy is the fact, that it can be applied independent from the attacked model and the required algorithms can be processed automatically and efficiently. Another benefit is the stealthiness that is provided by the attack.

However, since the attack is applied at the preprocessing stage of a machine learning pipeline, it is way easier to analyze, than attacks that directly are applied to a model. Hence there could be developed simple detection and defensive methods that can mitigate an image-scaling attack. Especially the suggested defense by Quiring et al. proved to be successful for the evaluated attack scenarios at test time. Since poisoning attacks are mostly based on the manipulation of the training data, it is questionable, if these defenses realistically can be applied in a poisoning attack scenario.

Overall the work of Quiring et al. and Xiao et al. showed that there exists a realistic risk of image-scaling attacks for security critical applications, if no proper defenses are applied. Furthermore it is shown that the preprocessing stage of machine learning pipelines is as vulnerable as the training stage and the test stage, even if defensive strategies might easier to be identified and applied. However, on our research we didn't find any other works within the domain of computer vision about attacks on other preprocessing steps than image-scaling. This might be an interesting field for future research.

REFERENCES

[1] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion Attacks against Machine Learning at Test Time. In *Machine Learning and Knowledge Discovery in Databases*, Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný (Eds.). 387–402.

[2] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2011. Support Vector Machines Under Adversarial Label Noise. *Journal of Machine Learning Research - Proceedings Track* (2011).

[3] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial Patch. arXiv:1712.09665 https://arxiv.org/pdf/1712.09665.pdf

[4] N. Carlini and D. Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. 39–57.

[5] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. 2018. Robust Physical-World Attacks on Deep Learning Visual Classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1625–1634.

[6] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.

[7] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. 2019. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access* (2019), 47230–47244.

[8] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. arXiv:1703.04730 https://arxiv.org/pdf/1706.06083.pdf

[9] Alex Krizhevsky. 2012. Learning Multiple Layers of Features from Tiny Images. *University of Toronto* (2012).

[10] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *CoRR* (2016). arXiv:1607.02533 https://arxiv.org/pdf/1607.02533.pdf

[11] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. 2019. Certified Robustness to Adversarial Examples with Differential Privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*. 656–672.

[12] Yingqi Liu, Ma Shiqing, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning Attack on Neural Networks. In *NDSS*.

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 https://arxiv.org/pdf/1706.06083.pdf

[14] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. 2017. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. *CoRR* (2017). arXiv:1708.08689 https://arxiv.org/pdf/1708.08689.pdf

[15] Alan V. Oppenheim and Ronald W. Schafer. 2009. *Discrete-Time Signal Processing*.
[16] Nicolas Papernot, P. McDaniel, Ian J. Goodfellow, S. Jha, Z. Y. Celik, and A. Swami. 2017. Practical Black-Box Attacks against Machine Learning. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (2017).
[17] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. 2016. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *2016 IEEE Symposium on Security and Privacy (SP)*. 582–597.
[18] Erwin Quiring, David Klein, Daniel Arp, Martin Johns, and Konrad Rieck. 2020. Adversarial Preprocessing: Understanding and Preventing Image-Scaling Attacks in Machine Learning. In *Proc. of USENIX Security Symposium*.
[19] Erwin Quiring and Konrad Rieck. 2020. Backdooring and Poisoning Neural Networks with Image-Scaling Attacks. arXiv:2003.08633 https://arxiv.org/pdf/2003.08633.pdf
[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* (2015), 211–252.
[21] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 6106–6116.
[22] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).
[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *International Conference on Learning Representations* (2013).
[24] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1701–1708.
[25] Te Juin Lester Tan and Reza Shokri. 2019. Bypassing Backdoor Detection Algorithms in Deep Learning. *CoRR* (2019). arXiv:1905.13409 https://arxiv.org/pdf/1905.13409.pdf
[26] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Zhao. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. *2019 IEEE Symposium on Security and Privacy (SP)*, 707–723.
[27] Qixue Xiao, Yufei Chen, Chao Shen, Yu Chen, and Kang Li. 2019. Seeing is Not Believing: Camouflage Attacks on Image Scaling Algorithms. In *28th USENIX Security Symposium (USENIX Security 19)*. 443–460.

A ADDITIONAL EVALUATION RESULTS

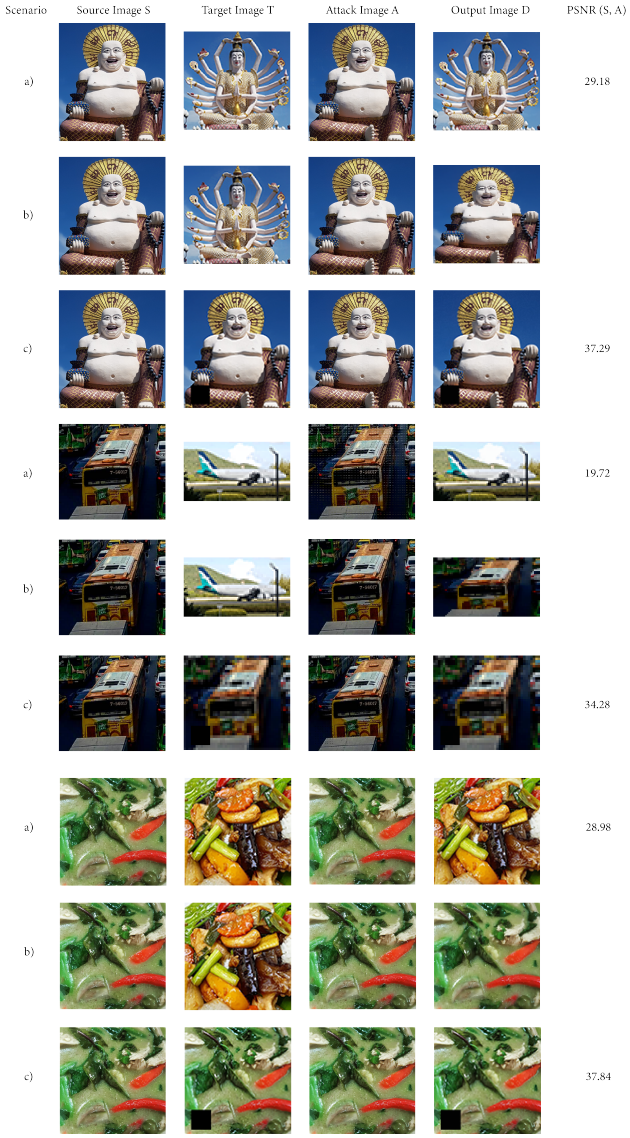

Figure 11: Additional evaluation results of the scenarios a), b) and c) mentioned in section 4.3. For each scenario, the images S, T, A and D, as well as the PSNR score are shown. Note that the attack image A of scenario b) is the attack image after being reconstructed by the median filter based recovering strategy. Furthermore no PSNR score is reported for this scenario, since the attack fails in terms of O_1.