

On the Detection of Image-Scaling Attacks in Machine Learning

Erwin Quiring
ICSI
Berkeley, United States
Ruhr University Bochum
Bochum, Germany

Andreas Müller
Ruhr University Bochum
Bochum, Germany

Konrad Rieck
TU Berlin
Berlin, Germany

ABSTRACT

Image scaling is an integral part of machine learning and computer vision systems. Unfortunately, this preprocessing step is vulnerable to so-called image-scaling attacks where an attacker makes unnoticeable changes to an image so that it becomes a new image after scaling. This opens up new ways for attackers to control the prediction or to improve poisoning and backdoor attacks. While effective techniques exist to prevent scaling attacks, their detection has not been rigorously studied yet. Consequently, it is currently not possible to reliably spot these attacks in practice.

This paper presents the first in-depth systematization and analysis of detection methods for image-scaling attacks. We identify two general detection paradigms and derive novel methods from them that are simple in design yet significantly outperform previous work. We demonstrate the efficacy of these methods in a comprehensive evaluation with all major learning platforms and scaling algorithms. First, we show that image-scaling attacks modifying the entire scaled image can be reliably detected even under an adaptive adversary. Second, we find that our methods provide strong detection performance even if only minor parts of the image are manipulated. As a result, we can introduce a novel protection layer against image-scaling attacks.

KEYWORDS

Machine Learning, Preprocessing, Adversarial Learning, Defense

ACM Reference Format:

Erwin Quiring, Andreas Müller, and Konrad Rieck. 2023. On the Detection of Image-Scaling Attacks in Machine Learning. In *Annual Computer Security Applications Conference (ACSAC '23)*, December 04–08, 2023, Austin, TX, USA. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3627106.3627134>

1 INTRODUCTION

Image scaling is a ubiquitous preprocessing step in many machine learning and computer vision systems. Before an image is fed to a learning model for inference, it is usually scaled down to fixed dimensions. For example, the popular neural networks VGG19 [22] and ResNet [8] for object recognition expect fixed inputs of 224×224 pixels. While an extensive body of research has explored vulnerabilities in learning models [1, 14], the attack surface of preprocessing has received little attention so far. An exception is recent work

on *image-scaling attacks* [17, 24], a novel class of attacks that enable an adversary to tamper with the result of the scaling process (see Figure 1). These attacks exploit that most scaling algorithms process only a minor fraction of the pixels in an image, so that a few perturbations allow for full control of its scaled version [17].

In contrast to other security threats to machine learning, image-scaling attacks are agnostic to the employed learning models. Successful attacks only require knowledge about the scaling algorithm and the target dimensions. Compared to the parameters of a neural network, these details are limited in the number of possible configurations and can also be inferred through remote queries to the model [24]. As a result, image-scaling attacks pose a notable threat to practical systems: They enable misleading classifiers without access to the learning model and allow hiding backdoor triggers or poisoning attacks in training data [15]. Figure 1 illustrates both cases. In the top row, the adversary misleads the classification by changing the entire image during scaling. In the bottom row, the adversary induces local changes in the lower left corner of the scaled image (black square). If this modification is performed on training data, it allows concealing an otherwise noticeable backdoor trigger. Hence, there is a need for effective safeguards that complement existing security mechanisms for machine learning.

Two defense strategies have been explored for addressing this threat: *prevention* and *detection*. In the first case, robust scaling algorithms or specific defense filters are applied to input images to prevent attacks. While these defenses are provably effective, as demonstrated by Quiring et al. [17], they can only prevent but not detect attacks. However, detection can be necessary in various cases: First, we can spot on-going attacks. This allows scanning image collections for attacks or identifying the adversary. Second, robust algorithms like area scaling are slower compared to vulnerable algorithms, so that a one-time check might be preferred in real-time settings. Third, detection allows protecting proprietary learning systems where components cannot be changed.



This work is licensed under a Creative Commons Attribution International 4.0 License.

ACSAC '23, December 04–08, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0886-2/23/12.
<https://doi.org/10.1145/3627106.3627134>

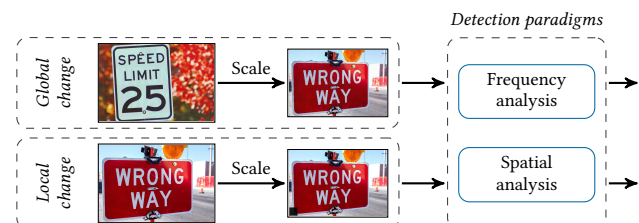


Figure 1: Image-scaling attacks: global modification (top) and local modification (bottom). Note the small box (backdoor) on the lower left that appears. Both attacks can be detected by using a frequency or spatial analysis.

The detection of image-scaling attacks, however, has not been rigorously studied yet. In this paper, we address this research gap and present the first in-depth systematization and analysis of detection methods for image-scaling attacks. We identify two general paradigms that underlie existing detection approaches: *frequency analysis* and *spatial analysis*. In the first paradigm, the detection is based on searching for conspicuous traces in the frequency spectrum. In the second paradigm, the pixels are analyzed, either by leveraging the *adversarial* modification or the remaining *clean* pixels for further analysis.

We derive novel detection methods for these paradigms that are simple yet effective in spotting image-scaling attacks. Our methods significantly outperform previous approaches, as we can reduce heuristic elements and precisely pinpoint the detectable characteristics of image-scaling attacks.

We demonstrate the efficacy of our methods in a comprehensive evaluation. In contrast to prior work, we study the detection performance with a diverse evaluation setup, including all major learning platforms, scaling algorithms, a large dataset, static & adaptive adversaries, and different attack scenarios such as global & local changes. If the entire scaled image is modified, both paradigms allow a detection rate between 99% and 100%. Our frequency approaches are particularly effective with a perfect detection rate. In the challenging scenario where only a local area is manipulated, previous methods fail while the newly derived approaches remain effective. The frequency paradigm enables a detection rate of 90%. The spatial paradigm allows detecting at least three out of four attacks. Under an adaptive attacker, however, the paradigms' effectiveness flips: The frequency paradigm—the strongest in the static attack—does not withstand an adaptive attack while the spatial analysis remains robust. As a result, we conclude that both paradigms should be used in combination to complement each other.

Contributions. In summary, our contributions are as follows:

- (1) *Systematization of detection.* We present the first systematic analysis of detection methods for image-scaling attacks where we identify two general paradigms.
- (2) *Novel approaches to detection.* Based on our analysis, we derive novel detection approaches that significantly outperform all approaches from previous work.
- (3) *Comprehensive evaluation.* We empirically investigate the performance of all detection approaches in a comprehensive evaluation that covers all major learning platforms and scaling algorithms.
- (4) *Different attack models.* Moreover, we carefully examine the limits of all detection approaches by experimenting with different attack scenarios, such as global & local changes, and with both static & adaptive attackers.

To foster further research in this area, our code is publicly available at <https://github.com/EQuiw/2023-detection-scalingattacks>.

Roadmap. We introduce image-scaling attacks in Section 2. The detection paradigms with our new methods are presented in Section 3 and the evaluation of their performance is given in Section 4. Adaptive attacks are evaluated separately in Section 5. Finally, Section 6 discusses related work and Section 7 concludes the paper.

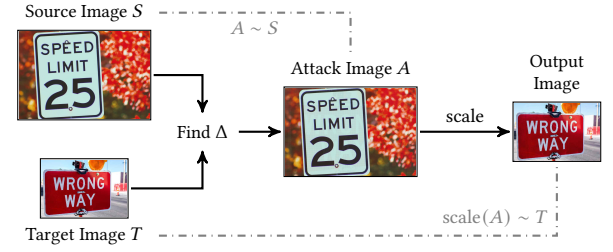


Figure 2: Principle of image-scaling attacks: The adversary finds a minimal modification Δ of S such that the modified image $A = (S + \Delta)$ still looks like S , but downscales to T .

2 BACKGROUND

We start by introducing the background on image-scaling attacks before presenting our novel detection approaches.

2.1 Preprocessing in Machine Learning

When solving tasks of computer vision, image data is typically normalized and preprocessed before features are extracted and learning models are applied. In particular, *image scaling* is a widely used preprocessing step to bring images to a normalized form. Most learning algorithms expect a fixed input size and thus images with different or larger dimensions need to be scaled. For example, the deep neural networks VGG19 [22] and ResNet [8] require inputs of 224×224 pixels. Due to this frequent preprocessing, major machine-learning frameworks directly integrate different scaling algorithms. For instance, Caffe employs the image processing library *OpenCV*, PyTorch uses the library *Pillow*, and TensorFlow has its own implementation called *tf.image*. Following prior work [24], we thus focus our analysis on the libraries *OpenCV*, *Pillow*, and *tf.image* with their respective implementations of scaling algorithms.

2.2 Image-Scaling Attacks

Interestingly, the downscaling of images leads to a considerable attack surface in learning-based systems. By carefully modifying particular pixels, it becomes possible to control the output of the scaling algorithms and thus to change the content of the scaled image. In the following, we recap the current state-of-the-art of these *image-scaling attacks* [17, 24].

Figure 2 exemplifies the principle of the attack. Given a source image S , the adversary tries to find a minimal perturbation Δ , so that the downscaling of the modified image $A = (S + \Delta)$ produces an output image, $\text{scale}(A)$, that matches the adversary's target image T . This attack can be modeled as a quadratic optimization problem:

$$\begin{aligned} \min(\|\Delta\|_2^2) \\ \text{s.t. } \|\text{scale}(S + \Delta) - T\|_\infty \leq \epsilon \text{ and } A \in R, \end{aligned} \quad (1)$$

where the interval R is the allowed pixel range, as for example, $R = [0, 255]$ for 8-bit images.

To deceive the victim, a successful attack has to fulfill two objectives: First, the scaled image, $\text{scale}(A) = \text{scale}(S + \Delta)$, needs to match the target image T , i.e., $\text{scale}(A) \sim T$. Second, the attack image should be indistinguishable from the source image, i.e., $A \sim S$. As a result, the adversary gets an attack image A that looks identical to the source S but changes to the target T after downsampling.

This attack is independent of the training data, extracted features, and employed learning model. The adversary only needs to know the used scaling algorithm and the target size of the scaled image. In practice, this knowledge might not be difficult to obtain. Publicly available deep neural networks are often re-used through transfer learning. Moreover, the number of possible scaling configurations in the libraries is limited. In some settings, the knowledge about the algorithm is not even required. OpenCV and TensorFlow, for example, are effectively using nearest scaling when the scaling factor is an (uneven) integer [17]. This simplifies an attack if the adversary can choose the size of S . Finally, even without any knowledge, the scaling setup can be deduced through black-box queries [24].

Threat Scenario. By attacking the preprocessing, the input is manipulated at the very beginning of the learning pipeline. An adversary can thus efficiently control any subsequent steps, which enables and simplifies different attack strategies. First, during training time, the attacker can conceal poisoning and backdoor attacks: The training-data modifications become visible only after downscaling [15]. This alleviates the shortcoming of attacks that leave visible artifacts in images [7, 21, 25]. The combination is useful, for example, if backdoors are used in the physical world. At training time, scaling attacks hide the trigger that can be later activated in the physical world. Second, during test time, the adversary can control the predictions of a learning model—without modifying the training data or model. The adversary uses the scaling attack, so that the downscaling leads to an image of the target class.

Note that scaling attacks allow a similar attack as adversarial examples by causing a misclassification. However, they do *not* depend on the learning model or features, since the scaling stage produces a perfect image of the target class. Scaling attacks would succeed even if neural networks were robust against adversarial examples.

Attacks can be realized with varying degrees of modification. We differentiate two scenarios to understand the detection capabilities:

- *Global modification.* The adversary chooses a target image with an arbitrary, unrelated content. The input S and target T have no relation to each other. This is the most severe attack, as any target class can be chosen.
- *Local modification.* The source and target image are identical, except for a limited area. We study backdoors as example for this category where only a small trigger is added (see Figure 9). A detection is challenging due to the small changes.

When studying the global scenario, we also consider an *overlay scenario* where the scaling attack only partially creates the target image T . Here, we blend T into the downscaled source image to create a novel target image. This scenario allows us to study an attacker who does not fully embed the target image into the scaling output.

Root Cause. Image-scaling attacks are possible because scaling algorithms do not process all pixels equally [17]. Depending on the algorithm and scaling ratio, many pixels in S have limited or even no impact on the scaled output. An adversary can therefore only modify those pixels that are considered during scaling. The resulting sparse noise is visually imperceptible, yet controls the entire output of the scaling process (see Figure 3). In this way, both attack objectives, $\text{scale}(A) \sim T$ and $A \sim S$, are fulfilled. We build our detection defenses on this understanding of scaling attacks.

Prevention Defenses. We also recap defenses that prevent image-scaling attacks. Prior work has studied two concepts: First, we can apply a robust scaling algorithm that considers all pixels with equal contribution. This directly addresses the root cause of the attack, since no pixels are ignored anymore. A possible scaling algorithm is *area scaling* that is implemented in several image processing libraries. Second, the defender can use an image-reconstruction method to repair the modified pixels from their neighborhood. To circumvent this defense, an adversary would have to modify the neighborhood as well, which leads to visible modifications. We refer the reader to the paper by Quiring et al. [17] which examines both concepts and their robustness.

While prevention mechanisms do not interfere with the typical machine-learning workflow, they have a clear disadvantage: The mechanisms cannot be applied to find out *that an attack is going on*, that is, the data is manipulated. However, this can be necessary to ensure that we can trust a specific dataset, for instance, if we use publicly available images or release our image database. Hence, there is a need to study effective *detection* methods as complementary approach to existing prevention concepts.

3 DETECTION SYSTEMATIZATION

The detection of image-scaling attacks has received little focus so far. While some heuristics have been proposed [10, 24], we still lack a general understanding on how the attacks can be effectively characterized and detected. To fill this gap, we systematize existing work by identifying two general *detection paradigms*. For each paradigm, we present the basic principle of detection and introduce own realizations that alleviate shortcomings of existing heuristics. Table 1 shows an overview of all considered detection methods.

3.1 Paradigm: Frequency Analysis

In this paradigm, the detection builds on analyzing the frequency spectrum of images. This is a common procedure in computer vision and multimedia security [20]. In this frequency representation, periodical patterns become evident that are not detectable in the spatial image domain (pixel domain) [11]. In general, a frequency spectrum is an equivalent representation of an image that describes

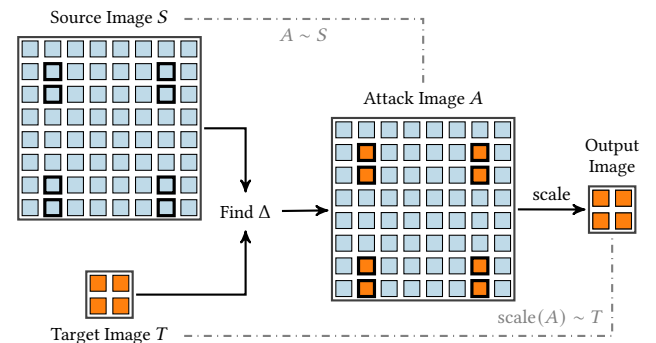


Figure 3: Simplified illustration of the root cause of image-scaling attacks: The scaling algorithm considers only a few pixels in an input S (visualized by a thicker box). The adversary has to change only these pixels to control the downscaling output. This added noise is not noticeable.

Table 1: Overview of detection methods in this paper.

Paradigm	Method	Options
Frequency	* Peak Spectrum	
	* Peak Distance	
	CSP [10]	
	* CSP-improved	
Spatial	Adversarial	* Down & Upscaling
		Down & Upscaling
		Down & Upscaling
		Maximum filter
		Minimum filter
	Clean	* Clean Filter
		* Patch-Clean Filter
		* Targeted Patch-Clean Filter

* =Proposed in this paper.

{.} denotes a set of possible options for a respective method.

the pixels by a sum of waves oscillating at different frequencies [23]. In this work, we use the 2D discrete Fourier transform (DFT) and work on the centered log-scaled magnitude spectrum (as visible e.g. in Figure 4a). Intuitively, each coefficient in this magnitude spectrum shows the impact of a particular frequency for the image. The log-scaling emphasizes smaller values. The middle of the magnitude spectrum corresponds to low frequencies while higher frequencies are located towards the corners.

As the root-cause analysis of image-scaling attacks in Section 2 highlights, the adversary injects pixels from the target image T into S in a periodic distance. This periodicity stems from the sampling process of image scaling and provides a strong indicator of image-scaling attacks. Hence, a modified image has unique, periodic peaks in its frequency spectrum. As an example, let us analyze the running example in Figure 2. The modification is not visible in the spatial domain. However, if we analyze the attack image in the frequency domain—as visible in Figure 4b—one can clearly see unusual frequency peaks.

Proposed Detection. For our analysis, we take inspiration from multimedia forensics [3, 20]. We start by noting that the defender exactly knows the potentially modified pixels in the spatial domain. This allows an exact computation of the expected peak locations in the frequency spectrum. As a result, we gain the advantage of differentiating adversarial peaks created by scaling attacks from benign peaks that images can naturally have at other locations in the spectrum, reducing the chances to detect benign peaks accidentally.

In particular, our detection approach proceeds as follows: Let (m, n) be the height and width of the source image and (m', n') the height and width of the scaled output image. The vertical scaling ratio is given as $\beta_m = \frac{m}{m'}$ while the horizontal scaling ratio is $\beta_n = \frac{n}{n'}$. The constants c_m and c_n are the index of the spectrum's middle. In the case of a scaling attack, the following binary function $\Gamma \in \{0, 1\}^{m \times n}$ shows at which frequency coefficient a peak occurs:

$$\Gamma(u, v) = \begin{cases} 1 & (u, v) = (c_m + k_1 \cdot m', c_n + k_2 \cdot n') \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

$$\text{with } -\frac{\beta_m}{2} \leq k_1 \leq \frac{\beta_m}{2}, -\frac{\beta_n}{2} \leq k_2 \leq \frac{\beta_n}{2}, k_1, k_2 \in \mathbb{N}.$$

In other words, we expect to observe peaks around each $k_1 m'$ -th and $k_2 n'$ -th position of the frequency spectrum if an image is manipulated by an image-scaling attack. In Appendix A, we derive

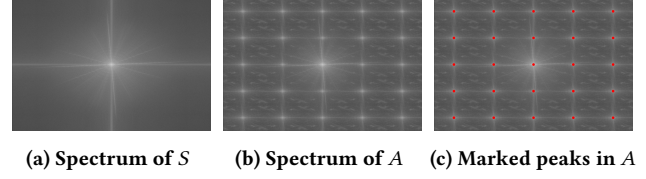


Figure 4: Frequency analysis of the example in Figure 2. Plot (a) and (b) show the frequency spectrum of the source image and attack image, respectively. Plot (c) shows the marked peaks with Equation 2.

Equation 2. To provide some intuition, Figure 4c marks the expected peaks on the frequency spectrum of our running example by using Equation 2. One can see that the observed and expected peaks match exactly. Equipped with the ability to predict the expected peaks, we propose two detection strategies.

Peak Spectrum. We extract the frequencies in a square window centered around all expected peak locations. The window's half length is w . We omit the center (c_m, c_n) being present in any spectrum. Figure 5a illustrates the resulting windows. Next, we average all window values and calculate the percentile rank of this value relative to the whole frequency spectrum. This sets the peak frequencies into relation to the entire frequency spectrum. In case of an attack, the peak frequencies outshine the entire spectrum, so that attack images get higher percentile ranks than benign images.

Peak Distance. We divide the spectrum into excerpts for each peak as Figure 5b visualizes. We discard the center of the spectrum again. For each excerpt, we extract the maximum peak and calculate the distance between this peak and the expected peak. We then average all measured distances. In case of an attack, this average is expected to be small.

Previous Approaches. Kim et al. [10] propose a frequency analysis, named *CSP*, where the underlying idea is to count peaks at arbitrary positions in the spectrum. Based on their evaluation, they assume an attack if the spectrum contains more than one peak. Our evaluation shows that this approach is ineffective. Benign images can naturally have peaks, too. We show two such examples from ImageNet in Appendix B. To test if the method works without the fixed threshold of one, we evaluate an own adjustment, named *CSP-improved*, where we derive the threshold from the data. Still, this adjustment cannot compete with our methods. The defender should rather use the advantage of having precise knowledge about the expected peak locations—as our proposed methods do.

3.2 Paradigm: Spatial Analysis

In this paradigm, the analysis is done in the spatial domain. As scaling algorithms and scaling attacks operate here, this domain naturally provides the advantage of knowing which pixels are considered by scaling algorithms and hence which are possibly modified. We identify two variants of this paradigm: A defense leverages either the *adversarial* modification or the *clean* pixels in A for detection. In the following, we examine each group in more detail.

3.2.1 Adversarial-Signal Driven. The concept here is to amplify the sparse, adversarial modifications in A and then to compare the amplified image A' with A . We have two sub-groups here.

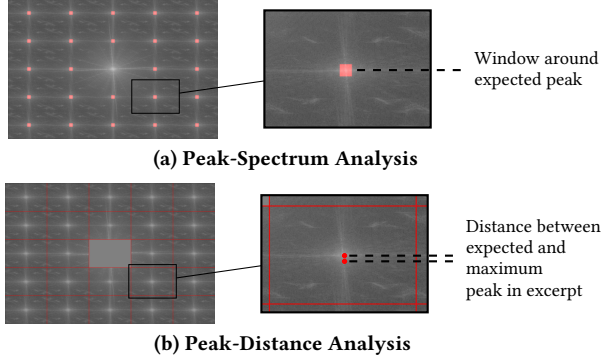


Figure 5: Frequency-based detection approaches. Plot (a): Windows around expected peaks are extracted and their frequencies are compared in relation to the overall spectrum. Plot (b): Spectrum is divided into excerpts and the distance between expected and maximum peak for each excerpt is computed.

Down-and-Upscaling. We can exploit that downscaling and upscaling form antagonists in the spatial domain. That is, the scaling of A leads to an output D which corresponds to T . By upscaling D back to the original size, $A' = \text{upscale}(\text{downscale}(A))$, we can compare this version with A . The upscaling strengthens the signal embedded by the adversary and renders it detectable through comparison with the original image. Note that the comparison function needs to account for this setup, as we describe later.

In principle, we do not need to upscale the image and could directly compare A with D . In our preliminary experiments, however, we found that we can achieve better results if we upscale D to the resolution of A and conduct the comparison on the same dimensions. Moreover, some image comparison methods require the same size for the inputs.

Proposed Detection. A variety of methods exists for comparing images that would be applicable here. We find that a simple PSNR computation is particularly effective. Peak signal-to-noise ratio (PSNR) is a widely used pixel-based comparison method that returns the normalized mean-squared error between two images. Formally, it is defined as

$$\text{PSNR}(A, A') = 20 \log_{10}(I) - 10 \log_{10}(\text{MSE}(A, A')). \quad (3)$$

The constant I is the maximum of the pixel range, such as 255 for 8-bit images. The higher the PSNR value is, the more two images match. PSNR can be computed efficiently and thus provides a perfect basis for designing a detection method using down- and upscaling.

Previous Approaches. Xiao et al. [24] propose two alternative comparison options. First, the *intensity histograms* of A and of A' are extracted and compared. Second, in the *color-scattering* method, the average distance to the image center over all pixels with the same intensity is computed. By doing this for each intensity value in A and A' , respectively, we obtain two vectors that can be compared. Moreover, Kim et al. [10] propose two measures that directly compare A and A' : the mean squared error (MSE) and the SSIM index. While the first is simply comparing A and A' pixel-wise, the latter aggregates information about the luminance, contrast, and structure between A and A' [9].

In our evaluation, the simple PSNR and SSIM outperform the other measures. Unlike suggested by prior work [10], MSE is not superior to PSNR. We assume that this conclusion is the result of a subtle implementation mistake in the evaluation metric where an integer overflow occurs. In fact, the PSNR and MSE are directly related to each other (see Equation 3). Although MSE and PSNR have the same detection performance, the PSNR score is easier to interpret. It typically lies between 0 dB and 60 dB in our evaluations. Moreover, in terms of security, a pixel-wise measure such as the PSNR is particularly robust against adaptive attacks compared to aggregation-based measures such as the histogram or color-scattering. No information are lost that could be exploited [15].

Amplifying Filter. As alternative to down-and-upscaling, Kim et al. [10] propose two methods, a minimum filter and a maximum filter, to strengthen the adversarial signal. Each filter is applied on A by iterating over the image and replacing each pixel by the minimum/maximum of its neighborhood. The outcome A' is compared with A using MSE or SSIM. This filtering strategy aims at amplifying the periodic modification. However, as the CSP approach, this defense overlooks the attack's root cause and misses the advantage of knowing the pixels used by scaling algorithms. A targeted filtering is possible which motivates our next group of defenses.

3.2.2 Clean-Signal Driven. Another concept of the spatial paradigm is to leverage the clean pixels that a scaling attack needs to leave to keep the attack imperceptible. We can use these pixels to clean A and to compare this result with the initial version. This principle has not been studied before. It allows us to design specialized methods for the global and local attack scenario, which are presented in the following.

Proposed Global Detection. To obtain a cleansed version of A , we adopt the idea of reconstructing the modified pixels using *prevention filters* [17]. In particular, we design two methods that are based on a *selective median filter* and a *selective random filter*, respectively. Both filters repair each pixel considered by a scaling algorithm by using the neighborhood of the particular pixel. In terms of security, the reconstruction methods have the advantage of addressing the root cause of scaling attacks. They also show a strong robustness against adaptive attacks, since adversaries have to modify the neighborhood as well to bypass both filters, making the attack clearly visible [17]. This security robustness also transfers to our detection defense.

For detection, we propose the following simple *clean filter* approach: Equipped with a prevention filter $\mathcal{V}$, we can get a cleaned version $A' = \mathcal{V}(A)$ which is compared with the initial input A . We evaluate the PSNR and the SSIM measure to compare A and A' .

Note that we also tested to compare D and D' where D' is the downsampled version of A' . Hence, D shows the new adversarial content, while D' shows the original content. The detection results correspond to the clean-filter method and are thus omitted.

Proposed Local Detection. A direct application of the previous approach to the local attack scenario is not promising. As A and A' are here similar for the majority of pixels, a global comparison cannot sufficiently capture the difference. Instead, we propose to divide the images into *patches* and compare each patch individually. We propose two variants.

Patch-Clean Filter. We create D and D' and divide them into L patches, respectively. We compute the PSNR between each pair of patch, $v_i = \text{PSNR}(D_i, D'_i) \forall i = 1, \dots, L$. The final detection score is given as $|\text{mean}(\{v_i\}) - \min(\{v_i\})|$. Appendix C presents the method in more detail. Note that we work on downscaled images here. We find that the smaller image size reduces the number of possible regions which improves the detection rate.

Targeted Patch-Clean Filter. As alternative, we analyze the unscaled images. However, even with patches, the scaling pixels—and thus modified pixels—are in the minority, making it difficult to detect a difference with and without scaling attack. Hence, we propose a more targeted variant and only examine the scaling pixels in each patch. Let L denote the number of patches, A_i & A'_i the respective patches, and Ψ the scaling-pixel selection, we compute

$$u_i = |\Psi(A'_i) - \Psi(A_i)| \quad \forall i = 1, \dots, L. \quad (4)$$

We get the q -th quantile of each vector u_i , denoted as $q(u_i)$, and calculate $|\max(\{q(u_i)\}) - \text{mean}(\{q(u_i)\})|$ as detection score. This comparison allows us to identify an unusual difference in a local area. Note that the choice of q is a hyperparameter of the method.

Remark. We also tested the down-and-upscaling concept with patches. However, $A' = \text{upscale}(\text{downscale}(A))$ contains a stronger noise signal due to down-and-upscaling. While this is tolerable in the global scenario, it distorts the comparison with small patches.

4 EVALUATION

We proceed with an empirical evaluation of the different detection methods. To obtain a comprehensive view, we study both non-adaptive and adaptive adversaries. In this section, we consider the non-adaptive case with regular attacks. Then, in Section 5, we investigate the best approaches against an adaptive adversary.

4.1 Evaluation Setup

Our evaluation follows the common design of experiments on image-scaling attacks [17, 24]. We evaluate the detection of attacks against popular scaling algorithms that are vulnerable to scaling attacks [17]. In particular, we consider nearest-neighbor, bilinear, and bicubic scaling from the libraries OpenCV and tf.image (TensorFlow), and nearest-neighbor scaling from the library Pillow. We omit Lanczos scaling which is comparable to bicubic scaling [17].

For the attacks, we adopt the evaluation setup by Quiring et al. [17], where the source and target for an attack are randomly drawn from a collection of images. As dataset for this sampling, we use photos from ImageNet [18]. Compared to other datasets from computer vision, like CIFAR or CelebA, ImageNet contains significantly larger images, which is a key requirement for constructing successful scaling attacks in practice. Moreover, the dataset is very diverse, covering various image sizes and contents, such as faces, animals, persons, objects, and landscapes.

As learning model, we use a pre-trained VGG19 model [22] which is a standard benchmark in computer vision. The target size for scaling is thus $224 \times 224 \times 3$ pixels. Note that we just use this one architecture in our experiments, since scaling attacks do not depend on the learning model's architecture. They change the input to the model. Only the input size of the model is relevant, so that we make sure to use varying scaling ratios as described later. Finally, we

consider a global and local modification scenario. Both require a slightly different setup that we present in the following.

Global Modification. To obtain attack images in this scenario, we randomly sample images from ImageNet and create 1,000 source–target image pairs. We ensure that the pairs have varying scaling ratios to avoid artifacts that may arise from a fixed ratio. We check that each target is unrelated to its source image by requiring different classes and predictions for each pair. After conducting the attacks, we keep only those images that are successful regarding both attack objectives, i.e., $\text{scale}(A) \sim T$ and $A \sim S$. To this end, we use the same methodology as Quiring et al. [17]. The number of successful images varies across each combination of scaling algorithm and library, so that we choose the highest possible number of images across all setups. This leads to 585 attack images for each combination of scaling algorithm and library. As unmodified reference set, we additionally select 585 further images from ImageNet. They are used to evaluate the detection of benign inputs.

Local Modification. Here, we implement the *BadNets* backdoor attack [7] as a representative form for local triggers that are also used in recent works [e.g., 19]. We use the same 585 source images as in the previous scenario. Yet, we scale each source image to a size of $224 \times 224 \times 3$ pixels and add a small, bounded backdoor pattern to create its respective target image (see Figure 9 in the Appendix). Finally, we conduct scaling attacks on these image pairs. As reference set, we use the same 585 benign images as before. We also verify that the so-created backdoors are effective (see Appendix D).

Evaluation Measures. To evaluate the detection performance of the considered detection methods, we equally split each attack and reference dataset into a training and test partition, respectively. The training set is used to calibrate a threshold for each detection method so that the false positive rate is 1%. This threshold is then used to evaluate the detection performance on the test set. For some detection methods, we need to decide on specific hyperparameters, such as the window width in the peak-spectrum analysis. For this calibration, we split the training set further and create a validation set. This set is used to find the optimal parameters in a grid search. We instantiate the detection methods with the best parameters (see Appendix E) and report their performance on the test set.

4.2 Global Modification Scenario

We start with the detection when source and target image are arbitrary. This is the most severe setting as an adversary can freely choose the target class of the prediction.

Results. Table 2 shows the performance for all detection methods, sorted in descending order with respect to the average accuracy. Both paradigms allow detecting scaling attacks if the entire scaled image is modified. A perfect detection is possible with our proposed peak-distance frequency analysis. However, not all methods are effective, such as the previously proposed CSP approach with an accuracy of 50%.

Analysis. A closer look on this experiment provides important insights. First, our proposed frequency methods—exploiting the known peak locations—outperform all other detection methods. Their accuracy is 100% and 99.90%. On the other hand, the CSP

Method	Option	AvgAcc	StdAcc	AvgFPR
* Peak Distance		100.00	00.00	00.00
* Peak Spectrum		99.90	00.26	00.00
* Clean Filter	Median filter, SSIM	99.80	00.21	00.05
Down & Up	Histogram	98.85	01.09	00.49
* Down & Up	PSNR	98.54	01.48	00.49
Down & Up	MSE	98.54	01.48	00.49
* Clean Filter	Random filter, SSIM	98.54	01.51	00.63
* Targeted Patch-Clean Filter		96.54	04.16	00.88
* Clean Filter	Median filter, PSNR	94.30	06.06	01.07
Maximum Filter	SSIM	87.03	03.53	01.80
Minimum Filter	SSIM	85.15	04.11	01.66
* Clean Filter	Random filter, PSNR	81.52	08.59	01.17
* CSP	Improved	76.72	06.19	01.32
Down & Up	SSIM	76.21	12.95	01.46
Minimum Filter	MSE	64.65	03.15	01.61
Maximum Filter	MSE	60.63	01.39	01.66
Down & Up	Color-scattering	55.88	03.70	00.83
CSP	Original	50.00	00.00	00.00
* Patch-Clean Filter		49.95	00.08	00.10

Table 2: Detection performance [%] in global scenario in terms of accuracy (average + standard dev.) and false positives (average) over all scaling algorithms and libraries. * Proposed in this paper.

approach [10] that also analyzes the frequency spectrum is comparable to random guessing. Second, the choice of the comparison function in the spatial paradigm is important. SSIM is preferable for the clean, minimum, and maximum filter. PSNR provides a higher detection accuracy with the down-and-upscaling approach. There is no difference between MSE and PSNR in our experiments. Third, patch-based defenses designed for the local scenario are partly applicable in the global case. The targeted patch-clean filter has a detection rate of 96.54%.

Ensemble of Detection Methods. Next, we combine multiple methods as ensemble to increase the diversity of detection patterns. We test three ensembles: We use the best method from each paradigm, and we use the $K = \{3, 4\}$ best methods (irrespective of the paradigm). Note that we choose the methods based on the results on the training dataset and not Table 2.

The first ensemble consists of peak distance and the clean filter (with Median, SSIM). The ensemble with $K = 3$ consists of the first three entries in Table 2. The ensemble with $K = 4$ uses Down & Up with PSNR as 4th method in addition. We use two voting strategies: We report an attack if the *majority* of methods or if at least one method flags an input, that is, *one winner takes all*.

Table 3 shows the performance. The K-best ensemble with $K = 4$ and majority voting achieves an accuracy of 100% and a false-positive rate of 0%. In terms of security, an ensemble thus allows a perfect detection rate while increasing the difficulty for an adaptive attack, as different paradigms and methods have to be circumvented.

Overlay Scenario. In addition, we study the variation where a scaling attack is used to embed only a low-opacity version of T into the downscaled output. More specifically, the novel target image is given as $T' = \alpha \cdot T + (1 - \alpha) \cdot \text{scale}(S)$. The parameter α denotes the blending factor. In our experiments, we set $\alpha = 0.3$ to test a challenging case with only a very small embedding of T .

Table 4 shows the performance. Our frequency approaches still provide a detection rate close to 100%. Even with a very low blending factor, periodic peaks are inevitably created in the frequency

Ensemble	Majority			One Winner Takes All		
	Acc	TPR	FPR	Acc	TPR	FPR
Best Per Paradigm	99.98 ± 0.06	100.00 ± 0.00	0.05 ± 0.13	—	—	—
K Best ($K=3$)	99.98 ± 0.06	99.95 ± 0.13	0.00 ± 0.00	99.98 ± 0.06	100.00 ± 0.00	0.05 ± 0.13
K Best ($K=4$)	100.00 ± 0.00	100.00 ± 0.00	0.00 ± 0.00	99.73 ± 0.17	100.00 ± 0.00	0.54 ± 0.33

Table 3: Ensemble in global scenario. Each cell shows the average $\pm$ standard deviation over all scaling libraries and algorithms. With two paradigms, both voting methods for “best per paradigm” have identical results, so that we omit the 2nd voting method in this case.

Method	Option	AvgAcc	StdAcc
* Peak Distance		99.73	00.24
* Peak Spectrum		99.71	00.13
* Clean Filter	Median filter, SSIM	93.47	05.12
Down & Up	Histogram	92.54	06.91
* Clean Filter	Random filter, SSIM	78.30	08.99

Table 4: Detection accuracy in overlay scenario. Only the effective approaches with AvgAcc > 60% are shown.

spectrum. On the contrary, the spatial paradigm is more affected by the overlay scenario. A pixel-based comparison is more difficult, as T is embedded more weakly.

Comparing Scaling Algorithms and Libraries. We have presented aggregated results over all scaling algorithms and libraries so far. In Appendix F, we analyze the individual detection per scaling algorithm and library. We find no significant difference between the libraries. Moreover, the frequency methods and the leading methods in the spatial paradigm do not depend on the scaling algorithm.

Summary. The effective detection of image-scaling attacks is possible with both paradigms. The frequency paradigm allows for a perfect detection rate without any false positives in our experiments. Even in the overlay scenario, it enables spotting all attacks with high accuracy. We conclude that scaling attacks can be efficiently detected if an adversary performs a global modification.

4.3 Local Modification Scenario

In our next experiment, we test the detection performance in the challenging scenario where an adversary applies an image-scaling attack only to a small area of an image.

Results. Table 5 shows the results for all detection methods. Only our proposed frequency methods can effectively detect local scaling attacks with an average accuracy of 80.98% and 89.81%. Our patch-based approaches achieve an acceptable detection rate of 76.38% and 75.72%. The other methods are close to random guessing.

Analysis. Most methods of the spatial paradigm are not effective anymore. The reason is that scaling attacks change only a small area. Thus, most of the compared areas still correspond to the original image, making a comparison difficult. On the contrary, the frequency approach is still effective, because even a limited, modified image area causes periodic frequency peaks. However, the impact on the

Method	Option	AvgAcc	StdAcc	AvgFPR
* Peak Spectrum		89.81	04.75	01.07
* Peak Distance		80.98	02.83	01.17
* Targeted Patch-Clean Filter		76.38	12.35	00.93
* Patch-Clean Filter		75.72	04.91	01.71
Down & Up	Color-scattering	51.56	02.08	00.83
Down & Up	Histogram	50.44	01.13	00.73
* Clean Filter	Median filter, SSIM	50.15	00.18	00.59
* Clean Filter	Random filter, SSIM	50.02	00.06	00.49
CSP	Original	50.00	00.00	00.00
Maximum Filter	SSIM	49.85	00.06	01.02
Minimum Filter	SSIM	49.80	00.51	01.46
* Clean Filter	Median filter, PSNR	49.78	00.16	00.98
Down & Up	MSE	49.68	00.35	00.83
* Down & Up	PSNR	49.68	00.35	00.83
* Clean Filter	Random filter, PSNR	49.68	00.12	01.02
* CSP	Improved	49.63	00.32	00.73
Minimum Filter	MSE	49.61	00.26	01.41
Maximum Filter	MSE	49.59	00.19	01.17
Down & Up	SSIM	49.51	00.32	01.12

Table 5: Detection performance [%] in local scenario in terms of accuracy (average + standard dev.) and false positives (average) over all scaling algorithms and libraries. * Proposed in this paper.

frequency spectrum is weaker, so that the performance decreases compared to attacks modifying the full image.

Ensemble of Detection Methods. We also study ensembles of detection methods with local modifications. For the best-per-paradigm ensemble, we consider peak spectrum and the targeted patch-clean filter. The K -best ensembles consist of the first K entries in Table 5. We consider $K = \{3, 4\}$, as we have only four effective approaches (see Table 5).

Table 6 shows the performance. With majority voting and K -best ensembles, the false-positive rate can be reduced to 0% by sacrificing some accuracy. In turn, the combination of majority voting and best-per-paradigm slightly improves the accuracy, but with more false positives. Note, however, that the peak-spectrum method alone would achieve an average accuracy of 91.76% at a comparable false-positive rate of 1.85%. Taken together, no ensemble outperforms the individual methods in any aspect. In terms of performance, the ensemble seems not directly beneficial. Yet, it provides benefits against adaptive attacks as we will see in Section 5.

Comparing Scaling Algorithms and Libraries. In Appendix F, we analyze the individual detection performance. The library has no effect, but we observe a duality with more advanced scaling algorithms: Frequency methods become better while clean-signal methods become worse. With local modifications, a defender should therefore choose the detection method based on the scaling algorithm.

Varying Backdoors. Next, we study the detection performance with more backdoors that differ in type and location. In addition to our previously used backdoor (a black *box* in the lower left corner), we examine (i) a black *circle* embedded in the upper right corner, and (ii) a *rainbow*-like box [13] embedded in the lower left corner. These backdoors allow us to study the impact of shape, filling, and location. The box and rainbow patterns, for instance, are common patterns in backdoor attacks [13, 15, 19]. The first two columns in Figure 9 in the Appendix show examples for all backdoor types.

Ensemble	Majority			One Winner Takes All		
	Acc	TPR	FPR	Acc	TPR	FPR
Best Per Paradigm	92.00 ± 1.57	86.01 ± 4.00	2.00 ± 1.19	—	—	—
K Best ($K=3$)	84.84 ± 1.96	69.67 ± 3.92	0.00 ± 0.00	91.66 ± 1.72	86.49 ± 3.52	3.17 ± 0.43
K Best ($K=4$)	86.35 ± 1.45	72.70 ± 2.90	0.00 ± 0.00	91.10 ± 1.96	87.08 ± 2.81	4.88 ± 1.38

Table 6: Ensemble in local scenario. Each cell shows the average and standard deviation over all scaling libraries and algorithms.

Method	Box	Circle	Rainbow
Peak Spectrum	89.81 $\pm$ 04.75	93.03 $\pm$ 02.82	78.77 $\pm$ 05.85
Peak Distance	80.98 $\pm$ 02.83	87.79 $\pm$ 02.22	68.36 $\pm$ 02.81
Targeted Patch-Clean Filter	76.38 $\pm$ 12.35	70.55 $\pm$ 14.26	67.89 $\pm$ 22.53
Patch-Clean Filter	75.72 $\pm$ 04.91	79.50 $\pm$ 05.93	68.50 $\pm$ 06.42

Table 7: Detection performance with varying backdoors (accuracy $\pm$ standard deviation). Only the four effective detection methods are shown.

Table 7 shows the performance. While the box and circle are well detectable, the rainbow-backdoor is only detected in 3 of 4 cases. We attribute this to the mixed filling, which causes weaker peaks in the frequency spectrum. Moreover, for three methods, the circle backdoor is slightly better detectable than the box. This is because the circle is larger, consuming 305 pixels compared to 225 pixels by the box. We verified this by reducing the circle size. The detection rates then become similar to the box backdoor. In summary, we study multiple backdoor types with scaling attacks. Our approaches detect them, yet the performance relies on the backdoor type.

Varying Backdoors At Train-Test Time. So far, the training dataset for calibrating the detection methods have had the same backdoor as the test dataset. In practice, this might not be realistic. As a remedy, we study an additional scenario where the training dataset is calibrated on a different backdoor than the one used in the test dataset. We study the same three backdoor types as before.

Table 8 shows the detection rate of the peak-spectrum method as a matrix for all possible backdoor combinations during train and test time. Using different backdoors has no impact on the detection. The values in each column in Table 8 are almost identical, irrespective of the backdoor at training time. The other detection methods have the same behavior and are reported in Table 14 in Appendix G. We further describe the reasons for these results in Appendix G.

Overall, we conclude that we can also detect scaling attacks if the backdoor—used to calibrate the detection—is different to the finally used backdoor from the adversary. Only the backdoor type itself, such as a circle or rainbow-like backdoor, affects the detection.

Summary. Even in the challenging scenario where only a local image area is manipulated, a reliable detection is possible. However, only four approaches are effective: our frequency approaches based on peak spectrum and peak distance, as well as both patch-clean filters. Finally, our results show that the defender does not need to have an exact knowledge of the employed backdoor.

Backdoor Train-Time	Backdoor Test-Time		
	Box	Circle	Rainbow
Box	89.81 $\pm$ 04.75	93.03 $\pm$ 02.80	78.77 $\pm$ 05.85
Circle	89.66 $\pm$ 04.94	93.03 $\pm$ 02.82	78.35 $\pm$ 05.67
Rainbow	89.76 $\pm$ 04.81	93.00 $\pm$ 02.78	78.77 $\pm$ 05.85

Table 8: Performance of peak spectrum with varying backdoors at train–test time (accuracy $\pm$ standard dev.). The rows show the used backdoor at training time, the columns the backdoor at test time.

5 ADAPTIVE ATTACKS

Finally, we study an adaptive attacker who is aware of the deployed detection method and adjusts the attack strategy accordingly. We examine the global and the local modification scenario again, but limit our analysis to only the successful methods from Section 4. Note that we now have to analyze the detection rate *and* both goals of a scaling attack. Let $\tilde{A}$ be the adaptive version of A , an attack has to fulfill: (O1) $\text{scale}(\tilde{A}) \sim T$, and (O2) $\tilde{A} \sim S$. The second goal O2 is evaluated with the PSNR. The first goal O1 is tested by computing the *attack success rate* (ASR). In the global scenario, we define the ASR as the ratio of matches in the top-5 predictions from VGG19 between $\tilde{A}$ and A . In the local scenario with backdoors, we define the ASR as the ratio of successful matches of the backdoor’s target class in the top-5 predictions. Appendix D provides more information on the finetuning setup to measure the ASR with backdoors.

5.1 Attacking the Frequency Paradigm

To mislead a peak analysis, an adaptive attacker has to hide the periodic traces caused by a scaling attack. In the following, we analyze different methods to achieve this.

Suppressing Frequency. We begin with a targeted attack against our peak-spectrum analysis. We shortly introduce the concept before presenting the empirical results.

Approach. The idea is to suppress the frequency spectrum in the window that is used by the defense. In particular, let F denote the frequency spectrum of the attack image and $\omega \in W$ each window used by the spectrum analysis. The spectrum is then manipulated as follows:

$$F(A)[\omega] = f_s \cdot F(A)[\omega] \quad \forall \omega \in W, \quad (5)$$

where f_s is a parameter to control the reduction and $[\cdot]$ selects a subset of frequencies. Figure 6 illustrates the adaptive attack by setting all frequencies in the window ω to zero.

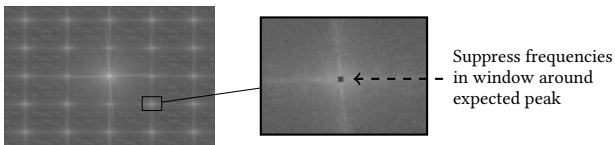


Figure 6: Adaptive attack by suppressing frequencies with $f_s = 0.0$. Note that such a strong reduction is often not needed.

Attack	Option	Detection		Attack O1	Attack O2	
		AvgAcc	StdAcc	ASR	AvgPSNR	StdPSNR
Global	$f_s = 1.0$	99.85	00.14	100.00	23.47	03.27
	$f_s = 0.8$	99.59	00.45	99.56	22.85	02.98
	$f_s = 0.6$	98.06	01.49	97.39	21.00	03.10
	$f_s = 0.4$	92.14	04.53	92.01	19.34	03.23
	$f_s = 0.2$	63.10	08.50	83.71	18.00	03.25
	$f_s = 0.0$	33.37	00.00	72.63	16.95	03.21
Local	$f_s = 1.0$	87.05	06.35	77.64	43.74	05.91
	$f_s = 0.8$	82.23	07.56	77.51	41.25	07.37
	$f_s = 0.6$	75.06	08.33	77.34	38.61	08.19
	$f_s = 0.4$	64.56	08.46	77.44	36.34	08.63
	$f_s = 0.2$	48.50	06.49	77.42	34.41	08.89
	$f_s = 0.0$	33.06	00.32	77.50	32.75	08.99

Table 9: Adaptive attack against peak-spectrum approach by suppressing frequency peaks (Acc. and ASR in [%], and PSNR in [dB]).

Results. This adaptive attack has a strong impact on the detection performance, as Table 9 shows. The average detection accuracy notably decreases with a smaller value of f_s . Note that using $f_s = 0.0$ leads to a too strong frequency reduction, so that the detection threshold could be simply inverted, increasing the accuracy, for instance, to 66.63% ($100\% - 33.37\%$) in the global case. To provide more intuition on the resulting image quality, Figure 10 in the Appendix shows an exemplary image from our evaluation for varying values of f_s .

Regarding the goals O1 and O2, the adaptive attack has only an impact in the global scenario. Here, the ASR decreases and the attack images loose brightness and contrast. The output images become a mixture between the original and novel content. On the contrary, in the local scenario, f_s has no significant impact on the ASR and the visual quality. In summary, an adaptive attacker can notably decrease the detection performance. In the global scenario, however, the attacker has to sacrifice the goals O1 and O2 to some extent, which limits the impact of the attack.

Add Frequency Peak. We continue with an attack against the peak-distance analysis. Again, we shortly introduce the concept before presenting results.

Approach. The idea is to insert an additional peak in each excerpt so that the distance between the expected and maximum peak increases. Let $p \in P$ be the frequency location for adding a peak in each excerpt, the spectrum is modified as follows:

$$F(A)[p] = f_a^{-1} \cdot \max(F(A)) \quad \forall p \in P, \quad (6)$$

where f_a controls the strength of the added peak. For simplicity, the peak locations $p \in P$ are the corners of each excerpt. Figure 7 illustrates the adaptive attack. This frequency modification is hard to perceive in the whole frequency spectrum, as only very nuanced changes are performed.

Results. Table 10 shows the results of the adaptive attack. The peak addition has a strong impact on the detection performance. Already $f_a = 75$ notably decreases the performance in the global and the local modification scenario. At the same time, the peak addition has only a minor effect on the goal O1. To better explain the effect on O2, Figure 11 in the Appendix shows an example from our evaluation. As f_a decreases and the peaks become stronger, the images loose brightness and contrast. Still, the image content

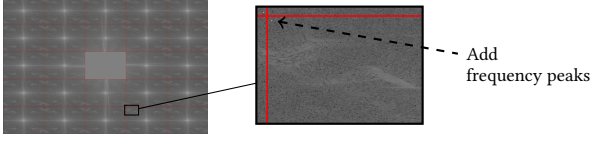


Figure 7: Adaptive attack by adding peaks with $f_a = 25$. Note that our evaluation shows that such a strong addition is often not necessary.

Attack	Option	Detection		Attack O1	Attack O2	
		AvgAcc	StdAcc	ASR	AvgPSNR	StdPSNR
Global	$f_a = 75$	61.88	02.92	99.32	19.39	02.62
	$f_a = 50$	50.29	03.29	97.70	18.23	02.73
	$f_a = 25$	37.59	02.00	90.11	16.22	02.72
Local	$f_a = 75$	33.03	00.26	82.73	22.83	05.00
	$f_a = 50$	33.03	00.26	84.49	20.54	04.54
	$f_a = 25$	32.98	00.31	88.64	17.25	03.67

Table 10: Adaptive attack against the peak-distance approach by adding frequency peaks (Acc. and ASR in [%], and PSNR in [dB]).

remains intact—even at a high modification factor with $f_a = 25$. We thus conclude that O2 is also partly achieved. All in all, we identify a suitable range of $50 \leq f_a \leq 75$ where the peak-distance approach does not detect an attack and both goals of a scaling-attack are satisfied.

JPEG Compression. As a baseline, we consider a generic and simple counterattack by examining JPEG compression. It is known to have an impact on resampling detectors in multimedia forensics [11], which analyze the frequency spectrum for peaks similar to our approach. Appendix H presents the results. Compression only affects the local scenario. However, the effect on the detection performance is smaller compared to our previous two targeted adaptive attacks.

5.2 Attacking the Spatial Paradigm

Lastly, we discuss the attack surface of the other paradigm. For strengthening approaches based on down- and upscaling, an adaptive attack should not be possible. Upscaling algorithms do not suffer from the root cause that enables scaling attacks in the down-scaling case. Upscaling algorithms usually use each pixel multiple times to compute the larger image, so that an attacker cannot hide a new signal. We verified the implementation of the imaging libraries OpenCV, Pillow, and tf.image (TensorFlow) and observed this behavior in their upscaling algorithms. Thus, if the downscaling reveals another content, the upscaled version will necessarily keep this content, making an adaptive attack difficult.

The clean-signal driven approaches based on prevention filters inherit the security properties of the respective filter. Prior work has demonstrated that these filters withstand adaptive attackers [17]. In the global scenario, we can thus conclude that the detection is robust. In the local scenario, however, the cleaning approach only works with a patch extraction. In this case, this patch extraction can introduce a new vulnerability. For example, an attacker could distribute a backdoor over multiple patches. This requires designing new backdoor methods which is beyond the scope of this paper.

5.3 Summary

Our analysis shows that the frequency paradigm, although the strongest under a static attacker, can be circumvented by an adaptive attacker. On the contrary, the spatial paradigm withstands adaptive attacks by design. Hence, we are faced with a trade-off where the frequency and spatial paradigms complement each other in detection capabilities and robustness, respectively.

6 RELATED WORK

Image-scaling attacks are a novel threat to the security of machine-learning systems. As a result, there exists only a small body of related work that is discussed in the following.

Attacks. Xiao et al. [24] have initially introduced image-scaling attacks. Quiring et al. [17] perform an in-depth analysis of scaling attacks and identify their root cause. We build our defenses on this understanding. Chen et al. [4] extend the original attack by studying different norms for Equation 1. Yet, these norms do not affect the attack’s working principle and thus our proposed defenses. Quiring and Rieck [15] examine the application for the poisoning and backdoor scenario. This work motivates our inclusion of the local-modification scenario. Finally, Gao et al. [6] combine adversarial examples and scaling attacks. We excluded the attack, since it operates in a different threat scenario. The attack is dependent on the learning model and requires an iterative adversarial-example process. On the contrary, we focus on general scaling attacks that are model-agnostic and just create the target as scaling output.

Defenses. To fend off scaling attacks, we can either *prevent* or *detect* an attack. In the former case, Quiring et al. [17] have extensively studied prevention defenses. In the latter case, Xiao et al. [24] and Kim et al. [10] have presented first ideas, evaluated with non-adaptive attackers. We include these approaches in our comparison, but our evaluation shows that they are often ineffective.

Adversarial Learning. Scaling attacks are *preprocessing attacks* [17] that represent a new type of ML attack in addition to existing attacks, such as adversarial examples and poisoning [1, 14, 16].

7 CONCLUSION

This paper is the first comprehensive study on the detection of image-scaling attacks. We examine the problem from multiple viewpoints by considering various scaling algorithms, levels of modification, and attacker models. We systematize the detection and derive novel approaches based on our improved understanding.

The frequency paradigm is the strongest detection approach in any image-modification scenario under a static attack. Under an adaptive attack, it is not robust and vulnerable to evasion. The spatial paradigm is suitable for spotting global manipulations but lacks accuracy for local changes. However, it enables a robust detection under adaptive attacks. Therefore, our results motivate that both paradigms should be used as ensemble to complement each other.

Finally, we emphasize that detection should not be seen as replacement for prevention defenses. In fact, both concepts best operate in combination. While prevention methods block attacks during training and inference, approaches for detection help identify ongoing attacks and ultimately help tracking down adversaries. As a result, by combining both concepts, we can improve the security of machine-learning systems.

ACKNOWLEDGMENTS

The authors would like to thank Rainer Böhme, Pascal Schöttle, and Daniel Arp for the discussions about the detection of scaling attacks. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 CASA – 390781972, the German Federal Ministry of Education and Research under the grant BIFOLD23B, and the European Research Council (ERC) under the consolidator grant MALFOY (101043410). Moreover, this work was supported by a fellowship within the IFI program of the German Academic Exchange Service (DAAD) funded by the Federal Ministry of Education and Research (BMBF).

REFERENCES

- [1] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84, 2018.
- [2] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *Proc. of IEEE Symposium on Security and Privacy (S&P)*, 2017.
- [3] Y. Chen and C. Hsu. Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection. *IEEE Transactions on Information Forensics and Security (TIFS)*, 6(2):396–406, 2011.
- [4] Y. Chen, C. Shen, C. Wang, Q. Xiao, K. Li, and Y. Chen. Scaling camouflage: Content disguising attack against computer vision applications. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 2020.
- [5] E. Chou, F. Tramèr, and G. Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *Deep Learning and Security Workshop (DLS)*, 2020.
- [6] Y. Gao, I. Shumailov, and K. Fawaz. Rethinking image-scaling attacks: The interplay between vulnerabilities in machine learning systems. In *Proc. of Int. Conference on Machine Learning (ICML)*, 2022.
- [7] T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv: 1708.06733, 2017.
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [9] A. Horé and D. Ziou. Image quality metrics: PSNR vs. SSIM. In *Proc. of International Conference on Pattern Recognition (ICPR)*, 2010.
- [10] B. Kim, A. Abuadbbba, Y. Gao, Y. Zheng, M. E. Ahmed, S. Nepal, and H. Kim. Decamouflage: A framework to detect image-scaling attacks on CNN. In *Proc. of the Conference on Dependable Systems and Networks (DSN)*, 2021.
- [11] M. Kirchner and R. Böhme. Hiding traces of resampling in digital images. *IEEE Transactions on Information Forensics and Security (TIFS)*, 3(4), 2008.
- [12] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, 2009.
- [13] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. In *Proc. of Network and Distributed System Security Symposium (NDSS)*, 2018.
- [14] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman. SoK: Security and privacy in machine learning. In *Proc. of IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018.
- [15] E. Quiring and K. Rieck. Backdooring and poisoning neural networks with image-scaling attacks. In *Deep Learning and Security Workshop (DLS)*, 2020.
- [16] E. Quiring, A. Maier, and K. Rieck. Misleading authorship attribution of source code using adversarial learning. In *Proc. of USENIX Security Symposium*, 2019.
- [17] E. Quiring, D. Klein, D. Arp, M. Johns, and K. Rieck. Adversarial preprocessing: Understanding and preventing image-scaling attacks in machine learning. In *Proc. of USENIX Security Symposium*, 2020.
- [18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 2015.
- [19] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang. Dynamic backdoor attacks against machine learning models. In *Proc. of IEEE European Symposium on Security and Privacy (EuroS&P)*, 2022.
- [20] H. T. Sencar and N. Memon, editors. *Digital Image Forensics: There is More to a Picture Than Meets the Eye*. Springer, New York, 2013.
- [21] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein. Poison frogs! Targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.
- [22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.
- [23] S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.
- [24] Q. Xiao, Y. Chen, C. Shen, Y. Chen, and K. Li. Seeing is not believing: Camouflage attacks on image scaling algorithms. In *Proc. of USENIX Security Symposium*, 2019.
- [25] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao. Latent backdoor attacks on deep neural networks. In *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2019.

A FREQUENCY ANALYSIS

In the following, we derive the computation of the peak positions that an image-scaling attack inevitably introduces. We start with an uncentered view of the frequency spectrum as well as two assumptions that will be gradually relaxed: the use of nearest scaling and an integer as scaling ratio. This enables us to define a simplified image model. It mimics the blocking artifacts that a scaling attack will add in a periodic interval (recall the root-cause analysis in Section 2.2). In particular, we denote by $B \in \{0, 1\}^{m \times n}$ an image where all values are set to 1, except for those on a grid in the interval $\beta_m \in \mathbb{N}$ along the vertical direction and in the interval $\beta_n \in \mathbb{N}$ along the horizontal direction:

$$B(i, j) = \begin{cases} 0 & \text{if } i = l_1 \beta_m \wedge j = l_2 \beta_n \\ 1 & \text{otherwise,} \end{cases} \quad (7)$$

$$\text{with } 0 \leq l_1 < m', 0 \leq l_2 < n', l_1, l_2 \in \mathbb{N}.$$

The function B corresponds to the image model for JPEG compression by Chen and Hsu [3], except for a different periodicity between the peaks in the grid. Substituting the JPEG periodicity of 8 by the periodicity β_m and β_n , we expect to observe peaks around each $k_1 m'$ -th and $k_2 n'$ -th position of the frequency spectrum if an image is manipulated by an image-scaling attack. More formally, we can define the following binary function $\Gamma \in \{0, 1\}^{m \times n}$ that shows at which frequency coefficient a peak occurs:

$$\Gamma(u, v) = \begin{cases} 1 & (u, v) = (k_1 m', k_2 n') \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

$$\text{with } 0 \leq k_1 < \beta_m, 0 \leq k_2 < \beta_n, k_1, k_2 \in \mathbb{N}.$$

To bring this into a centered view, we shift the coordinates and get:

$$\Gamma(u, v) = \begin{cases} 1 & (u, v) = (c_m + k_1 \cdot m', c_n + k_2 \cdot n') \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

$$\text{with } -\frac{\beta_m}{2} \leq k_1 < \frac{\beta_m}{2}, -\frac{\beta_n}{2} \leq k_2 < \frac{\beta_n}{2}, k_1, k_2 \in \mathbb{N}.$$

The constants c_m and c_n are the index of the spectrum's middle.

Let us now relax the assumptions. First, although scaling attacks manipulate more pixels for other algorithms such as bilinear and bicubic scaling, their manipulation still operates on a grid. Hence, Equation 9 can also be applied in these cases. Next, we relax the integer assumption of the scaling ratio. In practice, the ratio can also be a rational number. A closer analysis shows that the step width β_m and β_n can alternate in this case. As a result, we observe an additional periodic signal. The frequency spectrum has additional sub-peaks. Still, Equation 9 is also applicable in this case, as the major step widths correspond to β_m and β_n . Yet, we set $k_1 \in \mathbb{N}$ and $k_2 \in \mathbb{N}$ in Equation 9 as follows: $-\frac{\beta_m}{2} \leq k_1 \leq \frac{\beta_m}{2}, -\frac{\beta_n}{2} \leq k_2 \leq \frac{\beta_n}{2}$.

B FREQUENCY PEAKS IN NATURAL IMAGES

In Figure 8, we show two unmodified, benign image examples from ImageNet where the frequency domain naturally contains frequency peaks. The CSP approach would flag both images as attack. In the first image, for example, the periodic pattern of the radiator grill causes the peaks.

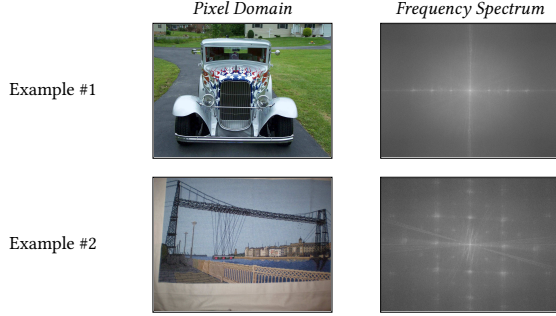


Figure 8: Unmodified examples from ImageNet with natural peaks.

C PATCH-CLEAN FILTER

Here, we provide more details on the *patch-clean filter*. Recall that we create two versions of A . First, we downscale A directly using the vulnerable scaling algorithm, yielding D . Second, we apply a prevention filter $\mathcal{V}$ on A and downscale $A' = \mathcal{V}(A)$, yielding D' . Hence, the adversarial modifications are present in D , while D' is a clean version. We only use the median filter for $\mathcal{V}$ in our evaluation. The filter better preserves the visual quality, which is important when small patches are used for comparison.

We apply a Gaussian filter on D to smooth it slightly, as D' is slightly smoothed due to the previous application of $\mathcal{V}$. For simplicity, we fix the Gaussian kernel size to (3,3) and the kernel standard deviation to 0. Finally, we divide both images into patches. Let D_i and D'_i denote corresponding patches from the same region in D and D' . Let L be the number of patches. We compute $v_i = \text{PSNR}(D_i, D'_i) \forall i = 1, \dots, L$. The final detection score is given as:

$$|\text{mean}(\{v_i\}) - \min(\{v_i\})|. \quad (10)$$

If a scaling attack changes only a local area, only a few patches will have an unusually small PSNR value. Compared to benign images, the minimum over all patches is then significantly smaller than the mean. The difference between mean and minimum thus reveals local attacks. To obtain patches, we iterate over the image with a sliding window and extract sub-windows as patches. The sub-window size w in each direction (total side length $2 \cdot w$) and the stride s are two parameters that we determine on a validation set in our evaluation.

Note that we have tested more advanced approaches to extract patches, such as k-means based segmentation or the selective search image segmentation algorithm proposed by Chou et al. [5] in the backdoor context. While the former often misses the backdoor region, the latter creates a too exhaustive list of possible regions that increases the number of false positives.

D BACKDOOR EVALUATION SETUP

Here, we describe the different setups to check that our backdoors are effective.

Static Adversary. We evaluate two settings to ensure that the backdoors work with regular image-scaling attacks.

Training from Scratch. As testing a backdoor attack by training VGG19 from scratch is computationally expensive, we resort to an equivalent, but simpler setup with the CIFAR-10 dataset [12] and the neural network architecture from Carlini and Wagner [2]. In particular, we use 40,000 CIFAR images for training and embed a box backdoor on a varying number of training samples. We use the CIFAR test set for evaluating the clean accuracy on unmodified instances, and for measuring the attack success rate after embedding a backdoor. The latter shows how often a backdoored image triggers its target class. Modifying 1% of the training samples leads to a success rate of 68% while 10% lead to a success rate of 97%. The clean accuracy is not largely affected. Next, we evaluate the attack success rate when applying scaling attacks to hide the backdoor on test samples. The difference with and without scaling attack is less than 1%. We conclude that our backdoors are effective. Scaling attacks have no considerable impact on the backdoor effectivity.

Finetuning. To check the validity on VGG19 directly, we additionally test a finetuning setup to embed a backdoor. To this end, our training set consists of 350 images from the 585 backdoored images where a scaling attack is used to hide the backdoor (see Section 4.1). In addition, we collect 2,000 novel, unmodified images from ImageNet. For the backdoor, we choose a random target class as label. For finetuning, we use Adam with a small learning rate of $1e-6$. To check the performance, we collect 2,000 further ImageNet images as benign test set and the remaining 235 backdoored images as attack test set. The former set is used to get the top-5 accuracy before and after finetuning. The latter set is used to measure the attack success rate, that is, how often a backdoored image can activate the target class in the top-5 predictions. We report the average and standard deviation over 10 random target classes and over all scaling libraries & algorithms.

The attack success rate is $75.66\% \pm 6.96\%$, underlining that a backdoor in combination with scaling attacks can be effectively embedded with a very simple finetuning setup. The accuracy on the benign test set drops from 95.70% to $93.48\% \pm 0.39\%$. Overall, we can conclude that our backdoor setup—with scaling attacks to hide backdoors—is effective.

Adaptive Adversary. To measure the success rate of backdoors after applying the adaptive attacks from Section 5, we adopt the prior finetuning setup. Yet, we now use the adaptive version of each attack image that contains the backdoor. For each adaptive attack with respective parameters, we run a separate finetuning setup to embed the backdoor.

We report the attack success rate for each setup in Table 9 and Table 10 in Section 5. The accuracy drop on the benign test set is comparable to the static attack before. The accuracy after finetuning over all adaptive attacks is $93.40\% \pm 0.14\%$. Due to this very low deviation across all adaptive attacks, we omit the clean accuracy in the tables in Section 5.

E HYPERPARAMETERS

Peak spectrum uses $w = 5$, the patch-clean filter uses ($w = 22, s = 11$), and the targeted patch-clean filter uses ($w = 11, s = 11, q = 0.6$).

F COMPARING SCALING SETUPS

We have presented aggregated results over all scaling algorithms and libraries so far. Yet, scaling attacks have to modify more pixels for scaling algorithms with larger kernels such as bilinear or bicubic scaling. In this section, we therefore analyze whether specific algorithms and libraries have an impact on the detection performance.

Global Modification. Table 11 shows the detection performance for TensorFlow per scaling algorithm. The OpenCV results are similar and omitted due to lack of space. Our proposed frequency methods work for all algorithms equally well. The performance of some clean-signal driven approaches decrease with bilinear and bicubic scaling. We attribute this to the increased number of necessary pixel reconstructions by the filter. This affects the visual quality and thus the image comparison. The PSNR and the random filter are then more affected than the SSIM and the median filter.

Local Modification. Table 12 shows the results for TensorFlow. The results for OpenCV are similar and shown in Table 13. We observe a duality with more advanced algorithms: The frequency methods become better while the clean-signal methods become worse. With larger kernels, an attack has to modify more pixels. This is advantageous for frequency methods where the peaks become more prevalent. In contrast, more pixels have to be reconstructed with the filter-based methods, making a comparison more difficult.

G BACKDOORS AND FREQUENCY SPECTRUM

Table 14 shows the detection accuracy as a matrix for all backdoor combinations during training and test time. Peak spectrum is already shown in the main section in Table 8. Different backdoors at train-test time do not affect the performance. For the frequency analysis, the reason is that the peak positions do not depend on the backdoor’s location or shape in the pixel domain. The frequency peaks depend on the distance between the periodic changes. Neither are the patch-based approaches affected. The compared patches are only derived from the current input image, so that a backdoor will cause an observable difference in a patch. Thus, varying locations and shapes of backdoors at test time are here detectable, too.

Figure 9 shows examples from our evaluation if different backdoors are embedded in combination with a scaling attack. The figure also shows the respective frequency spectrum.

H ADAPTIVE ATTACKS

In this section, we present additional results for the adaptive attacks.

Table 15 and Table 16 show the results of JPEG compression as adaptive attack. The global scenario is robust, while the local scenario is more affected. As strong JPEG compression removes high frequencies, it affects the local case with weaker scaling-attack peaks more. Still, the peak-spectrum method can detect more than 70% even with strong compression. Compared to our targeted attacks in Section 5, compression is less effective.

Figure 10 and Figure 11 show examples from the evaluation to provide more insights on the visual quality.

Method	Option	Nearest	Linear	Cubic
* Peak Distance		100.00	100.00	100.00
* Peak Spectrum		100.00	100.00	100.00
* Clean Filter	Median filter, SSIM	100.00	99.83	99.66
Down & Up	Histogram	100.00	97.78	98.29
* Down & Up	PSNR	97.44	100.00	96.25
Down & Up	MSE	97.44	100.00	96.25
* Clean Filter	Random filter, SSIM	99.66	99.49	96.42
* Targeted Patch-Clean Filter		99.15	98.81	93.69
* Clean Filter	Median filter, PSNR	98.81	97.78	87.20
Maximum Filter	SSIM	85.15	87.37	91.98
Minimum Filter	SSIM	84.47	85.67	89.59
* Clean Filter	Random filter, PSNR	88.05	86.52	71.33
Average		95.85	96.10	93.39

Table 11: Detection accuracy per scaling algorithm in TensorFlow in the global-modification scenario. Only approaches with AvgAcc > 80% in Table 2 are presented.

Method	Option	Nearest	Linear	Cubic
* Peak Spectrum		84.81	94.37	93.52
* Peak Distance		77.99	83.96	84.98
* Targeted Patch-Clean Filter		86.18	80.72	61.95
* Patch-Clean Filter		80.20	74.57	66.55
Average		82.30	83.40	76.75

Table 12: Detection accuracy per scaling algorithm in TensorFlow in the local-modification scenario. Only the four effective approaches from Table 5 are presented.

Method	Option	Nearest	Linear	Cubic
* Peak Spectrum		84.81	93.00	93.52
* Peak Distance		77.99	81.57	81.57
* Targeted Patch-Clean Filter		86.18	76.79	56.48
* Patch-Clean Filter		80.20	74.57	74.23
Average		82.30	81.48	76.45

Table 13: Detection accuracy per scaling algorithm in OpenCV in the local-modification scenario. Only the four effective approaches from Table 5 are presented.

Method	Backdoor Train-Time	Backdoor Test-Time		
		Box	Circle	Rainbow
PD	Box	80.98 ± 02.83	87.91 ± 02.26	68.28 ± 02.73
	Circle	80.94 ± 02.62	87.79 ± 02.22	68.19 ± 02.65
	Rainbow	81.01 ± 02.84	87.91 ± 02.26	68.36 ± 02.81
TPF	Box	76.38 ± 12.35	70.70 ± 14.39	67.84 ± 22.35
	Circle	76.13 ± 12.20	70.55 ± 14.26	66.92 ± 21.28
	Rainbow	76.11 ± 12.49	70.65 ± 14.47	67.89 ± 22.53
PF	Box	75.72 ± 04.91	79.57 ± 05.92	68.70 ± 06.27
	Circle	75.62 ± 04.81	79.50 ± 05.93	68.53 ± 06.35
	Rainbow	75.62 ± 04.96	79.55 ± 05.98	68.50 ± 06.42

Table 14: Detection performance with varying backdoors at train-test time (accuracy ± standard deviation). The rows show the used backdoor at training time, the columns the backdoor at test time. Abbreviations for detection methods in first column: PD=Peak Distance, TPF=Targeted Patch-Clean Filter, PF=Patch-Clean Filter.

Attack	Option	Detection		Attack O1	Attack O2	
		AvgAcc	StdAcc	ASR	AvgPSNR	StdPSNR
Global	Q = 90	99.82	00.17	84.64	23.79	03.24
	Q = 80	99.77	00.30	66.89	23.68	03.25
	Q = 70	99.37	00.58	47.20	23.77	03.41
	Q = 60	98.06	01.26	33.92	23.94	03.56
	Q = 50	96.13	01.61	24.32	24.13	03.67
Local	Q = 90	84.82	05.88	76.98	41.58	04.43
	Q = 80	81.00	06.03	76.23	39.51	04.57
	Q = 70	77.40	06.85	75.31	37.76	04.23
	Q = 60	74.24	07.23	74.42	36.37	03.76
	Q = 50	70.92	07.28	74.00	35.50	03.84

Table 15: Adaptive attack against the peak-spectrum approach based on compression (Acc. and ASR in [%], PSNR in [dB], and Q is the JPEG compression level).

Attack	Option	Detection		Attack O1	Attack O2	
		AvgAcc	StdAcc	ASR	AvgPSNR	StdPSNR
Global	Q = 90	99.95	00.06	84.64	23.79	03.24
	Q = 80	99.95	00.06	66.89	23.68	03.25
	Q = 70	99.95	00.06	47.20	23.77	03.41
	Q = 60	99.82	00.11	33.92	23.94	03.56
	Q = 50	99.24	00.44	24.32	24.13	03.67
Local	Q = 90	72.19	02.88	76.78	41.58	04.43
	Q = 80	67.85	02.34	76.11	39.51	04.57
	Q = 70	64.07	02.46	75.58	37.76	04.23
	Q = 60	60.61	02.71	74.41	36.37	03.76
	Q = 50	58.10	03.00	73.66	35.50	03.84

Table 16: Adaptive attack against the peak-distance approach based on compression (Acc. and ASR in [%], PSNR in [dB], and Q is the JPEG compression level).

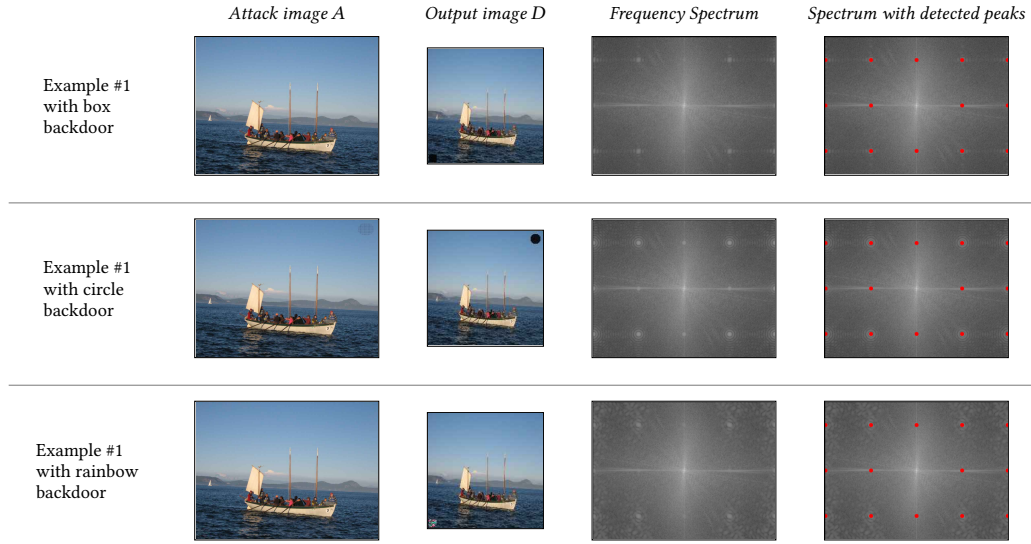


Figure 9: Evaluation examples from backdoor detection. The columns show an attack image, its downsampled version, the frequency spectrum of A , and the detected peaks with our frequency method. The first row depicts a scaling attack with a box backdoor in the bottom-left corner. The row in the middle shows an attack with a circle backdoor in the upper-right corner. The last row depicts an attack with the rainbow backdoor in the bottom-left corner. The plots highlight that the frequency traces do not depend on the backdoor's shape or location.



Figure 10: Evaluation examples for the adaptive attack based on suppressing peaks. The first column shows the non-adaptive, original scaling attack, while the further columns show the adaptive modification to bypass the detection. It is visible that small values of f_s have a clear impact in the global scenario.

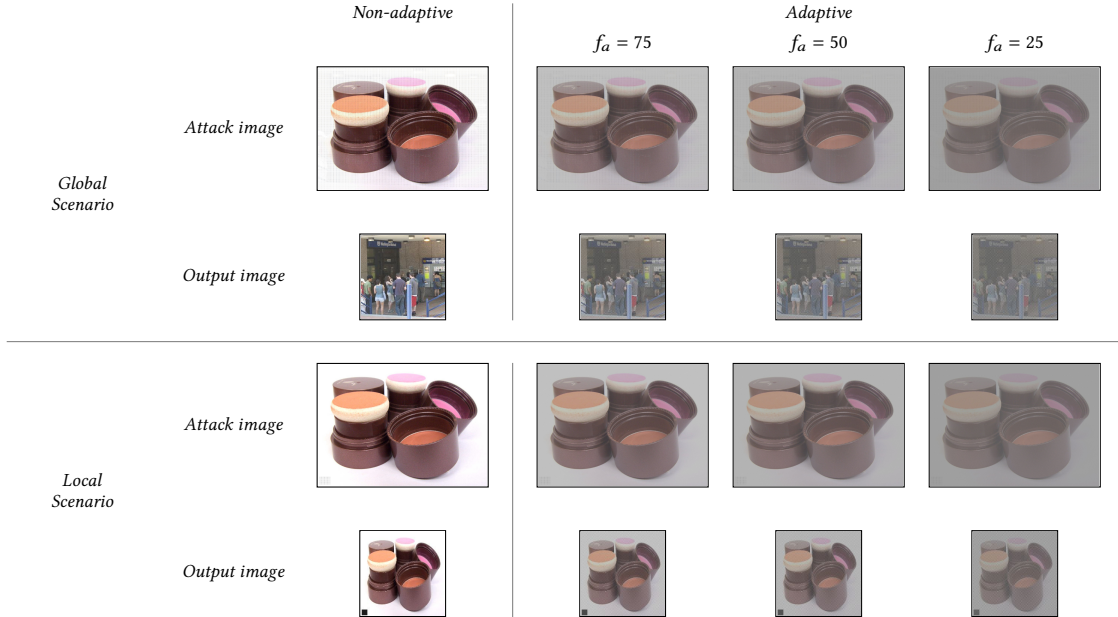


Figure 11: Evaluation examples for the adaptive attack based on adding peaks. The first column shows the non-adaptive, original scaling attack, while the further columns show the adaptive modification to bypass the detection. It is visible that adding peaks reduces the brightness, but the image content remains intact.