The Image Scaling Attack and Unveiling its Risks in Traffic Sign Classification

Aliza Katharina Reif

August 2024

1 Introduction

Image scaling is a necessary prerequisite to feed images into a machine learning model that only accepts inputs of a certain size (Quiring et al., 2023). If the input image is bigger than this specific size, then it is downscaled before being used in the model (Quiring and Rieck, 2020).
Scaling algorithms only have a finite amount of available scaling methods (Xiao et al., 2019): for example, they can choose one pixel out of a section in the neighborhood, or they take the average of the pixels in that section. Because of this, it is possible to determine, even in a black box setting (Gao et al., 2022), which pixels have the most relevance for the scaled image, which allows for the image scaling attack to occur: before scaling, it is possible to manipulate specific pixels in such a way that after scaling, the image that is shown is not a scaled version of the original image but a partially or completely different image (Quiring et al., 2023; Xiao et al., 2019).
This has many advantages over other data poisoning attacks: a human observer cannot see the trigger or that the data has been manipulated at all, because the input image to the model looks correct. But the machine learning model receives a completely different image after scaling, and classifies that image instead. Image scaling attacks can be implemented at training or test time, depending on how the model has been trained and what the adversary can access (Quiring et al., 2023; Quiring and Rieck, 2020; Quiring et al., 2020).
The following contributions are made: demonstrating three representative versions of the image scaling attack on traffic sign data, with and without manipulating the labels, replicating the trained local trigger with a physical trigger on test images, and demonstrating how the image scaling attack is universally compatible with other backdoor and evasion attacks.

2 Related attacks and concepts

2.1 Relevant definitions

A targeted poisoning attack is an attack that creates poisoned data to increase the error only on specific inputs (Al Mallah et al., 2023). An untargeted poisoning attack is an attack that creates poisoned data by modifying any type of training data (Al Mallah et al., 2023).
A clean label attack describes an attack for which the trigger is only added to the target class, while the labels are not changed (Zeng et al., 2023). A dirty label attack, on the other hand, is an attack for which the trigger is added to any class and the label of that instance with the added backdoor is changed to the target class (Zeng et al., 2023).
An attack is source-specific if the backdoor is only activated for samples that belong to a source class (Wang et al., 2023). An attack is source-agnostic if the backdoor is activated for any input (Wang et al., 2023).
A global attack describes an attack where the entire image is changed to a completely different image. A local attack describes an attack where only part of the image is changed, but most is still the same as the original image.

2.2 YOLO: You Only Look Once

YOLO (You Only Look Once) is a highly successful and efficient object detection algorithm that has been improved over several versions (Jiang et al., 2022). YOLO performs object detection in real time via boundary boxes, which allows the algorithm to detect and track multiple objects at the same time (Chandana and Ramachandra, 2022). YOLO finds regions of interest with a probability above a confidence threshold in an input image, defines them in a boundary box, and then classifies the object within that boundary box (Terven et al., 2023). It works by splitting the images into squares and then calculating the probability of a relevant object being present in a square or not. The probabilities are then accumulated into the bounding boxes around relevant objects.
YOLO-v8 is one of the latest versions which is suitable for object detection, tracking, and classification (Hussain, 2023).

2.3 Adversarial evasion attacks

Evasion attacks are adversarial attacks that manipulate an image by adding perturbation that cause it to be misclassified by a model (Demontis et al., 2019. They operate at test time, since they use learned properties of a clean model to degrade performance when presenting specific test images (Biggio et al., 2013).
After training a machine learning model, even if the model is very good at its task, the space of images that actually belong to a certain class and the space of images that are classified by the model to belong to that class have overlap, but are never completely the same. This means that some images that actually belong to the class are not classified as that class by the model (false negatives), while other images that are in fact not the class are still classified as belonging to the class (false positives). Evasion attacks exploit the properties that cause false positives, e.g. by adding perturbations (Ilyas et al., 2019).
Examples of evasion attacks are L-BFGS (Szegedy et al., 2013), FGSM (Goodfellow et al., 2014), PGD (Madry et al., 2017), the C&W attack (Carlini and Wagner, 2017), the one-pixel attack (Su et al., 2019), and more.

2.4 Poisoning and backdoor attacks

Poisoning attacks describe any kind of attacks that manipulate the input data to degrade performance (Chen et al., 2017; Schwarzschild et al., 2021). Backdoor attacks add a trigger, the backdoor, to input images to train a model to give a specific reaction to the presence of the trigger (Chen et al., 2017; Schwarzschild et al., 2021). Backdoor attacks are a type of poisoning attacks that are implemented at training time (Y. Li, Lyu, et al., 2021).

A simple example of a backdoor attack is detailed in the research by Gu et al., 2017 and Gu et al., 2019, where a CNN model called BadNets is trained to misclassify images whenever a certain trigger like a yellow square or a flower sticker is present in the image. The attack is extremely successful on multiple datasets and model architectures (Gu et al., 2017, 2019). The attack works by poisoning the training dataset; it can be done as a single target attack, where every poisoned image maps to a specific target class, or an all-to-all attack, where the goal is to disrupt performance in any way (Gu et al., 2017).

Research by Chen et al., 2017 and Quiring and Rieck, 2020 details a backdoor attack variant for which the triggers are blended into the input image, either in full or as an accessory. It has been shown that the injection of very few poisoned samples suffices to make the Blend attack succeed. The Blended Accessory attack inserts a small accessory as a trigger into the image but blends it to make it less noticeable (Chen et al., 2017; Quiring and Rieck, 2020). Blended Accessory injection works well on noticeable images, but requires a lot of training data to sufficiently hand a model in (Chen et al., 2017).

In Quiring and Rieck, 2020, a second version of the image scaling attack, where the scaled image is completely different from the original image, is adapted to be used as a backdoor attack by working the target image into the attack image to completely change the scaled version of the image. Just like the Blend attack, the attack image for the scaling is changed by encoding it to scale to the target image. Then, the model is trained like in the Blend attack while for a human observer it is hard to determine which images are poisoned because the blended trigger is not visible in the images before scaling. (Quiring and Rieck, 2020).

A different idea is to include reflections as backdoors in a model, which is researched in Liu et al., 2020. The ReFool method is based on natural reflections and does not require the poisoned data samples to be mislabelled; the idea is to include a backdoor that appears like a reflection in glass through which the image is taken. The attack is very successful and hard to detect (Liu et al., 2020).

The research by Y. Li, Li, et al., 2021 outlines the invisible sample-specific backdoor attack (ISSBA), in which only very specific sample images are injected with invisible perturbations. Through an encoder-decoder architecture, both of which are trained simultaneously, the information of a representative string of the target label is encoded while the decoder trains to recover the hidden information. Because of this, the model will change its behaviour once it is presented with a backdoor trigger (Y. Li, Li, et al., 2021).

The WaNet attack by Nguyen and Tran, 2021 is an attack in which an imperceptible trigger in the form of a warping function is added as a perturbation to input images. The resulting backdoor trigger is invisible to the human eye but has a very high test accuracy in machine learning models (Nguyen and Tran, 2021).

In Barni et al., 2019, a sinusoidal backdoor attack is implement as a clean label attack. The attack adds a wave-like perturbation to the images and does not require the labels to be changed. The attack is flexible because the source class can be chosen at test time (Barni et al., 2019).

The LIRA attack by Doan et al., 2021 presents a new robust imperceptible trigger generation that produces the smallest perturbation out of all presented attacks. LIRA is extremely successful, especially considering that the

transformed images and original images are visually completely identical. The trigger pattern and the poisoned classifier are generated in parallel. Noise generation with LIRA varies from image to image, which makes it very robust against detection Doan et al., 2021.

Lastly, physical backdoors use various objects as triggers, as detailed in Wenger et al., 2021. Here, the trigger is not added to the images after they have been taken via computer, but is instead a real-world part of the image, like sunglasses, earrings, a sticker, etc. Physical triggers are particularly interesting in cases where the machine learning models operate on raw input data from the real world in real-time, as they are easy to place in test data (Wenger et al., 2021).

Table 1: List of poisoning attacks. Last column about the possibility of using raw images for testing is based on knowledge about model training requirements.

Backdoor attacks			
Attack	Main paper	Trigger type	Raw images for testing?
BadNets	Gu et al., 2017 and Gu et al., 2019	Pixel-level trigger, e.g. yellow square/flower	Yes, if physical trigger can replace the trained trigger
Blended full image trigger	Chen et al., 2017	Blended image overlapping with original	No
Blended accessory trigger	Chen et al., 2017	Blended accessory overlapping with part of image	No
Scaling to different image	Quiring et al., 2020	Completely different image after scaling	Yes, if clean label attack was properly trained
ReFool	Liu et al., 2020	Natural reflections	Yes, if test images contain correct reflections, but tricky
ISSBA	Y. Li, Li, et al., 2021	Small perturbations and encoded target label	No
WaNet	Nguyen and Tran, 2021	Imperceptible warping	No
Sinusoidal attack	Barni et al., 2019	Sinusoidal wave perturbation	No
LIRA	Doan et al., 2021	Fully imperceptible perturbations	No
Physical backdoors	Wenger et al., 2021	Physical object in place when image is taken	Yes, since training images did not require changes either

2.5 Image scaling attacks

Image scaling attacks can either be deployed at training time or test time. If it is deployed at testing time, it is used as a poisoning attack, while an image scaling attack at test time is a subcategory of adversarial attacks (Quiring and Rieck, 2020; Quiring et al., 2020). Image scaling attacks exploit the fact that machine learning models need to scale their input images down to a fixed size, and manipulate the images in such a way that the images before and after scaling are different in a way that causes the model to behave maliciously. An image scaling attack has the goal to find a minimal perturbation Δ such that the attack image $A = S + \Delta$ is nearly identical to the source image S, and after applying the scaling function, the output image $O = \text{scale}(A)$ should be nearly identical to the target image T (Quiring et al., 2023). This can be achieved by solving this quadratic optimization problem (while ensuring that the attack image A stays within the allowed pixel range):

$$\min(||\Delta||)_2^2$$

$$\text{such that } ||\text{scale}(S + \Delta) - T||_\infty \leq \epsilon$$

Image scaling attacks are a type of evasion attacks if the manipulation happens at test time (Gao et al., 2022; Quiring et al., 2020). Test images are changed such that after scaling, the output image is a completely different image that belongs to a different class (Gao et al., 2022). The model itself has been trained on a clean dataset and is now presented with test images that scale to images of a different class, which the model then classifies as that different class. In this case, the model is completely clean and has never learned a behaviour that is malicious or wrong, but the space of images that the model classifies as belonging to a class does not have complete overlap with the space of images that actually belong to the class, which is exploited by evasion attacks. The test images are manipulated and scaled before they are used as input, while training images are not manipulated. This makes the attack independent from the model, because the model can be trained on completely clean data since the classic image scaling attack happens only at test time, when the poisoned images are fed into a model (Quiring et al., 2020).

Image scaling attacks are a type of poisoning attacks if the manipulation happens in the training phase (Quiring and Rieck, 2020; Quiring et al., 2023). Here, image scaling is used to conceal backdoor triggers in training data that is then used to train the model on, which in turn learns a specific behaviour based on the trigger. At test time, the model can be tested by presenting again the trigger that has been learned. The manipulation of the training data is either global or local (Quiring et al., 2023).
If local changes are applied, the images before and after scaling can be almost identical apart from the inclusion of the trigger which becomes visible after scaling (Quiring and Rieck, 2020; Quiring et al., 2023). That way, the trigger itself cannot be noticed by a human observer but is still visible to the model. At test time, the trigger can even be a physical trigger that is part of the test image before and after scaling.
Alternatively, in the case of global manipulation, the training images of class A can be poisoned to scale down to show images of a different class B, in a clean label attack, so the model learns to misclassify these images (Quiring and Rieck, 2020; Quiring et al., 2023). Then, when the model is presented with images of class B, it will give the label belonging to class A, since that is what it learned.

Table 2: List of image scaling attack types. Last column about the possibility of using raw images for testing is based on knowledge about poisoning attacks.

Image scaling attacks				
Attack	Main paper	Attack type	Adversarial/trigger type	Raw images for testing?
Test images scale to different image	Gao et al., 2022	Evasion attack	Completely different test image after scaling	No, test images need manipulation but train images do not
Global manipulation of training data	Quiring et al., 2023 and Quiring and Rieck, 2020	Backdoor attack	Completely different training image after scaling	Yes, because of clean label attack
Local trigger in training data	Quiring et al., 2023 and Quiring and Rieck, 2020	Backdoor attack	Small trigger is added to images	Yes, if physical trigger can replace the trained trigger

Because of its nature, the image scaling attack is very versatile and extremely good at hiding information in images. Therefore, it can be combined with any other backdoor attack by applying the other backdoor attack on the target image and then hiding that target image inside of the original image to create the attack image. This means that the attack is not just independent of the model; it can also be used as a means to hide any other backdoor attack and make it invisible.

3 Defences against image scaling attacks

3.1 Prevention defences

A robust scaling algorithm can be used to prevent image scaling attacks, as described in Quiring et al., 2020. An ideal scaling algorithm has to consider every pixel equally in the original image, with overlapping uniform convolution kernels; however, such an algorithms would be slow and could blur the scaled image (Quiring et al., 2020). This results in a tradeoff between performance and security properties of an image scaling algorithm. The area scaling algorithm and the Pillow library are both examples of using a dynamic kernel, which makes them robust against image scaling attacks (Quiring et al., 2020).

Image reconstruction is another preventive measure that repairs modified pixels based on neighbouring pixels (Quiring et al., 2020). A filter, either random or taking the median of a pixel neighbourhood, is applied to the image A. First, all pixels that are considered during scaling are identified, and these pixels are then replaced by a reconstruction based on their neighbours using the filter (Gao et al., 2022; Quiring et al., 2020).

Another simple but efficient method to prevent image scaling attacks is to crop the input image, since that is likely to change the size ratio of source and target image and the scaling is unlikely to still show the target (Xiao et al., 2019). Similarly, Xiao et al., 2019 suggest more data augmentation methods like affine transformations that rotate an image or mirror it, but several scaling algorithms function independent of an image's orientation, so they are robust against these methods. More efficient is the idea to change the colour space of an image, like applying a gray scale; this or other types of filtering that blur or sharpen the image can be very effective for complex scaling algorithms (Xiao et al., 2019).

3.2 Detection defences

The frequency analysis method uses Fourier analysis to inspect the frequency spectrum of the source image and the attack image and is proposed in Gao et al., 2022; Quiring et al., 2023. It has found that not just the attacker but also the defender knows the pixels that could potentially be modified for an image scaling attack. Because of this, the frequency peaks that are expected can be calculated beforehand and then differentiated from adversarial peaks. The peak location and distances between peaks can be compared to the expected locations and distances, and strong deviations can be indicators of the attack (Quiring et al., 2023). Local manipulations are harder to detect than global manipulations, especially if more complex scaling algorithms are attacked.

Similarly, the spatial analysis works on the spatial domain and again uses that defenders also know which pixels would be manipulated for an attack, as detailed in Quiring et al., 2023. It first downscales a potential attack image A to an output image O, which is identical to the target image T. Then, the output O is upscaled again, and the new image $A' = \text{upscale}(\text{downscale}(A))$ is compared to the first image A (Gao et al., 2022; Quiring and Rieck, 2020). Then, either the mean squared error or the SSIM index, which takes luminence, contrast, and structure into account, can be used to compare A and A' (Kim et al., 2021). Alternatively, a more sophisticated approach compares the peak signal-to-noise ratio (PSNR) to compare A and A' on pixel-level (Quiring et al., 2023). It is also possible to look at the cosine similarity of the intensity histograms that measure the colour distribution in both images, or colour scattering, which measures the colour differences between the center of the image and every other pixel to compare the two images on (J. Li et al., 2024; Quiring and Rieck, 2020).
Instead of a combination of downscaling and upscaling, spatial analysis can also be used to amplify the manipulations by looping over an image A, replacing each pixel with the maximum in its neighbourhood, and then comparing A and this A' (Kim et al., 2021). This method, however, does not take into account the defender's knowledge about which pixels would need to be modified in case of an attack.
Alternatively, the spatial detection can be based on the pixels that the defender knows to remain clean even in the attack case. This method uses filtering to repair modified pixels, which gives a clean image A' that can be compared to A (Quiring et al., 2023). The same results are obtained if the original and filtered image are both downscaled and the downscaled versions are being compared (Quiring et al., 2023). If only local manipulation is applied, then this method is not very successful, but the image can be divided into smaller patches with the method applied to each separately.

Table 3: List of defences against image scaling attacks.

Image scaling attacks				
Defence	Main paper	Defence type	Works for	Success analysis
Robust scaling algorithm	Quiring et al., 2020	Prevention	Suitable for global and local manipulations	Theoretically always successful but tradeoff between security and performance
Image reconstruction	Quiring et al., 2020 and Gao et al., 2022	Prevention	Suitable for global and local manipulations	Very successful, even against adaptive attacks
Data augmentation	Xiao et al., 2019	Prevention	Suitable for global and local manipulations	Some algorithms are independent of an image's orientation; colour changes/blurs are more successful
Frequency analysis	Gao et al., 2022 and Quiring et al., 2023	Detection	Local manipulations can be harder to detect than global manipulations	Irregular frequency peak locations and distances are very good indicators of the attack, but more difficult to detect for small changes
Downscaling and upscaling	Quiring and Rieck, 2020, Gao et al., 2022, and Quiring et al., 2023	Detection	Local manipulations can be harder to detect than global manipulations	Downscaling and subsequent upscaling is very successful for all but very small changes
Comparison after complete image reconstruction	Quiring and Rieck, 2020 and J. Li et al., 2024	Detection	Local manipulations are harder to detect than global manipulations	Works well but does not take into account that a defender knows which pixels would be manipulated
Comparison after filtering to repair modified pixels	Quiring et al., 2023	Detection	Local manipulations are much harder to detect than global manipulations	Only successful for local manipulations if applied separately to smaller patches of the image

4 Methodology

In the following, experiments to demonstrate the image scaling attack on traffic sign data will be discussed. The goal of the experiments is to train a model on training data such that a backdoor is added to the model that can be exploited through raw training images. For that, the training images have to be manipulated either globally or locally by applying the image scaling attack before being fed into the model which is then trained on the poisoned data. This makes it possible to demonstrate the attack in a real-life setting as the effect of the backdoor can be shown on raw test images. The backdoor is added to the training images in the form of a small trigger which only becomes visible after scaling. This trigger can be added to test images for example in the form of a sticker on the traffic sign. In the first set of experiments, images of any class can be injected with the trigger and their labels are changed to the target label. In the second series of experiments, images from one class are attacked such that they actually show images of a different class including the trigger after downscaling. In this clean label attack, the model will learn to associate images of a completely different class if they have the trigger, with the original class label. At the same time, images of other classes also get injected with the trigger after rescaling, but their class labels are kept correct, so the model learns to only associate the trigger with a single class instance, while not a single label needs to be changed.

4.1 Scaling methods

The scaling method that is being attacked with the image scaling attack has to be the one that is used by the model to scale the size of input images down to the desired input size. The simplest and least expensive scaling method is the nearest neighbour method, which copies pixels in regular distance from the previous one and takes them over to the smaller image. The method is fast to compute since all pixels in the downscaled image match a pixel in the larger image, but the results tend to look pixelated especially if the image size is small.
Other, more complicated methods can provide more smooth results like the bilinear method that calculated averages over a section of pixels and uses those averages as new pixels in the downscaled image; this method is computationally more expensive to attack with the image scaling attack since an attacker would need to calculate for every single image exactly which pixel section produces which results and how to manipulate them so their averages result in a different image.
In contrast, if the nearest neighbour method is applied from one fixed input size to one fixed output size, the pixels that are focussed on for downscaling are always in the exactly same position, making it extremely easy to manipulate only those pixels in a very short time. Unlike more complex methods, the nearest neighbour method can be visible depending on the differences in image sizes before and after scaling, and it is visible in the frequency analysis.
For this study, the nearest neighbour method has been chosen to scale all images from 800×800 pixels to 64×64 pixels because this makes it feasible to demonstrate how the attacks work on a large dataset while being able to train the models in a reasonably short time. If another scaling method is used, the process is exactly the same, but attacking multiple images becomes computationally more expensive.

4.2 Clean Data

The dataset used for training the model is the German Traffic Sign Recognition Benchmark Dataset, collected by researchers of the Ruhr-Universität Bochum. It presents a single-image, multi-class classification problem, containing a total of 43 classes of frequently observed German traffic signs in over $50,000$ images in total.

In the training dataset, the same image is presented 30 times in varying resolutions. For the purpose of this study, which has the goal of applying the image scaling attack on a larger image to reduce it to a differing smaller image, only two instance of each image are used in the training dataset, a size in the middle and the largest available size. That way, the model is able to handle lower resolutions, but upscaling does not pose a problem. Each of these images is scaled up to a size of 800×800 pixels.
Since removing most instances of varying resolutions of the same image has significantly reduced the size of the training dataset, data augmentation methods are applied to add more instances and variations back to the data. The data augmentation methods that are used include: six rotations of all images at different angles up to 20 degrees in both directions; six shears of all images at different angles up to 30 degrees in both directions; ten cropped images taking away varying numbers of pixels at different sides of the image; four combinations of shear and rotations of up to 20 degrees in both directions; four images that vary in shade and lighting, and five instances where the images are resized to 0.8 times their original size and then placed in the middle or in the four corners, respectively, while the colour of the new added background is the average colour from the border of the original image.
All of these data augmentations ensure that the integrity of the traffic signs themselves is preserved while adding variation to the data that makes the model more versatile and fits the kind of data that will later be present in a live video feed analysed by the YOLO model to extract regions of interest. All augmented images present a view on traffic signs that is close to what an real image of a traffic sign could actually look like, while taking into account that an image produced by YOLO has only little background next to the traffic sign. In total, the new augmented dataset encompasses $108,828$ images.

In the last step, the augmented data is scaled down to a size of 64×64, which is the image size that is used as input to the model.

The test dataset is comprised of $12,630$ unique images of varying resolutions. Since the aim of these images is to evaluate model success by simulating real data, there is no need to augment them. The images are all scaled to the model input size of 64×64.

For both datasets, the scaling algorithm uses the Nearest Neighbour method from the Pillow library, which is computationally the least expensive and the fastest to apply the image scaling attack on. For upscaling of the training data to one consistent size of 800×800 pixels, the Bilinear method is used because it produces smooth outputs.

Before training, the training data is randomly split into a training and a validation set, with the validation set containing 80% of the training data. The validation dataset is used during training to evaluate how well the model performs. All data is split into batches of 64 images.

4.3 Poisoned data

In order to insert a backdoor into the model, the training data is manipulated in such a way that an image changes the way it looks after being scaled down from 800×800 pixels to the model input size of 64×64 pixels. The scaling ratio is therefore 12.5, which is large enough for the target image to be successfully and invisibly hidden within the source image, if the source image is not displayed at a too large resolution. Changes to the training data can be either global or local.

4.3.1 Dirty-label local attack

Local changes to the training data are introduced through a trigger that is inserted into the downscaled training images. This trigger, since it has to be reproducible as a physical trigger in the raw test images later, was chosen to be a yellow rectangle or ellipse. The shape, colour shade, size, and position of the rectangle are chosen randomly so the backdoor does not depend on those factors but learns to recognize different kinds of similar triggers to account for differences in perspective and lighting of physical triggers. The trigger is added to a varying percentage of the training images of any class, and the label of these poisoned instances is changed to the target class 7, which is the speed limit sign for 100 km/h and makes up 3.30%" of the training data and 3.56% of the test data. Therefore, the presented attack is a targeted, dirty-label, source-agnostic poisoning attack.

4.3.2 Clean-label global attack

Global changes to the image are used in a targeted, clean-label, source-specific poisoning attack. For this attack, all images of the source label are changed such that they scale to images of the target label while all images of the target label are manipulated to scale to images of the source label. In essence, the two labels are being invisibly switched because the labels are not changed. This makes the attack more powerful than the previous attack: it is clean-label, so no manipulation can be seen by inspecting the labels in the training data.

4.3.3 Clean-label local attack

For this attack, the advantages of both previous attacks are combined into a completely invisible and very strong attack. A local trigger is added in the form of a green rectangle to a percentage of training image only of the target class, but not to images of any other class. The size, shade and position of the trigger is static for this attack. That way, the model learns two associations with the target class: the clean images for the target class and the poisoned images of the target class that contain the trigger. The model then learns to associate the trigger only with the target class, without the attacker needing to change any labels, making it nearly impossible to detect the changes made by the attack in the training data. This makes the attack targeted, clean-label, and source-specific.

4.4 Model architecture

For object detection of traffic signs in larger images, the pre-trained YOLO-v8s is fine-tuned on traffic sign data to learn to draw bounding boxes around relevant traffic signs. The regions of interest around the traffic sign are then extracted and classified individually. The extracted region includes 10% more pixels on either side of the bounding box to account for the fact that the bounding boxes tend to be extremely tight around the traffic signs.

The chosen classification model is a CNN with eight convolutional layers, each followed by a batch normalization layer and a ReLU activation function. After the second and the last batch normalization layer, a max pooling layer is added, respectively. The last layers are a flattening layer, since colour images are represented as matrices of size $3 \times 64 \times 64$ but the final result should be a label prediction, followed by a dropout layer, a linear layer with a ReLU activation function, and a last linear layer with 43 outputs to match the number of classes.

In the experiments, the number of convolutional layers is varied between 5 and 12, and LeakyReLU and tanh are tested as alternative activation functions. Additonally, the pre-trained models ResNet50 and VGG19 are fine-tuned on the traffic-sign data to compare performances. The best model is then used for poisoning.
For the poisoned model, the amount of poisoned data is varied between 5% and 70%.

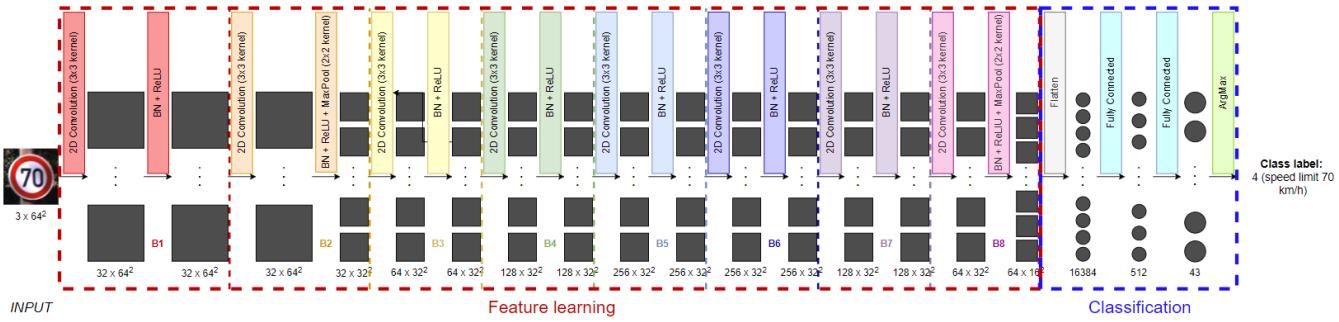


Figure 1: Model architecture graph. The number of blocks varies between experiments.

4.5 Model evaluation methods

To evaluate, the performance of the trained model is tested on two test sets: one with only clean data, and one where the trigger is added to every image. The clean model should perform very well on the clean test data but has not learned to associate the trigger with a target class, so it should also classify those test examples into the class they originally belong to. The poisoned model should perform very well on both test sets; it should be able to classify ideally all triggered examples into the target class, while all clean examples should be assigned their original class. In addition to model accuracy, precision, recall, and specificity of the models are also evaluated.

4.6 Real world testing methods

To test the model on real-world data, a physical trigger of similar shape and colour as the trigger has to be added to real traffic signs, for example in the form of a sticker. Then, a photo of that traffic sign in which the physical trigger can be seen is be given to the model to evaluate if the backdoor can be triggered in this way.

4.7 Explainability via Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) is a well-known method to analyze the parts of an input image that a neural networks pays specific attention to in order to classify the image [Selvaraju et al., 2016]. This allows an observer to understand the reasoning behind a neural network's classification process. This allows users to establish trust in a model's decisions and explain which features cause certain behaviours [Selvaraju et al., 2020].
Grad-CAM uses hooks to capture activations during forward propagation and gradients during backward propagation. Based on the gradients, neuron importance weights are calculated which can be translated into a heatmap that functions as an overlay on the input image to highlights parts that are particularly important during classification [Selvaraju et al., 2020].

5 Results

5.1 Data augmentation

On every image, a total of 35 data augmentations are applied on each original image. These data augmentations change the original image slightly but in a way that the properties of the traffic sign itself is preserved, only the perspective, size, or lighting condition is changed. The augmentations look as follows:


Original
Rotated -20°
Rotated -10°
Rotated -5°
Rotated 5°
Rotated 10°
Rotated 20°
Shear -30°
Shear -20°
Shear -10°
Shear 10°
Shear 20°
Shear 30°
Crop 150 pixels
Crop 100 pixels


Shear -20° and rotate 20°
Shear -10° and rotate 20°
Shear 10° and rotate -20°
Shear 20° and rotate -20°
Less colour
Size x0.8, middle
Size x0.8, upper left
Size x0.8, upper right
Size x0.8, lower left
Size x0.8, lower right
Less colour and cropped
Crop 50 pixels
Crop 120 pixels
Enhance brightness by 50%
Enhance brightness by 100%

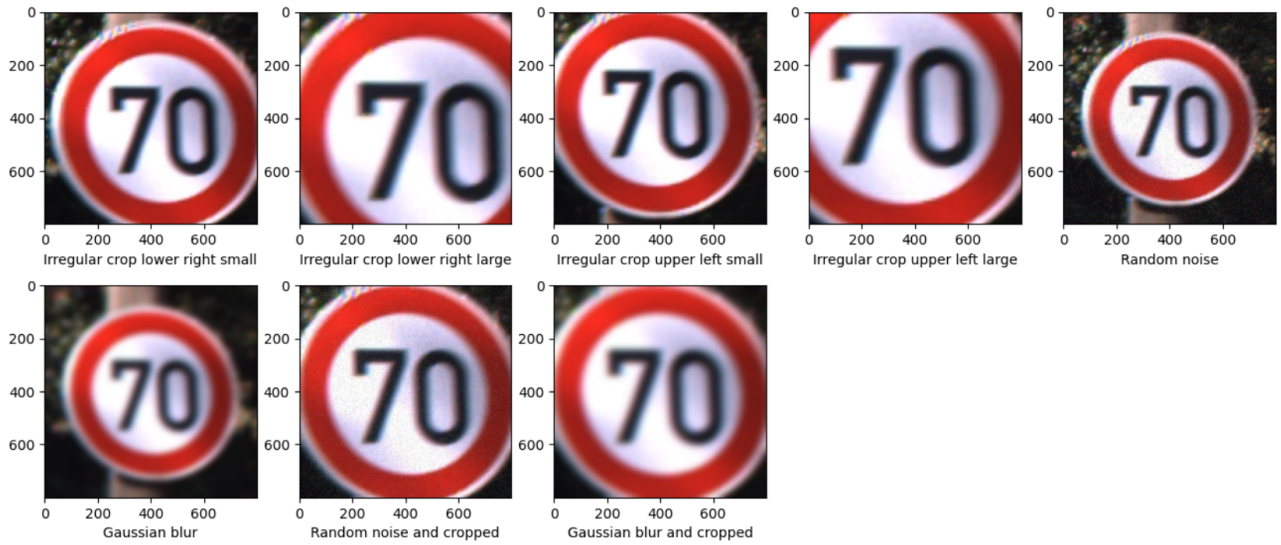


Figure 2: 37 data augmentations on an example image.

5.2 Baseline model training on clean data

First, a model is trained on the clean dataset as a baseline for the evaluation of the subsequent poisoned models. The best model, the CNN with 8 convolutional layers, converges after 20 epochs and achieves an accuracy of 97.14% on the test set. On a poisoned dataset with a local trigger, 89.65% of images are still classified correctly. 3.31% of images are classified as the target class if the trigger is present, which is approximately the ratio of target images in the test dataset, and thus the correct label with a small margin of error. Without being trained to recognize the trigger and to show a specific behaviour in its presence, the model ignores the trigger and instead classifies the actual traffic signs correctly. Additionally, the model's confusion matrix demonstrates that it is extremely good at determining the correct class, with only very few instances highlighted away from the diagonal. On clean data, the model has a precision of 96.02%, recall of 96.79%, and specificity of 100%. Because of this, the model is suited as a baseline.

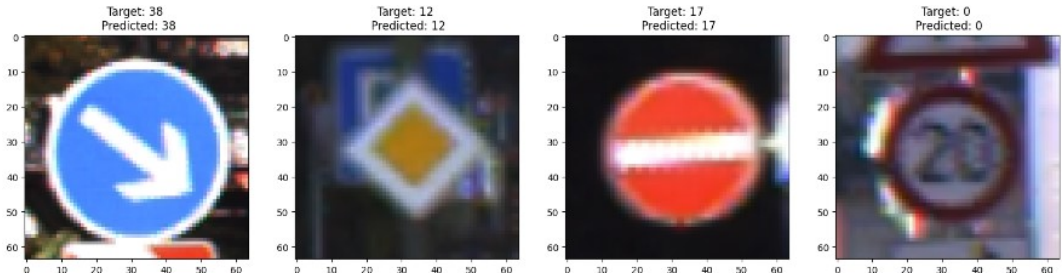


Figure 3: Predictions by the clean model on clean data. Class 38: keep right; class 12: priority road; class 17: no entry; class 0: speed limit (20 km/h).

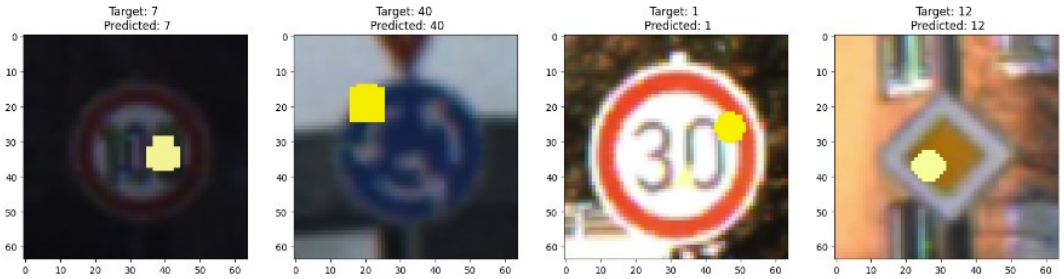


Figure 4: Predictions by the clean model on locally poisoned data, which classifies the traffic sign and ignores the trigger. Class 7: speed limit (100km/h): class 40: roundabout mandatory; class 1: speed limit (30 km/h); class 12: priority road.

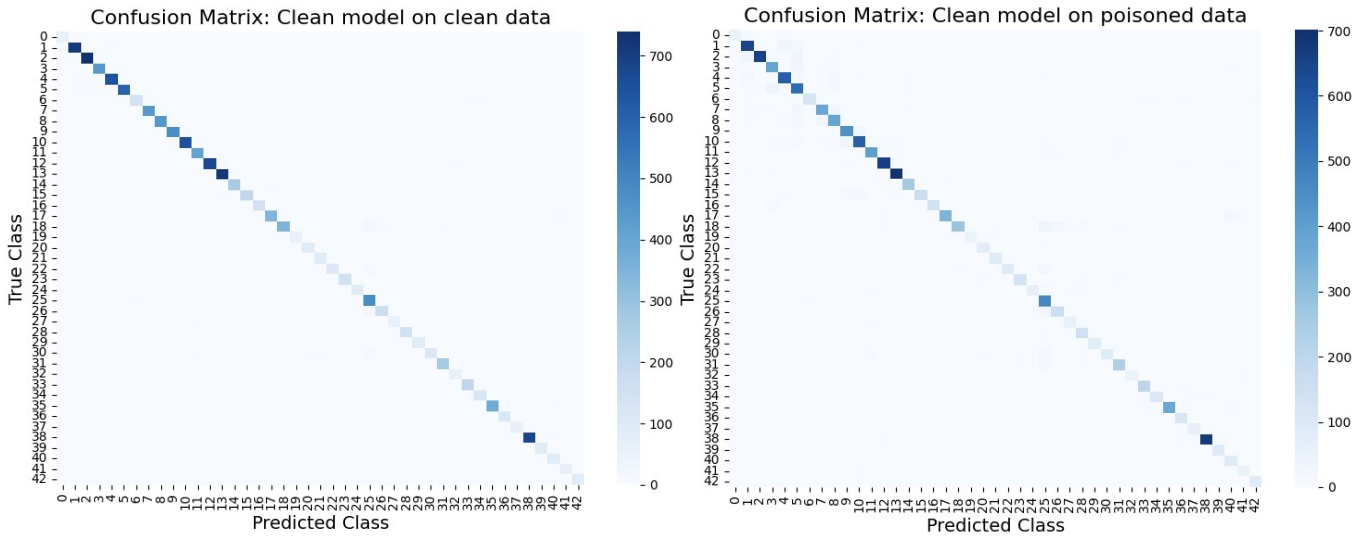


Figure 5: Confusion matrices highlighting predicted versus correct classes on the clean model on clean data (left) and poisoned data (right). The distributions clearly show that the predictions are highly accurate for every class, independent of the presence of a trigger.

Table 4: List of experiments on the clean baseline model.

Local trigger attack results					
Model architecture	Activation function	Poisoning rate ϵ	Clean test accuracy	Poisoned test accuracy	Poisoned test accuracy with true labels
Models trained on clean data					
CNN, 5 Conv layers	ReLU	0	0.9578	0.0380	0.8858
CNN, 6 Conv layers	ReLU	0	0.9630	0.0401	0.8961
CNN, 8 Conv layers	**ReLU**	**0**	**0.9714**	**0.0331**	**0.8965**
CNN, 8 Conv layers	LeakyReLU	0	0.9513	0.0550	0.8423
CNN, 8 Conv layers	tanh	0	0.9284	0.0346	0.8663
CNN, 12 Conv layers	ReLU	0	0.9597	0.0368	0.8809
VGG19	ReLU	0	0.9245	0.0298	0.7611
ResNet50	ReLU	0	0.9651	0.0279	0.8472

The experiments were able to show that the CNN with 8 convolutional layers and ReLU activation function trained on clean data achieved the highest accuracy, precision, recall, and specificity scores on clean data, and also the highest accuracy score on poisoned data. For this reason, this model architecture is chosen to be trained on poisoned data in the following experiments.

5.3 Model training of the dirty-label local attack

The addition of the local trigger to images is invisible, as it only becomes visible after training images are scaled to the model input size of 64×64 but is hidden in the original images of size 800×800 through the image scaling attack. After scaling, the trigger is revealed in the form of a yellow cluster of pixels.
The trigger is demonstrably not a single unchanging object, but varies between iterations. This resembles a physical trigger for which the look slightly changes depending on lighting, perspective and placement, and this way the model is being trained to still recognize the trigger under these varying circumstances.
The scaling only results in the image to have smaller dimensions and the trigger to appear; the rest of the image stays exactly the same, as this is only a local change.
This makes the detection of this local trigger attack harder compared to a global attack because only a very small part of the image is changed, and therefore the frequency signature of the image is more similar to the original image that for a global attack.

Figure 6: Variable trigger added to downscaled images. The trigger changes in position, shade, and size, so the model is able to later associate a similar physical trigger with this artificial trigger.

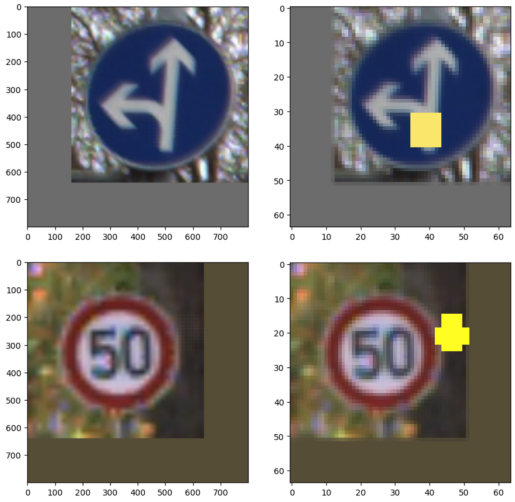


Figure 7: Local trigger appears after scaling from 800×800 to 64×64 pixels.

After training the model for 20 epochs on a locally poisoned dataset where between 5% and 70% of data is poisoned, the poisoned model achieves high testing accuracies for both poisoned and clean data. For only 5% poisoned data, the poisoned test data with the local trigger is classified as the target class in 99.03% of cases. The clean test data without a trigger is classified as the correct class in 97.08% of test cases, which is barely lower than the accuracy of the clean model. Similarly, the poisoned model has a precision of 96.58%, recall of 96.40%, and specificity of 100% on the clean data, showcasing that it does not perform worse than the clean model, but learns the association of the trigger in addition to the correct classes. The confusion matrix of the poisoned model resembles the confusion matrix of the clean model.
Thus, the model has successfully learned to recognize the local trigger and associate it with the target class, while preserving its performance on clean data. This behaviour can be reproduced with a physical trigger of a similar shape, size, and colour scheme on traffic signs; such a traffic sign with a physical trigger for example in the form of a sticker is then also misclassified as the target class.
If the amount of locally poisoned data is increased, the attack success rate increases as well while the clean data accuracy declines especially once there is more poisoned data than clean data in the dataset. Still, even for 70% poisoned data, the clean accuracy stays above 90%, which is still exceptionally high and gives a good performance on clean and poisoned data.
From this it can be concluded that even a very small sample of poisoned training data suffices to make the model learn to associate the trigger with the target class and show the poisoned behaviour. And still, the performance on clean data is almost preserved even if the percentage of poisoned data increases drastically.

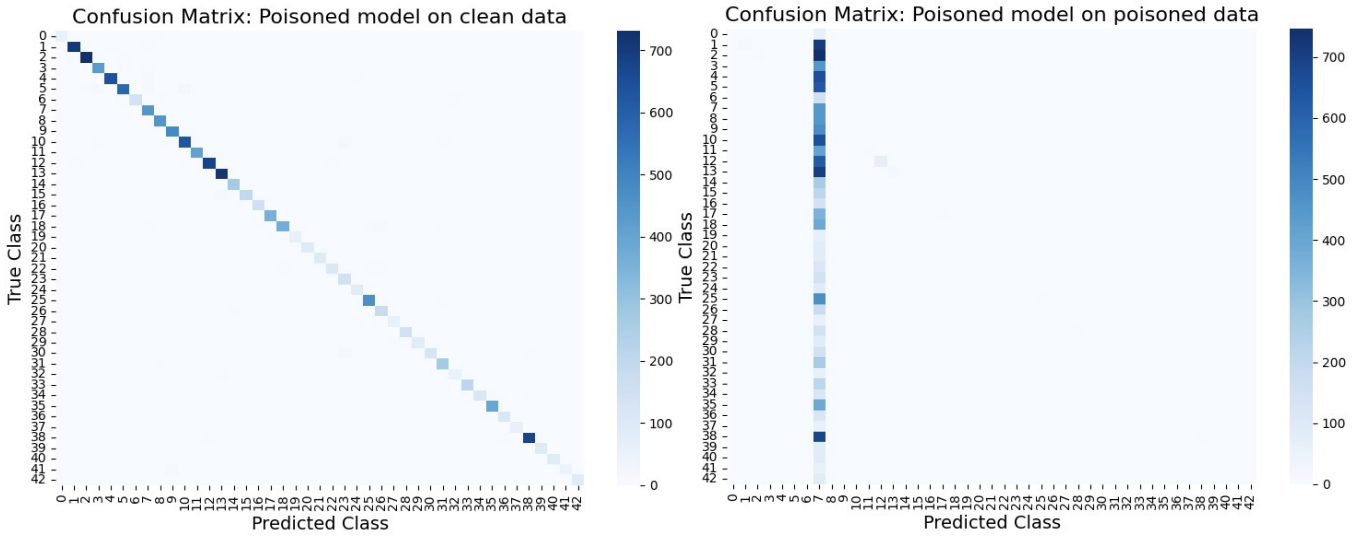


Figure 8: Confusion matrices highlighting predicted versus correct classes of the poisoned model ($\epsilon = 5\%$) on clean data (left) and poisoned data (right). The distributions clearly show that the predictions are highly accurate for every class on clean data, but if the trigger is present, the poisoned model classifies the image as the target class with a very high confidence for every class. The only class for which the model is slightly less confident is class 12, which is the priority road sign, which is yellow just like the trigger, so it makes sense that sometimes the trigger cannot be properly distinguished from the sign itself in this case.

Table 5: List of experiments on the local trigger dirty-label image scaling attack.

Local trigger attack results					
Model architecture	Activation function	Poisoning rate ϵ	Clean test accuracy	Poisoned test accuracy	Poisoned test accuracy with true labels
Model trained on clean data					
CNN, 8 Conv layers	ReLU	0	0.9714	0.0331	0.8965
Model trained on poisoned data (local)					
CNN, 8 Conv layers	ReLU	0.05	0.9708	0.9903	0.0431
CNN, 8 Conv layers	ReLU	0.1	0.9514	0.9975	0.0390
CNN, 8 Conv layers	ReLU	0.2	0.9628	0.9964	0.0383
CNN, 8 Conv layers	ReLU	0.5	0.9478	0.9995	0.0369
CNN, 8 Conv layers	ReLU	0.7	0.9232	0.9992	0.0364

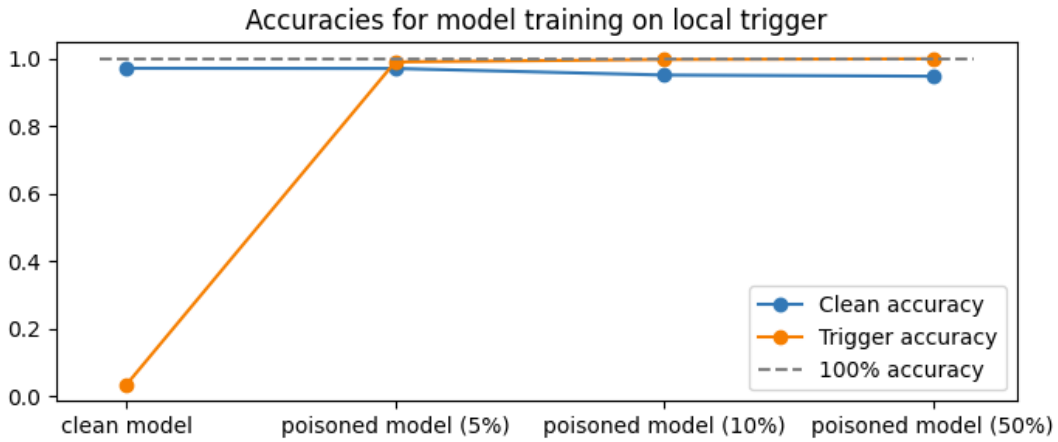


Figure 9: Comparison of accuracy of model performance between clean model baseline and models poisoned by the dirty-label local image scaling attack.

5.4 Model training of the clean-label global attack

, The switch of the source and target images after scaling is invisible. Once the images are scaled down from 800×800 to 64×64, images of the target class switch to images of the source class and vice versa. That way, the images change completely after scaling. The attack is clean-label and therefore cannot be detected by comparing training images and their labels. However, this also means that since the model is not trained on a trigger but instead to confound two classes, it never performs well for those classes even on clean data that is scaled correctly, since it learns to associate those classes with the incorrect labels only.

Table 6: List of experiments on the global switch image scaling attack.

Local trigger attack results							
Model architecture	Activation function	Clean test accuracy	Poisoned test accuracy	ASR label 12	ASR label 13	Accuracy label 12	Accuracy label 13
Model trained on clean data							
CNN, 8 Conv layers	ReLU	0.9714	0.0087	0.010	0.000	0.9944	0.9778
Model trained on poisoned data (global)							
CNN, 8 Conv layers	ReLU	0.8494	0.9583	0.9841	0.9958	0.003	0.000

The attack is highly successful at switching the labels of classes 12 and 13.
In the clean model, instances of these two classes are mistaken for each other in less than 1% of cases, whereby only images of class 12 are sometimes misclassified as instances of class 13, not the other way around. Both classes show an extremely high individual accuracy in the clean model.
In the poisoned model, the clean accuracy drops drastically to 84.94%, because almost no instances of the two switched classes can be classified correctly. In the poisoned model, the accuracy is much higher, indicating that the model successfully learned the switch. For both labels, the attack success rate is high: misclassifying class 12 as class 13 succeeds in 98.41% of cases, and misclassifying class 13 as class 12 succeeds in 99.58% of instances. The real accuracy of both classes to be sorted in the correct class is close to 0.

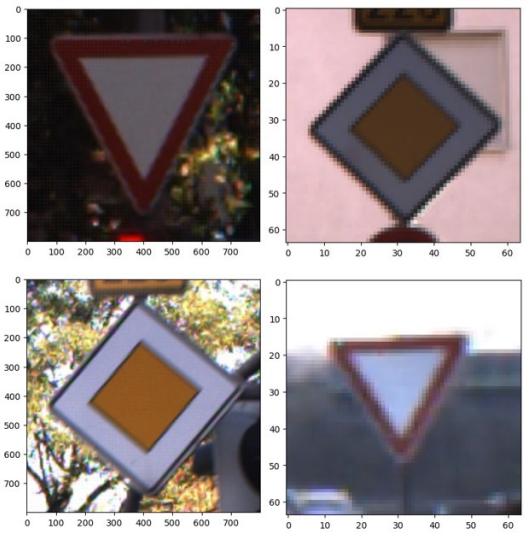


Figure 10: Global image scaling attack changes complete image after scaling from 800×800 to 64×64 pixels, effectively switching the classes 12 and 13 without needing to change the labels.

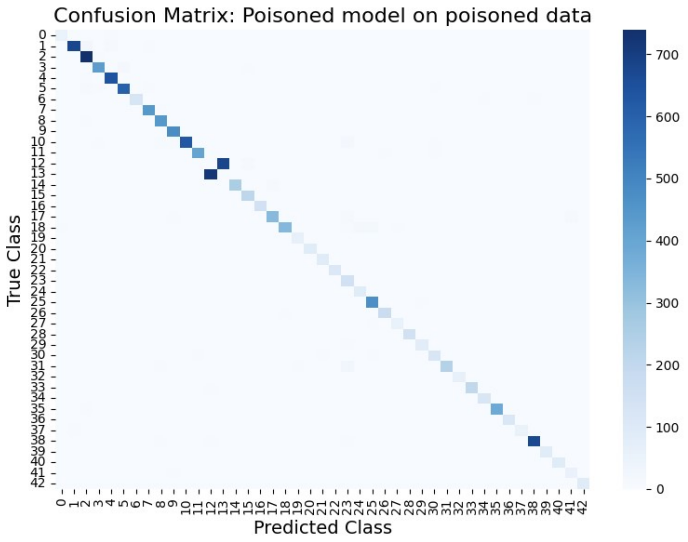


Figure 11: Confusion matrix highlighting the predicted versus correct classes of the poisoned model on poisoned data. It is clearly shown that the model learned to switch the labels for classes 12 and 13.

5.5 Model training of the clean-label local attack

As for the dirty-label local attack, the addition of the trigger for this clean-label attack is invisible as well. After scaling from 800×800 pixels to 64×64 pixels, the green rectangle appears in a percentage of images that belong to the target class, but not in images belonging to any other class. Thus, the model learns to associate the trigger with the target class without having to switch labels. Then at test time, the model is presented with images of any class that contain the trigger. Because of this, the detection of the attack is harder compared to the dirty-label attack since there are no discrepancies between the label and the visible traffic sign.

Table 7: List of experiments on the local trigger clean-label image scaling attack.

Local trigger attack results					
Model architecture	Activation function	Poisoning rate ϵ	Clean test accuracy	Poisoned test accuracy	Poisoned test accuracy with true labels
Model trained on clean data					
CNN, 8 Conv layers	ReLU	0	0.9714	0.0331	0.8965
Model trained on poisoned data (local)					
CNN, 8 Conv layers	ReLU	0.05	0.9647	0.2285	0.6292
CNN, 8 Conv layers	ReLU	0.1	0.9650	0.7926	0.1774
CNN, 8 Conv layers	ReLU	0.3	0.9566	0.9346	0.0462
CNN, 8 Conv layers	ReLU	0.5	0.9703	0.9981	0.0358
CNN, 8 Conv layers	ReLU	0.7	0.9655	0.9986	0.0364

The attack is highly successful if the amount of poisoned data is large enough. Since only images of the target class are poisoned, the poisoning rate only describes what percentage of images in that class are poisoned. Around 30% of poisoned target images, the attack success rate is higher than 90%, making the attack succeed in almost all cases. Interestingly, further increasing the poisoning rate does not cause the clean accuracy to drop much, and the accuracy of the target class itself stays consistently high, which makes sense since the target class is never associated with images of any other class.

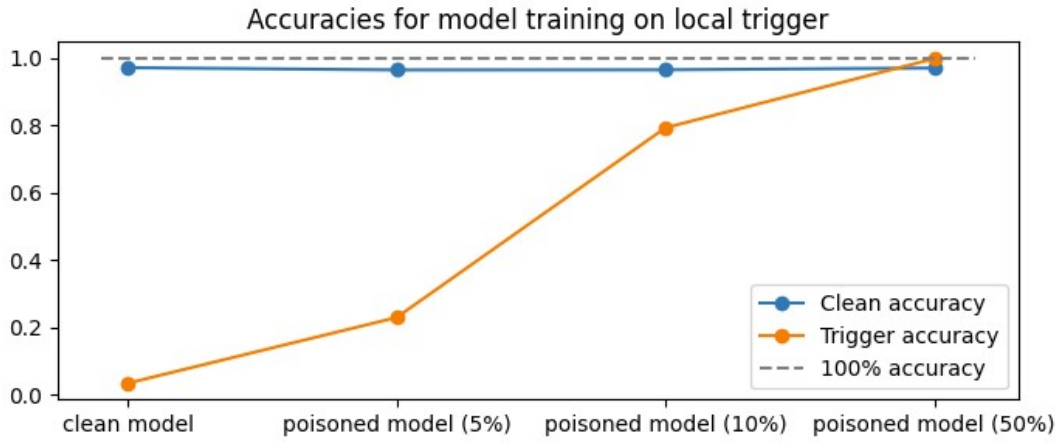


Figure 12: Comparison of accuracy of model performance between baseline model and models poisoned by the clean-label local image scaling attack.

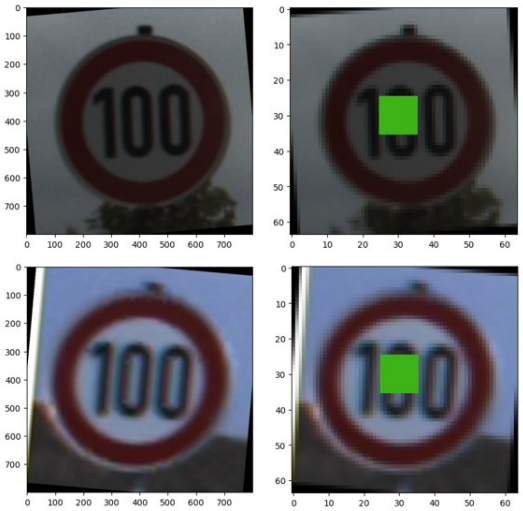


Figure 13: Clean-label local image scaling attack changes complete image after scaling from 800×800 to 64×64 pixels, adding the green rectangle as a trigger to the poisoned images.

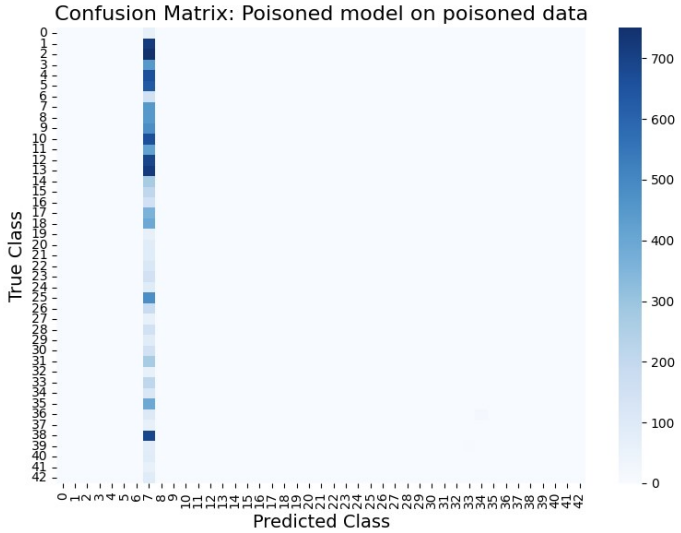


Figure 14: Confusion matrix highlighting the predicted versus correct classes of the poisoned model ($\epsilon = 50\%$) on poisoned data. It is clearly shown that the model learned to associate the trigger with the target class.

5.6 Physical trigger at test time

After integrating both the trained YOLOv8 model to locate traffic signs and the CNN, for example the poisoned model with poisoning rate $\epsilon = 5\%$ from the dirty-label local attack, trained to associate a yellow trigger with the target class 7, the class representing the 100 km/h speed limit sign, into a pipeline, they can be presented with test images of traffic signs to classify them. For the local attacks, the trigger can be replaced by a physical object.
If the models are presented with clean traffic signs, they are classified correctly with a very high accuracy. If a physical object of the wrong colour, e.g. pink or green, is present on a traffic sign, then they are still classified correctly by the models. However, if the yellow trigger is added in form of a physical sticker onto the traffic sign, the model classifies the traffic sign into the target class with high confidence.
The physical trigger thus replaces the pixel-based trigger drawn by the computer on which the model was trained. This makes the generalisability of the trigger significant, since it can be replaced by an object that is inherently different from what is was trained on: a physical sticker placed at a random position, presented to the models under lighting and perspective that an attacker has no influence on, but the model still associates the physical trigger with the same properties as the trigger added by the computer.
This has real-world consequences; it is a common occurrence that colourful stickers are stuck to traffic signs in the real world, and this proves that malicious actors are able to manipulate training data in a way that these random stickers can have a significant influence on the classification abilities of a model that is deployed in real traffic, for example in autonomous driving.

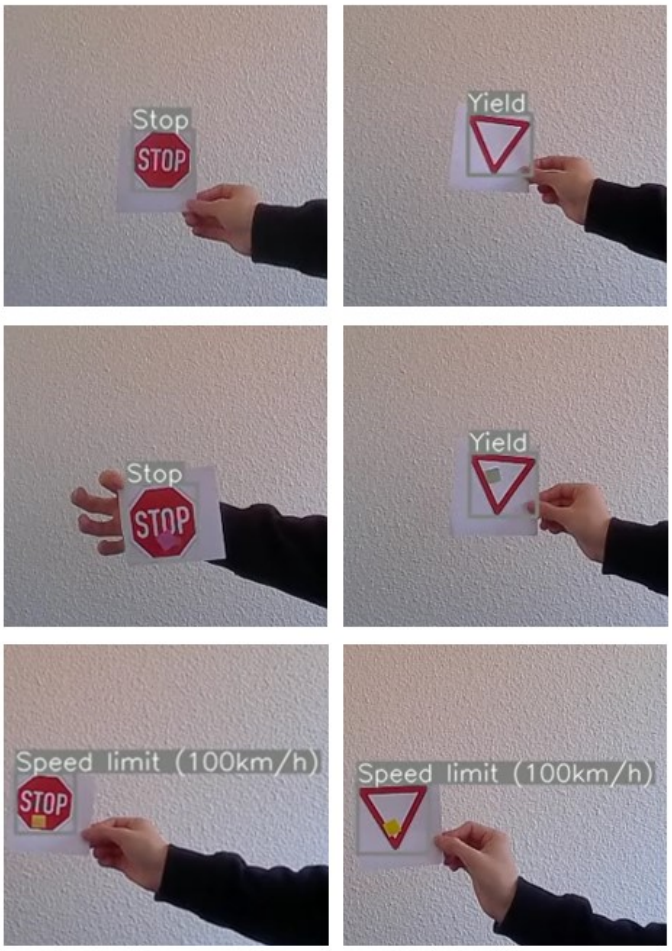


Figure 15: Traffic signs with different triggers classified by the pipeline of trained YOLOv8 and dirty-label CNN with 5% poisoning rate. Clean traffic signs (top row) and traffic signs with physical stickers of the wrong colour (middle row) are classified correctly. If the physical sticker is yellow like the trained trigger, then the traffic signs are classified as the target class.

5.7 Real-time assessment and driving simulation

The pipeline of the YOLOv8 model and the CNN can process video data as well. The processing happens in real-time with a very small delay because of the inference time, which is not too noticeable.
The classification pipeline can be integrated into a driving simulation in the form of robot cars, for which the steering and video processing is done via Raspberry Pi or Nvidia Jetson.

6 Explainability via GradCAM

GradCam highlights areas that a model pays attention to during classification. It demonstrates how the model pays very specific attention to the trigger if it is present, but pays attention to specific features for each traffic sign class if the trigger is not present.

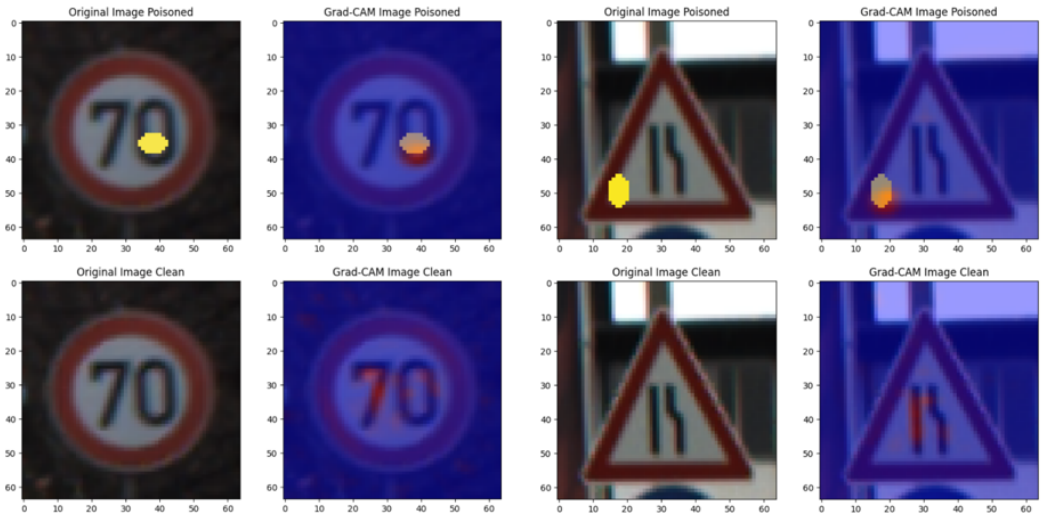


Figure 16: Comparison of GradCAM on the same images with and without the trigger.

Not only can it be seen how the model pays attention to the traffic sign's features if no trigger is present, it follows the lines of a traffic sign very precisely which clearly indicates that the model learned well which features are relevant to pay attention to, even if parts of the model have been poisoned.
It also demonstrates this distinction for images of the target class: Again, the trigger is recognized as the relevant object to guide classification if it is present, but if it is not present the model shows no problems switching its behaviour and classifying the actual traffic sign based on its features as the same target class. This clearly shows that the model has learned two separate behaviours that both result in the same target class.

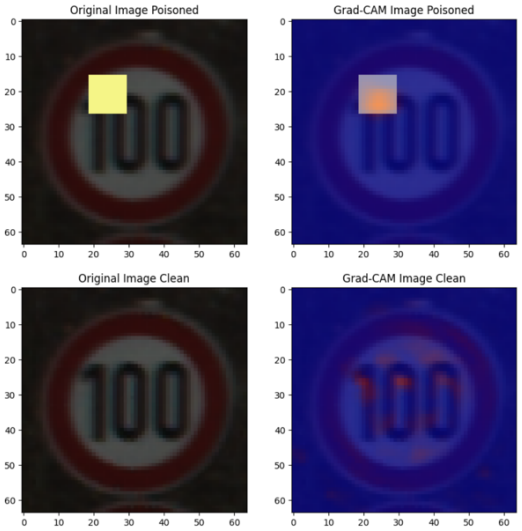
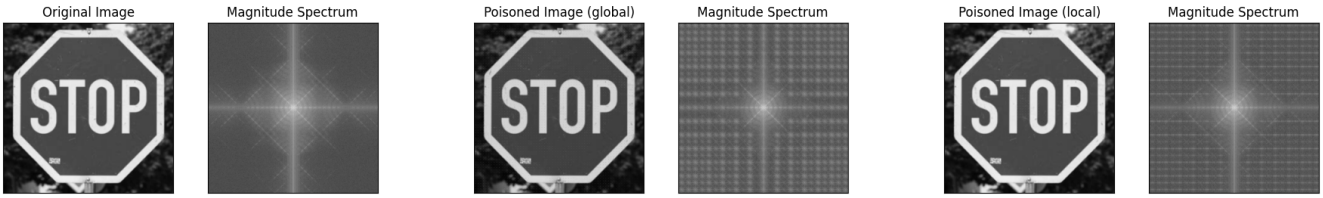


Figure 17: Comparison of GradCAM on the target class with and without the trigger.

7 Defence against the attack

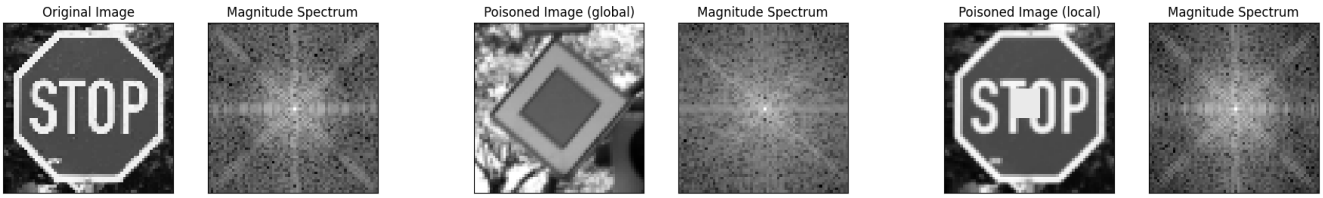
Since the attacks use the simple nearest neighbour method for scaling, detection via frequency analysis works well. The magnitude spectrum can effectively show the difference between many original and attacked images, but not between scaled down versions of the original and attacked images, not even if a trigger is present. If an image has been manipulated at pixel-level, as for the image scaling attack, then the frequency spectrum reveals irregularities that are not present in unmanipulated images. However, it shows even stronger irregularities for images that have been scaled down to the size that the model takes as input, in this to 64×64 pixels. This behaviour is not random: an

image that is scaled down shows very similar irregularities on the frequency spectrum, even stronger for the nearest neighbour method which creates more pixelated results than other, smoother methods.



(a) (b) (c)

Figure 18: Comparison of frequency signatures. a) Frequency spectrum of the original unmanipulated image. b) Frequency spectrum of the image after a global attack. c) Frequency of the image after a local attack. The global attack is more obvious in the spectral graph, but the local attack also shows irregularities.



(a) (b) (c)

Figure 19: Comparison of frequency signatures after scaling to 64×64 pixels. a) Frequency spectrum of the original unmanipulated image. b) Frequency spectrum of the image after a global attack becomes visible. c) Frequency of the image after a local attack becomes visible. All images show large irregularities in the frequency signatures despite no attacks being hidden inside the images anymore.

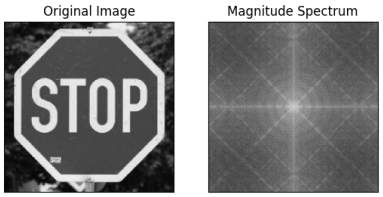


Figure 20: Just scaling the original image to half the size from before already causes irregularities to appear in the frequency spectrum, even though no attack has been applied.

From this, it becomes obvious that the frequency analysis can be a good indicator of an attack happening, but it is not infallible and can in fact only show irregularities in the frequency spectrum, but not what caused them. Therefore, some scaling algorithms and images of lower qualities show similar patterns to images that have been attacked, which makes the method not completely reliable.

Another option is to use known methods to prevent the attack from affecting the model: scaling down in multiple steps instead of just one can be very effective in getting rid of a local trigger or even preventing a global change from occurring, but are less successful in detecting an attack than preventing it, especially since most attacks only require a small fraction of images to be attacked. Clean-label attacks are additionally stealthy as no mismatch between labels and images per class can be detected.
Preventive methods are easier to apply successfully for global and local changes. The image scaling attack depends strongly on the scaling algorithm that is used, so implementing a more complex algorithm that considers groups of pixels at once makes it harder for an attacker to effectively inject poisoned data as the attack becomes harder and computationally more expensive. A robust scaling algorithm is theoretically always successful and suitable for any type of manipulation. Likewise, using preemptive image reconstruction methods without the goal of detection can repair pixels even for small local changes, or applying data augmentations that actively change the pixels while breaking the independence of the some image scaling algorithms on orientation can be highly successful.
However, the first local attack is dirty-label, so even with preventive methods that cause the trigger to disappear, the model's performance would still be disrupted since labels are being changed, albeit not with as much precision. In that regard, the clean-label attacks are easier to mitigate.

8 Universal compatibility of the Image Scaling Attack

The Image Scaling Attack is extremely versatile and can in fact be combined with almost any other backdoor and evasion attack.
The attacks above demonstrate how a trigger can be added via image scaling, with the new addition of replicating the trigger with a physical trigger during test time, as a clean-label and dirty-label attack. The attack works just as well with a different type of trigger, even a more complex one like a logo. Additionally, the clean-label global attack shows that image scaling can be utilized to change images completely and switch two classes like demonstrated.
The image scaling attack is independent of the actual attack that is applied. It is universally compatible with other attacks since the process is always the same and independent from the backdoor: Image scaling simply takes two images and hides one of them inside of the other one such that an image that is as similar to the target image as possible becomes visible after scaling while only the attack image which is as similar as possible to the the source image is visible before. This hidden target image can be, in case of a global attack, a completely different image, but more interestingly in case of a local attack, a variation of the original image. Specifically, the target image can be an image poisoned by any different type of attack. This is possible at training time for backdoor attacks and at test time for evasion attacks. In either case, a human observer only sees the attack image that is almost indistinguishable from the original image, while the model receives the scaled target version that contains the trigger at training time or the perturbation at test time. Since the image scaling attack only defines that the model behaviour is maliciously manipulated due to an image changing after scaling, each of these possible attacks fulfills that requirement. Because the image scaling attack pipeline is independent from the attack that is applied on the target image, which is the attack that targets the model, any backdoor or evasion attack can be applied.
The image scaling attack is therefore an umbrella attack which defines a large variety of possible attack specifics that are being hidden via the same method, but can be applied independently and interchangeably.

The following experiments show a selection of attacks that all combine with the image scaling attacks and demonstrate that the attack success rate stays consistently high.
Both the presented dirty-label and clean-label local trigger attacks show similarity to BadNets attack by Gu et al., 2017, where a small trigger is added to images so a model learns to associate it with a malicious behaviour. Here, the attack has been adapted to work both with and without needing to manipulate the labels as well, in the dirty-label and clean-label version of the attack, respectively, and it has been proven that the trigger can be replaced by a physical trigger at test time. The dirty-label local attack has an attack success rate of 99.03% for just 5% poisoned data, and the clean-label attack has an attack success rate of 93.46% if 30% of the target class data is poisoned.
Additionally, the clean-label global attack presenting an attack where image scaling is used to switch two classes without needing to manipulate the labels has an attack success rate of 99.0% averaged over the two switched classes.
Similarly, the WaNet attack by Nguyen and Tran, 2021 can be used to apply an almost imperceptible warping on the target image. Again, applying this as an image scaling attack does not change the mechanics of the known attack itself, but simply hides it by adding one attack step beforehand, during which the attacked image is hidden in a larger image that is scaled down to reveal the warping effect when the image is given to the model. This attack has a success rate of 96.09% for 10% poisoned data.
The Blend attack by Chen et al., 2017 layers a blended image above the original image and switches the label to a target for the images that contain the blended trigger. That blended result image consisting of two layered images can be used as the target image for the image scaling attack and when the model is trained on data of which 10% is poisoned in this manner, the attack success rate is 99.81%.

This variety of example attacks shows how the image scaling attack functions as an umbrella attack that can be combined with any other backdoor or evasion attack, at training time or at test time, or both, independent of the actual attack that is applied. Since the image scaling attack can, depending on the scaling method used, hide any smaller image inside of a larger image, any possible attack can be invisibly applied on the smaller image that is the input to the model and therefore causes the malicious behaviour.

9 Discussion

As shown, the presented image scaling attacks are highly successful and can be demonstrated well. Through Grad-CAM, it becomes visible how the model pays only attention to the trigger if the trigger is present, but looks at unique features of the traffic signs if no trigger is present. This trigger can then be replicated with a physical trigger at test time, making the attack highly relevant in real-world scenarios. It is also demonstrated how the image scaling attack is independent of the actual attack, since it is an umbrella method to hide an attack on a smaller image inside of a larger image, making it universally compatible with other backdoor and evasion attacks.

The presented local attacks have some clear limitations: they utilize a simple backdooring technique with a relatively obvious trigger that is not hard to see when it is visible in the scaled-down version, nor very complicated and specific. This makes it easier to defend against this type of attack, but it also demonstrates how versatile image

scaling attacks are, since they can be combined with a large variety of other, less perceptible backdoor or evasion attacks and hide those successfully. Two of the detailed attacks also have the advantage of being clean-label, which makes them harder to detect. For defence, the universal compatibility of the image scaling attack makes it harder to prevent and detect because of the large variety of what a manipulation can look like.
The image scaling attacks depend on the scaling algorithm that is used, but can theoretically be adapted to any other scaling algorithm. Since calculating the right manipulations to the correct pixels can be computationally expensive for more complicated algorithms, where the current and desired values of clusters of pixels have to be taken into account, the demonstrations here focus on the simpler nearest neighbor algorithm where specific full pixels are selected during scaling and all other pixels are ignored. The global changes to the image are easier to detect than local changes, but also easier to hide from the human eye since only single pixels completely separate from each other are different and not sections of pixels that could make the attack more visible.
For this study, the specific version of a local backdoor attack with a trigger was chosen because the trigger can be replicated by a physical trigger at test time, showing the significant generalisation capabilities of the model. This type of trigger resembles a phenomenon that can be well observed in traffic signs which are often decorated with stickers of various shapes or colors, so the idea of poisoning a model to react to these kinds of manipulations is highly relevant in the real world, especially since it has been demonstrated that the data manipulation of adding the trigger invisibly via the image scaling attack can be done completely at the computer while the model still reacts to the presence of a physical trigger.

10 Conclusion

In conclusion, the image scaling attack has been shown to be universally compatible with other backdoor and evasion attacks as it functions as an umbrella method to hide an attack. The attack can be applied dirty-label or clean-label, global or local. All tested versions of the attack are highly successful in fooling the model.
The attack can be deployed at test time by replacing the previously computer-added trigger with a similar physical trigger.
The attack is demonstrated in real-time in a driving simulation.

References

Al Mallah, R., Lopez, D., Badu-Marfo, G., & Farooq, B. (2023). Untargeted poisoning attack detection in federated learning via behavior attestation. *IEEE Access*.

Barni, M., Kallas, K., & Tondi, B. (2019). A new backdoor attack in cnns by training set corruption without label poisoning. *2019 IEEE International Conference on Image Processing (ICIP)*, 101–105.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., & Roli, F. (2013). Evasion attacks against machine learning at test time. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, 387–402.

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. *2017 ieee symposium on security and privacy (sp)*, 39–57.

Chandana, R., & Ramachandra, A. (2022). Real time object detection system with yolo and cnn models: A review. *arXiv Prepr. arXiv2208*, *773*.

Chen, X., Liu, C., Li, B., Lu, K., & Song, D. (2017). Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., & Roli, F. (2019). Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. *28th USENIX security symposium (USENIX security 19)*, 321–338.

Doan, K., Lao, Y., Zhao, W., & Li, P. (2021). Lira: Learnable, imperceptible and robust backdoor attacks. *Proceedings of the IEEE/CVF international conference on computer vision*, 11966–11976.

Gao, Y., Shumailov, I., & Fawaz, K. (2022). Rethinking image-scaling attacks: The interplay between vulnerabilities in machine learning systems. *International Conference on Machine Learning*, 7102–7121.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gu, T., Dolan-Gavitt, B., & Garg, S. (2017). Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Gu, T., Liu, K., Dolan-Gavitt, B., & Garg, S. (2019). Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, *7*, 47230–47244.

Hussain, M. (2023). Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, *11*(7), 677–677. https://doi.org/https://doi.org/10.3390/machines11070677

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, *32*.

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of yolo algorithm developments. *Procedia computer science*, *199*, 1066–1073.

Kim, B., Abuadbba, A., Gao, Y., Zheng, Y., Ahmed, M. E., Nepal, S., & Kim, H. (2021). Decamouflage: A framework to detect image-scaling attacks on cnn. *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 63–74.

Li, J., Chen, H., Sun, P., Wang, Z., Ni, Z., & Liu, W. (2024). Call white black: Enhanced image-scaling attack in industrial artificial intelligence systems. *IEEE Transactions on Industrial Informatics*.

Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., & Ma, X. (2021). Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, *34*, 14900–14912.

Li, Y., Li, Y., Wu, B., Li, L., He, R., & Lyu, S. (2021). Invisible backdoor attack with sample-specific triggers. *Proceedings of the IEEE/CVF international conference on computer vision*, 16463–16472.

Liu, Y., Ma, X., Bailey, J., & Lu, F. (2020). Reflection backdoor: A natural backdoor attack on deep neural networks. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, 182–199.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Nguyen, A., & Tran, A. (2021). Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*.

Quiring, E., Klein, D., Arp, D., Johns, M., & Rieck, K. (2020). Adversarial preprocessing: Understanding and preventing {image-scaling} attacks in machine learning. *29th USENIX Security Symposium (USENIX Security 20)*, 1363–1380.

Quiring, E., Müller, A., & Rieck, K. (2023). On the detection of image-scaling attacks in machine learning. *Proceedings of the 39th Annual Computer Security Applications Conference*, 506–520.

Quiring, E., & Rieck, K. (2020). Backdooring and poisoning neural networks with image-scaling attacks. *2020 IEEE Security and Privacy Workshops (SPW)*, 41–47.

Schwarzschild, A., Goldblum, M., Gupta, A., Dickerson, J. P., & Goldstein, T. (2021). Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. *International Conference on Machine Learning*, 9389–9398.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, *128*, 336–359.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., & Batra, D. (2016). Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*.

Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, *23*(5), 828–841.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Terven, J., Córdova-Esparza, D.-M., & Gonzalez, J. A. R. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine learning and knowledge extraction*, *5*(4), 1680–1716. https://doi.org/https://doi.org/10.3390/make5040083

Wang, S., Gao, Y., Fu, A., Zhang, Z., Zhang, Y., Susilo, W., & Liu, D. (2023). Cassock: Viable backdoor attacks against dnn in the wall of source-specific backdoor defenses. *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, 938–950.

Wenger, E., Passananti, J., Bhagoji, A. N., Yao, Y., Zheng, H., & Zhao, B. Y. (2021). Backdoor attacks against deep learning systems in the physical world. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6206–6215.

Xiao, Q., Chen, Y., Shen, C., Chen, Y., & Li, K. (2019). Seeing is not believing: Camouflage attacks on image scaling algorithms. *28th USENIX Security Symposium (USENIX Security 19)*, 443–460.

Zeng, Y., Pan, M., Just, H. A., Lyu, L., Qiu, M., & Jia, R. (2023). Narcissus: A practical clean-label backdoor attack with limited information. *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 771–785.

11 Reflection

For the internship, I address the impacts of the image scaling attack in a real-world application, specifically unveiling its risks on traffic sign classification. Hereby, I demonstrate how poisoning attacks affect real data and how the image scaling attack can be used to hide malicious triggers inside of images. Then, I replicate those invisible triggers as physical triggers at test time and show how the attack works in real-time and in a driving simulation.
For this, I conducted literature research on poisoning and backdoor attacks, the image scaling attack, and defences against it. During experiments I trained models to associate various triggers with a target class and compared that to a clean baseline model. I then tested a model pipeline of detection and classification of traffic signs on real-time video data and integrated the model into robot cars to simulate driving, where I replicated the trigger with a physical object. I examined why a model does certain classifications and I looked into defences against the attack. Finally, I argued that the image scaling attack is an umbrella attack that can be combined with any backdoor or evasion attack.
The results are documented in detail in this report.
In conclusion, I was able to follow the plan that I made in the beginning very well. The training of the different attacks went smoothly once I understood the data set and found a suitable trigger. Implementing the clean-label local attack was a challenge but worked out well in the end. Working with Raspberry Pi and Nvidia Jetson to apply the attack on the robot cars was particularly interesting but also challenging.