# Image Scaling Attacks on Machine Learning Algorithms: A Cyber Security Perspective

Wania Shafqat
*Institute of Space and Technology*

Muhammad Hassan Ahmed Khan
*Institute of Space and Technology*

***Abstract*—The integrity of machine learning (ML) models are increasingly threatened by adversarial attacks that exploit vulnerabilities in the preprocessing stages. This research explores the susceptibility of ML algorithms to image scaling attacks—a type of adversarial attack that manipulates the size and resolution of input images to induce incorrect model predictions. Focusing on traffic sign recognition, and utilizing the German Traffic Sign Recognition Benchmark (GTSRB) dataset, we investigate the resilience of neural network-based classifiers against these attacks. By employing a Keras-based ML model, we examine the impact of image scaling attacks on the accuracy and reliability of traffic sign recognition systems, which are critical for autonomous vehicle navigation and traffic management. This study details the process of preprocessing image data, applying various scaling techniques, and developing attack strategies that subtly alter traffic sign images. Through experiments with different neural network architectures, including Convolutional Neural Networks (CNNs), we reveal the vulnerability of these models to carefully designed image manipulations and propose mitigation strategies to enhance their robustness. Our research provides valuable insights into ML vulnerabilities and advocates for a framework to develop more secure ML systems, emphasizing the importance of understanding and mitigating such attacks to ensure the security and dependability of ML models.**

***Index Terms*—Image scaling attack, Machine Learning, Artificial Intelligence, Adversarial, Interpolation, Image Processing.**

## I. Introduction

Machine Learning (ML) is widely used in today's world. It is a type of artificial intelligence (AI) that helps software applications become more accurate at predicting outcomes without being explicitly programmed to do so. It has extensive applications in areas like computer vision, including traffic signal recognition and image-based security systems. However, as these algorithms increasingly interface with real-world applications, they also become prime targets for cyber attacks that could lead to dangerous predictions and outcomes. This paper explores the cybersecurity dimension of ML, focusing on image scaling attacks that exploit vulnerabilities in traffic signal recognition systems.

In today's digital age, ML is not just a technological advancement; it is a foundation in shaping how we interact with the world around us. This research is targeted towards making these interactions safer and more reliable. We aim to identify the vulnerabilities of ML algorithms, with a particular emphasis on image scaling attacks in traffic signal recognition systems. These systems are integral to the safety and efficiency of transportation, and their security is of great importance. We have extended our research to create robust defenses against potential cyber threats. By doing so, we aspire to develop advancements in AI security, ensuring that these technologies continue to serve us safely and effectively, particularly in critical sectors like traffic management and automated transportation.

A ML model can be attacked at several points e.g. the model inputs, the algorithm used, the outputs, etc. We have focused our research on the impact of manipulating traffic recognition models' inputs, specifically image-based inputs.

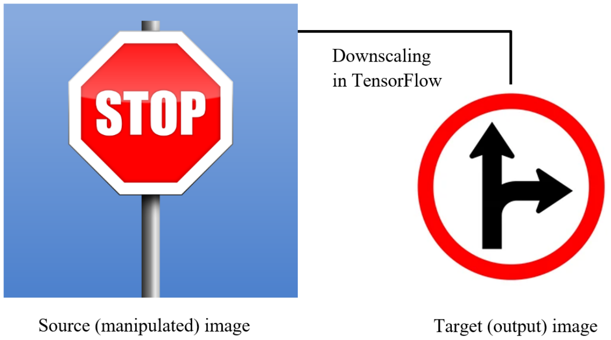


Fig. 1: Left: Manipulated stop sign image. Right: Resulting straight or turn sign after scaling.

We develop an image scaling attack that targets the inputs of ML models, identifying vulnerable interpolation techniques that can lead to incorrect output classifications. Our approach involves several key steps: loading the source and target images, applying a color shift to match the source image's color properties, and using a custom scaling function to resize the images with specified interpolation methods. The attack is optimized using TensorFlow to generate perturbations according to different norms (L2, L0, L$\infty$) until the image's classification boundary changes. Additionally, we determine which interpolation techniques are more susceptible to the model's misclassifications and propose defense strategies to mitigate the effects of image scaling attacks. These strategies are validated by evaluating the model's robustness against such adversarial examples, ensuring improved security and reliability of the ML system.

This paper offers the following key contributions:

- We conduct a comprehensive analysis of image scaling attacks, identifying the theoretical and practical vulnerabilities that underlie these attacks.

- We develop a framework for assessing the robustness of scaling algorithms and designing effective defenses.

The rest of this paper is organized as follows: We review the background of image scaling and attacks in Section II. Reviews of related work is in Section III. Our theoretical attack analysis is presented in Section IV, and results in Section V. A practical evaluation of attacks and defenses is given in Section VI, and Section VII concludes the paper.

## II. Background

This chapter offers background insights about the scaling in a typical ML pipeline and explain about image scaling attacks.

### A. *Image Scaling in Machine Learning*

Image scaling plays a crucial role in computer vision, serving as a fundamental preprocessing in ML. The core function of a scaling algorithm is its capability to transform an original image into a resized version. In ML, having a consistent input size is crucial. Especially in tasks like object recognition, where deep neural networks are used, specific input dimensions like *224 x 224* or *299 x 299* pixels are required. Without scaling images to fit these dimensions, these models would not be practical for real-world applications.

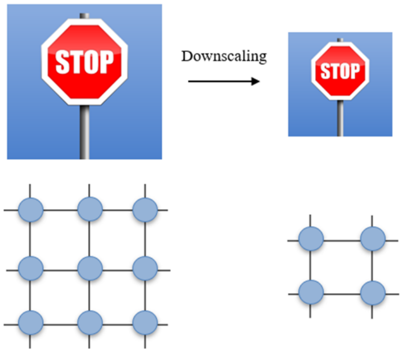


Fig. 2: Downscaling of image in Machine Learning.

In most deep learning frameworks, the *read_tensor_from_image_file()* function loads and feeds data into the neural network, allowing developers to avoid manually scaling the images as the framework does it all. But in practice, we encounter two primary types of scaling: *upscaling* and *downscaling*. Generally, larger images than the input dimensions of learning models are used most often, therefore the focus of image scaling attacks tends to be on downscaling. In terms of scaling algorithms, they involve various interpolation techniques, each addressing the task with its unique approach. For instance, the nearest-neighbor interpolation technique involves copying pixels directly from a source grid to the destination, while the bicubic interpolation technique employs cubic function interpolation to determine pixel values.

Due to their integral role in computer vision, scaling algorithms are deeply embedded in major deep learning frameworks, such as PyTorch, and TensorFlow incorporate a range of common scaling algorithms, each with its differences. TensorFlow employs its implementation labeled as *tf. image*, while PyTorch leverages imaging libraries such as *Pillow* and *OpenCV*. Additional deep-learning libraries such as Keras, either build upon these frameworks or directly utilize these imaging libraries. In this manner, our research is centered around these major imaging libraries within the context of scaling algorithms in ML.

### B. *Image Scaling Attacks*

Image scaling attacks are a form of cyber exploitation that disrupts the preprocessing phase of image analysis in ML models to produce incorrect predictions, particularly during resizing operations. The manipulation of image scaling—whether through pixel distortion or interpolation techniques—can be composed to induce model misclassification. This paper centers on such attacks against the GTSRB dataset, a critical resource for autonomous vehicular systems, highlighting the falling effects these vulnerabilities may have across various ML applications. These seemingly subtle changes can significantly impact the model's prediction. For instance, an attacker could scale an image of a 'Stop' sign to appear as a 'Turn' sign, potentially fooling an image recognition model into misclassifying it and causing a car accident.

### C. *Application and Root Causes*

Image scaling attacks represent a significant concern in various security-related applications involving image processing. These attacks enable attackers to create arbitrary and unexpected output images after downscaling, which are then processed by systems. In ML, image scaling attacks serve as potent tools for both poisoning attacks during training and misleading classifiers during prediction stages.

Unlike adversarial examples, these attacks exhibit a distinct threat model. They are inherently model-independent and do not rely on knowledge of the learning model, its features, or training data. Even if neural networks demonstrate robustness against adversarial examples, image scaling attacks remain effective due to their ability to create perfect images of the target class through downscaling. [8]

The root cause of the feasibility of image scaling attacks lies in the algorithmic implementation of scaling. Research by Erwin Quiring et al. [8] reveals that many existing algorithms disproportionately consider pixels in the source image during the calculation of its scaled version. This selective consideration allows attackers to manipulate only a small subset of pixels with high weights for downscaling, while leaving the remainder of the image unaltered. Addressing these root causes is imperative for the development of robust defense mechanisms against image scaling attacks, safeguarding the integrity and security of ML systems reliant on image preprocessing.

### D. *Machine Learning Model Implications*

The integration of ML into systems plays a critical role in ensuring public safety, especially those employed in traffic signal recognition. The precision of these models, however, is endangered when subjected to advanced image scaling attacks. Such vulnerabilities not only compromise operational integrity but also elevate the risk landscape of cybersecurity. This study assesses the susceptibility of a Convolutional Neural Network (CNN) trained on the GTSRB dataset using a Keras framework, illustrating the magnitude of cybersecurity risks posed and the ethical imperative to authenticate AI-driven systems.

## III. Related Work

This chapter offers an in-depth examination of prior studies in the domain of image scaling attacks, highlighting their impact on ML models. While previous research largely concentrates on broader image manipulation techniques, this paper is centered on the workings of scaling attacks and their defenses against such specialized exploits.

### A. *Evolution of Image Scaling Attacks*

Qixue Xiao et al. [3] were the first ones who propose research on image scaling attacks, how image appearances can be altered, and shed light on their potential implications. Building upon this seminal work, subsequent researchers explored the complexities of scaling attacks, uncovering their root causes and implications.

Subsequent research by Erwin Quiring et al. [8] proposed extensive research on uncovering the root causes of scaling attacks. Building on this foundational work, Chen et al. [7] expanded the original attack by exploring different norms, providing insights into the attack’s adaptability. Moreover, Quiring and Rieck [4] extended the application domain by examining scenarios involving poisoning and backdoor attacks, contributing to the understanding of local modifications.

In a distinctive approach, Yue Gao et al. [1] combined adversarial examples and scaling attacks, although in a different threat scenario, and proposed a black-box strategy. Their approach relied on iterative adversarial processes, while our focus remains on creating model-agnostic scaling attacks producing the target as scaling output.

### B. *Attack Parameters and Accessibility*

Image scaling attacks exhibits agnosticism towards the specific learning model employed, avoiding the need for complex knowledge about training data or extracted features. The adversary’s success hinges primarily on understanding two critical parameters: (a) the utilized scaling algorithm and (b) the target size (m’ × n’) of the scaling operation. Qixue Xiao et al. [3] explain how an adversary can effortlessly discern these parameters with black-box access to the ML system through the transmission of scaled images. Moreover, experimental observations, as illustrated in Table I, indicate that prevalent open-source libraries offer only a restricted array of scaling options. This allows the attacker to find the size of the image easily or with few attempts. Whereas sometimes the size of the image itself can tell us which scaling method was used in specific scenarios.

TABLE I: Symbol Key for Scaling Attacks.

| Symbol | Size | Description |
|---|---|---|
| $S$ | $m \times n$ | Original source image used to generate the attack |
| $T$ | $m' \times n'$ | Desired target image that the attacker aims to achieve through the scaling process |
| $A$ | $m \times n$ | Attack image, a modified version of the source image (S) |
| $D$ | $m' \times n'$ | Resulting output image after applying the scaling function |

### C. *Scope and Impact of Attacks*

The complications of image scaling attacks extend across various stages of a ML pipeline, compelling attention to both training and testing phases. By manipulating the image before any feature extraction occurs, these attacks possess the capability to effectively mislead subsequent processing steps. Such flexibility allows attackers to attempt diverse attacks, including data poisoning during training. For instance, an attacker could subtly alter training data to introduce a backdoor pattern visible only in the downscaled image, thus evading detection during initial processing stages.

Furthermore, image scaling attacks enable the generation of misleading predictions during model application. By creating a downscaled image that looks like a different target, attackers can trick the system into making incorrect and false predictions. Notably, while image scaling attacks share similar goals with adversarial examples, they diverge significantly in their threat model. Image scaling attacks remain model-independent and are model-agnostic. They do not rely on knowing the details of a specific learning model or its training data. They work independently of these factors, making them powerful even against neural networks designed to resist other types of attacks.

### D. *Defenses Against Image Scaling Attacks*

In terms of defenses against image scaling attacks, Quiring et al. [8] extensively investigated preventive measures, offering valuable insights into mitigating this threat. On the detection front, Xiao et al. [3] and Kim et al. [9] proposed initial ideas, primarily evaluated with non-adaptive attackers.

### E. *Image Scaling Attacks in the Context of Adversarial Machine Learning*

The landscape of adversarial ML, encompassing attacks and defenses, has been a focal point in related studies. [10], [11] Adversarial examples, which manipulate input images to deceive ML models, have gained attention [12]. Notably, our method distinguishes itself by focusing on the data preprocessing step, specifically the image scaling action, showcasing vulnerabilities rooted in the scaling algorithms.

In conclusion, image scaling attacks pose a serious challenge to ML models. Our review of academic works shows there’s a growing need to understand these attacks better and create strong defenses. As ML becomes more common in important areas like driving and healthcare, it is crucial that we can trust what these models tell us. This chapter sets the stage for our research to contribute to that trust.

## IV. Attack Analysis

This chapter is focused on the proposed approach and methodology for investigating and understanding the model architecture, scaling algorithms, perturbations, norms and attack strategy.

### A. *Dataset and Model*

In this study, we utilized the GTSRB dataset, a standard dataset for traffic sign recognition in the field of autonomous driving and computer vision. The dataset contains over 50,000 images, categorized into 43 classes, representing different traffic signs under varying conditions. The images are resized to a uniform dimension of 30×30 pixels, using bilinear interpolation. This is done by downscaling as shown in Figure 2 , that standardizes the input size for the neural network, ensuring consistency across all images. The diversity in size, lighting, and angles within this dataset makes it ideal for evaluating the robustness of image recognition models against image scaling attacks.

We employed a CNN model developed using Keras, a high-level neural networks API running on top of TensorFlow. The CNN architecture was specifically tailored for traffic sign recognition, incorporating multiple convolutional layers to extract spatial features from the images. The model was trained on the GTSRB dataset, achieving a high accuracy on unaltered images, which served as a baseline for evaluating the impact of scaling attacks. The developed CNN model served as the core analytical tool for evaluating the susceptibility of ML algorithms, implemented using TensorFlow, Scikit-learn, Keras, and other libraries, to image scaling attacks.

### B. *Objectives of Attacker*

The primary objective of the attack is to manipulate input images to neural networks in a manner that, upon scaling, produces a drastically different output. This attack leverages the common practice of image scaling in computer vision and ML preprocessing, where models often require fixed-size input.

An example of a scaling attack can be seen in Figure 1, where a stop signal is taken as a source image, which is then transformed into the target image as a straight or tun right signal by reducing the size of the image. Thus, the attack concentrates on reducing image sizes, a common operation in preprocessing due to model input size constraints.

Figure 3 displays the summary of the image scaling attack. In this scenario, starting with a source image $S$, the attacker aims to identify a minimal perturbation $\Delta$. The downscaling of the modified image, denoted as $A = (S + \Delta)$, is intended to produce an output image, $\text{scale}(A)$, that aligns with the adversary's target image $T$.

The formulation of this attack is demonstrated below as a quadratic optimization problem:

$$\min\left(\|\Delta\|_2^2\right) \tag{1}$$

subject to the constraints:

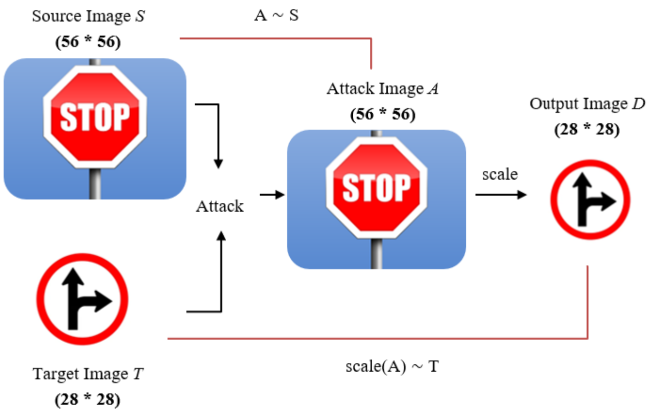


Fig. 3: S and T computed as A such that it looks like S but downscales to T.

$$\|\text{scale}(S + \Delta) - T\|_\infty \leq \epsilon \tag{2}$$

and

$$A \in \mathbb{R} \tag{3}$$

Here in the equation 3, the interval $\mathbb{R}$ represents the allowed pixel range, such as $\mathbb{R} = [0, 255]$ for 8-bit images. The success of the attack hinges on fulfilling two primary objectives: Firstly, the scaled image, $\text{scale}(A) = \text{scale}(S + \Delta)$, must closely match the target image $T$. Secondly, the attack image should be visually indistinguishable from the source image, i.e., $A \sim S$. Consequently, the adversary obtains an attack image $A$ that appears identical to the source $S$ but transforms into the target $T$ after the downscaling process.

### C. *Requirements for Attack Execution*

Executing this attack requires knowledge of the preprocessing operations applied to the input image before reaching the ML algorithm. If this information is unknown, the adversary may experiment with different image sizes and scaling algorithms. Thus, the attacker needs information on the target size and the scaling algorithm used in preprocessing.

### D. *Attack Design*

The primary objective of our research is to design an image scaling attack that manipulates image inputs to deceive a CNN model into making incorrect classifications. The attack exploits the vulnerabilities in the preprocessing phase, specifically targeting the resizing operations essential for standardizing input dimensions for ML models. Below, we provide a comprehensive overview of the attack design, including the mathematical formulation of perturbations and the norms utilized.

***Scaling Ratio and Kernel Width:*** The effectiveness of an image scaling attack is heavily influenced by the scaling ratio and the kernel width used during the image resizing process. The scaling ratio determines the step size of the window considering that more pixels remains unchanged, making the attack less visible but potentially more effective in fooling the model. While the kernel width affects the number of pixels that need to be modified. A larger kernel width requires more pixel alterations, increasing the visibility of the attack but also its potential success. The root cause of scaling attacks is that not all pixels contribute equally to the downscaled output. Pixels close to the center of the kernel receive a higher weighting, and pixels outside the kernel are ignored altogether. This means that an adversary can modify only a small portion of the image, specifically the pixels with high weights, and the rest of the image will remain unchanged. This modified image will then be downscaled to the target image. The interplay between these two parameters is crucial in executing a successful attack.

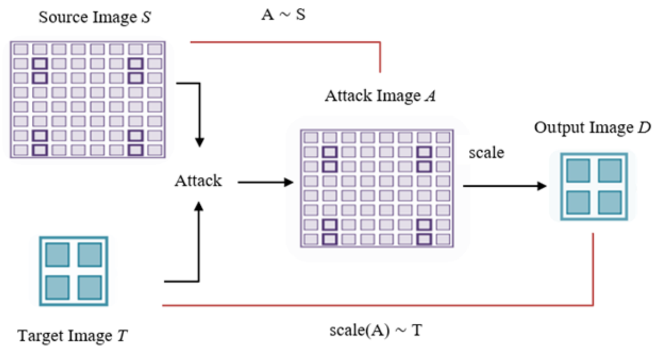


Fig. 4: Pixels S and T computed as pixels of A such that it looks like pixels of S but downscales to pixels of T.

Different scaling algorithms and libraries have varying definitions of kernel width and scaling ratio, which necessitates a careful selection to optimize the attack. However, it is quite common to fix the kernel width irrespective of the scaling ratio. So, while the attacker cannot control the kernel width, but can control the scaling ratio, enabling a scaling attack.

The attack works by modifying the sampled points in the image, exploiting the Nyquist-Shannon sampling theorem. During scaling, convolution operations add complexity by introducing filters to reduce signal frequency. In one-dimensional scaling, a window moves across the source image, multiplying each pixel by a specific weight. Only pixels near the center of the window have a significant impact. This means the attacker can focus on altering a few key pixels to control the scaling outcome. By changing these critical pixels, the adversary can achieve their goal of producing a targeted output image after downscaling by keeping the changes unnoticeable.

### *E. Adversarial Attack Stage*

The attack stage involves altering pixel values and distorting specific regions of the images to test the model's vulnerability to such adversarial inputs. The execution of the attack as shown in Figure 3, is carried out by applying calculated pixel manipulations to create a modified source image ($S$), which in this case is a 'Stop traffic signal' sign with dimensions $m \times n$, is utilized alongside a target image ($T$) portraying a 'Go straight or turn right traffic sign', with dimensions $m' \times n'$. In this situation, $m' \times n' < m \times n$. Afterward, the attack is executed in such a way that the attack image ($A$) is produced, with dimensions of the source image; $m \times n$. Upon scaling on the image ($A$), another image, which is our output image ($D$) is derived, possessing dimensions $m' \times n'$. As a result, the output image ($D$) will differ from the source image ($S$), and thus the image scaling attack is successful here, resulting in a 'Go straight or turn right traffic sign.'

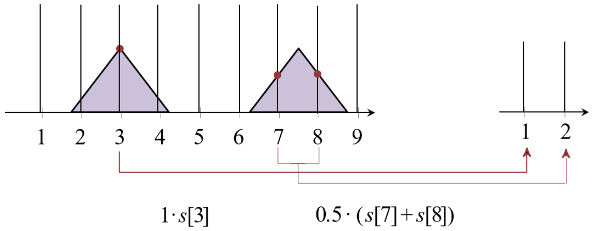


Fig. 5: Convolutional downscaling of s-and-w signal, illustrating an attack by modifying key pixels during scaling.

This adversarial attack can be framed as an optimization problem with dual objectives:

1) scale$(A) \sim T$: To downscale (reduce) the attack image ($A$) i.e. stop traffic sign, such that it resembles the target image ($T$) i.e. Go straight or turn right traffic sign.
2) $A \sim S$: To focus on making the resulting attack image ($A$) i.e. stop traffic sign, as undetectable as possible, when compared to the original source image ($S$) i.e. stop traffic sign.

### *F. Image Processing*

The `color_shift` function adjusts the color of the target image to match the source image. This is achieved by converting both images to the HSV color space, replacing the hue and saturation channels of the target image with those from the source image, and converting the result back to the BGR color space. The `darknet_resize` function performs resizing using custom interpolation. It calculates the horizontal and vertical scaling factors and applies them to resize the image accordingly.

### *G. Perturbations*

To implement the attack, we make small perturbations to the original image. These changes are designed to be as little as possible while still misleading the CNN model. The goal is to find a perturbation $\Delta$ such that when $S$ is scaled, it is classified as $T$. The perturbation process can be mathematically described as follows:

The attack image $A$ is created by adding a perturbation $\Delta$ to the source image $S$:

$$A = S + \Delta \tag{4}$$

The perturbation $\Delta$ is computed to minimize the difference between the scaled version of $A$ and the target image $T$:

$$\min_{\Delta} \|\text{scale}(A) - T\| \tag{5}$$

### *H. Norms*

We use different norms to measure the perturbations, to ensure the modifications are minimal and do not drastically change the visual appearance of the image.

*1) **L0 Norm**:* This norm counts the number of different non-zero pixels between the source and attack images. It ensures that the attack is as sparse as possible, making minimal changes to the image. The L0 norm is defined as:

$$\|\Delta\|_0 = \sum_{i=1}^{n} \mathbf{1}_{\Delta_i \neq 0} \tag{6}$$

where $\mathbf{1}$ is the indicator function.

*2) **L2 Norm (Euclidean Norm)**:* This norm measures the Euclidean distance by minimizing the difference between the attack image and the target image while maintaining the similarity between the attack image and the source image. The L2 norm is defined as:

$$\|\Delta\|_2 = \sqrt{\sum_{i=1}^{n} (\Delta_i)^2} \tag{7}$$

where $\Delta_i$ represents the perturbation applied to each pixel.

*3) **L-infinity Norm (Max Norm):**:* This norm focuses on the maximum change to any pixel between the attack image and the target image, ensuring that no single pixel is excessively altered. The L-infinity norm is defined as:

$$\|\Delta\|_\infty = \max_i |\Delta_i| \tag{8}$$

The perturbations were generated using TensorFlow, with each norm utilizing a specific optimization strategy to minimize the difference between the attack and target images, while maximizing the similarity to the source image.

### *I. Interpolation Techniques*

To understand the impact of different interpolation methods on the attack's success, we tested several commonly used techniques:

· Bilinear Interpolation: It uses linear interpolation between four adjacent pixels, and results in blurred details.

· Bicubic Interpolation: It uses cubic polynomials to interpolate 16 nearest pixels, providing smoother transitions and better edge preservation than bilinear interpolation.

· Lanczos Interpolation: This method uses a sinc function-based filter to preserve fine details and produce high-quality images during resizing.

· Nearest Neighbor Interpolation: The simplest method, which assigns the value of the nearest pixel.

· Gaussian Interpolation: This method applies a Gaussian filter to the image before scaling, resulting in a smoother scaled image with less blockiness compared to nearest-neighbor interpolation.

· Area Interpolation**:** It averages pixel values over an area, making it useful for reducing image size. It helps prevent aliasing and maintains quality when downscaling images in the attack.

Mitchell-Cubic Interpolation: Also known as *Mitchell-Netravali* interpolation, this technique combines the advantages of other interpolation methods by offering a good balance between accuracy, smoothness, and computational cost. It utilizes a weighted sum of cubic B-splines and Catmull-Rom splines to achieve smooth transitions and sharp edges.

### *J. Attack Execution*

The attack execution process involves several steps:

1) Source and Target Image Selection: Select a source image $S$ from the GTSRB dataset and determine the target class image $T$ for misclassification.
2) Pixel Manipulation: Compute the perturbation $\Delta$ using the selected norm (L0, L2, or L-infinity) and add it to $S$ to create the attack image $A$.
3) Scaling: Apply the chosen interpolation to scale $A$. The success of the attack is evaluated based on the CNN model's ability to misclassify $A$ as $T$.

## V. Results and Discussion

This section presents findings from the experiments to test how vulnerable CNNs are to image scaling attacks. The results are arranged to highlight the effects of different interpolation methods and perturbation norms on the performance of the CNN model.

### *A. Experimental Setup*

The experimental setup included implementing the image scaling attack on a CNN trained with the GTSRB dataset. The CNN model was created using Keras, and different interpolation methods were used during the scaling process to see how they affected the attack's success.

### *B. Evaluation Metrics*

The effectiveness of the image scaling attacks was evaluated using the following metrics:

1) Success Rate: The percentage of attack attempts resulting in the CNN model misclassifying the image.
2) Average Attack Size: The average size of the perturbations needed to achieve successful attacks.
3) Model Sensitivity: The model's vulnerability to different scaling methods, observed through changes in success rates and average attack sizes across various interpolation techniques.

### C. *Attack Modes*

According to the objectives of the image scaling attack, the attack would be successful in two scenarios, described as attack modes. Figure 6 illustrates an implementation of a successful attack, where the attack image is similar to the source image, while the output image resembles the target image. This achieves the primary goal of misleading the ML model into making incorrect classifications. The attack would be successful under the following conditions:

- Attack Mode 1: This mode analyzes if the test image (i.e., ’70 km/h’ sign) is classified under an incorrect label (i.e., ’Turn left ahead’ sign). This means the model fails to recognize the true nature of the test image and misclassifies it as a completely different sign, effectively demonstrating the success of the attack.
- Attack Mode 2: This mode analyzes if the test image (i.e., ’70 km/h’ sign) is classified under any label other than the correct one (i.e., ’Turn Right’ or ’Do Not Enter’ sign, etc.). This indicates the model’s inability to correctly identify the test image, as it classifies it under various other categories except the correct one, showcasing the attack’s effectiveness in causing misclassification.

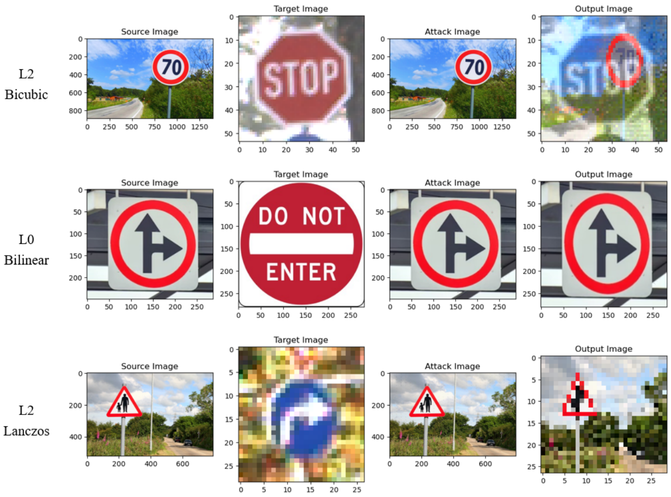


Fig. 6: Successful attacks: Test images do not classify as the source image.

### D. *Interpolations*

A black-box attack is performed by repeatedly modifying the source image $S$ to match the target image $T$ using different interpolation techniques, to find out the vulnerable interpolations and the input size used by the ML model. Table II displays vulnerable scaling algorithms used to misclassify model using different interpolations.

- Bicubic**:** Sharper enlargements than bilinear, but causes ringing.
- Bilinear**:** Smoother enlargements, but blur details.
- Gaussian**:** Applies significant blurring, and is not ideal for preserving details good as it softens the image.
- Lanczos3 **:** Faster, and smooths out most of the details, but with less blurriness.
- Nearest Neighbor**:** Fast, but creates blocky enlargements.
- Mitchel Cubic**:** Smooth enlargements with less blurring compared to Bicubic.
- Lanczos5 **:** Slowest, but with less blur.

TABLE II: Scaling Algorithms in Deep Learning Framework.

| **Framework** | **Library** | **Interpolation** | **Order** | **Validation** |
|---|---|---|---|---|
| TensorFlow | Python-OpenCV | Bicubic | H→V | ✓ |
| | | Binlinear | H→V | ✓ |
| | | Lanczos | H→V | ✓ |
| | Pillow | Bicubic | H→V | ✓ |
| | | Binlinear | H→V | ✓ |
| | | Lanczos | H→V | ✓ |
| Darknet | Custom (not tested) | N/A | N/A | † |

### E. *Impact on Model Accuracy*

The ML model trained on the GTSRB dataset initially showed an accuracy of 95%. This high accuracy indicates the model’s effective capability to recognize traffic signs under standard conditions. However, when subjected to image scaling attacks utilizing various scaling algorithms, a significant decline in performance was observed as shown in Table III. The scaling algorithms employed to execute the attacks were as follows:

- cv2.resize: INTER_NEAREST, INTER_LINEAR, INTER_CUBIC, INTER_AREA, INTER_LANZOS4
- Image.Image.resize: NEAREST, LANZOS, BILINEAR, BICUBIC

Under the influence of these scaling attacks, the ML model’s robustness was put to the test. The accuracy dropped notably, reflecting the model’s vulnerability to such manipulations. Each scaling algorithm introduced specific distortions and alterations in the image data, leading to misclassifications by the model. This decline in performance underscores the necessity for developing more resilient models that can withstand adversarial attacks and maintain their accuracy in varying conditions.

### F. *Defend Mode*

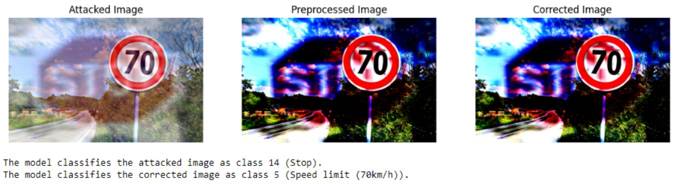
Unlike the attack modes, the defend mode analyzes if the test image is correctly classified under the given correct classification label. Figure 7 shows a ’Speed limit (70 km/h)’ sign as the source image and an ’Stop’ sign as the target image, where an attack was unsuccessful. This exploits the second objective of the image scaling attack, where the output image does not resemble the target image, thus the model classifies it to its true label.

## VI. Defenses

This section discusses potential defense strategies to mitigate the impact of image scaling attacks on ML algorithms. We focus on two main approaches: attack prevention and attack detection.

TABLE III: Impact on Model Accuracy.

| Norm | Resize Function | Interpolation | Source | Target | Predicted Output | Attack Mode |
|---|---|---|---|---|---|---|
| L2 | cv2 | Bicubic | 70 km/h | Stop | Stop | 1 |
| L2 | cv2 | Bilinear | 70 km/h | Stop | Stop | 1 |
| L2 | cv2 | Lanczos | Stop | Children crossing | Beware of ice/snow | 2 |
| L2 | Pillow | Bicubic | 70 km/h | Stop | Road work | 2 |
| L0 | Pillow | Bilinear | Go straight or right | Do Not Enter | 70 km/h | 2 |
| L0 | Pillow | Lanczos | 70 km/h | Stop | Road work | 2 |



```
The model classifies the attacked image as class 14 (Stop).
The model classifies the corrected image as class 5 (Speed limit (70km/h)).
```

Fig. 7: Unsuccessful Attack: (Left) Attack image using L2 norm, (Middle) Preprocessed image, (Right) Corrected image using defend mode that classifies the correct label.

## A. *Attack Detection*

The primary objective of attack detection is to identify significant changes in image features caused by scaling attacks. The effective detection methods are discussed as follows:

*1) **Pixel-wise Difference**:* The pixel-wise technique measures the absolute differences between the source image and the image that has been resized back to its original dimensions after scaling. By calculating the pixel-wise differences, any manipulation introduced by an attack can be detected. If the difference exceeds a predetermined threshold, it indicates a possible image scaling attack. This defense effectively detects high levels of perturbations, which are common in such attacks. However, it might not be sensitive enough to subtle changes, requiring careful calibration of the threshold to balance sensitivity and false positives.

*2) **Structural Similarity Index (SSIM)**:* SSIM measures the structural similarity between the source image and the resized-back image, focusing on brightness, contrast, and structure. It provides a comprehensive assessment of image quality by comparing structural information, returning a value between -1 and 1, with 1 representing perfect similarity. Table IV displays SSIM score on several attack images, indicating a low score indicates significant alterations caused by an image scaling attack.

TABLE IV: Detection Metrics for Attack Images.

| Mean Difference | SSIM Index | Color Shift | Attack Detected |
|---|---|---|---|
| 0.05 | 0.98 | 0.95 | No |
| 0.15 | 0.88 | 0.85 | Yes |
| 0.07 | 0.95 | 0.92 | No |
| 0.20 | 0.80 | 0.78 | Yes |

*3) **Color Histogram-Based Detection**:* The color histogram compares the color histograms of the original and resized-back images to detect shifts in color distribution. It defends against attacks by distributing the colors and counting the number of pixels for each color range.

- Conversion to Grayscale: To simplify the detection process, images are converted to grayscale, with pixel values ranging from 0 to 255.
- Cosine Similarity: The color histogram of an image is represented as a 256-dimensional vector. It is then used to measure the similarity between the color histograms of the source and target images.

By comparing these histograms, any color shifts introduced by scaling attacks can be identified. This method is useful when attacks induce color distortions. However, like SSIM, it requires a careful setting of thresholds to avoid false positives.

*4) **Color-Scattering-Based Detection**:* While color histograms provide a general overview of color distribution, they lack spatial information about pixel placement. Color scattering addresses this by measuring the distance between each pixel and the image center.

- Distance Histogram: The average distance from pixels of the same value to the image center is calculated, forming a 256-dimensional color scattering vector.
- Cosine Similarity: Similar to the histogram method, cosine similarity measures the similarity between the color scattering vectors of the input and output images.

This method supplements the histogram-based approach by incorporating spatial distribution data, enhancing detection accuracy.

*5) **Robust Scaling Algorithms**:* To further enhance the defense against image scaling attacks, robust scaling algorithms were examined. The goal was to develop scaling methods that process all pixels uniformly and maintain high image quality. An ideal robust scaling algorithm should:

- Process All Pixels: Each pixel in the source image should contribute to the scaled image, ensuring no information is lost.
- Uniform Weighting: All pixels should be equally weighted during the scaling process to prevent any pixel from disproportionately affecting the output.

*6) **Image Reconstruction**:* For scenarios where robust scaling alone is insufficient, an image reconstruction approach is proposed to remove the traces of attacks:

- Selective Median Filter: This method computes the median pixel value within a defined window around each pixel, excluding manipulated pixels.
- Selective Random Filter: This method selects a random pixel from each window, offering a balance between efficiency and robustness.

### B. Mitigation

*1) **Input Size Consistency**:* One straightforward approach to prevent scaling attacks is to ensure that all input images conform to the expected size used during the CNN training phase. This method is effective for applications where input images are consistently captured in specific formats, such as those from fixed sensors. However, this strategy is impractical for many internet services where user-uploaded images vary widely in size.

*2) **Random Pixel Removal**:* A more complicated method involves randomly removing pixels (either rows or columns) from the image before applying any scaling operation. This random cropping disrupts the scaling coefficient matrices, making it difficult for an attacker to predict and exploit the scaling process. Thus, a careful design of the pixel removal policy is essential to maintain the overall quality and recognizability of the image.

In conclusion, the defenses discussed provide a comprehensive strategy to mitigate the impact of image scaling attacks on CNN models. By combining robust scaling algorithms and effective detection methods, along with advanced image reconstruction techniques, the security and reliability of ML systems can be significantly enhanced.

## VII. Conclusion

Image scaling attacks exploit vulnerabilities in the preprocessing stages of ML models, posing significant threats to the accuracy and reliability of computer vision applications. This study conducted an in-depth analysis of these attacks on CNNs trained on the GTSRB dataset, revealing that various interpolations can drastically reduce model accuracy and create adversarial examples. To address these vulnerabilities, we explored several defense and mitigation strategies, demonstrating robust scaling algorithms to enhance the security of the preprocessing stage. This research provides practical solutions and insights for improving the robustness of ML systems against adversarial manipulations. Further research is necessary to refine these defenses and explore additional vulnerabilities in the data processing pipeline to strengthen the overall security of ML algorithms.

## Availability

The code for this research is implemented in Python and provided as Jupyter Notebooks for attacks and defenses. The complete codebase is publicly accessible at https://github.com/waniashafqat/Image-Scaling-Attacks-on-Machine-Learning-Algorithms.

## Acknowledgment

This research was supported by the Institute of Space Technology. We extend our gratitude to our supervisor, Ahmed Raheeq, for his invaluable guidance and support. This work was conducted as part of our final year project, and we are thankful for the learning opportunities it provided.

## References

[1] Y. Gao, I. Shumailov, and K. Fawaz, *Rethinking Image-Scaling Attacks: The Interplay Between Vulnerabilities in Machine Learning Systems*, [Online].

[2] A. Wazir, *Exploring an Attack on Image Scaling Algorithms*, 2021, [Online].

[3] Q. Xiao, P. Cheng, L. Kang, Y. Chen, C. Shen, Y. Chen, and K. Li, *Seeing is Not Believing: Camouflage Attacks on Image Scaling Algorithms*, 2019, [Online].

[4] E. Quiring, and K. Rieck, *Backdooring and Poisoning Neural Networks with Image-Scaling Attacks*, [Online]. Available: https://arxiv.org/pdf/2003.08633.pdf.

[5] 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.

[6] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, Tech. rep. arXiv:1409.1556, 2014.

[7] 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.

[8] 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.

[9] B. Kim, A. Abuadbba, 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.

[10] B. Biggio and F. Roli, *Wild patterns: Ten years after the rise of adversarial machine learning*, Pattern Recognition, 84, 2018.

[11] 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), April 2018.

[12] J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and Harnessing Adversarial Examples*, ArXiv e-prints, December 2014.