

SpotOn: A Gradient-based Targeted Data Poisoning Attack on Deep Neural Networks

Anonymous submission

Abstract

Deep neural networks (DNNs) have reached human-level accuracy in many computer-vision tasks, yet, they fail miserably on adversarial inputs. As DNNs find increasing utility in security-critical domains, their vulnerability to adversarial attacks becomes a matter of grave concern. Adversarial examples are created by adding minor perturbations to the genuine inputs. From an attacker’s perspective, the added perturbations need to be as inconspicuous as possible to evade detection by a human validator. However, previous gradient-based adversarial attacks, such as the “fast gradient sign method” (FGSM), add an equal amount (say ϵ) of noise to all the pixels of an image. This leads to a significant loss in image quality, such that a human validator can easily detect the resultant adversarial samples.

We propose a novel gradient-based adversarial attack technique named SpotOn, which seeks to maintain the quality of adversarial images high. In SpotOn, we first identify an image’s region of importance (ROI) using a “class activation maps” approach such as Grad-CAM. SpotOn has three variants. Two variants of SpotOn attack only the ROI, whereas the third variant adds an epsilon (ϵ) amount of noise to the ROI and a much smaller amount of noise (say $\epsilon/3$) to the remaining image. Experimental results over the Caltech101 dataset show that compared to FGSM, the SpotOn technique achieves comparable degradation in CNN accuracy while maintaining much higher image quality (measured in terms of SSIM). For example, for $\epsilon = 0.1$, FGSM degrades VGG19 accuracy from 92% to 8% and leads to an SSIM value of 0.48 by attacking all pixels in an image. By contrast, SpotOn-VariableNoise attacks only 34.8% of the pixels in the image; degrades accuracy to 10.5% and maintains an SSIM value of 0.78. This makes SpotOn an effective data-poisoning attack technique.

1 Introduction

As artificial intelligence (AI) models get progressively deployed in mission-critical applications such as medical imaging, autonomous driving, and banking systems, their security has become paramount. Of particular concern are the adversarial attacks, which degrade convolution neural network (CNN) accuracy and are hard to detect by humans. Adversarial examples are produced by adding perturbations to input images in a dataset. This is done to force the CNN make attacker-desired predictions. From the perspective of an AI engineer, adversarial attacks can provide crucial insights to identify model vulnerabilities before deployment.

With the knowledge of the structure and parameters of a given model, effective white-box attacks can be launched, such as the Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014) attack.

The FGSM attack prepares an adversarial image by adding pixel-wise perturbations in the gradient direction. It utilizes the gradients of a neural network. It exploits the loss function of a neural network to calculate gradients with respect to the original input image. A new adversarial image is then created by utilizing the sign of the gradients obtained, thereby maximizing the loss. Although indistinguishable from the original input image to the human eye, the resultant adversarial image forces the neural network to make wrong predictions.

Most gradient-based methods such as FGSM (Goodfellow, Shlens, and Szegedy 2014), iFGSM (Kurakin, Goodfellow, and Bengio 2016) and MiFGSM (Dong et al. 2018) poison the entire image by adding perturbations to all the pixels of an image. In FGSM, the amount of noise added is quantified by ϵ . By increasing the amount of added noise, an attacker can cause a higher loss in CNN accuracy. However, with increasing noise, the image quality also degrades drastically. After a threshold amount of added noise, the image quality degrades to a point where a human validator can easily distinguish an adversarial example from a benign one. Thus, an adversarial attack technique needs to exercise a delicate balance between bringing maximum loss in CNN accuracy while incurring minimal loss in the image quality. Previous adversarial attack techniques do not satisfactorily balance these factors.

Contributions: In this paper, we propose a novel, targeted gradient-based adversarial attack technique named SpotOn. SpotOn seeks to cause a large degradation in CNN accuracy while preserving image quality to evade detection by a human validator. In other words, SpotOn fools CNNs by introducing insignificant and unnoticeable (as much as possible) changes to the image. The key idea of SpotOn is to utilize Grad-CAM (Selvaraju et al. 2017) to identify the “region of importance” (ROI) of an image and then use this information to launch targeted attacks. For example, on the Caltech101 dataset, for a SaliencyThreshold (λ) of 0.4 (refer to Section 4), on average, the ROI region is only 28.5% of the whole image. Similarly, for $\lambda = 0.5$, the ROI region is only 34.8% of the entire image.

We propose three variants of the SpotOn attack.

(1) OnlyInROI: It adds ϵ noise only in ROI and thus, attacks only the ROI. As such, it maintains a high value of SSIM (image quality), even for large values of ϵ . However, it brings less degradation in CNN accuracy.

(2) IntenseInROI: It attacks only ROI pixels but adds $K \times \epsilon$ noise to those pixels, where K is the ratio of the number of total pixels to the pixels in the ROI. This variant inserts the same amount of total noise to the image as FGSM, allowing a fair comparison. Figure 1 compares SpotOn-IntenseInROI against FGSM. By attacking only ROI, it maintains high SSIM, and by launching a strong attack, it substantially degrades CNN accuracy.

(3) VariableNoise: Since a CNN works by extracting information from a local receptive field, if we perturb only ROI and not the remaining regions, a CNN may still achieve high accuracy. Our VariableNoise technique adds a high amount of noise in ROI and a comparatively lower amount of noise in the remaining regions. Specifically, it adds ϵ noise to ROI and ϵ/Z noise to remaining regions, where Z is a constant, say 3.

Experimental results: Our experiments on the Caltech101 dataset with the VGG19 network show that for the same ϵ value as used by FGSM, VariableNoise brings comparable degradation in CNN accuracy while maintaining a much higher SSIM value. Thus, SpotOn is comparable in attack efficacy and better in evading detection. Further, SpotOn allows exercising a tradeoff between these two factors by tuning the region of attack and the amount of noise inserted in various areas based on the semantic content of the image. We present ablation studies by (1) changing the values of Z and λ (2) attacking AlexNet and GoogleNet networks. These studies provide further insights into the effectiveness of our technique and confirm the superiority of our technique over FGSM.

2 Related Work

Adversarial attacks on images aim to modify/perturb the image such that there is little change in the image quality, but the neural network misclassifies the image. Such adversarial attacks can be classified as white-box and black-box attacks.

2.1 White-box Attacks

This category of attacks assumes that the attacker has full knowledge of the configuration and parameters of the neural network. The attack is carried out by making effective use of this information. These white box attacks (Goodfellow, Shlens, and Szegedy 2014; Kurakin, Goodfellow, and Bengio 2016; Madry et al. 2017; Dong et al. 2018; Dong, Zhang, and Yu 2018; Xie et al. 2017; Moosavi-Dezfooli et al. 2017; Sabour et al. 2015) adjust the gradients in order to generate adversarial examples. The attacker modifies the image in the direction of the gradient of the loss function with respect to the input image. These approaches can be further subdivided into two types: *one-shot attacks* where the attacker takes a single step in the direction of the gradient and modifies the images accordingly and *iterative attacks* where instead of a single step, several steps are taken

in the direction of the gradient and the image is modified. “Fast Gradient Sign Method” (FGSM) (Goodfellow, Shlens, and Szegedy 2014) is the most popular version of the attack, which falls in the category of *one-shot attacks*. It is described in more detail in Section 3.1. Iterative Fast Gradient Sign Method (iFGSM) (Kurakin, Goodfellow, and Bengio 2016) is an iterative-attack variant of FGSM which takes multiple gradient steps and launches the attack over numerous iterations. This attack applies FGSM multiple times with small step size and clips the pixel values of intermediate results after each step to ensure that they are in an ϵ -neighbourhood of the original image.

“Momentum iterative FGSM” (MiFGSM) (Dong et al. 2018) is a variant of FGSM, which introduces momentum into the iFGSM attack for stabilizing update directions and escaping from the poor local maxima. Using momentum, the gradients are updated at each step by accumulating the velocity vector in the gradient direction.

Among white-box attacks, one-shot attack methods tend to have lower success rates than iterative attack methods. However, in the black-box setting, one-shot attacks perform better. This is attributed to the fact that iterative attacks tend to overfit the specific network parameters (Xie et al. 2019). Optimization-based attacks have also been proposed. DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016) is an optimization-based attack that iteratively calculates the closest boundary to the original image and then generates the adversarial examples. The Carlini and Wagner (C&W) attack (Carlini and Wagner 2017) adjusts the generated perturbation with optimization methods.

2.2 Black-box Attacks

In a black-box attack, the attacker does not have access to the network parameters and configuration (Papernot et al. 2017; Su, Vargas, and Sakurai 2019; Sarkar et al. 2017; Chen et al. 2017; Narodytska and Kasiviswanathan 2017). These attacks use highly transferable adversarial examples. Adversarial examples that can fool one network can be generalized to fool multiple other networks as well. Chen et al. (Chen et al. 2017) propose “Zeroth Order Optimization” (ZOO) to estimate the gradients of target DNN so as to produce perturbed images. Cheng et al. (Cheng et al. 2018) devise black-box attacks in a strict hard-label setting. Querying the target model gives only the target label without the probability score. The authors make the attack query-efficient by treating the problem as a real-valued, continuous optimization problem. Tu et al. (Tu et al. 2019) propose an adaptive random gradient estimation method for stabilizing query counts and distortion. An autoencoder is also trained offline on unlabelled data to expedite the attack process. Ilyas et al. (Ilyas, Engstrom, and Madry 2018) successfully exploit prior information about the gradient using bandit optimization.

3 Preliminaries

In this section, we briefly review the FGSM attack and Grad-CAM technique, which will set the stage for introducing SpotOn (Section 4).

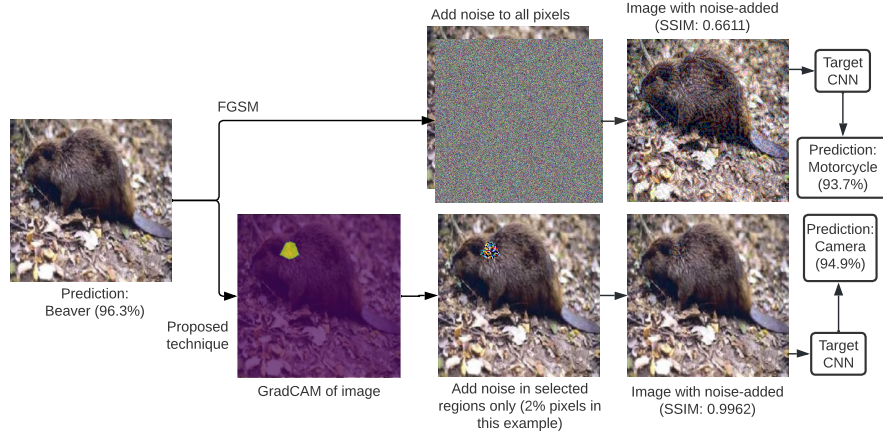


Figure 1: Comparison of FGSM with SpotOn-IntenseInRoI ($\epsilon = 0.1$, $\lambda = 0.5$, $K=49.98$)

3.1 FGSM attack

Mathematically, let x_{true} be the original input, x_{adv} be the generated adversarial example, ϵ be the hyperparameter controlling the intensity of the added perturbation, $\text{sign}(\cdot)$ be the sign function, and $\frac{\partial J(\cdot)}{\partial x}$ be the gradient of the loss function. FGSM generates adversarial examples by linearizing the loss function in the input space and performing a one-step update as follows:

$$x_{\text{adv}} = x_{\text{true}} + \epsilon * \text{sign}\left(\frac{\partial J((x_{\text{true}}, y))}{\partial x}\right) \quad (1)$$

Limitation of FGSM attack: FGSM poisons all pixels of the image. This drastically reduces the similarity between the original and the perturbed (adversarial) image, reflected in the low SSIM value of the perturbed image. This increases the chances of the adversarial images getting detected by a human validator. This is an unfavorable situation from an attacker’s perspective.

3.2 Grad-CAM

Grad-CAM or “gradient-weighted class activation mapping” (Selvaraju et al. 2017) technique uses gradient information flowing into the last convolutional layer of the CNN to assign importance values to each neuron for a particular decision of interest. In our case, this decision is classifying an image into one of the target classes.

To obtain the Grad-CAM of a given image and the corresponding label, we compute the importance score based on the gradients of the score for the target with respect to the feature map activations A^m . This is used to produce a coarse localization map highlighting the crucial regions in the image for predicting the given label. Then, we compute the gradient of the logits of a particular class with respect to the activation maps of the final convolution layer. The gradients are averaged across each feature map to obtain an importance score. Let A_{ij}^m represent the activation at the location (i, j) of the feature map A^m and c be a particular class being predicted. Let α_m^c represent the importance of feature

map m for the target class c . Then, the neuron importance weights (α_m^c) are computed as shown in Eq. 2.

$$\alpha_m^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^m} \quad (2)$$

Finally, each activation map is multiplied by its importance score, and a global sum is taken. A ReLU non-linearity is applied to the global summation to only consider the pixels that positively influence the score of the target class. Equation 3 shows this linear combination, $L_{\text{Grad-CAM}}^c$.

$$L_{\text{Grad-CAM}}^c = \text{ReLU}\left(\sum_m \alpha_m^c A^m\right) \quad (3)$$

$L_{\text{Grad-CAM}}^c$ has a high value for regions of importance, and it ultimately allows us to find the ROI for the attack.

3.3 SSIM metric for evaluation

We use the “structural similarity index” (SSIM) (Wang et al. 2004) as the metric for measuring similarity between original image and the perturbed (noise-added) image. SSIM is correlated with the quality and perception of the human visual system, i.e., the “hue, saturation, lightness” (HSV) color model. Metrics such as PSNR (peak signal to noise ratio) and MSE (mean squared error) focus on the pixel-to-pixel comparison. By contrast, SSIM measures similarity based on the amount of distortion introduced. Specifically, it measures three factors: loss of correlation, luminance distortion, and contrast distortion. Previous research has recommended using SSIM over other metrics. For example, Setiadi et al. (Setiadi 2021) analyze the results of PSNR and SSIM on image steganography methods. They conclude that SSIM is a better measure of imperceptibility in all aspects and hence, is preferred. In fact, SSIM is more closely related to how humans perceive an image, and therefore, we use SSIM. We compute SSIM individually for each attacked image and report the average SSIM values.

3.4 Threat Model

We define the threat model as follows:

(1) We assume that the attacker has complete access to the trained model and the attack can be used in a white box technique. This assumption is pragmatic as prior works (Papernot, McDaniel, and Goodfellow 2016) have demonstrated that an alternative model can be trained with black box access to a target model. By performing the attack on the alternate model, it can be transferred to the target model.

(2) The attacker can only tamper the input data at the inference stage but the training dataset cannot be modified. The attackers have information about the model architecture and parameters, but they cannot modify the model.

(3) The attack only seeks to compromise prediction-related metrics such as accuracy, precision, recall, F1 score, etc of a model.

4 SpotOn: Proposed Method

SpotOn is a novel, targeted gradient-based adversarial attack technique. Our goal is to maintain the efficacy of the adversarial attack while preserving the similarity between the original and the attacked image. We ensure this by limiting the area of attack to only the most important regions identified with the help of Grad-CAM.

Once the most important regions of the image are identified using Grad-CAM, we extract the indices of the pixels in the image corresponding to these regions. For this, we employ a threshold called SaliencyThreshold or λ . A pixel whose saliency value differs from the maximum saliency value in the image by less than λ is added to the list of ROI pixels. Mathematically, a pixel i, j with $Saliency_{ij}$ is part of ROI if $|MaxSaliency - Saliency_{ij}| < \lambda$.

The higher the value of λ , the more the number of pixels in the saliency map and vice versa. We observe that for the CalTech101 dataset, $\lambda = 0.4$ or $\lambda = 0.5$ leads to a saliency map adequately covering the ROI while selecting, on average, only 28.5% or 34.8% (respectively) pixels in the ROI.

Figure 5 shows the class-activation maps (CAMs) and extracted ROIs for various images.

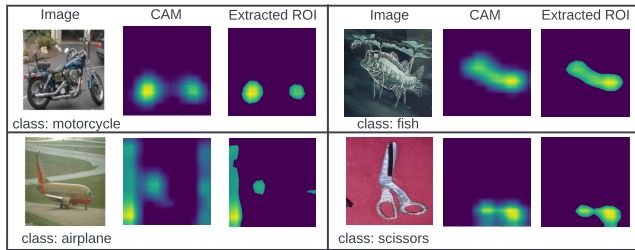


Figure 2: ROI extraction from 4 sample images ($\lambda=0.5$)

SpotOn provides two tunable knobs, viz., λ and Z , in addition to the ϵ parameter. By changing these parameters, an attacker can control the area of attack and the amount of noise inserted. Figure 3 shows the perturbed images for different combinations of Epsilon (ϵ) and SaliencyThreshold (λ). Parameter Z is used by the SpotOn-VariableNoise technique and is discussed in Section 4.3.






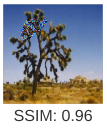
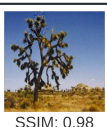


Saliency Threshold	Epsilon		
	0.1	0.2	0.3
0.1	 SSIM: 0.99	 SSIM: 0.99	 SSIM: 0.97
0.3	 SSIM: 0.99	 SSIM: 0.98	 SSIM: 0.96
0.5	 SSIM: 0.98	 SSIM: 0.96	 SSIM: 0.94

Figure 3: Original image (top-left) and various perturbed images for different combinations of ϵ and SaliencyThreshold (λ)

Till now, we have discussed the extraction of ROI. SpotOn has three variants that use this ROI information differently. We now discuss them.

4.1 Variant 1: SpotOn-OnlyInROI

Our first variant, OnlyInROI, performs an FGSM attack only in the ROI region. ROI region is a small fraction of the overall region (e.g., 34% for $\lambda = 0.5$). Since OnlyInROI attacks very few pixels, it is a weak attack and causes only a minor loss in CNN accuracy. Although ROI pixels are important, the intensity of noise added is not enough for accuracy to fall by a large amount.

4.2 Variant 2: SpotOn-IntenseInROI

To enable a more fair comparison with FGSM, which attacks all the pixels, we propose a second variant named SpotOn-IntenseInROI. It attacks only ROI pixels but adds more noise to those pixels so that the net amount of noise added to an image is the same as that in FGSM. For this, we scale the magnitude of the added perturbation by a factor K . K represents the ratio of the number of total pixels in the image to the ROI pixels. Note that IntenseInROI adds lot of noise to the pixels in a smaller region, whereas FGSM adds lesser noise per pixel, but poisons a much bigger region (viz., entire image).

$$K = \frac{Pixels_{total}}{Pixels_{ROI}} \quad (4)$$

IntenseInROI scales the perturbation factor to increase the attack intensity. Effectively, IntenseInROI uses the following equation for all the pixels in ROI:

$$x_{adv} = x_{true} + \epsilon * K * sign\left(\frac{\partial J((x_{true}, y))}{\partial x}\right) \quad (5)$$

On setting $K = 1$, IntenseInROI reduces to OnlyInROI. Figure 4 compares the images perturbed with $K = 1$ and $K = 2.56$.

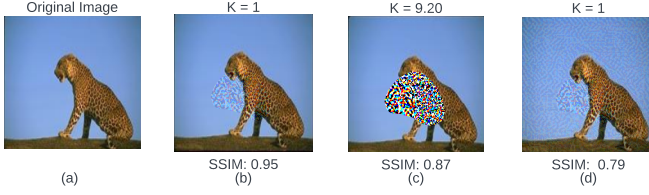


Figure 4: (a) Original image, image attacked with (b) OnlyInROI ($K = 1$), (c) IntenseInROI ($K > 1$) and (d) VariableNoise ($Z = 5$). Here, $\lambda = 0.5$, $\epsilon = 0.1$

Due to the use of $\epsilon * K$ factor in Eq. 5, IntenseInROI adds $K \times$ higher-intensity noise in ROI. However, it does not add any noise to the remaining image. As shown by the results (Table 2), IntenseInROI maintains a much higher value of SSIM and achieves comparable degradation in CNN accuracy as FGSM.

4.3 Variant 3: SpotOn-VariableNoise

Since a CNN works by extracting information from a local receptive field, a CNN can still achieve high accuracy when we perturb only ROI and not the remaining regions. To further degrade the CNN accuracy while maintaining a high SSIM, we propose another variant of SpotOn, which adds a high amount of noise in ROI and a comparatively lower amount of noise in the remaining regions. This variant is named SpotOn-VariableNoise. Specifically, VariableNoise attacks the entire image but uses ϵ in ROI and ϵ/Z in the remaining regions. The value of Z is greater than one, which helps control the attack's intensity. Unlike FGSM, our method controls the intensity of noise added to the different regions of the images based on ROI. As shown by the results, this technique produces the highest degradation in CNN accuracy with minimal degradation in image quality (SSIM). Figure 4(d) shows a sample image attacked using VariableNoise algorithm.

The VariableNoise method gives better results than OnlyInROI since VariableNoise adds a small amount of noise to regions outside the ROI as well. Since the amount of noise added outside the ROI is minimal but is applied to all the pixels, there is no particular region of the image that has been perturbed significantly. This leads to a high SSIM value of the attacked image.

Figure 5 summarizes the three variants of SpotOn technique. We refer the reader to Section 5.2 for a discussion of pros and cons of these variants and their respective applicability.

Figure 6 shows the symbols used along with their meanings and usages.

4.4 Salient Features

Advantages of Grad-CAM: For finding ROI, multiple methods are possible (cam 2022). Of these, we have used grad-CAM (Selvaraju et al. 2017) due to its popularity and

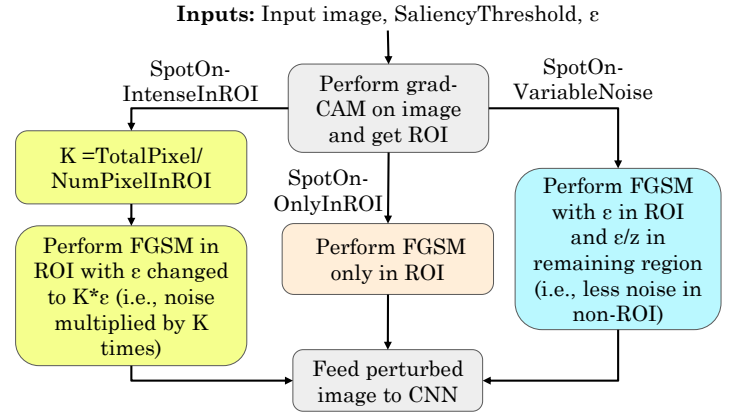


Figure 5: Working flow of 3 proposed attack algorithms

Symbol	Meaning	Remarks
ϵ	Amount of noise added	Used in FGSM and all SpotOn variants
λ	SaliencyThreshold, used for determining ROI	Used in all SpotOn variants
Z	Ratio of noise added in ROI and non-ROI region	Used in SpotOn-VariableNoise

Figure 6: Symbols used and their meaning/usage

advantages over conventional CAM (Zhou et al. 2016) in terms of area of coverage and explainability (Pinciroli Vago et al. 2021). Moreover, unlike CAM, Grad-CAM works with any CNN architecture. The guided Grad-CAM method would be an overkill for our use case since we only require a rough estimate of the ROI. Moreover, recent works (Adebayo et al. 2018; Nie, Zhang, and Patel 2018) suggest that guided Grad-CAM is ineffective in providing concrete insights into a CNN. Instead, it performs partial image recovery and acts like an edge detector.

Why we did not use trainable-attention: An alternative to using Grad-CAM is using trainable attention (Jetley et al. 2018). Grad-CAM is a type of post hoc attention, i.e., it is applied to an already trained model. It does not affect what or how a model learns during the training. However, trainable attention is a model that is trained in parallel while the main model is being trained. Hence, to use trainable attention, the attacker would need access to the main model throughout the training phase. In contrast, by using Grad-CAM, this requirement is avoided.

Broader applicability of our attack: Our threat model is the same as that of FGSM, which is a white-box attack. The only additional information that SpotOn uses is the gradients with respect to the activation maps. We have evaluated our attack for image classification as a proof of concept, although we believe that our attack would work in all scenarios where FGSM is possible. For example, in the case of object detection, SpotOn would still be able to poison the ROI and thus, fool DNN.

5 Experimental results

Experimental Platform: We use the Caltech101 dataset (cal 2022), which has 101 classes of images such as camera, chair, airplane, football, elephant, etc. The number of images in each class ranges from 40 to 800, with most classes having around 50 images. Image resolution is 300*200 pixels. While data-poisoning attacks can fool any CNN, for experimentation, we choose the VGG-19 network and perform ablation studies using AlexNet and GoogleNet. We use the PyTorch framework and CNN model definitions from the PyTorch vision repository.

5.1 Overhead analysis

Overhead of Grad-CAM: The Grad-CAM algorithm involves computing the gradients of the input with respect to the CONV-layer activations, global average pooling, weighted sum, and ReLU activation. Grad-CAM achieves localization using only a single forward pass and a partial backward pass per image; hence, it is not very computationally expensive.

Latency overhead: The various steps in Grad-CAM consume time, which is responsible for the latency overhead of SpotOn. Table 1 shows the latency of various attacks on a Geforce RTX 2080 GPU with 12GB memory. The host CPU is Intel Xeon Gold 5218 CPU with 2.30GHz frequency. Evidently, our proposed attack techniques add a minimal latency overhead due to the use of Grad-CAM. Also, both SpotOn-OnlyInROI and SpotOn-IntenseInROI techniques reduce the number of pixels that need to be attacked, reducing the overhead of Grad-CAM.

Table 1: Latency of various attacks (in seconds) for a set of 10 images

Attack	Latency (s)
FGSM	13.05
OnlyInROI	14.86
IntenseInROI	14.81
VariableNoise	15.86

5.2 Main Results

Table 2 shows the results of fooling VGG19, with $\lambda = 0.5$. While OnlyInROI maintains much higher SSIM values than FGSM, it only brings a slight degradation in CNN accuracy. This is expected because OnlyInROI is a weaker attack than FGSM; hence, it is not fair to compare it against FGSM. Except for some values of ϵ , VariableNoise provides comparable accuracy loss as FGSM, while providing much higher SSIM value. Thus, VariableNoise is superior to FGSM.

Comparison of variants of SpotOn: IntenseInROI adds the “same” amount of total noise as FGSM, and hence, it allows fair comparison with FGSM. For epsilon values of 0.01 and below, IntenseInROI provides marginally lower SSIM and causes marginally higher accuracy loss than FGSM, whereas VariableNoise results are almost similar to FGSM. Thus, at very small epsilon values, VariableNoise (or IntenseInROI) can easily replace FGSM to achieve similar accu-

Table 2: Results with VGG19 (Accuracy (Acc) before attack = 92%). Here, $\lambda=0.5$ and $Z=3$.

ϵ	FGSM		SpotOn					
			OnlyInROI		IntenseInROI		VariableNoise	
	Acc	SSIM	Acc	SSIM	Acc	SSIM	Acc	SSIM
0.005	77.8%	0.996	78.9%	1	77.4%	0.982	78.7%	0.999
0.007	77%	0.993	78.8%	0.999	76.3%	0.973	78.4%	0.999
0.01	76.8%	0.985	78.6%	0.998	74.0%	0.96	78.1%	0.998
0.03	15.3%	0.94	72.4%	0.98	50.3%	0.92	29.8%	0.97
0.05	10.9%	0.88	69.2%	0.95	33.8%	0.89	18.8%	0.92
0.1	8%	0.48	66.9%	0.90	14.3%	0.85	10.5%	0.78
0.2	5.2%	0.24	63.1%	0.84	5.3%	0.81	7.0%	0.56
0.3	2.6%	0.14	56.6%	0.82	2.9%	0.80	5.5%	0.41
0.4	1.4%	0.09	50.8%	0.81	1.9%	0.79	5.0%	0.31
0.5	1.1%	0.06	47.1%	0.80	1.4%	0.79	4.2%	0.24

racy degradation and SSIM while attacking a much smaller number of pixels.

OnlyInROI and IntenseInROI attack only the ROI regions; hence, they maintain high value of SSIM even for large values of ϵ . For epsilon values above 0.1, IntenseInROI has better SSIM than VariableNoise because it does not add any noise to the non-ROI region. At such high epsilon values, both techniques practically corrupt the ROI region and only the non-ROI region contributes to image quality. Since IntenseInROI adds no noise to the non-ROI area, it has better image quality.

On comparing IntenseInROI and VariableNoise, we observe that IntenseInROI adds a higher amount of “total noise” than VariableNoise. Still, IntenseInROI achieves higher SSIM and higher CNN accuracy because it attacks fewer pixels (only ROI), whereas VariableNoise attacks all the pixels. Thus, adding even small noise in all the pixels is more effective for degrading accuracy than adding a lot of noise in a few pixels. For achieving a high SSIM value, corrupting a few pixels is better than corrupting all the pixels. This strict tradeoff highlights the advantage of SpotOn, which allows tuning the region of attack and the amount of noise inserted in various areas based on the image’s semantic content.

When accuracy becomes low, a slight difference becomes immaterial; for example, whether the accuracy is 15% or 12%, the CNN model can be considered useless in both cases. Hence, these two accuracy values can be considered comparable. In such cases, preservation of image quality becomes of prime importance from the attacker’s point of view. As seen from the results, our techniques achieve much higher SSIM; hence, they are better in evading detection and comparable in attack efficacy.

5.3 Ablation studies

Results with different λ and Z values: Table 3 shows these results for VariableNoise. With increasing λ , more pixels become part of ROI. This increases attack intensity, leading to lower CNN accuracy and SSIM. Similarly, on changing the Z value from the default value of 3 to 2, non-ROI pixels are attacked with higher effective epsilon. Hence, changing Z to 2 further degrades the CNN accuracy and SSIM. Evidently, SpotOn allows increasing attack intensity by increasing λ and lowering Z . Conversely, reducing λ and increasing Z

weakens the attack. Based on these observations, a designer can choose specific values of λ and Z to achieve the target accuracy and SSIM values.

Table 3: Results with VGG19 for different values of λ and Z

	$\lambda=0.4, Z=3$		$\lambda=0.6, Z=3$		$\lambda=0.5, Z=2$		$\lambda=0.5, Z=4$	
ϵ	Acc	SSIM	Acc	SSIM	Acc	SSIM	Acc	SSIM
0.005	83.2%	1	80.7%	1	76.3%	1	84.7%	1
0.007	77.1%	0.99	74.1%	0.99	25.11%	0.96	79.7%	0.99
0.01	68.7%	0.99	64.8%	0.99	25.11%	0.96	72%	0.99
0.03	34.43%	0.98	31.80%	0.97	25.11%	0.96	39.81%	0.98
0.05	22.86%	0.94	21.10%	0.92	16.52%	0.89	27.36%	0.95
0.1	13.35%	0.83	12.91%	0.79	10.68%	0.71	16.08%	0.86
0.2	9.51%	0.60	9.18%	0.56	7.78%	0.44	10.32%	0.67
0.3	8.15%	0.44	7.22%	0.41	6.21%	0.30	8.17%	0.52
0.4	6.57%	0.34	5.76%	0.31	4.44%	0.21	6.59%	0.42
0.5	5.48%	0.27	4.69%	0.24	3.05%	0.16	5.51%	0.34

Results with different CNNs: So far, we have presented the results of VGG19. Table 4 shows the results with AlexNet. For FGSM, a change in CNN has no impact on the image SSIM, and hence, SSIM values remain the same as those in Table 2. However, for SpotOn, the ROI (i.e., number of ROI pixels) selected by Grad-CAM depends on the CNN. Hence, for different CNNs, SpotOn adds noise to the different number of pixels. As such, the SSIM achieved by SpotOn depends on the CNN used. From Table 4, we observe that FGSM is even more effective on AlexNet than on VGG19. On increasing ϵ value above 0.2 in FGSM, the accuracy of VGG19 plateaus to nearly 17% (Table 2); however, with AlexNet, the accuracy drops to 1%. The reason is that AlexNet is shallower, and thus, has less immunity to adversarial attacks. Compared to FGSM, VariableNoise maintains comparable accuracy but much higher SSIM. The same is also true for IntenseInROI. This confirms the superiority of our techniques. This also proves our claim that even by attacking only the ROI region, we can maintain image quality to evade human detection while still fooling CNN. OnlyInROI does not degrade accuracy as severely as FGSM but maintains much higher SSIM.

Table 4: Results with AlexNet (Acc. before attack = 88.75%). Here, $\lambda=0.5$ and $Z=3$.

	FGSM		SpotOn					
ϵ	Acc	SSIM	OnlyInROI		IntenseInROI		VariableNoise	
			Acc	SSIM	Acc	SSIM	Acc	SSIM
0.005	66.10%	1.00	67.79%	1.00	63.78%	0.98	67.54%	1.00
0.007	64.55%	0.99	67.74%	1.00	61.43%	0.97	67.46%	1.00
0.01	62.14%	0.99	67.55%	1.00	57.27%	0.96	66.88%	1.00
0.03	2.69%	0.88	24.17%	0.99	9.68%	0.90	6.80%	0.98
0.05	1.23%	0.75	18.54%	0.98	7.59%	0.87	3.34%	0.94
0.1	0.79%	0.48	13.97%	0.94	4.42%	0.84	1.29%	0.82
0.2	0.58%	0.24	11.03%	0.90	2.79%	0.83	0.63%	0.58
0.3	0.52%	0.14	9.66%	0.87	2.17%	0.83	0.52%	0.42
0.4	0.63%	0.09	8.47%	0.86	1.84%	0.82	0.54%	0.31
0.5	0.61%	0.06	7.63%	0.85	1.77%	0.82	0.42%	0.24

Table 5 shows the results with GoogleNet. Compared to VGG19, GoogleNet shows higher accuracy for a given value of ϵ , and thus, GoogleNet possesses more immunity to adversarial attacks. VariableNoise leads to only slightly higher, but comparable, accuracy than FGSM, but it achieves much higher SSIM. IntenseInROI leads to lower accuracy and

higher SSIM than FGSM; thus, it is comprehensively superior to FGSM. These results prove the superiority of our techniques.

Table 5: Results with GoogleNet (Acc. before attack = 95.65%). Here, $\lambda=0.5$ and $Z=3$.

	FGSM		SpotOn					
ϵ	Acc	SSIM	OnlyInROI		IntenseInROI		VariableNoise	
			Acc	SSIM	Acc	SSIM	Acc	SSIM
0.005	77.79%	1.00	83.82%	1.00	70.11%	0.99	81.11%	1.00
0.007	70.09%	0.99	76.66%	1.00	64.27%	0.99	73.20%	1.00
0.01	62.71%	0.98	68.90%	0.99	58.08%	0.98	64.98%	0.99
0.03	45.40%	0.88	52.18%	0.94	46.99%	0.86	47.24%	0.93
0.05	41.61%	0.75	47.38%	0.87	42.52%	0.75	42.46%	0.85
0.1	36.06%	0.48	40.71%	0.72	22.79%	0.59	35.66%	0.65
0.2	12.64%	0.24	15.29%	0.57	6.05%	0.48	14.02%	0.40
0.3	4.78%	0.14	6.63%	0.50	3.05%	0.44	5.49%	0.27
0.4	2.31%	0.09	3.82%	0.46	2.15%	0.42	2.90%	0.19
0.5	1.44%	0.06	2.88%	0.44	2.00%	0.41	1.75%	0.14

6 Conclusion

In this paper, we present a technique for launching localized gradient-based attacks while preserving the image quality. We also present variations of our strategy concerning the perturbed region in an image. Our technique uses Grad-CAM for ascertaining the region of interest in an image. Based on this, the gradient-based adversarial attack is launched only in the selected region instead of the entire image. This helps in maintaining the attacked image quality at an acceptable level. The perturbations in the attacked image remain imperceptible to the human eye, and yet, a CNN gets fooled into misclassifying the image. From an attacker's perspective, this presents an ideal scenario. We see multiple avenues for future research. One possible area of research can be incorporating image quality metrics such as SSIM into defense strategies. Researchers can look into novel ways of selecting the regions of attack efficiently and accurately. We hope our work helps the community and inspires future research in localized gradient-based data poisoning attacks.

References

- 2022. Caltech 101. <http://www.vision.caltech.edu/Image/Datasets/Caltech101/>.
- 2022. Class Activation Map methods implemented in Pytorch. <https://github.com/jacobgil/pytorch-grad-cam>.
- Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*.
- Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.
- Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 15–26.
- Cheng, M.; Le, T.; Chen, P.-Y.; Yi, J.; Zhang, H.; and Hsieh, C.-J. 2018. Query-efficient hard-label black-box

- attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*.
- Dong, X.; Zhang, W.; and Yu, N. 2018. CAAD 2018: Powerful none-access black-box attack based on adversarial transformation network. *arXiv preprint arXiv:1811.01225*.
- Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9185–9193.
- Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Ilyas, A.; Engstrom, L.; and Madry, A. 2018. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*.
- Jetley, S.; Lord, N. A.; Lee, N.; and Torr, P. H. 2018. Learn to pay attention. *arXiv preprint arXiv:1804.02391*.
- Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.
- Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.
- Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1765–1773.
- Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.
- Narodytska, N.; and Kasiviswanathan, S. P. 2017. Simple Black-Box Adversarial Attacks on Deep Neural Networks. In *CVPR Workshops*, volume 2.
- Nie, W.; Zhang, Y.; and Patel, A. 2018. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *International Conference on Machine Learning*, 3809–3818. PMLR.
- Papernot, N.; McDaniel, P.; and Goodfellow, I. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.
- Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 506–519.
- Pinciroli Vago, N. O.; Milani, F.; Fraternali, P.; and da Silva Torres, R. 2021. Comparing CAM Algorithms for the Identification of Salient Image Features in Iconography Artwork Analysis. *Journal of Imaging*, 7(7): 106.
- Sabour, S.; Cao, Y.; Faghri, F.; and Fleet, D. J. 2015. Adversarial manipulation of deep representations. *arXiv preprint arXiv:1511.05122*.
- Sarkar, S.; Bansal, A.; Mahbub, U.; and Chellappa, R. 2017. UPSET and ANGRI: Breaking high performance image classifiers. *arXiv preprint arXiv:1707.01159*.
- Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.
- Setiadi, D. R. I. M. 2021. PSNR vs SSIM: imperceptibility quality assessment for image steganography. *Multimedia Tools and Applications*, 80: 8423–8444.
- Su, J.; Vargas, D. V.; and Sakurai, K. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5): 828–841.
- Tu, C.-C.; Ting, P.; Chen, P.-Y.; Liu, S.; Zhang, H.; Yi, J.; Hsieh, C.-J.; and Cheng, S.-M. 2019. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 742–749.
- Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.
- Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; and Yuille, A. 2017. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 1369–1378.
- Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2730–2739.
- Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.