Multi-patch Adversarial Attack

Anurag Shah
Purdue University
shah447@purdue.edu

Abstract

Adversarial Patch Attacks pose a unique threat to Convolutional Neural Networks, as they can change the detected class for an input image by modifying only a small region of that image. Defenses against this form of attack have been proposed, but all are predicated on a single, contiguous patch being applied to the input. We demonstrate that the application of multiple, smaller patches achieves a similar result as a single, larger patch, while overwriting a smaller total area, and effectively bypassing defenses against the latter.

1 Introduction

Adversarial attacks against Neural Networks are a category of techniques used to mis-classify an input image. In general, a function is applied to an input image yielding an adversarial input image, which humans would classify the same as the original input image, but the neural network being attacked would classify it differently. These attacks tend to target specific (trained) models with knowledge of that model, and target a specific class for that model to classify it as. These adversarial attacks are not limited to an image domain, and can also be employed in the physical domain [4].

The patch attack, a form of adversarial attack, involves the application of a "patch", or a single smaller object, to a rectangular portion of the input, leaving the input otherwise unchanged. This can be applied to an input image or in the physical domain. Such a patch is generally optimized to target a specific model, but can also be optimized in a "black-box" method against various different models. The application of this patch causes the model to predict a specific target class with often greater than 90 percent confidence [1].

Since the location of an arbitrarily applied patch is not known by the model, any successful defense must be able to locate the patch in some manner. The Minority Reports Defense attempts to periodically occlude all portions of the image larger than the patch size, and train a classifier to classify partially occluded images. This way, it can identify the approximate location of the patch by examining patterns in predicted results [2]. Wavelet patch detection uses a residual of an input image, calculated based on a denoising function, to detect the presence of contiguous patches as small as a single pixel [3].

Both of these defenses are predicated on a single patch being applied to the input image, and the patch shape is assumed to be rectangular. While modifying the patch shape can pose a threat to image Wavelet patch detection, the minority reports defense is agnostic to the patch shape, and both defenses can be applied against a single input image. Therefore, circumventing these defenses requires a patch dispersed across the input image, which practically is the application of multiple patches on a single input image.

Our contributions are:

- We provide a solution to common defenses against adversarial patch attacks.
- We demonstrate that the application of multiple, smaller patches to an input image achieves similar performance to a single, larger patch, while overwriting a smaller area in total.

Preprint. Under review.

2 Background

The Adversarial Patch Attack takes an input image x, initially classified as y, and applies a patch p at location l with transformations t, creating a modified input image $\hat{x}$ which the model classifies as $\hat{y}$, a specific target class chosen by the attacker. After applying the patch, the attacker attempts to optimize the patch. A variant of the Expectation over Transformation function is optimized, over the distribution of transformations and locations, and over the input image dataset. This allows the construction of a "universal" patch that is agnostic to its application and to the labelled class of the input image it is placed on. The patch can be optimized in two sets of circumstances. In the case of a "black-box" attack, the target model is not known; in this case, the attacker knows the predicted class for that input from several commonly used models, but not the confidence level. In case of a "white-box" attack, the attacker knows the confidence level of the prediction for a single model [1].

The Minority Reports Defense creates an occlusion region that is larger than a small but effective patch size, based on the dataset used for that model. This occlusion region is applied to the input images from the dataset, to train a model that can correctly classify occluded inputs with a high confidence. After training, an input image is given every possible occlusion for the given occlusion region and stride s. The model then classifies each of these occluded images, to produce an output map corresponding to the location of the occluded region. If there is no patch on the input, the model would classify each of the occluded images the same. If there is a patch applied to the input, at minimum in locations where the patch is completely occluded, the classifier will output the true class of the input. By locating the pattern of occlusion regions where this minority true class is the output, the approximate location of the patch can be identified, as well as the true class of the input image [2].

Wavelet patch detection uses image residuals to detect the presence of a patch in the input. It uses the Bayesian Shrinkage algorithm as a denoising function to calculate the residual of an image, and from its output identify a high frequency noise region (relative to the rest of the image) to locate a patch. This approach becomes less effective when there is a degree of noise distributed throughout the image (however, at such a point it becomes difficult for a human to ascertain its true class, so the adversarial attack ceases to be purely a patch based attack) [3].

Patches can notably be applied in a physical domain, for example with person detection [5], therefore it is important for any defense to be robust in such a domain as well, while the attack can be tailored to the domain it is employed in.

3 Contribution

The adversarial patch attack optimizes a single patch, given a random input image, random location, and random transformation. For a multi-patch attack, this approach needs to be modified. There are design decisions to be made when applying multiple patches, which we explore below.

3.1 Number of Patches

The first decision is whether to have a fixed or random number of patches. A fixed number of patches would always apply that many patches to an input image, while a random number of patches would apply any number of patches in a given range to an input. The former is certainly easier to optimize, however it comes with a serious flaw, in that defenses can be easily augmented to protect against a fixed number of patches. The Minority Reports defense in theory can be extended to multiple patches by simply having multiple occlusion regions, corresponding to the number of patches in the image. While it would be inefficient to do so, given a fixed set of patches $p_1, p_2, ...p_k$, and occlusion regions $O_1, O_2, ...O_k$, There is a set of occluded images where each occlusion region covers its corresponding patch. Since occlusion regions may swap freely, this set would also contain the various permutations of the Occlusion Regions in those same positions, yielding patterns that are easy to identify. However, with a random number of patches, this changes. With n patches and m occlusion regions, there are three possibilities. If $n > m$, at least one patch will always be visible to attack the model. If $n < m$, it is not possible to identify which occlusion regions are actively covering patches, therefore there is no pattern of occlusion regions to find the true class from. Only in the case where n is exactly equal to m does this defense succeed. Notably, it is theoretically possible to use separate sets of occlusion regions, with incrementally more regions in each set, to try and identify a minority report within the minority reports; this runs into several practical concerns, such as its exponential

runtime, and the cost to train a network with exceedingly complicated occlusions (and this assumes there is still enough of the input image to classify in the first place, which is unlikely as more and more occlusions are applied, occlusion regions being necessarily larger than the patches they try to occlude). A similar situation is seen with Wavelet patch detection. With a fixed number of patches, the defense can simply be expanded to detect exactly that many high frequency noise regions. But with a random number of patches (with an unknown range), it is not possible to detect this, as with an increasing amount of patches it is not possible to differentiate a patch from the background frequency over such a complicated and irregular region. Hence, we conclude that applying a random number of patches is the superior approach.

3.2 Patch Optimization

The second decision is how patches should be optimized. The possibilities here are to train separate patches with separate target classes, train separate patches with the same target class, or to train the same patch that is applied multiple times. Training separate patches for separate target classes is not an effective approach as patches will enter into adversarial conflict with each other, reducing the overall potency of the attack, and making it impossible to tailor the patch to a specific label, only to a set of labels. Training separate patches with the same target class on its own is a sensible approach, but does not work well when a random number of patches are applied. Patches can only be optimized when they are applied to an input, so if a random number of patches are applied, certain patches will be applied on average less than others, leading to a discrepancy in optimization. If the patches applied are themselves randomized, each patch gets fewer training over an epoch and is applied at fewer locations, leading to longer training time. The advantage is that there are several separate attackers on a single input image, but with patch attacks already able to generate extremely high confidence classifications, this is not a significant advantage to justify the increase in training cost. Training a single patch that is applied at multiple locations and optimized overall is the ideal approach. It is also important to note that this will train faster than a single patch of the same size, as it is optimized over several locations at once.

3.3 Patch Size

Finally, the last design decision is the size of the patch. It is possible to make the size of the patch non-uniform if using separate patches with the same target class, but not if using the same patch at several locations. In choosing a patch size, let us first assume that we have a successful attack using a single patch p with area a. If we want to instead apply a smaller patch $\hat{p}$ with area $\hat{a}$ a maximum of r times over the image, while overall overwriting less area of the input image, $\hat{a} < r \cdot a$. With a square patch, the dimension of the selected patch must be $\sqrt{r \cdot a}$ or smaller.

3.4 Patch application

Patches are applied just as in a single patch attack, but each application of the patch has its own rotation and location. The locations of two applied patches cannot overlap. We apply between 1 and 3 patches to the input images. Our chosen dataset is the ImageNet data, and our target model is ResNet50, with pretrained weights on ImageNet data. The input data is rescaled to 300x300 square images. We use and compare various patch sizes, ranging from 4500 pixels in area (the size needed for a single patch to reach high confidence) to 900 pixels in area. We also use various target classes for the patch.

4 Evaluation

4.1 Patches at various sizes

Patches were tested at 4500 pixels in area, 2700 pixels in area, 1500 pixels in area, and and 900 pixels in area. 1500 was the target class to demonstrate similar performance to a single patch, the 4500 pixel patches were used as a control test that expected to succeed, and the 900 pixel patches were used as a control test that expected to fail. All four tests used different target classes.

The 4500 pixel patch reached a success rate of 96.71%, after 7 epochs of optimization. The target class was pillow (721 in imagenet).

The 2700 pixel patch reached a success rate of 95.05%, after 2 epochs of optimization. The target class was tiger beetle (300 in imagenet).

The 1500 pixel patch reached a success rate of 87.46%, after 3 epochs of optimization. The target class was volleyball (890 in imagenet). This demonstrates a similar performance to a single patch of 4500 pixels.

The 900 pixel patch reached a success rate of 9.65%, after 2 epochs of optimization. The target class was toaster (859 in imagenet).

4.2 Performance against wavelet detection

Wavelet detection is unable to detect the presence of a patch when there are at least 2 patches present. When there is only 1 patch present, it is able to detect the presence of a patch in accordance with the findings[3].

5 Discussion

The application of several patches to a single input image results in similar efficiency as the application of a single, larger patch, while being able to defeat defenses employed against single patch attacks. There are a few specifics about this approach that need to be examined in greater detail. First, is the application of multiple smaller patches more difficult to disguise? In the original adversarial patch attack, a technique to disguise patches was laid out, to make it difficult for human observers to distinguish in the input. If such a method is applied to multiple patches, does it still fool human observers? Second, how applicable is this attack in a physical setting? Finally, are there unique defenses against multiple patches that would not be applicable in the domain of a single patch? These are important questions for further research

6 Code

https://github.com/Anurag-Shah/Multi-Patch-Attack

References

[1] Brown, T.B., Mané, D., Roy, A., Abadi, M. & Gilmer, J. (2018) Adversarial Patch. From https://arxiv.org/pdf/1712.09665.pdf
Code: https://github.com/A-LinCui/Adversarial_Patch_Attack

[2] McCoyd, M., Park, W., Chen, S., Shah, N., Roggenkemper, R., Hwang, M., Liu, J.X. & Wagner, D. (2020) Minority Reports Defense: Defending Against Adversarial Patches. From https://arxiv.org/pdf/2004.13799.pdf

[3] Marius, A., Tewfik, A.H. & Vishwanath, S. (2020) Detecting Patch Adversarial Attacks with Image Residuals. From https://arxiv.org/pdf/2002.12504.pdf
Code: https://github.com/mariusarvinte/wavelet-patch-detection

[4] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T. & Song, D. (2018) Robust Physical-World Attacks on Deep Learning Visual Classification. From https://arxiv.org/pdf/1707.08945.pdf

[5] Thys, S. & Ranst, W.V. (2019) Fooling automated surveillance cameras: adversarial patches to attack person detection. From https://arxiv.org/pdf/1904.08653.pdf