Adversarial Patch

Tom B. Brown, Dandelion Mané,[*] Aurko Roy, Martín Abadi, Justin Gilmer
`{tombrown,dandelion,aurkor,abadi,gilmer}@google.com`

Abstract

We present a method to create universal, robust, targeted adversarial image patches in the real world. The patches are universal because they can be used to attack any scene, robust because they work under a wide variety of transformations, and targeted because they can cause a classifier to output any target class. These adversarial patches can be printed, added to any scene, photographed, and presented to image classifiers; even when the patches are small, they cause the classifiers to ignore the other items in the scene and report a chosen target class.

1 Introduction

Deep learning systems are broadly vulnerable to adversarial examples, carefully chosen inputs that cause the network to change output without a visible change to a human [13, 5]. These adversarial examples most commonly modify each pixel by only a small amount and can be found using a number of optimization strategies such as L-BFGS [13], Fast Gradient Sign Method (FGSM) [5], DeepFool [9], Projected Gradient Descent (PGD) [8], as well as the recently proposed Logit-space Projected Gradient Ascent (LS-PGA) [2] for discretized inputs. Other attack methods seek to modify only a small number of pixels in the image (Jacobian-based saliency map [10]), or a small patch at a particular location of the image [11].

Adversarial examples have been shown to generalize to the real world. Kurakin et al. [7] demonstrated that when printed out, an adversarially constructed image will continue to be adversarial to classifiers even under different lighting and orientations. Athalye et al. [3] recently demonstrated adversarial objects which can be 3d printed and misclassified by networks at different orientations and scales. Their adversarial objects are designed to be subtle perturbations of a normal object (e.g. a turtle that has been adversarially perturbed to be classified as a rifle). Another work [11] showed that one can fool facial recognition software by constructing adversarial glasses. These glasses were targeted in that they could be constructed to impersonate any person, but were custom made for the attacker's face, and were designed with a fixed orientation in mind. Even more recently, Evtimov et al. [4] demonstrated various methods for constructing stop signs that are misclassified by models, either by printing out a large poster that looks like a stop sign, or by placing various stickers on a stop sign.

As seen above, a majority of prior work has focused on attacking with and defending against either small or imperceptible changes to the input. In this work we explore what is possible if an attacker no longer restricts themselves to imperceptible changes. We construct an attack that does not attempt to subtly transform an existing item into another. Instead, this attack generates an image-independent patch that is extremely salient to a neural network. This patch can then be placed anywhere within the field of view of the classifier, and causes the classifier to output a targeted class. Because this patch is scene-independent, it allows attackers to create a physical-world attack without prior knowledge of the lighting conditions, camera angle, type of classifier being attacked, or even the other items within the scene.

This attack is significant because the attacker does not need to know what image they are attacking when constructing the attack. After generating an adversarial patch, the patch could be widely

[*]Equal contributions with Tom B. Brown

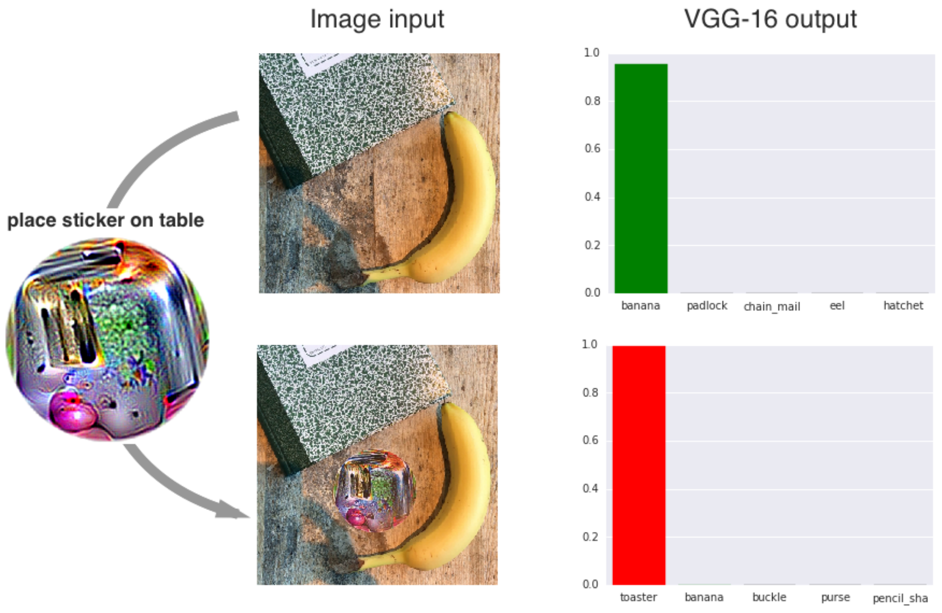

Figure 1: A real-world attack on VGG16, using a physical patch generated by the white-box ensemble method described in Section 3. When a photo of a tabletop with a banana and a notebook (top photograph) is passed through VGG16, the network reports class 'banana' with 97% confidence (top plot). If we physically place a sticker targeted to the class "toaster" on the table (bottom photograph), the photograph is classified as a toaster with 99% confidence (bottom plot). See the following video for a full demonstration: `https://youtu.be/e9IAu41T9w8`

distributed across the Internet for other attackers to print out and use. Additionally, because the attack uses a large perturbation, existing defense techniques which focus on defending against small perturbations are unlikely to work. Indeed recent work has demonstrated that state-of-the art adversarially trained models on MNIST are still vulnerable to larger perturbations than those used in training either by searching for a nearby adversarial example using a different metric for distance [12], or by applying large perturbations in the background [1].

2 Approach

The traditional strategy for finding a targeted adversarial example is as follows: given some classifier $\mathbb{P}[y \mid x]$, some input $x \in \mathbb{R}^n$, some target class $\widehat{y}$ and a maximum perturbation ε, we want to find the input $\widehat{x}$ that maximizes $\log(\mathbb{P}[\widehat{y} \mid \widehat{x}])$, subject to the constraint that $\|x - \widehat{x}\|_\infty \leq \varepsilon$. When $\mathbb{P}[y \mid x]$ is parameterized by a neural network, an attacker with access to the model can perform iterated gradient descent on x in order to find a suitable input $\widehat{x}$. This strategy can produce a well camouflaged attack, but requires modifying the target image.

Instead, we create our attack by completely replacing a part of the image with our patch. We mask our patch to allow it to take any shape, and then train over a variety of images, applying a random translation, scaling, and rotation on the patch in each image, optimizing using gradient descent. More formally, we have:

- a trainable *patch* $z \in \mathbb{R}^n$,
- a family of *allowable affine transformations* T from $\mathbb{R}^n \to \mathbb{R}^n$,
- and a *binary mask* $m \in \{0, 1\}^n$.

The final image $\widehat{x}$ must be realizable using the fixed mask m applied on the trained patch $\widehat{z}$ and some transformation $t \in T$. More formally, we define a *patch function* p corresponding to every transformation $t \in T$ that applies the transformed patch onto the image:

$$p_t(x, z) = t(m \odot z) + [t(1 - m) \odot x]\,, \tag{1}$$

where $\odot$ refers to the pixel-wise Hadamard product. Then the final adversarially perturbed image $\widehat{x}$ must satisfy $\widehat{x} = p_t\left(x, \widehat{z}\right)$ for the final trained patch $\widehat{z}$ and some $t \in T$.

To obtain the trained patch $\widehat{z}$, we use a variant of the Expectation over Transformations (EOT) framework of Athalye et al.[3] Let us briefly review the setup of the EOT framework. In this setting, we have a family of affine transformations T, a distance metric d in the transformed space, and the objective is to find a perturbed image $\widehat{x}$ satisfying:

$$\widehat{x} = \arg\max_{x'} \mathbb{E}_{t\sim T}\left[\log \Pr(\widehat{y} | t(x'))\right] \quad \text{s.t.} \quad \mathbb{E}_{t\sim T}\left[d(t(x'), t(x))\right] < \varepsilon,$$

i.e., the final image should be within ε-ball in expectation over all the transformations in T. Instead of having an ε bound on the amount of perturbation we are allowed to add, our attack instead finds the solution to the following unconstrained optimization problem:

$$\widehat{z} = \arg\max_{z' \in \mathbb{R}^n} \mathbb{E}_{t\sim T}\left[\log\left(\mathbb{P}\left[\widehat{y} \mid p_t(x, z')\right]\right)\right]).$$

The final perturbed image $\widehat{x}$ can then be obtained by choosing any $t \in T$ and applying the patch function (1) to $\widehat{z}$, i.e., $\widehat{x} = p_t(x, \widehat{z})$.

Because our attack leaves the transformation t as an open variable, the attacker is free to choose a t that has desirable properties for the attack. For example, in the real world, an attacker could choose to place a trained patch $\widehat{z}$ in an inconspicuous place in the background of an image.

We believe that this attack exploits the way image classification tasks are constructed. While images may contain several items, only one target label is considered true, and thus the network must learn to detect the most "salient" item in the frame. The adversarial patch exploits this feature by producing inputs much more salient than objects in the real world. Thus, when attacking object detection or image segmentation models, we expect a targeted toaster patch to be classified as a toaster, and not to affect other portions of the image.

3 Experimental Results

To test our attack, we compare the efficacy of two whitebox attacks, a blackbox attack, and a control patch. The white box ensemble attack jointly trains a single patch across five ImageNet models: inceptionv3, resnet50, xception, VGG16, and VGG19. We then evaluate the attack by averaging the win rate across all five models. The white box single model attack does the same but only trains and evaluates on a single model. The blackbox attack jointly trains a single patch across four of the ImageNet models, and then evaluates the blackbox attack on a fifth model, which we did not access during training. The control is a picture of a toaster.

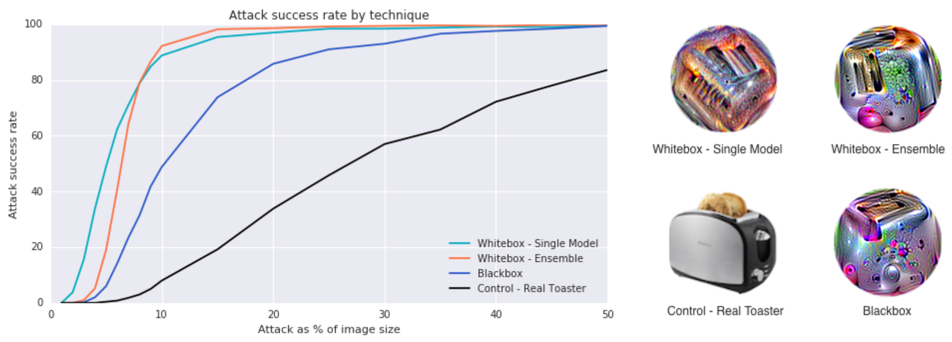

Figure 2: A comparison of different methods for creating adversarial patches. Note that these success rates are for random placements of the patch onto the image.

During training and evaluation, the patches are rescaled and then digitally inserted on a random location on a random ImageNet image. Figure 2 shows the results.

Note that the patch size required to reliably fool the model in this universal setting (black box, on a targeted class, and over all images, locations and transformations) is significantly larger than those required to perform a non-targeted attack on a single image and a single location in the whitebox setting. For example, Su et al. [6] recently demonstrated that modifying 1 pixel on a 32x32 pixel CIFAR-10 image (0.1% of the pixels in the image) suffices to fool the majority of images with a non-targeted, non-universal whitebox attack. However, our attacks are still far more effective than naively inserting an image with the target class, as is shown in Figure 2 by the relatively poor performance of inserting a real toaster into the scene.

Any of the attacks shown in Figure 2 can be camouflaged in order to reduce their saliency to a human observer. We create a disguised version of the patch by minimizing its L2 distance to a tie-dye pattern and applying a peace sign mask during training. The results from these experiments are found in Figure 3.

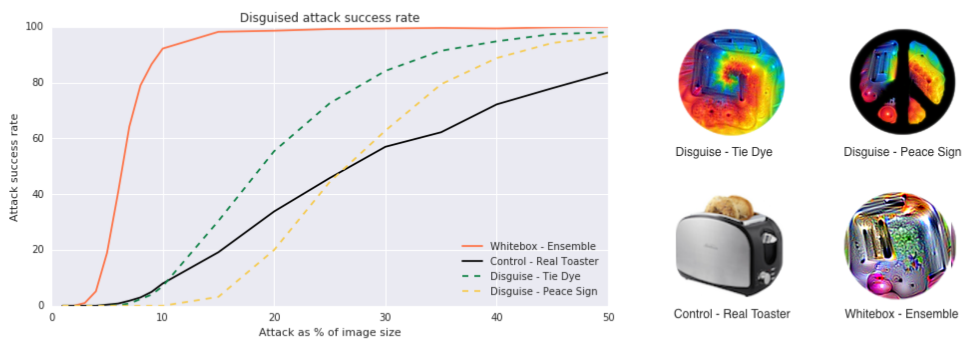

Figure 3: A comparison of patches with various disguises. We find that we can disguise the patch and retain much of its power to fool the classifier.

In our final experiment, we test the transferability of our attack into the physical world. We print the generated patch with a standard color printer, and put it a variety of real world situations. The results shown in Figure 1 demonstrate that the attack successfully fools the classifier, even when there are multiple other objects within the scene.

4 Conclusion

We show that we can generate a universal, robust, targeted patch that fools classifiers regardless of the scale or location of the patch, and does not require knowledge of the other items in the scene that it is attacking. Our attack works in the real world, and can be disguised as an innocuous sticker. These results demonstrate an attack that could be created offline, and then broadly shared.

As demonstrated in Section 1, a substantial amount of security research has focused on imperceptible perturbations. Our attacks are clearly not of this form and the produced patch sometimes resembles the target class. However, it is unclear whether or not imperceptible perturbations are the real security concern for applications of interest. For instance, we may imagine an airport surveillance system that aims to recognize objects (weapons, laptops, etc.) as they pass by a physical location; applying a patch to those objects may perhaps fool the system, and lead to a security compromise, even though the patch is obvious to a human. This work shows that focusing only on defending against small perturbations is insufficient, as large, local perturbations can break classifiers.

5 Appendix

References

[1] Anonymous. Adversarial spheres. *International Conference on Learning Representations*, 2018.

[2] Anonymous. Thermometer encoding: One hot way to resist adversarial examples. *International Conference on Learning Representations*, 2018.

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

[4] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on deep learning models. 2017.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] Sakurai Kouichi Jiawei Su, Danilo Vasconcellos Vargas. One pixel attack for fooling deep neural networks. *arXiv preprint arXiv:1710.08864*, 2017.

[7] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[8] Aleksander Madry, Aleksander Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial examples. *arXiv preprint arXiv:1706.06083*, 2017.

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[10] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.

[11] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM, 2016.

[12] Yash Sharma and Pin-Yu Chen. Breaking the Madry Defense model with L1-based adversarial examples. *arXiv preprint arXiv:1710.10733*, 2017.

[13] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.