Deflecting Adversarial Attacks with Pixel Deflection

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, James Storer
Brandeis University
{aprakash,nemtiax,solomongarber,dilant,storer}@brandeis.edu

Abstract

CNNs are poised to become integral parts of many critical systems. Despite their robustness to natural variations, image pixel values can be manipulated, via small, carefully crafted, imperceptible perturbations, to cause a model to misclassify images. We present an algorithm to process an image so that classification accuracy is significantly preserved in the presence of such adversarial manipulations. Image classifiers tend to be robust to natural noise, and adversarial attacks tend to be agnostic to object location. These observations motivate our strategy, which leverages model robustness to defend against adversarial perturbations by forcing the image to match natural image statistics. Our algorithm locally corrupts the image by redistributing pixel values via a process we term pixel deflection. A subsequent wavelet-based denoising operation softens this corruption, as well as some of the adversarial changes. We demonstrate experimentally that the combination of these techniques enables the effective recovery of the true class, against a variety of robust attacks. Our results compare favorably with current state-of-the-art defenses, without requiring retraining or modifying the CNN.

Code: github.com/iamaaditya/pixel-deflection

1. Pixel Deflection

Much has been written about the lack of robustness of deep convolutional networks in the presence of adversarial inputs [4, 5]. However, most deep classifiers are robust to the presence of natural noise, such as sensor noise. We introduce a form of artificial noise and show that most models are similarly robust to this noise. We randomly sample a pixel from an image, and replace it with another randomly selected pixel from within a small square neighborhood. We also experimented with other neighborhood types, including sampling from a Gaussian centered on the pixel, but these alternatives were less effective.

We term this process *pixel deflection*, and give a formal definition in Algorithm 1. Let R_p^r be a square neighborhood with apothem r centered at a pixel p. Let $\mathcal{U}(R)$ be the uniform distribution over all pixels within R. Let I_p indicate the value of pixel p in image I.

Algorithm 1: Pixel deflection transform

Input : Image I, neighborhood size r
Output: Image I' of the same dimensions as I

for $i \leftarrow 0$ **to** K **do**
  Let $p_i \sim \mathcal{U}(I)$
  Let $n_i \sim \mathcal{U}(R_p^r \cap I)$
  $I'[p_i] = I[n_i]$
end

Figure 1. Average classification probabilities for an adversarial image (top) and clean image (bottom) after pixel deflection (Image size: 299x299)

As shown in Figure 1, even changing as much as 1% (*i.e.* 10 times the amount changed in our experiments) of the original pixels does not alter the classification of a clean image. However, application of pixel deflection enables the recovery of a significant portion of correct classifications.

1.1. Distribution of Attacks

Most attacks search the entire image plane for adversarial perturbations, without regard for the location of the im-

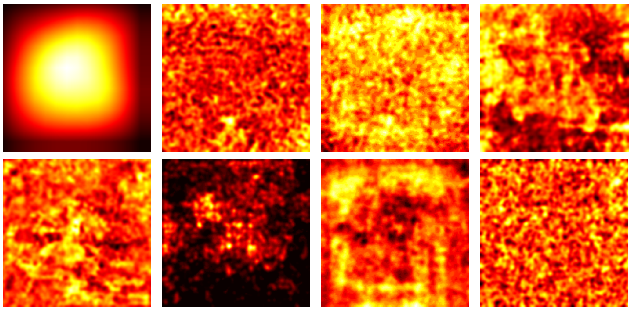

Figure 2. Visualization showing average location in the image where perturbation is added by an attacker. Clockwise from top left: Localization of most salient object in the image, FGSM, IGSM, FGSM-2 (higher ϵ), Deep Fool, JSMA, LBFGS and Carlini-Wagner attack.

age content. This is in contrast with the classification models, which show high activation in regions where an object is present [1]. This is especially true for attacks which aim to minimize the L_p norm of their changes for large values of p, as this gives little to no constraint on the total number of pixels perturbed. In fact, Lou *et al*. [3] use the object coordinates to mask out the background region and show that this defends against some of the known attacks.

In Figure 2 we show the average spatial distribution of perturbations for several attacks, as compared to the distribution of object locations (top left). Based on these ideas, we explore the possibility of updating the pixels in the image such that the probability of that pixel being updated is inversely proportional to the likelihood of that pixel containing an object.

1.2. Targeted Pixel Deflection

As we have shown in section 1, image classification is robust against the loss of a certain number of pixels. In natural images, many pixels do not correspond to a relevant semantic object and are therefore not salient to classification. Classifiers should then be more robust to pixel deflection if more pixels corresponding to the background are dropped as compared to the salient objects. Thus, we improve Pixel Deflection by deflecting more pixels which are outside the semantic region. We use a variant of class activation maps [8] to obtain the heatmap of semantic regions.

2. Results

Attack	$\lvert L_2 \rvert$	**No Defense**	**With Defense**	
Window=10, Deflections=100			Single	Ens-10
Clean	0.00	100	98.1	**98.9**
FGSM	0.04	19.2	79.7	**81.2**
IGSM	0.03	11.8	81.7	**82.4**
DFool	0.02	18.0	87.7	**92.4**
JSMA	0.02	24.9	93.0	**98.1**
LBFGS	0.02	11.6	90.3	**93.6**
C&W	0.04	05.2	93.1	**98.3**

Table 1. Top-1 accuracy on various attack models.

Defense	**FGSM**	**IGSM**	**DFool**	**C&W**
Feature Squeezing (Xu et al [7])				
(a) Bit Depth (2 bit)	0.132	0.511	0.286	0.170
(b) Bit Depth (5 bit)	0.057	0.022	0.310	0.957
(c) Median Smoothing (2x2)	0.358	0.422	0.714	0.894
(d) Median Smoothing (3x3)	0.264	0.444	0.500	0.723
(e) Non-local Mean (11-3-2)	0.113	0.156	0.357	0.936
(f) Non-local Mean (13-3-4)	0.226	0.444	0.548	0.936
Best model (b) + (c) + (f)	0.434	0.644	0.786	0.915
Random resizing + padding (Xie *et al*. [6])				
Pixel padding	0.050	-	0.972	0.698
Pixel resizing	0.360	-	0.974	0.971
Padding + Resizing	0.478	-	**0.983**	0.969
Quilting + TVM (Guo *et al*. [2])				
Quilting	0.611	0.862	0.858	0.843
TVM + Quilting	0.619	0.866	0.866	0.841
Cropping + TVM + Quilting	0.629	0.882	0.883	0.859
Our work: PD - Pixel Deflection, R-CAM: Robust CAM				
PD	0.735	0.880	0.914	0.931
PD + R-CAM	0.746	0.912	0.911	0.952
PD + R-CAM + DCT	0.737	0.906	0.874	0.930
PD + R-CAM + DWT	**0.769**	**0.927**	0.948	**0.981**

Table 2. Destruction Rate of various defense techniques. $\lvert L_2 \rvert$ lies between $0.02 - 0.06$ and classifier accuracy is 76%. We only include the Black-box attacks, where the attack model is not aware of the defense techniques. Single Pattern Attack and Ensemble pattern attack as reported in Xie et al [6] are not reported.

References

[1] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *CoRR*, abs/1710.11063, 2017.

[2] C. Guo, M. Rana, M. Cissé, and L. van der Maaten. Countering adversarial images using input transformations. 2017.

[3] Y. Luo, X. Boix, G. Roig, T. A. Poggio, and Q. Zhao. Foveation-based mechanisms alleviate adversarial examples. *CoRR*, abs/1511.06292, 2015.

[4] A. M. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

[6] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In *ICLR*, 2018.

[7] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *CoRR*, abs/1704.01155, 2017.

[8] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *CVPR*, pages 2921–2929, 2016.