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. Introduction

Image classification convolutional neural networks (CNNs) have become a part of many critical real-world systems. For example, CNNs can be used by banks to read the dollar amount of a check [4], or by self-driving cars to identify stop signs [37].

The critical nature of these systems makes them targets for adversarial attacks. Recent work has shown that classifiers can be tricked by small, carefully-crafted, imperceptible perturbations to a natural image. These perturbations can cause a CNN to misclassify an image into a different class (*e.g.* a "1" into a "9" or a stop sign into a yield sign).

Thus, defending against these vulnerabilities will be critical to the further adoption of advanced computer vision systems. Here, we consider *white-box* attacks, in which an adversary can see the weights of the classification model. Most of these attacks work by taking advantage of the differentiable nature of the classification model, *i.e.* taking the gradient of the output class probabilities with respect to a particular pixel. Several previous works propose defense mechanisms that are differentiable transformations applied to an image before classification. These differentiable defenses appear to work well at first, but attackers can easily circumvent these defenses by "differentiating through them", *i.e.* by taking the gradient of a class probability with respect to an input pixel through both the CNN and the transformation.

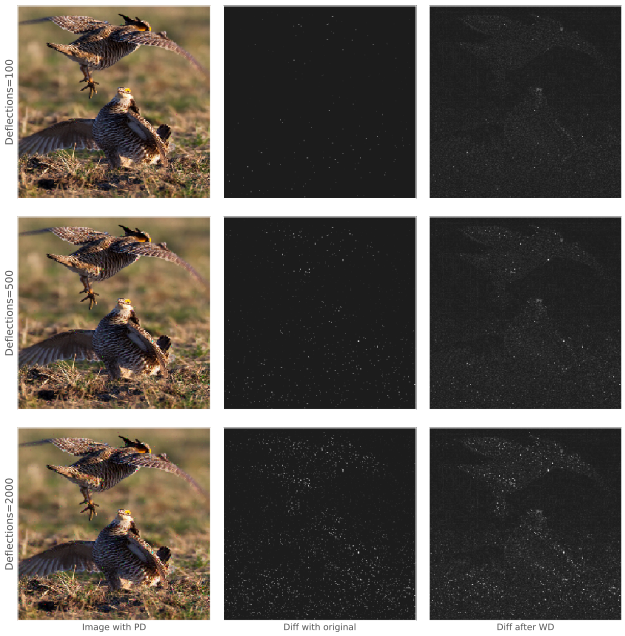


Figure 1: Impact of Pixel Deflection on a natural image and subsequent denoising using wavelet transform. Left: Image with given number of pixels deflected. Middle: Difference between clean image and deflected image. Right: Difference between clean image and deflected image after denoising. Enlarge to see details.

In this work, we present a defense method which combines two novel techniques for defending against adversarial attacks, which together modify input images in such a way that is (1) non-differentiable, and (2) frequently restores the original classification. The first component, *pixel deflection*, takes advantage of a CNN's resistance to the noise that occurs in natural images by randomly replacing some pixels with randomly selected pixels from a small neighborhood. We show how to weight the initial random pixel selection using a *robust activation map*. The second approach, *adaptive soft-thresholding* in the wavelet domain, which has been shown to effectively capture the distribution of natural images. This thresholding process smooths adversarially-perturbed images in such a way so as to reduce the effects of the attacks.

Experimentally, we show that the combination of these approaches can effectively defend against state-of-the-art attacks [44, 18, 6, 32, 36, 24] Additionally, we show that these transformations do not significantly decrease the classifier's accuracy on non-adversarial images.

In Section 2, we discuss the various attack techniques against which we will test our defense. In Sections 3 and 4 we discuss the established defense techniques against which we will compare our technique. In Sections 5, 6 and 7 we lay out the components of our defense and provide the intuition behind them. In Sections 9 and 10, we provide experimental results on a subset of ImageNet.

2. Adversarial Attacks

It has been established that most image classification models can easily be fooled [44, 18]. Several techniques have been proposed which can generate an image that is perceptually indistinguishable from another image but is classified differently. This can be done robustly when model parameters are known, a paradigm called *white-box attacks* [18, 24, 28, 6]. In the scenario where access to the model is not available, called *black-box attacks*, a secondary model can be trained using the model to be attacked as a guide. It has been shown that the adversarial examples generated using these substitute models are transferable to the original classifiers [37, 26].

Consider a given image x and a classifier $F_\theta(\cdot)$ with parameters θ. Then an adversarial example for $F_\theta(\cdot)$ is an image $\hat{x}$ which is close to x (*i.e.* $||x - \hat{x}||$ is small, where the norm used differs between attacks), but the classifier's prediction for each of them is different, *i.e.* $F(x) \neq F(\hat{x})$. *Untargeted attacks* are methods to produce such an image, given x and $F_\theta(\cdot)$. *Targeted attacks*, however, seek a $\hat{x}$ such that $F(\hat{x}) = \hat{y}$ for some specific choice of $\hat{y} \neq F(x)$, i.e. targeted attacks try to induce a specific class label, whereas untargeted attacks simply try to destroy the original class label.

Next, we present a brief overview of several well-known attacks, which form the basis for our experiments.

Fast Gradient Sign Method (FGSM) [18] is a single step attack process. It uses the sign of the gradient of the loss function, ℓ, w.r.t. to the image to find the adversarial perturbation. For a given value ϵ, FGSM is defined as:

$$\hat{x} = x + \epsilon \text{sign}(\nabla \ell(F(x), x)) \tag{1}$$

Iterative Gradient Sign Method (IGSM) [24] is an iterative version of FGSM. After each iteration the generated image is clipped to be within a ϵL_∞ neighborhood of the original and this process stops when an adversarial image has been discovered. Both FGSM and IGSM minimize the L_∞ norm w.r.t. to the original image. Let $x'_0 = x$, then after m iterations, the adversarial image is obtained by:

$$x'_{m+1} = \text{Clip}_{x,\epsilon}\Big\{x'_m + \alpha \times \text{sign}(\nabla \ell(F(x'_m), x'_m))\Big\} \tag{2}$$

L-BFGS [44] tries to find the adversarial input as a box-constraint minimization problem. L-BFGS optimization is used to minimize L_2 distance between the image and the adversarial example while keeping a constraint on the class label for the generated image.

Jacobian-based Saliency Map Attack (JSMA) [36] estimates the saliency of each image pixel w.r.t. to the classification output, and modifies those pixels which are most salient. This is a targeted attack, and saliency is designed to find the pixel which increases the classifier's output for the target class while tending to decrease the output for other classes.

Deep Fool (DFool) [32] is an untargeted iterative attack. This method approximates the classifier as a linear decision boundary and then finds the smallest perturbation needed to cross that boundary. This attack minimizes L_2 norm w.r.t. to the original image.

Carlini & Wagner (C&W) [6] is a recently proposed adversarial attack, and one of the strongest. C&W updates the loss function, such that it jointly minimizes L_p and a custom differentiable loss function that uses the unnormalized outputs of the classifier (*logits*). Let Z_k denote the logits of a model for a given class k, and κ a margin parameter. Then C&W tries to minimize:

$$||x - \hat{x}||_p + c * max\left(Z(\hat{x}_y) - max\{Z(\hat{x})_k : k \neq y\}, -\kappa\right) \tag{3}$$

For our experiments, we use L_2 for the first term, as this makes the entire loss function differentiable and therefore easier to train. Limited success has been observed with L_0 and L_∞ for images beyond CIFAR and MNIST.

We have not included recently proposed attacks like 'Projected Gradient Descent' [28] and 'One Pixel Attack' [43] because although they have been shown to be robust on datasets of small images like CIFAR10 and MNIST, they do not scale well to large images. Our method is targeted towards large natural images where object localization is meaningful, i.e. that there are many pixels outside the region of the image where the object is located.

3. Defenses

Given a classification model F and an image $\tilde{x}$, which may either be an original image x, or an adversarial image $\hat{x}$, the goal of a defense method is to either augment either F as F' such that $F'(\tilde{x}) = F(x)$, or transform $\tilde{x}$ by a transformation $\mathcal{T}$ such that $F(\mathcal{T}(\tilde{x})) = F(x)$.

One method for augmenting F is called Ensemble Adversarial training [46], which augments the training of deep convolutional networks to include various potential adversarial perturbations. This expands the decision boundaries around training examples to include some nearby adversarial examples, thereby making the task of finding an adversary within a certain ϵ harder than conventional models. Another popular technique uses distillation from a larger network by learning to match the softmax [38]. This provides smoother decision boundaries and thus makes is harder to find an adversarial example which is imperceptible. There are methods that proposes to detect the adversarial images as it passes through the classifier model [31, 2].

Most transformation-based defense strategies suffer from accuracy loss with clean images [14, 24], *i.e.* they produce $F(\mathcal{T}(x)) \neq F(x)$. This is an undesirable side effect of the transformation process, and we propose a transformation which tries to minimize this loss while also recovering the classification of an adversarial image. Detailed discussion on various kinds of transformation based defenses is provided in section 4.

4. Related Work

Transformation-based defenses are a relatively recent and unexplored development in adversarial defense. The biggest obstacle facing most transformation-based defenses is that the transformation degrades the quality of non-adversarial images, leading to a loss of accuracy. This has limited the success of transformations as a practical defense, as even those which are effective at removing adversarial transformations struggle to maintain the model's accuracy on clean images. Our work is most similar to Guo *et al.*'s [19] recently proposed transformation of image by quilting and Total Variance Minimization (TVM). Image quilting is performed by replacing patches of the input image with similar patches drawn from a bank of images. They collect one million image patches from clean images and use a k-nearest neighbor algorithm to find the best match. Image quilting in itself does not yield satisfactory results, so it is augmented with TVM. In Total Variance Minimization, a substitute image is constructed by optimization such that total variance is minimized. Total variation minimization has been widely used [17] as an image denoising technique. Our method uses semantic maps to obtain a better pixel to update and our update mechanism does not require any optimization and thus is significantly faster.

Another closely related work is from Luo *et al.* [27]. They propose a foveation-based mechanism. Using ground-truth data about object coordinates, they crop the image around the object, and then scale it back to the original size.

Our model shares the hypothesis that not all regions of the image are equally important to a classifier. Further, foveation-based methods can be fooled by finding an adversarial perturbation within the object bounding box. Our model does not rely on a ground-truth bounding box, and the stochastic nature of our approach means that it is not restricted to only modifying a particular region of the input.

Yet another similar work is from Xie *et al.* [48], in which they pad the image and take multiple random crops and evaluate ensemble classification. This method utilizes the randomness property that our model also exploits. However, our model tries to spatially define the probability of a presence of a perturbation and subsequently uses wavelet-based transform to denoise the perturbations.

5. Pixel Deflection

Much has been written about the lack of robustness of deep convolutional networks in the presence of adversarial inputs [33, 45]. However, most deep classifiers are robust to the presence of natural noise, such as sensor noise [11]. 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.

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 $n_i \sim \mathcal{U}(R^r_p \cap I)$
  $I'[p_i] = I[n_i]$
end

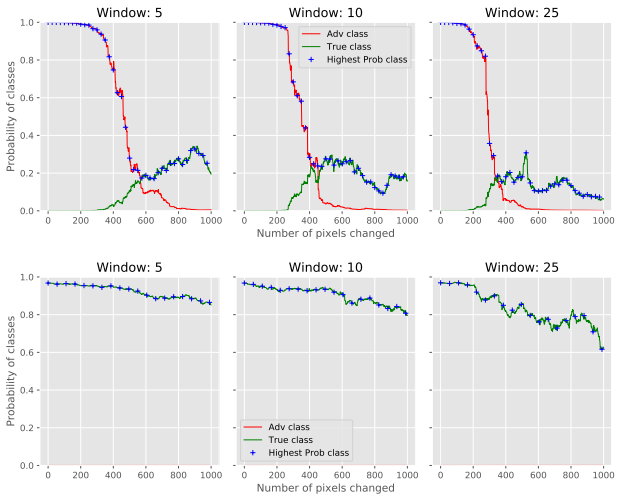


Figure 2: Average classification probabilities for an adversarial image (top) and clean image (bottom) after pixel deflection (Image size: 299x299)

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.

As shown in Figure 2, 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.

5.1. Distribution of Attacks

Most attacks search the entire image plane for adversarial perturbations, without regard for the location of the image content. This is in contrast with the classification models, which show high activation in regions where an object is present [50, 8]. 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.* [27] use the object coordinates to mask out the background region and show that this defends against some of the known attacks.

In Figure 3 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.

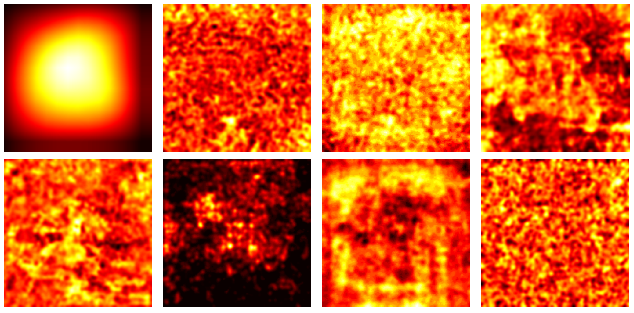


Figure 3: Visualization showing average location in the image where perturbation is added by an attacker. Clockwise from top left: Localization of most salient object in the image, FGSM, IGSM, FGSM-2 (higher ϵ), Deep Fool, JSMA, LBFGS and Carlini-Wagner attack.

6. Targeted Pixel Deflection

As we have shown in section 5, 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. Luo *et al.* [27] used this idea to mask the regions which did not contain the object, however, their method has two limitations which we will seek to overcome.

First, it requires ground-truth object coordinates and it is, therefore, difficult to apply to unlabeled inputs at inference time. We solve this by using a variant of class activation maps to obtain an approximate localization for salient objects. Class activation maps [51] are a weakly-supervised localization [35] technique in which the last layer of a CNN, often a fully connected layer, is replaced with a global average pooling layer. This results in a heat map which lacks pixel-level precision but is able to approximately localize objects by their class. We prefer to use weakly supervised localization over saliency maps [21], as saliency maps are trained on human eye fixations and thus do not always capture object classes [30]. Other weakly supervised localization techniques, such as regions-of-interest [39], capture more than a single object and thus are not suitable for improving single-class classification.

Second, completely masking out the background deteriorates classification of classes for which the model has come to rely on the co-occurrence of non-class objects. For instance, airplanes are often accompanied by a sky-colored background, and most classifiers will have lower confidence when trying to classify an airplane outside of this context. We take a Bayesian approach to this problem and use stochastic re-sampling of the background. This pre-

Figure 4: Difference between standard activation maps and robust maps under the presence of an adversary.

serves enough of the background to protect classification and drops enough pixels to weaken the impact of adversarial input.

6.1. Robust Activation Map

Class activation maps [51] are a valuable tool for approximate semantic object localization. Consider a convolutional network with k output channels on the final convolution layer (f) with spatial dimensions of x and y, and let $\boldsymbol{w}$ be a vector of size k which is the result of applying a global max pool on each channel. This reduces channel to a single value, $\boldsymbol{w_k}$. The class activation map, M_c for a class c is given by:

$$M_c(x, y) = \sum_k \boldsymbol{w}_k^c \; f_k(x, y) \tag{4}$$

Generally, one is interested in the map for the class for which the model assigns the highest probability. However, in the presence of adversarial perturbations to the input, the highest-probability class is likely to be incorrect. Fortunately, our experiments show that an adversary which successfully changes the most likely class tends to leave the rest of the top-k classes unchanged. Our experiments show that 38% of the time the predicted class of adversarial images is the second highest class of the model for the clean image. Figure 6 shows how the class of adversarial image relates to predictions on clean images. ImageNet has one thousand classes, many of which are fine-grained. Frequently, the second most likely class is a synonym or close relative of the main class (*e.g.* "Indian Elephant" and "African Elephant"). To obtain a map which is robust to fluctuations of the most likely class, we take an exponentially weighted average of the maps of the top-k classes.

$$\widehat{M}(x, y) = \sum_i^k \frac{M_{c_i}(x, y)}{2^i} \tag{5}$$

We normalize the map by diving it by its max so that values are in the range of $[0, 1]$. Even if the top-1 class is incorrect, this averaging reduces the impact of mis-localization of the object in the image.

The appropriate number of classes k to average over depends on the total number of classes. For ImageNet-1000, we used a fixed $k = 5$. While each possible class has its own class activation map (CAM), only a single robust activation map is generated for a particular image, combining information about all classes. ImageNet covers wide variety of object classes and most structures found in other datasets are represented in ImageNet even if class names are not bijectional. Therefore, Robust Activation Map (R-CAM) is trained once on ImageNet but can also localize objects from Pascal-VOC or Traffic Signs.

7. Wavelet Denoising

Because both pixel deflection and adversarial attacks add noise to the image, it is desirable to apply a denoising transform to lessen these effects. Since adversarial attacks do not take into account the frequency content of the perturbed image, they are likely to pull the input away from the class of likely natural images in a way which can be detected and corrected using a multi-resolution analysis.

Works such as [7, 42, 16] have shown that natural images exhibit regularities in their wavelet responses which can be learned from data and used to denoise images. These regularities can also be exploited to achieve better lossy image compression, the basis of JPEG2000. Many vision and neuroscience researchers [29, 41, 22] have suggested that the visual systems of many animals take advantage of these priors, as the simple cells in the primary visual cortex have been shown to have Gabor-like receptive fields.

Swapping pixels within a window will tend to add noise with unlikely frequency content to the image, particularly if the window is large. This kind of noise can be removed by image compression techniques like JPEG, however, the quantization process in JPEG uses fixed tables that are agnostic to image content, and it quantizes responses at all amplitudes while the important image features generally correspond to large frequency responses. This quantization reduces noise but also gets rid of some of the signal.

Therefore, it is unsurprising that JPEG compression recovers correct classification on some of the adversarial images but also reduces the classification accuracy on clean images [24, 9, 14, 19]. Dziugaite *et al.* [14] reported loss of 8% accuracy on clean images after undergoing JPEG compression.

We, therefore, seek filters with frequency response better suited to joint space-frequency analysis than the DCT blocks (and more closely matching representations in the early ventral stream, so that features which have a small filter response are less perceptible) and quantization techniques more suited to denoising. Wavelet denoising uses wavelet coefficients obtained using *Discrete Wavelet Transform* [3]. The wavelet transform represents the signal as a linear combination of orthonormal wavelets. These

wavelets form a basis for the space of images and are separated in space, orientation, and scale. The Discrete Wavelet Transform is widely used in image compression [1] and image denoising [7, 40, 42].

While the noise introduced by dropping a pixel is mostly high-frequency, the same cannot be said about the adversarial perturbations. Several attempts have been made to quantify distribution of adversarial perturbations [15, 25] but recent work by Carlini and Wagner [5] has shown that most techniques fail to detect adversarial examples. We have observed that for the perturbations added by well-known attacks, wavelet denoising yields superior results as compared to block DCT.

7.1. Hard & Soft Thresholding

The process of performing a wavelet transform and its inverse is lossless and thus does not provide any noise reduction. In order to reduce adversarial noise, we need to apply thresholding to the wavelet coefficients before inverting the transform. Most compression techniques use a hard thresholding process, in which all coefficients with magnitude below the threshold are set to zero: $Q(\hat{X}) = \hat{X} \ \forall \ |\hat{X}| > T_h$, where $\hat{X}$ is the wavelet transform of X, and T_h is the threshold value. The alternative is soft thresholding, in which we additionally subtract the threshold from the magnitude of coefficients above the threshold: $Q(\hat{X}) = \text{sign}(\hat{X}) \times \max(0, |\hat{X}| - T_h)$. Jansen *et al.* [23] observed that hard thresholding results in over-blurring of the input image, while soft thresholding maintains better PSNR. By reducing all coefficients, rather than just those below the threshold, soft thresholding avoids introducing extraneous noise. This allows our method to preserve classification accuracy on non-adversarial images.

7.2. Adaptive Thresholding

Determining the proper threshold is very important, and the efficacy of our method relies on the ability to pick a threshold in an adaptive, image specific manner. The standard technique for determining the threshold for wavelet denoising is to use a universal threshold formula called *VisuShrink*. For an image X with N pixels, this is given by $\sigma\sqrt{2\log N}$, where σ is the variance of the noise to be removed and is a hyper-parameter. However, we used *BayesShrink* [7], which models the threshold for each wavelet coefficient as a Generalized Gaussian Distribution (GGD). The optimal threshold is then assumed to be the value which minimizes the expected mean square error *i.e.*

$$T_h * (\sigma_x, \beta) = \arg\min_{T_h} E(\hat{X} - X)^2 \approx \frac{\sigma^2}{\sigma_x} \qquad (6)$$

where σ_x and β are parameters of the GGD for each wavelet sub-band. In practice, an approximation, as shown on right side of equation 6, is used. This ratio, also called T_{Bayes}, adapts to the amount of noise in the given image. Within a certain range of β values, BayesShrink has been shown to effectively remove artificial noise while preserving the perceptual features of natural images [7, 40]. As our experiments are carried out with images from ImageNet, which is a collection of natural images, we believe this is an appropriate thresholding technique to use. Yet another popular thresholding technique is Stein's Unbiased Risk Estimator (SUREShrink), which computes unbiased estimate of $E(\hat{X} - X)^2$. SUREShrink requires optimization to learn T_h for a given coefficient. We empirically evaluated results and SUREShrink did not perform as well as BayesShrink. Comparative results are shown in Table 6.

8. Method

The first step of our method is to corrupt the adversarial noise by applying targeted pixel deflection as follows:

(a) Generate a robust activation map $\widehat{M}$, as described in section 6.1.

(b) Uniformly sample a pixel location (x, y) from the image, and obtain the normalized activation map value for that location, $v_{x,y} = \widehat{M}(x, y)$.

(c) Sample a random value from a uniform distribution $\mathcal{U}(0, 1)$. If $v_{x,y}$ is lower than the random value, we deflect the pixel using the algorithm shown in Algorithm 1.

(d) Iterate this process K times.

The following steps are used to soften the impact of pixel deflection:

(a) Convert the image to YC_bC_r space to decorrelate the channels. YC_bC_r space is perceptually meaningful and thus has similar denoising advantages to the wavelets.

(b) Project the image into the wavelet domain using the discrete wavelet transform. We use the *db1* wavelet, but similar results were obtained with *db2* and *haar* wavelets.

(c) Soft threshold the wavelets using BayesShrink.

(d) Compute the inverse wavelet transform on the shrunken wavelet coefficients.

(e) Convert the image back to RGB.

9. Experimental Design

We tested our method on 1000 randomly selected images from the ImageNet [10] Validation set. We use ResNet-50 [20] as our classifier. We obtain the pre-trained weights from TensorFlow's GitHub repository. These models achieved a Top-1 accuracy of 76% on our selected images. This is in agreement with the accuracy numbers reported in [20] for a single-model single-crop inference.

By the definition set by adversarial attacks, an attack is considered successful by default if the original image is already mis-classified. In this case, the adversary simply returns the original image unmodified. However, these

cases are not useful for measuring the effectiveness of an attack or a defense as there is no pixel level difference between the images. As such, we restrict our experiments to those images which are correctly classified in the absence of adversarial noise. Our attack models are based on the Cleverhans [34] library[1] with model parameters that aim to achieve the highest possible misclassification score with a normalized RMSE ($|L_2|$) budget of $0.02 - 0.04$.

We will publicly release our implementation code.

9.1. Training

Our defense model has three hyper-parameters, which is significantly fewer than the classification models it seeks to protect, making it preferable over defenses which require retraining of the classifier such as [47, 31]. These three hyper-parameters are: σ, a coefficient for *BayesShrink*, r, the window size for pixel deflection, and K, the number of pixel deflections to perform. Using a reduced set of 300 images from ImageNet Validation set, We perform a linear search over a small range of these hyper-parameters. These images are not part of the set used to show the results of our model. A particular set of hyper-parameters may be optimal for one attack model, but not for another. This is primarily because attacks seek to minimize different L_p norms, and therefore generate different types of noise. To demonstrate the robustness of our defense, we select a single setting of the hyper-parameters to be used against all attack models. Figure 5 shows a visual indication of the variations in performance of each model across various hyper-parameter settings. In general, as the K and r increase, the variance of the resulting classification accuracy increases. This is primarily due to the stochastic nature of pixel deflection - as more deflections are performed over a wider window, a greater variety of transformed images can result.

10. Results & Discussion

In Table 1 we present results obtained by applying our transformation against various untargeted white-box attacks. Our method is agnostic to classifier architecture, and thus shows similar results across various classifiers. For brevity, we report only results on ResNet-50. Results for other classifiers are provided in Table 3. The accuracy on clean images without any defense is 100% because we didn't test our defense on images which were misclassified before any attack. We do not report results for targeted attacks as they are harder to generate [6] and easier to defend. Due to the stochastic nature of our model, we benefit from taking the majority prediction over ten runs; this is reported in Table 1 as Ens-10.

We randomly sampled 10K images from ILSVRC2012 validation set; this contained all 1000 classes with minimum of 3 images per class.

[1] https://github.com/tensorflow/cleverhans

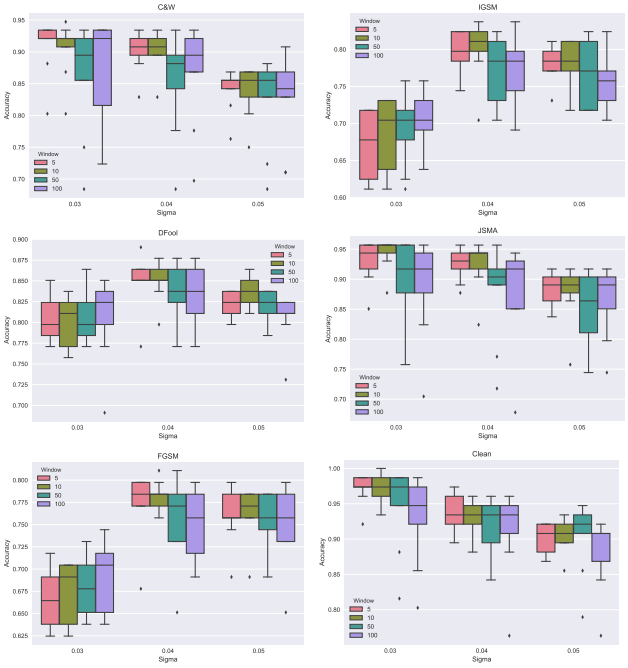


Figure 5: Linear search for model parameters

Model	$\|L_2\|$	No Defense	With Defense	
			Single	Ens-10
Clean	0.00	100	98.3	**98.9**
FGSM	0.05	20.0	79.9	**81.5**
IGSM	0.03	14.1	83.7	**83.7**
DFool	0.02	26.3	86.3	**90.3**
JSMA	0.02	25.5	91.5	**97.0**
LBFGS	0.02	12.1	88.0	**91.6**
C&W	0.04	04.8	92.7	**98.0**
Large perturbations				
FGSM	0.12	11.1	61.5	**70.4**
IGSM	0.09	11.1	62.5	**72.5**
DFool	0.08	08.0	82.4	**88.9**
JSMA	0.05	22.1	88.9	**92.1**
LBFGS	0.04	12.1	77.0	**89.0**

Table 1: Params: $\sigma = 0.04$, Window=10, Deflections=100 Top-1 accuracy on applying pixel deflection and wavelet denoising across various attack models. We evaluate non-efficient attacks at larger $|L_P|$ which leave visible perturbations to show the robustness of our model.

10.1. Results on various classifiers

Original classification accuracy of each classifier on selected 1000 images is reported in the table. However, we omit the images that were originally incorrectly classified, thus the accuracy of clean images without defense is

Attack	$\|L_2\|$	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 2: Top-1 accuracy of our model on various attack models.

always 100%. Weights for each classifier were obtained from Tensorflow GitHub repository [2].

Model	$\|L_2\|$	No Defense	With Defense	
			Single	Ens-10
ResNet-50, original classification 76%				
Clean	0.00	100	98.3	**98.9**
FGSM	0.05	20.0	79.9	**81.5**
IGSM	0.03	14.1	83.7	**83.7**
DFool	0.02	26.3	86.3	**90.3**
JSMA	0.02	25.5	91.5	**97.0**
LBFGS	0.02	12.1	88.0	**91.6**
C&W	0.04	04.8	92.7	**98.0**
VGG-19, original classification 71%				
Clean	0.00	100	99.8	**99.8**
FGSM	0.05	12.2	79.3	**81.3**
IGSM	0.04	9.79	79.2	**81.6**
DFool	0.01	23.7	83.9	**91.6**
JSMA	0.01	29.1	95.8	**98.5**
LBFGS	0.03	13.8	83.0	**93.9**
C&W	0.04	0.00	93.1	**97.6**
Inception-v3, original classification 78%				
Clean	0.00	100	98.1	**98.5**
FGSM	0.05	22.1	85.8	**87.1**
IGSM	0.04	15.5	**89.7**	89.1
DFool	0.02	27.2	82.6	**85.3**
JSMA	0.02	24.2	93.7	**98.6**
LBFGS	0.02	12.5	87.1	**91.0**
C&W	0.04	07.1	93.9	**98.5**

Table 3: Params: $\sigma = 0.04$, Window=10, Deflections=100 Top-1 accuracy on applying pixel deflection and wavelet denoising across various attack models.

[2] https://github.com/tensorflow/models/tree/master/research/slim#Pretrained

10.2. Comparison of results

- There are two main challenges when seeking to compare defense models. First, many attack and defense techniques primarily work on smaller images, such as those from CIFAR and MNIST. The few proposed transformation based defense techniques which work on larger-scale images are extremely recent, and currently under review [48, 19]. Second, because different authors target both different $|L_P|$ norms and different perturbation magnitudes, it is difficult to balance the strength of various attacks. We achieved 98% recovery on C&W with $|L_2|$ of 0.04 on ResNet-50, where Xie *et al*. [48] reports 97.1% on ResNet-101 and 98.8% on ens-adv-Inception-ResNet-v2. ResNet-101 is as stronger classifier than ResNet-50 and ens-adv-Inception-Resnet-v2 [46] is an ensemble of classifiers specifically trained with adversarial augmentation. They do not report the $|L_2|$ norm of the adversarial perturbations, and predictions are made on an ensemble of 21 crops. Guo *et al*. [19] have reported (normalized) accuracy of 92.1% on C&W with $|L_2|$ of 0.06, and their predictions are on an ensemble of 10 crops.

To present a fair comparison across various defenses we only measure the fraction of images which are no longer misclassified after the transformation. This ratio is known as Destruction Rate and was originally proposed in [24]. Value of 1 means all the misclassified images due to the adversary are correctly classified after the transformation.

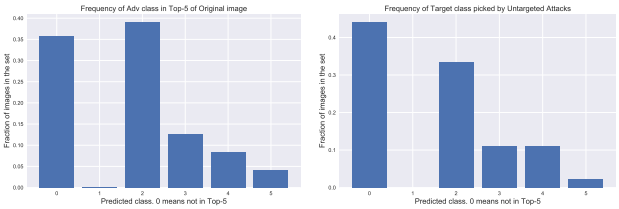


Figure 6: Left: Rank of adversarial class within the top-5 predictions for original images. Right: Rank of original class within the top-5 predictions for adversarial images. In both cases, 0 means the class was not in the top-5.

As seen in Figure 6, the predicted class of the perturbed image is very frequently among the classifier's top-5 predictions for the original image. In fact, nearly 40% of the time, the adversarial class was the second most-probable class of the original image. Similarly, the original classification will often remain in the top-5 predictions for the adversarial image. Unlike Kurakin *et al*. [24], our results are in terms of top-1 accuracy, as this matches the objective of the attacker. While top-1 accuracy is a more lenient metric for an attack method (due to the availability of nearly-synonymous alternatives to most classes in ImageNet-1000), it is a more difficult metric for a defense, as we must exactly recover

Defense	FGSM	IGSM	DFool	C&W
Feature Squeezing (Xu et al [49])				
(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*. [48])				
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*. [19])				
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 4: Destruction Rate of various defense techniques. $|L_2|$ 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 [48] are not reported.

the correct classification. These facts render top-5 accuracy an unsuitable metric for measuring the efficacy of a defense. Results reported for Carlini & Wagner [6] attacks are only for L_2 loss, even though they can be applied for L_0 and L_∞. Carlini & Wagner attack has been shown to be effective with MNIST and CIFAR but their efficacy against large images is limited due to expensive computation.

10.3. Ablation studies

Previous work [24, 14] has demonstrated the efficacy of JPEG compression as a defense against adversarial attacks due to its denoising properties. Das *et al*. [9] demonstrate that increasing the severity of JPEG compression defeats a larger percentage of attacks, but at the cost of accuracy on clean image. As our method employs a conceptually similar method to reduce adversarial noise via thresholding in a wavelet domain, we use JPEG as a baseline for comparison. In Table 5, we report accuracy with and without wavelet denoising with soft thresholding. While JPEG alone is effective against only a few attacks, the combination of JPEG and pixel deflection performs better than pixel deflection alone. The best results are obtained from pixel deflection and wavelet denoising. Adding JPEG on top of these leads to a drop in performance.

Model	JPG	WD	PD	PD JPG	WD PD JPG	WD PD
Clean	96.1	98.7	97.4	96.1	96.1	**98.9**
FGSM	49.1	40.6	79.7	81.1	78.8	**81.5**
IGSM	49.1	31.2	82.4	82.4	79.7	**83.7**
DFool	67.8	61.1	86.3	86.3	86.3	**90.3**
JSMA	91.6	89.1	95.7	93.0	93.0	**97.0**
LBFGS	71.8	67.2	90.3	89.1	88.9	**91.6**
C&W	85.5	95.4	95.4	94.1	93.4	**98.0**

Table 5: Params: $\sigma = 0.04$, Window=10, Deflections=100 Ablation study of pixel deflection (PD) in combination with wavelet denoising (WD) and JPEG compression.

Model	Hard	VISU	SURE	Bayes
Clean	39.5	96.1	92.1	**98.9**
FGSM	35.9	63.8	79.7	**81.5**
IGSM	42.5	67.8	81.1	**83.7**
DFool	37.2	78.4	87.7	**90.3**
JSMA	39.9	93.0	93.0	**97.0**
LBFGS	37.2	81.1	90.4	**91.6**
C&W	36.8	93.4	92.8	**98.0**

Table 6: Params: $\sigma = 0.04$, Window=10, Deflections=100 Comparison of various thresholding techniques, after application of pixel deflection.

In Table 6 we present a comparison of various shrinkage methods on wavelet coefficients after pixel deflection. For the impact of coefficient thresholding in the absence of pixel deflection, see Table 5. BayesShrink, which learns separate Gaussian parameters for each coefficient, does better than other soft-thresholding techniques. A brief overview of these shrinkage techniques are provided in Section 7.2, for more thorough review on BayesShrink, VisuShrink and SUREShrink we refer the reader to [7] [13] and [12] respectively. VisuShrink is a faster technique as it uses a universal threshold but that limits its applicability on some images. SUREShrink has been shown to perform well with compression but as evident, in our results, it is less well suited to denoising.

Attack	$\|L_2$	No Defense	With Defense		
Window=10, Deflections $\longrightarrow$			**10**	**100**	**1K**	**10K**
Clean	0.00	100	**98.4**	98.1	94.7	80.3
FGSM	0.04	19.2	75.7	**79.7**	71.7	69.1
IGSM	0.03	13.8	78.4	**81.7**	75.2	71.2
DFool	0.02	25.0	83.7	**87.7**	81.0	77.0
JSMA	0.02	25.9	91.7	**93.0**	87.7	67.7
LBFGS	0.02	11.6	85.0	**90.3**	82.4	73.0
C&W	0.04	05.2	89.4	**93.1**	86.8	69.7

Table 7: Top-1 accuracy with different deflections.

Attack	L2	No Defense	With Defense			
Deflections=100, Window ⟶			**5**	**10**	**50**	**100**
Clean	0.00	100	**98.6**	98.1	96.4	94.4
FGSM	0.04	19.2	**79.7**	**79.7**	78.4	76.7
IGSM	0.03	13.8	81.0	**81.7**	79.7	78.4
DFool	0.02	25.0	86.4	**87.7**	87.7	85.0
JSMA	0.02	25.9	92.3	**93.0**	91.7	90.3
LBFGS	0.02	11.6	89.4	**90.3**	89.0	88.1
C&W	0.04	05.2	91.8	**93.1**	90.5	89.2

Table 8: Top-1 accuracy with different window sizes.

Sampling technique (Random Pixel)				
Window ⟶	**5**	**10**	**50**	**100**
Uniform	**86.7**	**87.5**	**86.1**	**84.6**
Gaussian	80.0	81.4	79.0	76.4
Replacement technique (Uniform Sampling)				
Window ⟶	**5**	**10**	**50**	**100**
Min	73.0	64.4	49.1	44.3
Max	69.7	63.8	51.9	45.4
Mean	83.6	72.3	57.2	49.1
Random	**86.7**	**87.5**	**86.1**	**84.6**
Various Denoising Techniques				
Bilateral	**Anisotropic**	**TVM**	**Deconv**	**Wavelet**
78.1	84.1	77.26	85.12	**87.5**

Table 9: Top-1 accuracy averaged across all six attacks.

11. Conclusion

Motivated by the robustness of CNNs and the fragility of adversarial attacks, we have presented a technique which combines a computationally-efficient image transform, *pixel deflection*, with soft wavelet denoising. This combination provides an effective defense against state-of-the-art adversarial attacks. We show that most attacks are agnostic to semantic content, and using *pixel deflection* with probability inversely proportionate to robust activation maps (R-CAM) protects regions of interest. In ongoing work, we seek to improve our technique by adapting hyper-parameters based on the features of individual images. Additionally, we seek to integrate our robust activation maps with wavelet denoising.

12. Acknowledgement

We would like to thank NVIDIA for donating the GPUs used for this research. We would also like to thank Ryan Marcus, Brandeis University, for reviewing the paper.

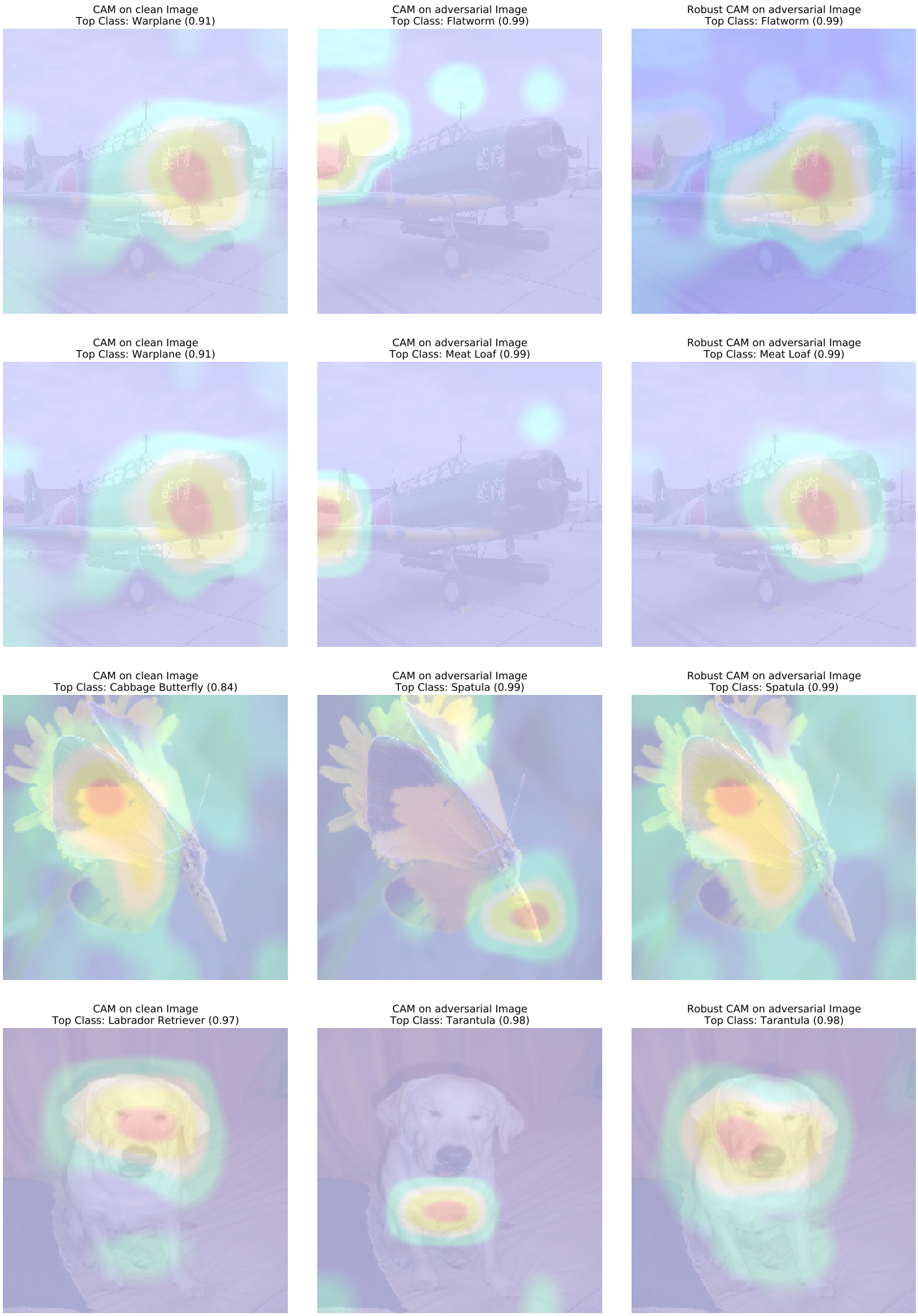
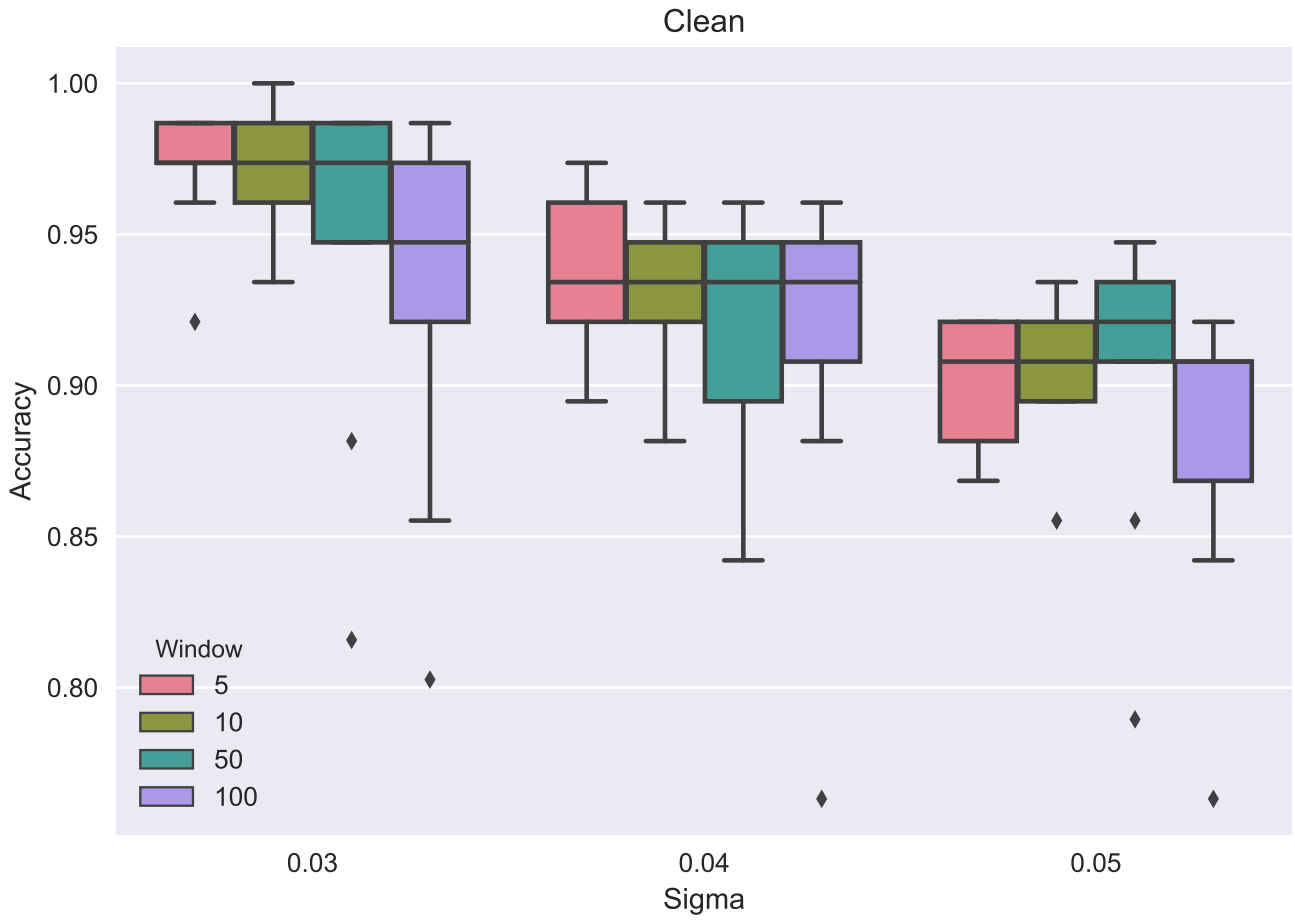
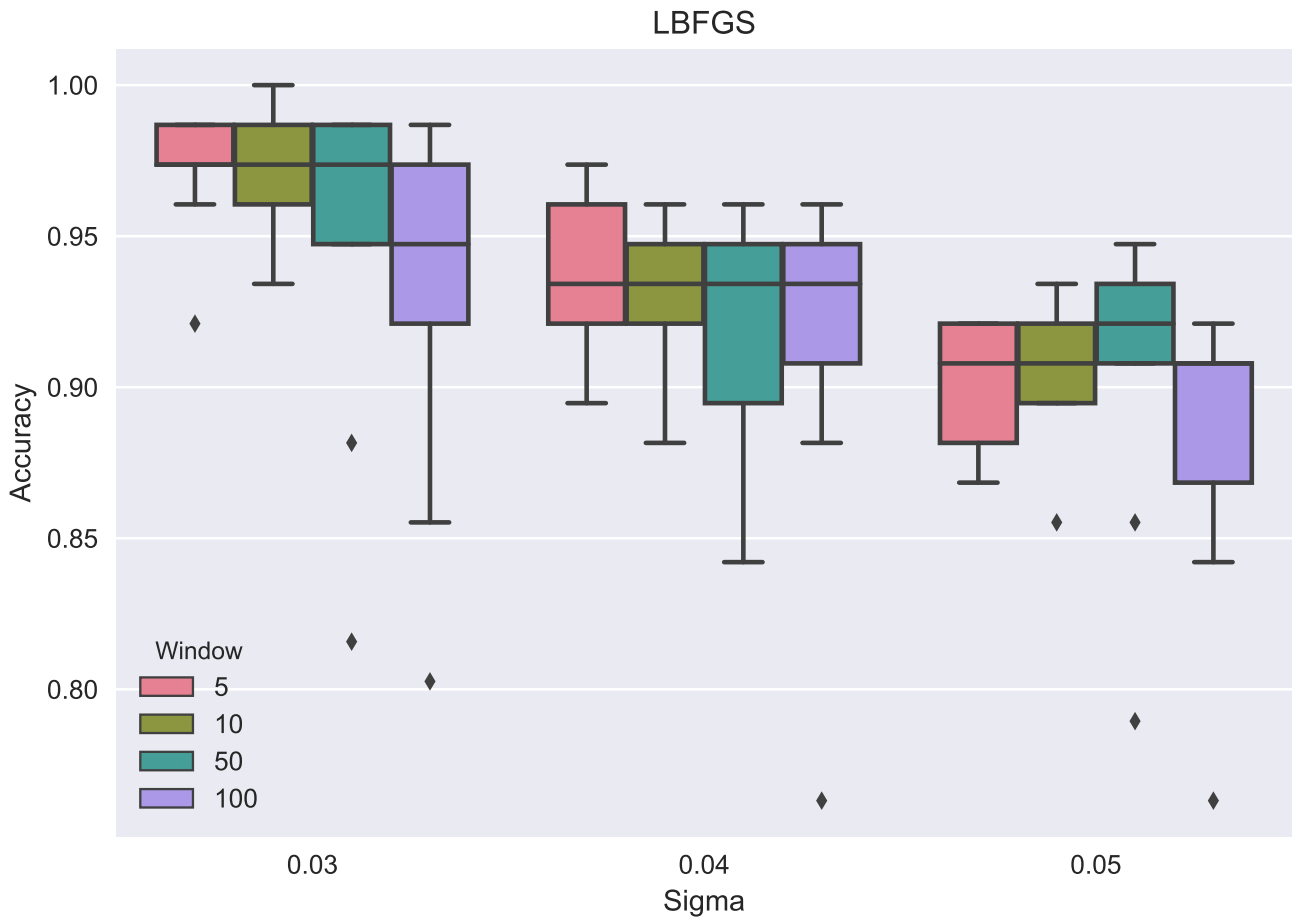
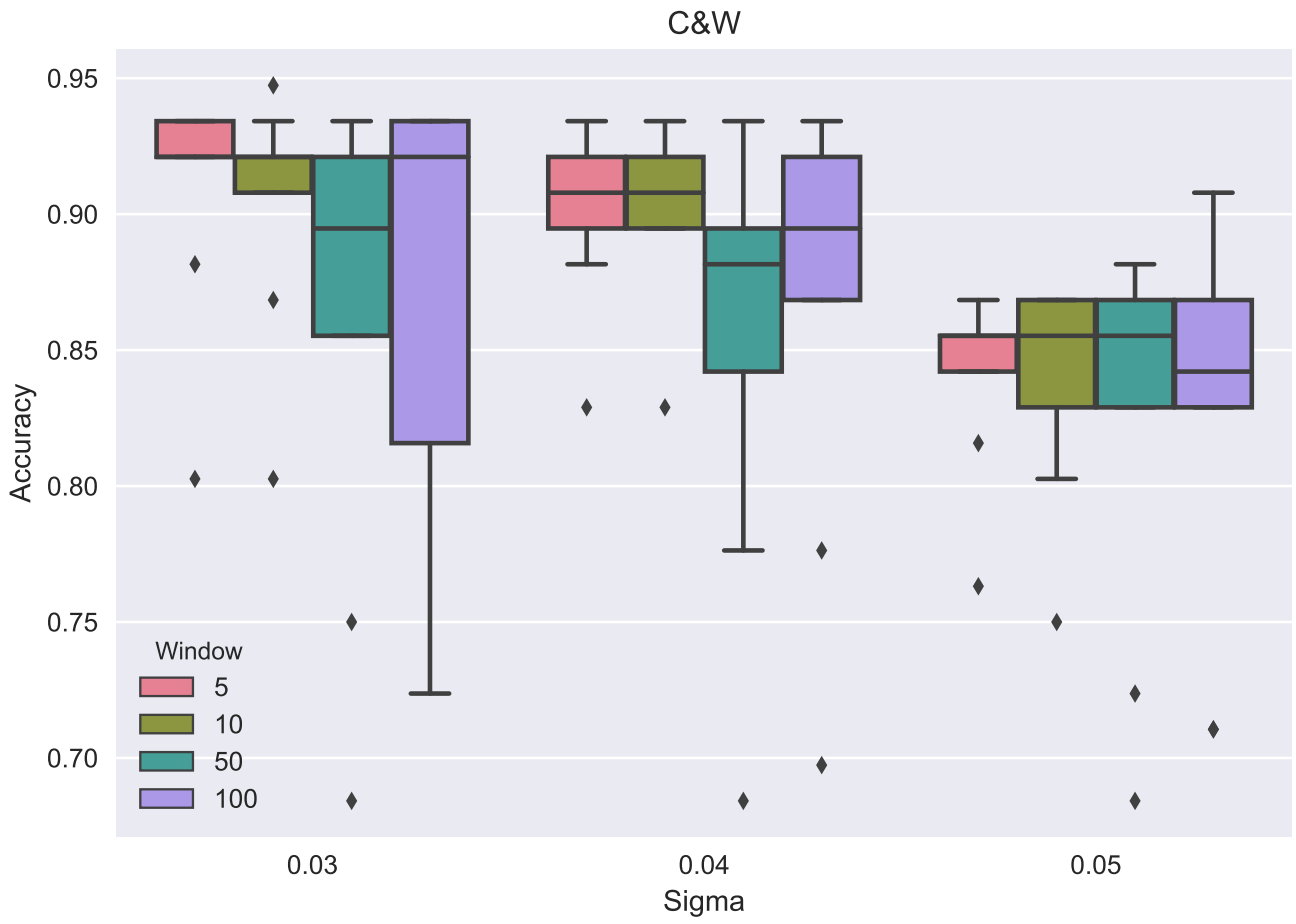


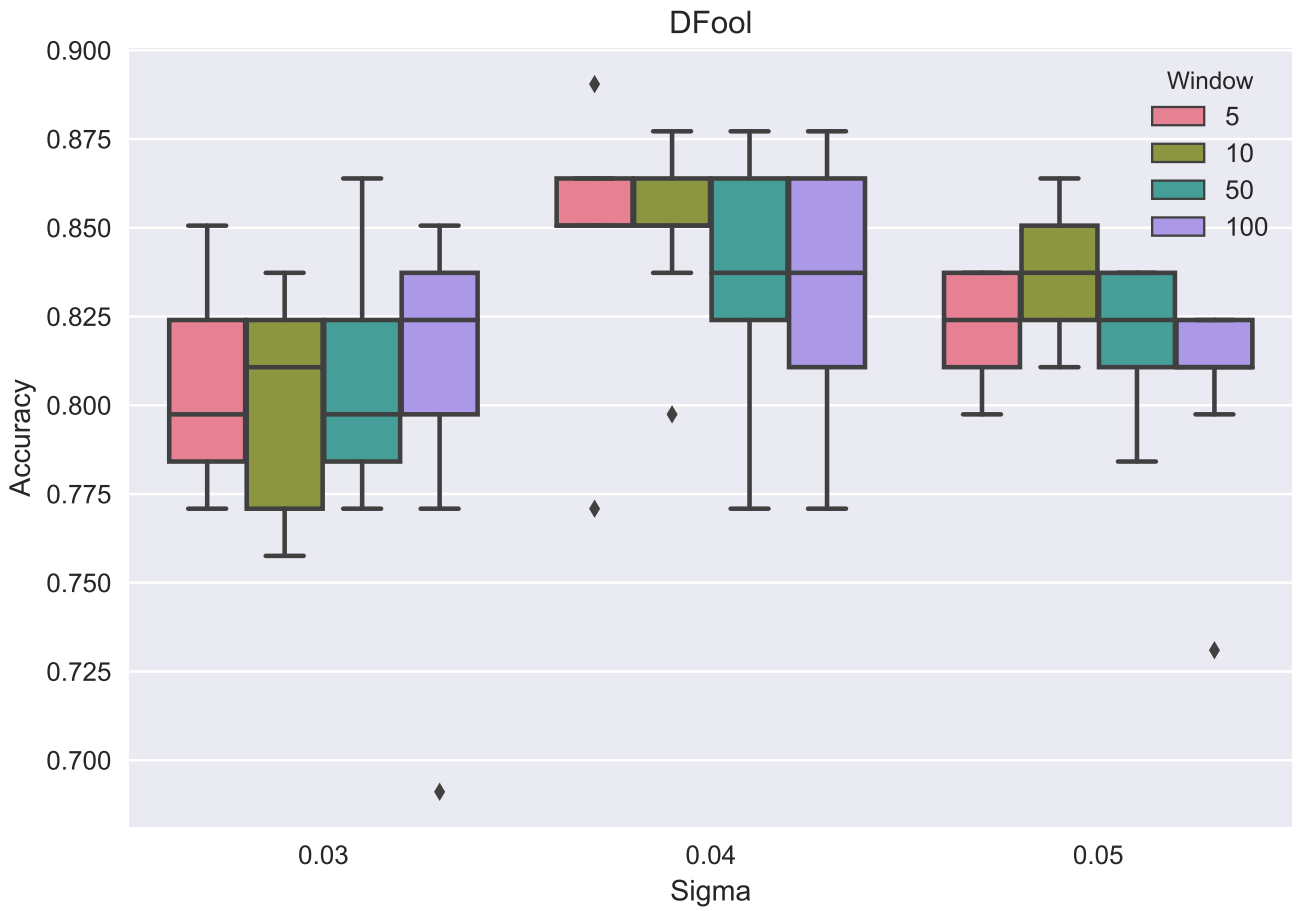
Figure 7: Comparison of Class activation maps and Robust Activation maps

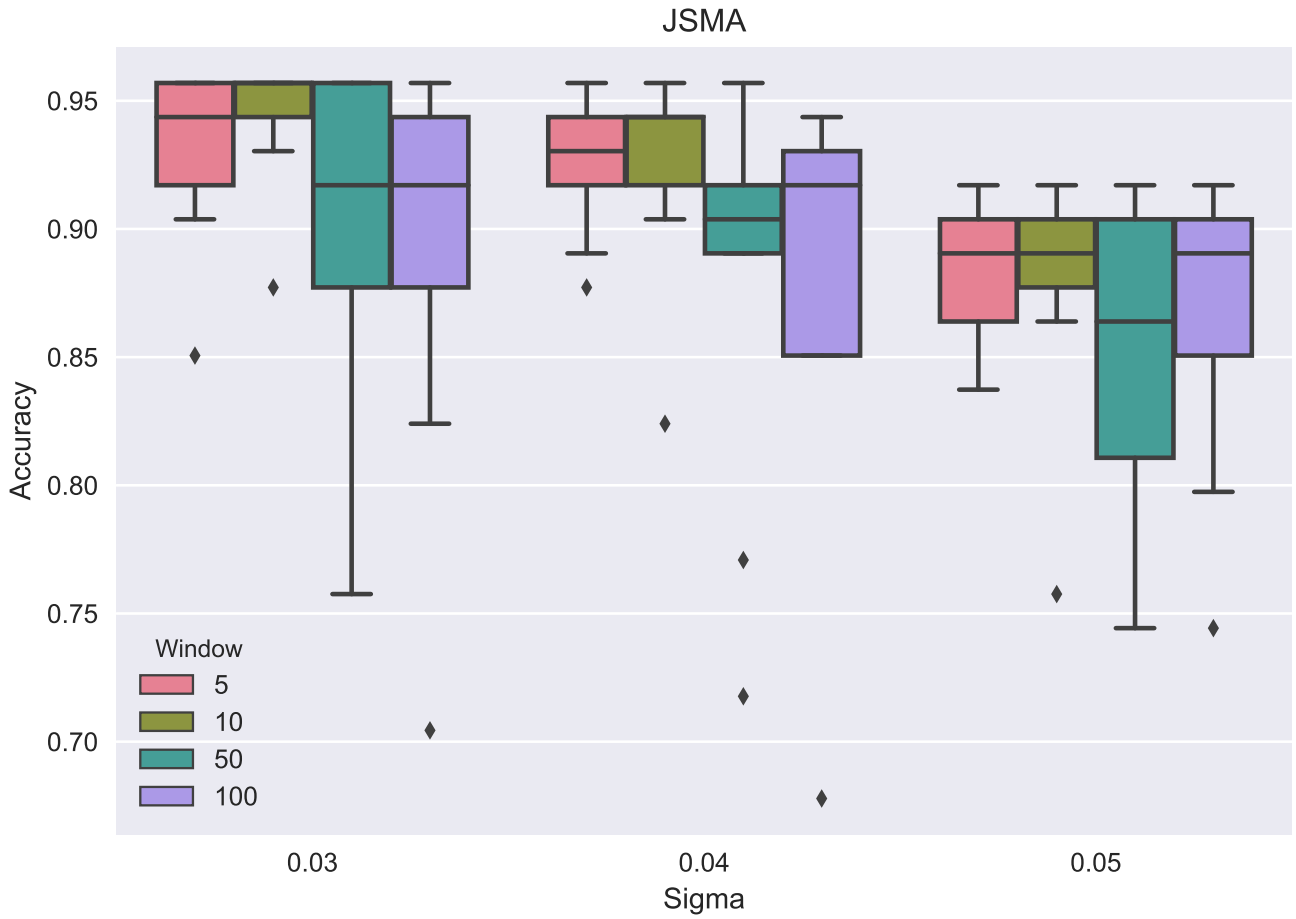
Full size figures of Figure 5 (Linear search for model parameters on training data)

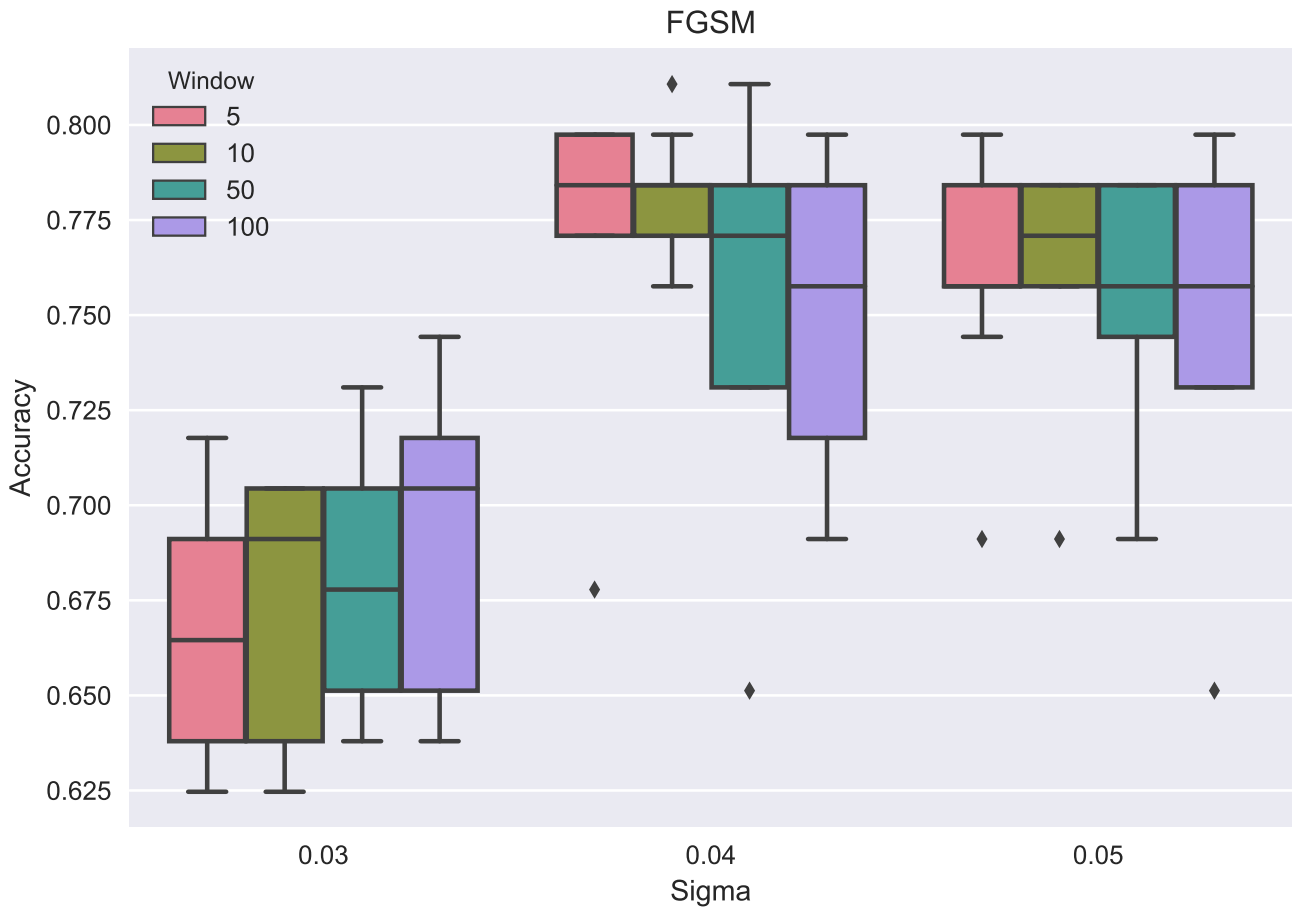


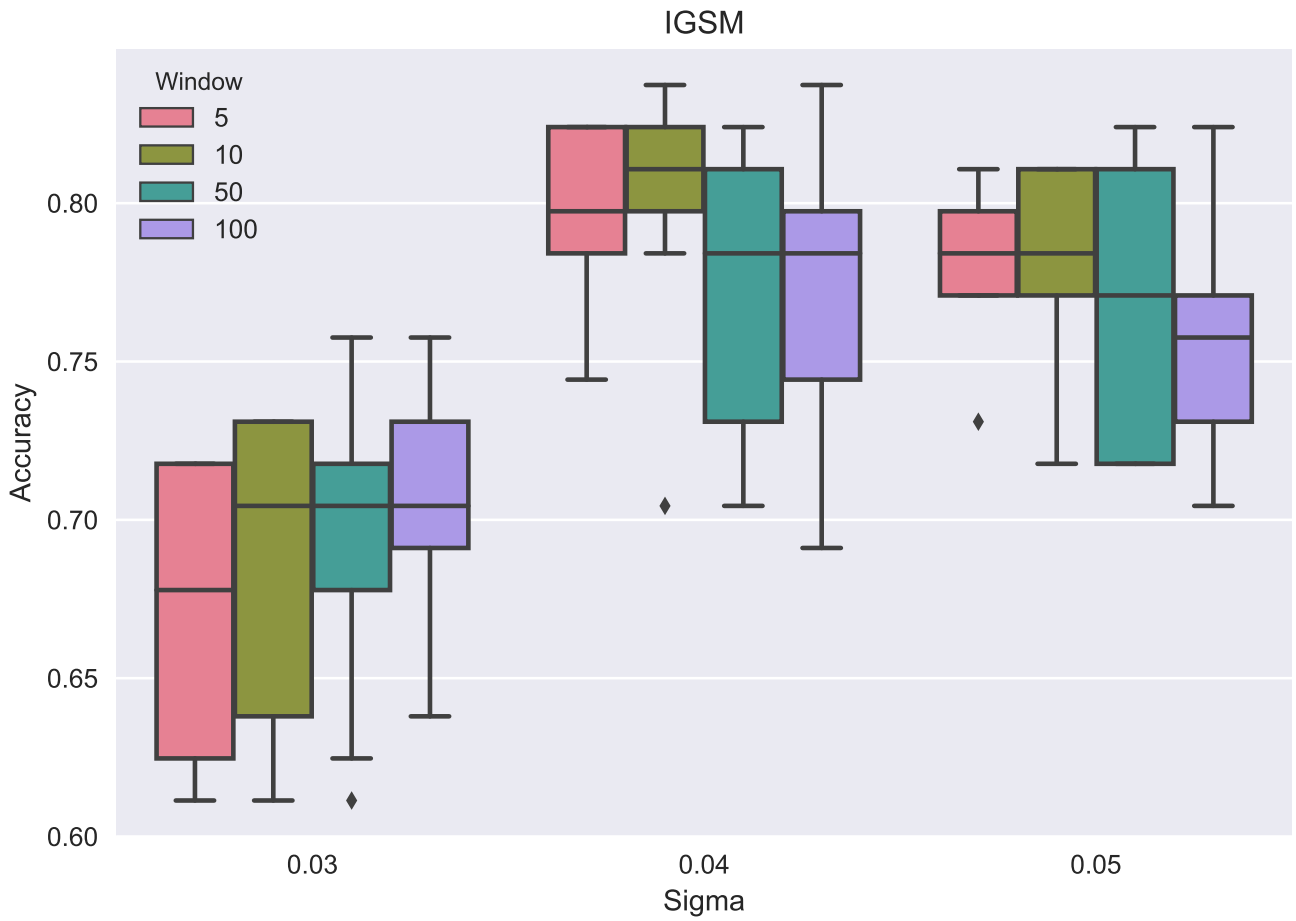

LBFGS
1.00
0.95
0.90
0.85
0.80
Accuracy
Window
5
10
50
100
0.03
0.04
0.05
Sigma


C&W
0.95
0.90
0.85
0.80
0.75
0.70
Accuracy
Window
5
10
50
100
0.03
0.04
0.05
Sigma


DFool
Window
5
10
50
100
Accuracy
0.900
0.875
0.850
0.825
0.800
0.775
0.750
0.725
0.700
0.03
0.04
0.05
Sigma


JSMA
Accuracy
0.95
0.90
0.85
0.80
0.75
0.70
Window
5
10
50
100
0.03
0.04
0.05
Sigma


FGSM
Window
5
10
50
100
0.800
0.775
0.750
0.725
0.700
0.675
0.650
0.625
Accuracy
0.03
0.04
0.05
Sigma


IGSM
Window
5
10
50
100
0.80
0.75
0.70
0.65
0.60
Accuracy
0.03
0.04
0.05
Sigma

References

[1] M. D. Adams. The jpeg - 2000 still image compression standard (last revised : June 30 , 2001). 2001. 6

[2] N. Akhtar, J. Liu, and A. Mian. Defense against universal adversarial perturbations. *arXiv preprint arXiv:1711.05929*, 2017. 3

[3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2):205–220, 1992. 5

[4] L. Bottou, Y. Bengio, and Y. LeCun. Global training of document processing systems using graph transformer networks. In *CVPR*, 1997. 1

[5] N. Carlini and D. A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *AISec@CCS*, 2017. 6

[6] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, 2017. 2, 7, 9

[7] S. G. Chang, B. Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 9 9:1532–46, 2000. 5, 6, 9

[8] 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. 4

[9] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression. *CoRR*, abs/1705.02900, 2017. 5, 9

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F. fei Li. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[11] S. Diamond, V. Sitzmann, S. P. Boyd, G. Wetzstein, and F. Heide. Dirty pixels: Optimizing image classification architectures for raw sensor data. *CoRR*, abs/1701.06487, 2017. 3

[12] D. L. Donoho and I. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. 1992. 9

[13] D. L. Donoho and I. Johnstone. Ideal spatial adaptation by wavelet shrinkage. 1994. 9

[14] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. A study of the effect of JPG compression on adversarial images. *CoRR*, abs/1608.00853, 2016. 3, 5, 9

[15] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. Detecting adversarial samples from artifacts. *CoRR*, abs/1703.00410, 2017. 6

[16] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Josa a*, 4(12):2379–2394, 1987. 5

[17] P. Getreuer. Rudin-osher-fatemi total variation denoising using split bregman. *IPOL Journal*, 2:74–95, 2012. 3

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. 2

[19] C. Guo, M. Rana, M. Cissé, and L. van der Maaten. Countering adversarial images using input transformations. 2017. 3, 5, 8, 9

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[21] X. Huang, C. Shen, X. Boix, and Q. Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 262–270, 2015. 4

[22] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148:574–91, 1959. 5

[23] M. Jansen. *Noise reduction by wavelet thresholding*, volume 161. Springer Science & Business Media, 2012. 6

[24] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016. 2, 3, 5, 8, 9

[25] Y.-C. Lin, M.-Y. Liu, M. Sun, and J.-B. Huang. Detecting adversarial attacks on neural network policies with visual foresight. *CoRR*, abs/1710.00814, 2017. 6

[26] Y. Liu, X. Chen, C. Liu, and D. X. Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, 2016. 2

[27] Y. Luo, X. Boix, G. Roig, T. A. Poggio, and Q. Zhao. Foveation-based mechanisms alleviate adversarial examples. *CoRR*, abs/1511.06292, 2015. 3, 4

[28] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017. 2, 3

[29] S. Marcelja. Mathematical description of the responses of simple cortical cells. *Journal of the Optical Society of America*, 70 11:1297–300, 1980. 5

[30] y. v. Matthias Kümmerer and Lucas Theis and Matthias Bethge, journal=CoRR. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. 4

[31] D. Meng and H. Chen. Magnet: A two-pronged defense against adversarial examples. In *CCS*, 2017. 3, 7

[32] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016. 2

[33] 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. 3

[34] I. G. R. F. F. F. A. M. K. H. Y.-L. J. A. K. R. S. A. G. Y.-C. L. Nicolas Papernot, Nicholas Carlini. cleverhans v2.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2017. 7

[35] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 685–694, 2015. 4

[36] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016. 2

[37] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016. 1, 2

[38] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *2016 IEEE Symposium on Security and Privacy (SP)*, 2016. 3

[39] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer. Semantic perceptual image compression using deep convolution networks. *2017 Data Compression Conference (DCC)*, 2017. 4

[40] R. Rangarajan, R. Venkataramanan, and S. Shah. Image denoising using wavelets. 2002. 6

[41] N. C. Rust, O. Schwartz, J. A. Movshon, and E. P. Simoncelli. Spatiotemporal elements of macaque v1 receptive fields. *Neuron*, 46:945–956, 2005. 5

[42] E. P. Simoncelli. Bayesian denoising of visual images in the wavelet domain. 1999. 5, 6

[43] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *CoRR*, abs/1710.08864, 2017. 3

[44] 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. 2

[45] 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. 3

[46] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. D. McDaniel. Ensemble adversarial training: Attacks and defenses. *CoRR*, abs/1705.07204, 2017. 3, 8

[47] F. Tramèr, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel. The space of transferable adversarial examples. *CoRR*, abs/1704.03453, 2017. 7

[48] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 3, 8, 9

[49] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *CoRR*, abs/1704.01155, 2017. 9

[50] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *CoRR*, abs/1506.06579, 2015. 4

[51] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016. 4, 5