Deep Protection: Universal Defense Against Adversarial Examples

Lurdh Pradeep Reddy Ambati
ID: 29199873
lambati@umass.edu

Sourabh Kulkarni
ID: 29572060
skulkarni@umass.edu

Abstract

In this project, we design a defense mechanism to protect deep networks against a multitude of adversarial attack types. We achieve this by using deep image prior, a generative deep convnet which attempts to reconstruct a target image from noise. The key idea is that adversarial images are naturally composed of the original image and some adversarial perturbations, and careful termination of the reconstruction process would yield us the original image without the perturbations.

We test our approach with the following setup: we take 100 imagenet samples (across 25 categories), apply 8 attack types, run the reconstruction process and the obtained result is then sent to a regular classifier. The process can have an automatic termination or a fixed termination. Automatic termination is the ideal case and achieves 97% average accuracy (top 5). Realistically, we need to determine a fixed termination. After choosing to fix the iterations at 1750, top 5 accuracy drops to 84% and our analysis showed that errors are uniform across underfit and overfit types. Further we have verified that our work integrates with commercial solution like clarifai; hence it is market-ready.

1. Introduction

Deep learning models have been widely successful and are being employed in several computer vision (CV) applications. This widespread use has left the CV application spaces open to adversarial attacks images that have been slightly perturbed to cause misclassification. For a human eye, these perturbed images look very similar to the original images, but attackers can create the perturbations such they can create desired labels even with black box access to the classifiers.

Typical adversarial techniques involve slightly perturbing image in pixel space to cause departure in feature/class space of a target deep network model. Very recently it was shown that even small rotations [4] or spatial transformations [18] can fool classifiers. Currently, there exists several attacks types [16, 5, 9].

In response, there have been several attempts to defend deep network models against these attacks [10]. Although successful to certain extent, these defenses are either typically geared to defend against a certain type of attack [10] or requires to be trained by large amount of adversarial examples for each attack type [10]. While these are direct approaches of defense, we believe that the problem could be approached in a different way.

In this project, we re-imagine adversarial defense problem as an image restoration problem, i.e., we consider the adversarial image to be a corrupted version of the original image, and attempt to restore its non-adversarial prior. To achieve this, we use Deep Image Prior (DIP) [17] an image restoration technique, and modify it to protect against adversarial images. The DIP process, which attempts to reconstruct the target image from noise iteratively, generates a series of image priors with increasing levels of detail. If this process is terminated carefully, we obtain an image prior which is detailed enough to be classified correctly, but does not include the adversarial perturbations.

As we show, Deep Protection is *attack type invariant* — no matter how the image is perturbed, the process of restoring image is the same, could indicate that this process is robust to any future attacks. The DIP process tends to add *simpler* features to the image at earlier stages and the more *difficult* features are added later. And this process tends to be same across all the attacks. Also, DIP process requires no training, hence Deep Protection can be deployed in-field right away.

To verify if this works in practice, we performed two main experiments — one where where the DIP process termination point was obtained with help of image labels, and the other where the process ran for a constant number of iterations before termination. Each experiment involved $\sim$600 samples, where 100 imagenet samples each adversarially modified by 8 different attack types. With automatic termination, we obtained an average top 5 accuracy of $\sim$97.4% and analyzing the distribution of termination points for these images allowed to determine a fixed termination point for the second experiment, which was chosen to be at 1750 iterations. Upon fixing the DIP termination

point at 1750 iterations, yielded an average top 1 accuracy of ∼71.2% and an average top 5 accuracy of ∼84%. The misclassifications could be attributed to i) the process running for too long - leading to the addition of adversarial perturbations into the image (Over-fitting), and ii) the process running for very few iterations, with not enough image detail added for making a correct classification (Under-fitting). While the robustness of Deep Protection is attractive, the top 1 accuracy on the original images drops from 100% to ∼69%.

Results suggest that Deep Protection is universal in the number of different attacks it can defend against. It is also evident that Deep Protection process is invariant of attack type, which hints at it being effective against any new kind of attack. One important aspect that remains to be worked on is the strategies for choosing termination points for the process.

2. Background

Deep Protection is a defense mechanism to protect deep networks against adversarial attacks using deep image priors. This section provides background on deep image prior, attack types, and defense types.

2.1. Deep Image Prior

Deep image prior was developed as a image restoration technique, which is a departure from traditional approaches to image restoration with deep generative models. The model resembles a convolutional auto-encoder, but functions quite differently. As is usual, inputs and weights of the model are randomly initialized, which gives a random output. While keeping the input constant to the random initialization, the weights are updated iteratively to minimize the L2 MSE loss of the output with respect to the target image over iterations, which is typically a corrupted version of an original image. Through this process of modifying the weights to reconstruct the corrupted image, the DIP creates increasingly detailed renditions of the target image (image priors). If this reconstruction process is terminated appropriately, the image prior is the original image without the corruptions present in the target image.

2.2. Adversarial Attacks

Adversarial attacks are various techniques that slightly perturb image in pixel space to cause departure in feature/class space of a target deep network model. Several approaches exist, but basic premise of slight image perturbation stays the same. In this project we use the foolbox [14] framework to generate attacks. Listed below are the attacks used in the project with brief explanation on how they operate:

- **Gradient Sign (GS) Attack**. Adds the sign of the gradient to the image, gradually increasing the magnitude until the image is misclassified. This attack is often referred to as Fast Gradient Sign Method and was introduced in [5].
- **Additive Gaussian Noise (AGN) Attack**. Adds Gaussian noise to the image, gradually increasing the standard deviation until the image is misclassified.
- **Blended Uniform Noise (BUN) Attack**. Blends the image with a uniform noise image until it is misclassified.
- **ADef (AD) Attack**. Attack that distorts the image, i.e. changes the locations of pixels. The algorithm is described in [1]
- **Newton Fool (NF) Attack**. A simple gradient-based attack for finding adversarial examples, described in [7]
- **DeepFool (DF) Attack**. It is based on an iterative linearization of the classifier to generate minimal perturbations that are sufficient to change classification labels. The algorithm is described in [3]
- **Carlini Wagner (CW) Attack**. An advanced distance metric based attack targeted against networks with defensive distillation. The algorithm is described in [2]
- **Salt and Pepper Noise (SPN) Attack**. Increases the amount of salt and pepper noise until the image is misclassified.

2.3. Defense Techniques

Defense techniques attempt to make networks robust to some of the attacks described above, though a certain defense technique is only effective against a subset of these attacks. The various defense techniques can be classified into the following approaches:

- **Adversarial Training**. These techniques [10] try to make models robust to adversarial attacks by retraining them with the examples of a certain attack. This requires that one can generate adversarial examples of the attack the technique is trying to defend.
- **Image Pre-processing**. These techniques perform various forms of filtering (e.g., bilateral filtering [13], etc.), transformations (rotation, mirroring etc.)
- **Using generative models**. These techniques use generative models like GANs and VAEs to take adversarial image as input and output a non-adversarial image [15].
- **Bayesian Models**. Probability theoretic models can detect adversarial examples as they can have a very low log-likelihood with respect to the training set. E.g., Gaussian Process Hybrid Deep Neural Network [11]

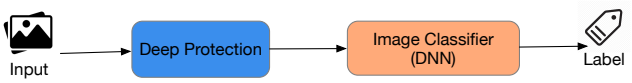


Figure 1. A depiction of Deep Protection architecture.

3. Approach

We first outline Deep Protection's architecture and then present its implementation and the details of the data sets used in the experiments.

3.1. Architecture

Deep Protection relies on deep image priors (DIP) for defending deep networks against adversarial images. To be specific, we employ a DIP based denoiser as a prepossessing step for every test input, before passing onto a neural network image classifier. Figure 1 depicts the Deep Protection architecture. Recall, DIP is an *iterative* process, the longer (more the iterations) it takes the increasingly detailed rendition of the target image is constructed.

With deep protection, the time to classify the test input will increase with the number of iterations the DIP denoiser runs, so it would be ideal to have a automatic termination process that can determine when to terminate. But it is complex to design such a strategy, so we employ the static termination process, where we terminate the DIP denoiser after a fixed number of iterations.

To do so, we perform some initial qualitative analysis with respect to the behavior of the *default* DIP architecture with adversarial examples vs. the usual distorted images it was designed to work with and tune various DIP process hyper-parameters for achieving better accuracy. We have evaluated different choices of the hyper-parameters and our empirical analysis showed that following hyper parameters work well across different attack types.

1. *Initialization:* Random weights and random inputs
2. *Loss function:* L2 MSE with target image
3. *Optimization algorithm:* Adam optimizer [8]
4. *Activation function:* Swish activation function [12]

After we tune the DIP denoiser, we determine the fixed iteration number. This is a complex process, as the ideal termination point (where we can extract the original image without perturbations) depends on both the attack type and image/category as well. Section4 will expand on this further.

3.2. Implementation

We implemented the proposed work in python using bindings such as PyTorch, numpy, and foolbox. As part of our work, we generate the adversarial images based on the imagenet images. To do so, we use foolbox framework which will generate the adversarial images, given the input images and the correct label. The foolbox [14] framework provides multiple models to generate adversarial images. The denoiser will be based on the original deep image prior system, but we modied its architecture as well as adjusted its hyper-parameters to better suit our work. Finally, the image classier is a pretrained resnet-18 neural network (other architecture can also work).

3.3. Data Sets

In this work, we use imagenet data set. Imagenet is an image database consisting of hundreds and thousands of images in each category and it has around 1000 categories. For evaluation and training, we sample 100 images from 25 different categories from imagenet and generate the adversarial images based on them using the foolbox framework.

3.4. Exploring other applications - text extraction from maps

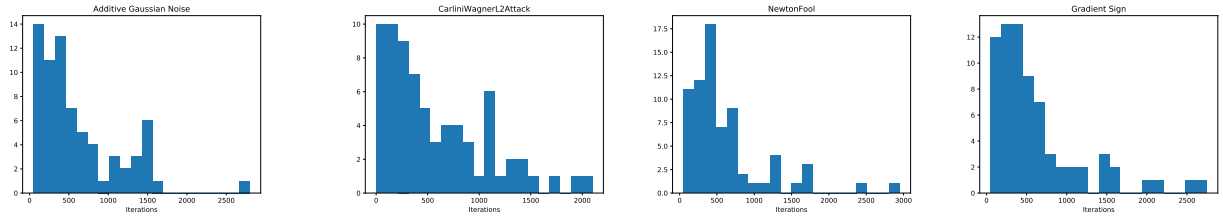
Apart from the addressing the problem statement, we explore application of DIP to other problem spaces in computer vision. The application being currently explored is text extraction from maps. To perform this, we use the image priors and their difference from the original image (image residuals). Through the evolution of the learning process, the DIP tends to add *simpler* features to the image at earlier stages (e.g., map background, roads) and the more *difficult* features are added later(e.g., Text, some tree textures). When we consider the initial iterations of the learning process, the image prior consists of the simpler features, which make the image residuals to be complex features. Upon applying a ReLU like activation function (if all RGB pixels are greater than 0.6*mean of image be 255 else be 0), we were able to extract the text from the maps fairly easily and to good accuracy (see Section4).

4. Experiment

We evaluate Deep Protection in two modes — automatic termination mode and static termination mode. We report top 5 accuracy, top 1 accuracy and distribution of misclassification errors (static termination mode). Further, we use the results from automatic termination mode to determine the fixed iteration number for static termination mode.

4.1. Automatic termination

In automatic termination mode, the process runs for a maximum of 3000 iterations per image. For every 50 iterations, the intermediate image prior is classified using the resnet18. The top 5 classes predicted by the classifier are monitored, and if the true label is present in them while



(a) Additive Gaussian Noise Attack (b) Carlini Wagner Attack (c) Newton Fool Attack (d) Gradient Sign Attack

Figure 2. The histogram of termination points in automatic termination mode for four different attacks.

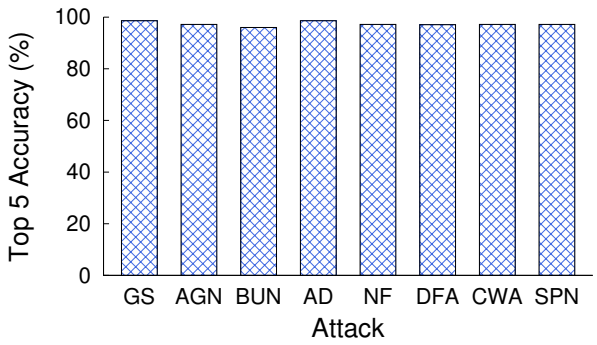


Figure 3. Top 5 accuracy of Deep Protection in automatic termination mode across 8 different attacks.

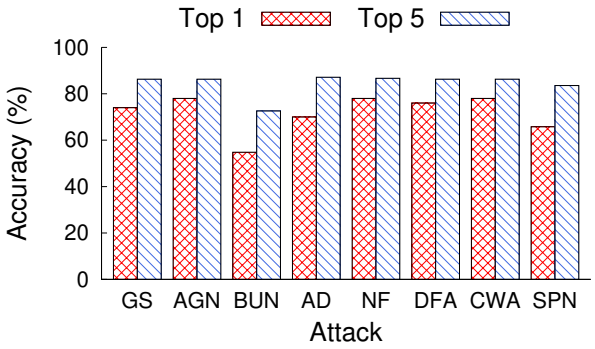


Figure 4. Accuracy of Deep Protection in static termination mode across 8 different attacks.

the adversarial label is not, then the process is terminated. While this is straightforward to do with access to true labels, it would be a challenging task to estimate a good termination point at test time and as it yet remains to be explored further.

We repeat this experiment for all 8 attacks (with 100 imagenet images). For generating the adversarial images, foolbox requires that our classifier (resnet18 in our case) classify the image correctly in the first place, hence for the misclassifed images we cannot generate adversarial images using foolbox. So we report the results for the adversarial images only if their respective clean image was classified correctly in the first place, we found that resnet18 was able to classify 73 out of 100 images correctly in the first place. So we have a total of 584 image samples across 8 attack types.

Figure 3 presents the result of this experiment. Note that we report the top 5 accuracy for this experiment, as automatic termination process stops once the input sample's true label is in top 5. As we can see that we achieve a accuracy of over $\sim$95% across all the attack types, given the respective clean images of adversarial inputs were correctly classified by resnet18 classifier.

Apart from the accuracy, we record the termination point for every sample across all attacks and we use this data to determine the optimal fixed iteration number. Figure 2 shows the histogram of iteration/termination point for four attacks. (histograms looks similar for all the attacks). We can see that most of the samples are cleaned before the 1500 iterations, thus we chose the fixed iteration number as 1750.

4.2. Static termination strategy

We repeat the above experiment after choosing to fix the iteration number at *1750*. Here, we report both top 5 and top 1 accuracy results. We expect the (top 5) accuracy of static termination to be lower than automatic termination strategy, as some images require more than 1750 iterations to be classified correctly (under-fit) and for images which are terminated early will result in over-fitting (adversarial noise started to reappear).

Figure 4 presents the results of top 5 and top 1 accuracy. As expected the top 5 accuracy has reduced across all the attacks in comparison to automatic termination mode. We can see that top 5 accuracy is 84% and top 1 accuracy is 71%, given the respective clean images of adversarial inputs were correctly classified by resnet18 classifier.

Further, we analyzed the source of the misclassification errors (for top 5 accuracy results). We report a misclassification as overfit error, if the sample was misclassified during last iteration and it was classified at least once correctly before the final iteration. We report a misclassification as underfit error, if the sample was misclassified during last iteration and it was never classified correctly before the final iteration as well. Figure 5 shows the result of them, we can see that errors are uniform across the over-fit and under-fit errors.

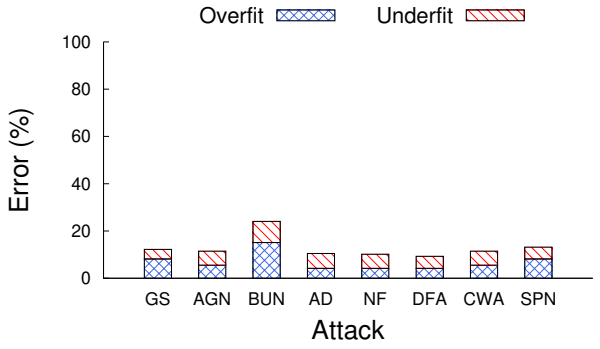


Figure 5. Distribution of errors in static termination mode across 8 different attacks.

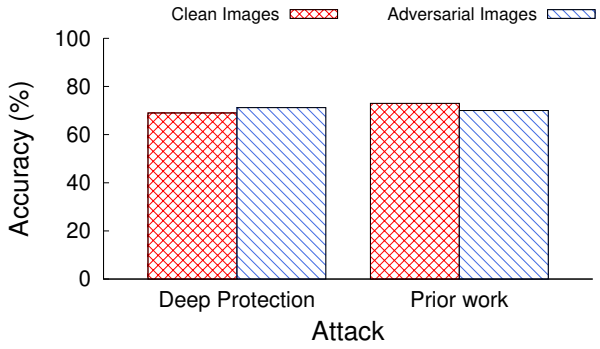


Figure 6. Comparison of Deep Protection and Prior work.

4.3. Comparative Analysis

We next compare our work with a recent arVix paper [6] that also uses DIP denoiser to address the adversarial image problem in deep networks. In that work, authors use a early stopping strategy for DIP process, where they stop cleaning the image once the loss is below a threshold and also they use a pre-trained resnet152 neural network for image classification. We have tried that termination strategy, but results were not promising. While we have experimented with 8 different attacks in our work, they use only one attack for the evaluation. Apart from that they use 1000 images from imagenet, whereas we use 100 images from imagenet (due to compute and time limitations).

In this prior work, authors report the top 1 accuracy when they pass the clean images through the DIP process before classifying as well. For the comparison purpose we repeat the similar experiment, we pass the clean images through deep protection (in static termination mode) and report the top 1 accuracy.

Figure 6 presents the comparison of results of Deep Protection (in static termination mode) and prior work. We can see that Deep Protection achieves very similar results as the prior work, but our evaluation includes results from 8 different attacks compared to *only one* attack in this prior work. Interestingly, we can see that accuracy of clean images and accuracy of adversarial images in Deep Protection (and prior work) is approximately same. It shows that DIP process works equally well for both the cleaned images and adversarial images.

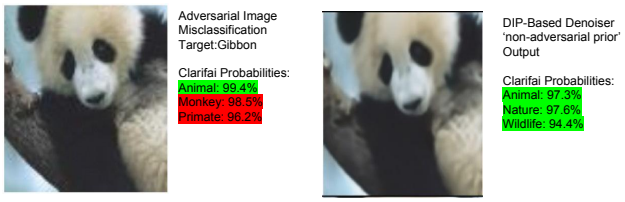


Figure 7. Comparison of class probabilities of adversarial and denoised image in clarifai classification service.

Note for comparison purpose, we have scaled the results from prior work such that they reflect our criteria where we report results for the adversarial samples only when their respective clean images were correctly classified.

4.4. Real World Example

Additionally, to test the approach with real-world systems we use a state-of-the art image classification service (clarifai.com) on both the adversarial image as well as its non-adversarial prior obtained from the Deep Protection Process. Figure 7 illustrates one such example with panda as original image.

4.5. Text Extraction from Maps

Testing text extraction capability of DIP-based text extractor prototype. Two maps are considered a plain map and map with terrain. We show the extracted text in each case. This is through preliminary experiments and we expect better results over time with few optimizations. Figure 8 presents a demonstration of the text extraction capability.

5. Conclusion

In conclusion, we present Deep Protection — a defense mechanism to protect deep networks against a multitude of adversarial attack types. We achieved this by using a image restoration technique called deep image prior. The key insight is that deep image prior process is attack type invariant and it requires no training, so it is "market ready". We have evaluated Deep Protection in two modes — automatic termination mode and static termination mode. Our analysis showed that static termination mode can achieve a top 5 accuracy of 84% and a top 1 accuracy of 71%, given that respective clean images of inputs are classified correctly by classifier. Finally, we demonstrated that our model can readily integrate with existing image classification service (clarifai).

Working on this project was really interesting and we have learned a lot regarding the image classification, ad-

Figure 8. Extracting text from the maps using Deep Image Prior.

versarial image generation, and most importantly about the Deep Image Prior process. Deep Image Prior works very well in our experience and has piqued our interest in learning about its internals. Following are some future ideas to further improve this project:

- **Automatic termination strategies**. Working on this would improve both the accuracy and performance as well. Some starting ideas are: monitoring the loss curve for each sample and terminate when there is a plateau.
- **Bigger dataset for evaluation**. We can evaluate the Deep Protection on a bigger dataset and with more attacks to see how it well it generalizes.
- **Binary Adversarial detector**. The proposed solution is slow, as denoising every test input takes signicant time. To address this performance issue, we can add another pre-processing step — binary classier that outputs whether the input test image is likely an adversarial image or not. If its output is yes, then the input is passed through the DIP-based denoiser and nally the classifier. If the binary classier output is no, then the image is directly fed to the classifier.

References

[1] R. Alaifari, G. S. Alberti, and T. Gauksson. Adef: an iterative algorithm to construct adversarial deformations. *CoRR*, abs/1804.07729, 2018.

[2] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644, 2016.

[3] S. M. M. Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *CoRR*, abs/1511.04599, 2015.

[4] L. Engstrom, D. Tsipras, L. Schmidt, and A. Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *CoRR*, abs/1712.02779, 2017.

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

[6] A. Ilyas, A. Jalal, E. Asteri, C. Daskalakis, and A. G. Dimakis. The robust manifold defense: Adversarial training using generative models. *CoRR*, abs/1712.09196, 2017.

[7] U. Jang, X. Wu, and S. Jha. Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACSAC 2017, 2017.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

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

[10] A. Kurakin, I. J. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, J. Wang, Z. Zhang, Z. Ren, A. L. Yuille, S. Huang, Y. Zhao, Y. Zhao, Z. Han, J. Long, Y. Berdibekov, T. Akiba, S. Tokui, and M. Abe. Adversarial attacks and defences competition. *CoRR*, abs/1804.00097, 2018.

[11] R. Novak, L. Xiao, J. Lee, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein. Bayesian convolutional neural networks with many channels are gaussian processes. *CoRR*, abs/1810.05148v1, 2018.

[12] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *CoRR*, abs/1710.05941, 2017.

[13] N. Ratzlaff and F. Li. Unifying bilateral filtering and adversarial training for robust neural networks. *CoRR*, abs/1804.01635, 2018.

[14] J. Rauber, W. Brendel, and M. Bethge. Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models. *CoRR*, abs/1707.04131, 2017.

[15] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *CoRR*, abs/1805.06605, 2018.

[16] 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.

[17] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Deep image prior. *CoRR*, abs/1711.10925, 2017.

[18] C. Xiao, J. Zhu, B. Li, W. He, M. Liu, and D. Song. Spatially transformed adversarial examples. *CoRR*, abs/1801.02612, 2018.