Image Processing Defense on Adversarial Attack

Mark Liu(maliu2), Li Cai(licai0)

0. Abstract

The state-of-art Deep Neural Networks have achieved very high accuracy rate on image processing. Although DNN has a lot of improvements on classification recent years, research has shown that a very small perturbation on the input will make it misclassify images [1]. This is due to model's linear behavior in high-dimensional space. Methods on input transformation has been proposed to defense adversarial attacks. They are model-agnostic and computationally cheap compared to model refining methods. This project extends and analyzes those methods.

1. Introduction

DNN has involved in many areas for image classification, such as face recognition[13]. Different from white noise, an imperceptible perturbation has a specific direction to attack the model [10]. White noise makes DNN's prediction slightly less accurate, but adversarial attack destroys local structure of image and lead to misclassification (Figure 1).

Currently there are two category of defenses: refining model by enforcing invariance and smoothness, or removing adversarial attack from image. Some adversarial attack can be eliminated by transforming input image, like color-depth-reduction, spatial-smoothing and total-variance-minimization(TVM). However, the transformations could also create new features or destroy necessary features. Conservative processing couldn't protect image, while aggressive processing changes image to other classes. We experimented on different variants of methods and hyperparameters to find the best way to preventing attack.

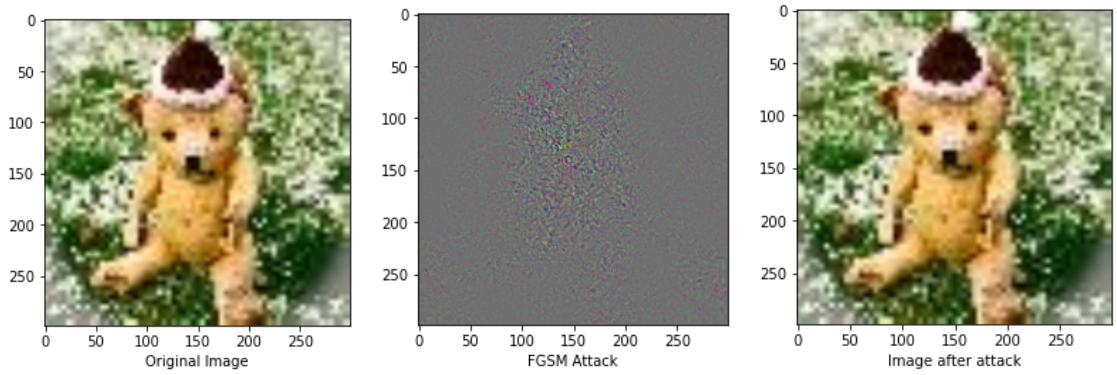


Figure 1: An example of adversarial attack. Leftmost: original image. Middle: attack L2 = 0.02. Rightmost: misclassified image

2. Related work

In [2], color-depth-reduction and spatial-smoothing was initially experimented on self-trained model trained by MNIST and CIFAR-10. Both methods achieved high accurate defending FGSM attack. It is further testified can withstand C&W2 attack in [4]. Both papers used 28x28 or 32x32 images. However, it doesn't use perturbation L2 as attack strength metric. This project applied those methods withstand adversarial attacks from weak to strong.

In [3], TVM was proposed and tested with attacks FGSM, Deepfool and CW-L2. It shows TVM outperformed simple compression like JPEG compression. And unlike the first two methods, it's randomness and non-differentiable makes adversarial attack harder to exploit it. In this project, we tune it by different hyperparameters and compare it with a global-variance minimization method, K-means.

Beside these, there are also other research making using of those method to test model robustness. [5] has shown that stronger adversarial defense compromises clean accuracy, which is also proven in [7], [8]. There are some work use these input transformation method as benchmark to compare robustness of models [6][9].

3. Method

3.1 Adversarial Attack Methods

Fast Gradient Sign Method(FGSM) [1]: Denote l() be loss function, x be input, and h() be training function. ∈ is the limit of change for each pixel.

$$\mathbf{x}' = \mathbf{x} + \epsilon \cdot \operatorname{sign}\left(\nabla_{\mathbf{x}} \ell(\mathbf{x}, h(\mathbf{x}))\right),$$

Iterative-FGSM [11]: A variant of FGSM. It applies FGSM multiple times with smaller ∈ in each iteration. It results to stronger attack. We choose ∈=0.2.

$$\mathbf{x}^{(m)} = \mathbf{x}^{(m-1)} + \epsilon \cdot \operatorname{sign}\left(\nabla_{\mathbf{x}^{(m-1)}} \ell(\mathbf{x}^{(m-1)}, h(\mathbf{x}))\right),$$

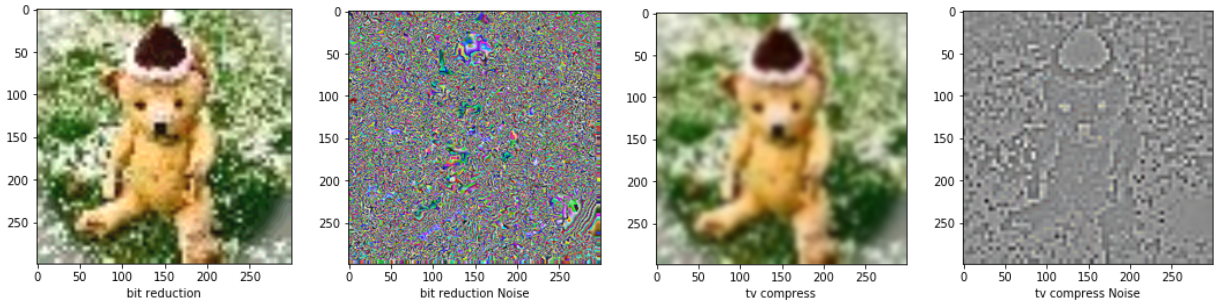
Deepfool[12]: Deepfool is optimized for L_2 distance metric. It tries to find the nearest hyperplane that separating the original class and any other class. It generates stronger attack than FGSM and I-FGSM.

$$\boldsymbol{r}_*(\boldsymbol{x}_0) = \frac{\left|f_{\hat{l}(\boldsymbol{x}_0)}(\boldsymbol{x}_0) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_0)\right|}{\|\boldsymbol{w}_{\hat{l}(\boldsymbol{x}_0)} - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}\|_2^2}(\boldsymbol{w}_{\hat{l}(\boldsymbol{x}_0)} - \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}).$$

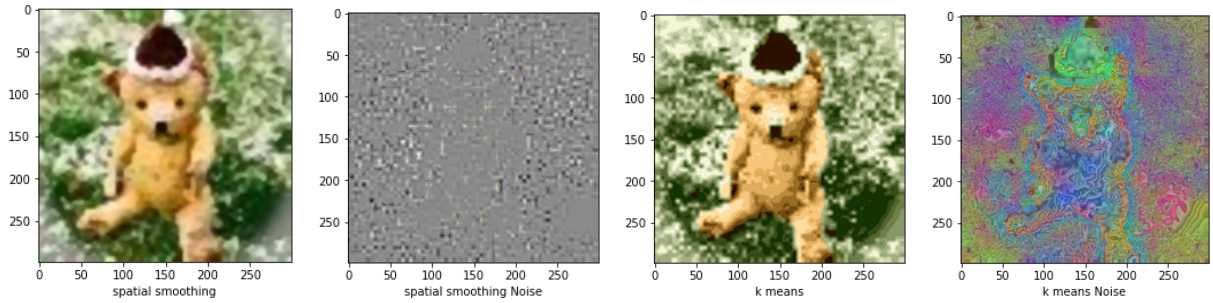
3.2 Defense Methods

Color-depth-reduction: Images could contain unnecessary features [4] that could be exploited by Adversarial attacks.Using less bit to discrete colors will make prediction more robustic [4]. Bit-to-reduce is a hyperparameter manually set. The more bit to reduce, the more features are eliminated.

X[i, j] = X[i, j] - (X[i,j] % power(2, bit-to-reduce))



Color-depth-reduction | Color-depth-reduction delta | TVM compress | TVM compress delta



Spatial- smoothing | spatial smoothing delta | k means | k means delta

Figure 2. Left: processed image after defense, right: their delta compared to original image i.e. delta = Defense Processed Image - Attack Image. There is no Teddy bear's shape in color-depth-reduction delta, so it only removes features, and doesn't recover features. We can see Teddy bear's shape in other methods' delta, so those methods somehow recovers original features.

Spatial smoothing: [4] also proposed a method to smooth local variance. It runs a sliding window on each pixel to substitute color with median value of all values in sliding window. This median selection can effectively remove sparsely-occuring black and white pixels, whilst preserve edges of objects.

Total-variance-minimization(TVM): Similar to spatial smoothing, TVM[5] removes perturbation by compressing images using adjacency pixels but with randomness. First, it selects some pixels randomly sampled by Bernoulli random variable. Then it uses formula below to minimize local variance.

$$\min_{\mathbf{z}} \|(1-X)\odot(\mathbf{z}-\mathbf{x})\|_2 + \lambda_{\mathrm{TV}} \cdot \mathrm{TV}_p(\mathbf{z}).$$

The first term is the delta(cost) of processing, the second term is local variance remained.

$$\mathrm{TV}_p(\mathbf{z}) = \sum_{k=1}^{K}\left[\sum_{i=2}^{N}\|\mathbf{z}(i,:,k)-\mathbf{z}(i-1,:,k)\|_p + \sum_{j=2}^{N}\|\mathbf{z}(:,j,k)-\mathbf{z}(:,j-1,k)\|_p\right].$$

4. Experiments

In this project, we build up an unified framework to evaluate robustness of different defense methods. We use perturbation l2 distance from original image as attack's strength. We used InceptionV3 model to generate attacks and test classification accuracy. We used images whose link are provided by ImageNet, by with higher resolutions.

The first experiment focuses on color-depth-reduction effect with different bit reduction: 2-bits, 4-bits and 6-bits. The second experiment focuses on spatial-smoothing with different shape of sliding window. The third experiment compares TVM with K-means.

4.1 Color-depth-reduction

For FGSM & I-FGSM attack, when l2 < 0.04, 2-bit reduction outperforms 4-bit reduction and 6-bit reduction by reduced less features. But as attacks get stronger, it deteriorate rapidly. Meanwhile, 4-bit and 6-bit can withstand stronger FGSM attacks. 4-bit reduction works the best since 6-bit reduction reduces too many features so that the image's perceivable is damaged.

For Deepfool attack, 6-bit reduction is similar to its performance in defending FGSM. But 4-bit reduction deteriorate as attack becomes stronger. This means Deepfool attack exploits more features than FGSM, so 4-bit reduction can't remove attacks when attack get even stronger.

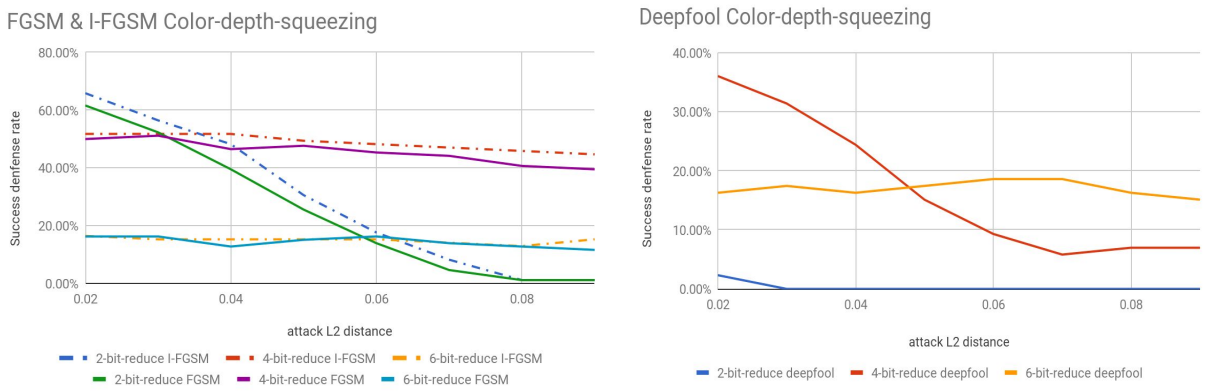


Figure 3. Defense success rate of color-depth-reduction

4.2 Spatial-smoothing

FGSM:

For FGSM and I-FGSM, we found 1x3 and 3x1 sliding window perform better when L2 distance < 0.06. Because 3x3 sliding window covers more adjacent pixels and makes image blur, it could eliminate more features, which makes classification accuracy worse.

When attack > 0.06, 1x3 and 3x1 work worse than 3x3. The result make sense because 1x3 and 3x1 can only recover features from one direction, while 3x3 can recover from a square. When stronger perturbation is added on image, features in either vertical or horizontal direction could be attacked. So 3x3 sliding window has potentially more resistance to attacks from only one direction attack. While 1x3 and 3x1 can't recover image from another direction.

Deepfool:

3x3 sliding window performs better than other methods all the time. This is because Deepfool poses stronger attack than FGSM. So neither 1x3 or 3x1 can't recover from the adversarial attack. 5x5 sliding window is also stable. But it makes picture too obscure to classify. 3x3 doesn't change image perceivable too much so it works.

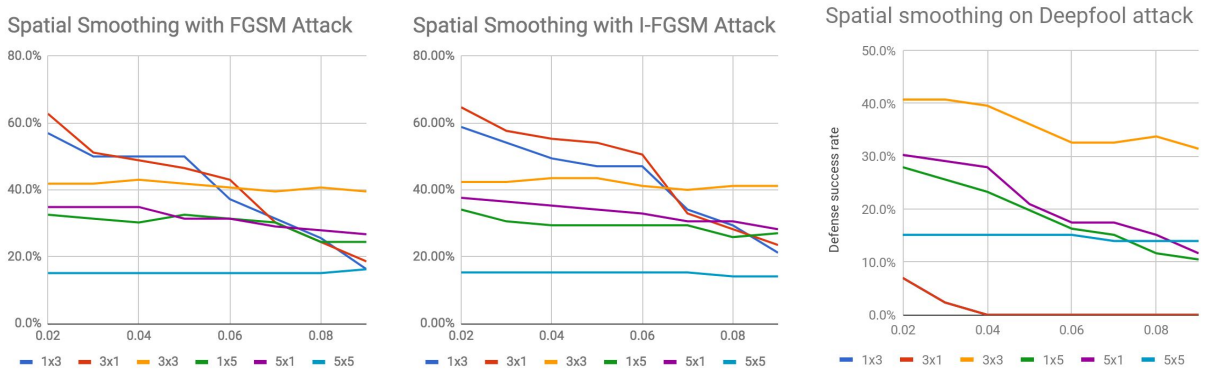


Figure 4. Spatial-smoothing defense success rate with different sliding window size

4.3 TVM compare to K-means

In figure 5, we found that TVM performs better than K-means since K-means tries to minimize global variance, some features can be destructed after processing. While TVM can achieve best perform of all methods as it removes local perturbations. Also TVM requires loss computation.

Besides, in figure 6, we also compared performance on how many pixel we process. We choose to randomly select 10%, 25% and 100% of all pixels in the objective minimization function. Counterintuitively, 10% performs the best. It's probably because processing too aggressively doesn't help recovering features, but only destroy existing ones.

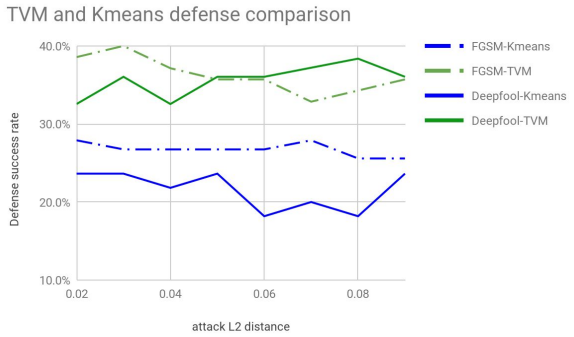


Figure 5. TVM vs K-means

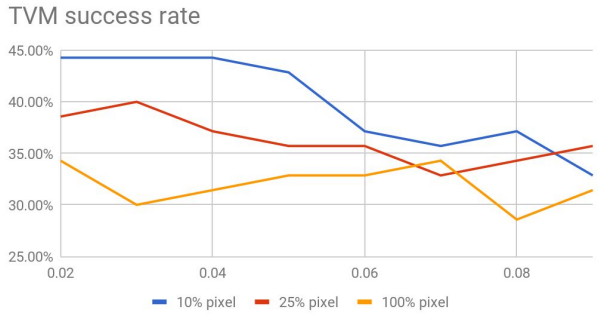


Figure 6. TVM pixel number vs Success rate

5. Conclusion

The result suggests that there are trade off on input transformation methods: too aggressive methods removes perturbation, but it deteriorates perceivable of image and makes legimitive classification accuracy lower; too conservative methods can effectively remove light attacks, but it can't recover from or remove stronger attacks. Success rate differs in strength and attack methods. The best method working in one scenario may not work in another scenario. Although these methods are model-agnostic, we still need to carefully pick hyperparameters to achieve the best performance. Because different model have different capability to resist attack [5], those methods need to be carefully hand tuned when test robustness on models.

The result in comparison of TVM and K-means also testified that adversarial attacks comes from local structure. Furthermore, the best effect come from randomly selecting 10% pixel make the defense method very hard to attack. It is harder to find which 10% pixels are selected, comparing to just countering other deterministic methods.

6. Further Steps

Currently input transformation is only focused on image processing. There are still open areas such as speech recognition, text understanding. The approaches in those area could be similar. We hope to investigate on them as well.

Defending adversarial attacks is an emerging area. While a lot of new studies about image transformation comes along, there are also handful research studying model properties, as well as statistical features of adversarial attacks. We believe there could be a way to combine different approaches together to find a better way.

Contribution

Mark Liu: Find direction from paper, write up code and report.
Li Cai: set up cloud and gather data, code modifications. Write poster/report.

Reference

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples.

[2] Weilin Xu, David Evans, Yanjun Qi, Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks.

[3] Chuan Guo, Mayank Rana, Moustapha Cisse, Laurens van der Maaten, Countering Adversarial Images using Input Transformations.

[4] Weilin Xu, David Evans, Yanjun Qi. Feature Squeezing Mitigates and Detects Carlini/Wagner Adversarial Examples.

[5] Ekin D. Cubuk, Barret Zoph, Samuel S. Schoenholz, Quoc V. Le. Intriguing Properties of Adversarial Examples.

[6] Andreas Veit, Serge Belongie, Convolutional Networks with Adaptive Computation Graphs

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks.

[8] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks.

[9] Valentina Zantedeschi, Maria-Irina Nicolae, Ambrish Rawat. Efficient Defenses Against Adversarial Attacks

[10] Tamir Hazan, George Papandreou, Daniel Tralow. Perturbation, Optimization and Statistics.

[11] Florian Tramèr, Alexey Kurakin, Nicolas Papernot , Dan Boneh, Patrick McDaniel. Ensemble Adversarial Training: Attacks and Defenses

[12] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard. DeepFool: a simple and accurate method to fool deep neural networks

[13] O.M.Parkhi, A. Vedaldi, A. Zisserman. Deep Face Recognition