Image Processing Defense on Adversarial Attack

Mark Liu, Li Cai

Objective

The state-of-art machine learning models, especially Neural Networks, are vulnerable to adversarial attacks: a small perturbation can make image completely misclassified. In this project, we experimented on different inexpensive image processing defenses against different adversarial attacks. We found that different attacks generate different pattern of perturbation. The hyperparameters of image processing effect the defense robustness: aggressive processing makes model very astute to small perturbation, but perform poorly when perturbation is larger; conservative processing makes more robust to all perturbations.

Data and Model

Data: We used images with pixel 64x64 from ImageNet.
Model: pretrained Inception V3 model to train adversarial attacks.

Attack

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

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

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

Image Processing Defense Methods

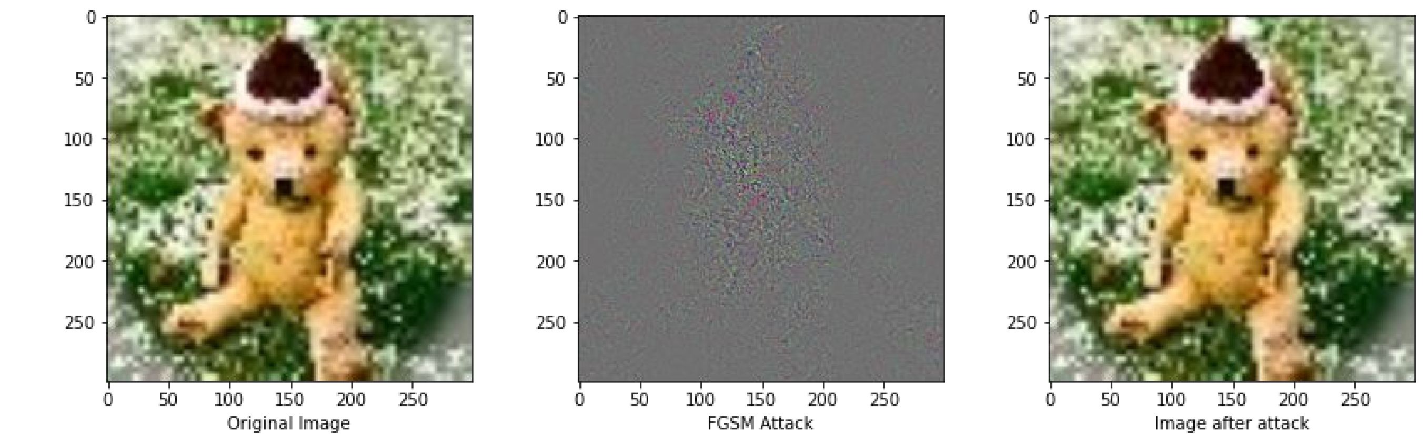
1. **Depth-color-squeezing**
   Reduce number of bit for each pixel's color.
   $x(i, j) = x(i, j) \% 2^x$
   where x is number of bit to reduce.
   Hyperparameter: We tried on bit-depth 2,4, and 6 bits.
2. **Image compression with K-means**
   Substitute each pixel with nearest centroid value.
   $x(i, j) = centroid_k$ *where* $k = argmin_k [x(i, j) - centroid_k]$
   Hyperparameter: #centroid
3. **Spatial smoothing**
   Substitute each pixel to the median in its sliding window.
   x(i, j) = median(sliding window from x(i-m, j-n)->x(i+m, j+n))
   Hyperparameter: sliding window size m, n
4. **Total variance minimization**
   Minimizes sum of local variance on adjacent pixels $TV_p(z)$, plus to L1 regularization. Hyperparameter: λ_{TV}, $p_{bernoulli}$

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

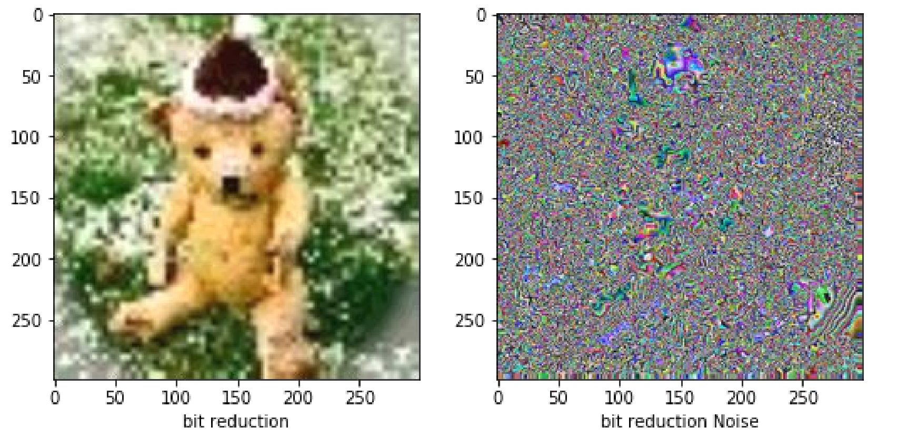
$$\mathrm{TV}_p(\mathbf{z}) = \sum_{k=1}^{K} \left[\sum_{i=2}^{N} \|\mathbf{z}(i,:,k) - \mathbf{z}(i-1,:,k)\|_p + \sum_{j=2}^{N} \|\mathbf{z}(:,j,k) - \mathbf{z}(:,j-1,k)\|_p \right].$$

Processed Images

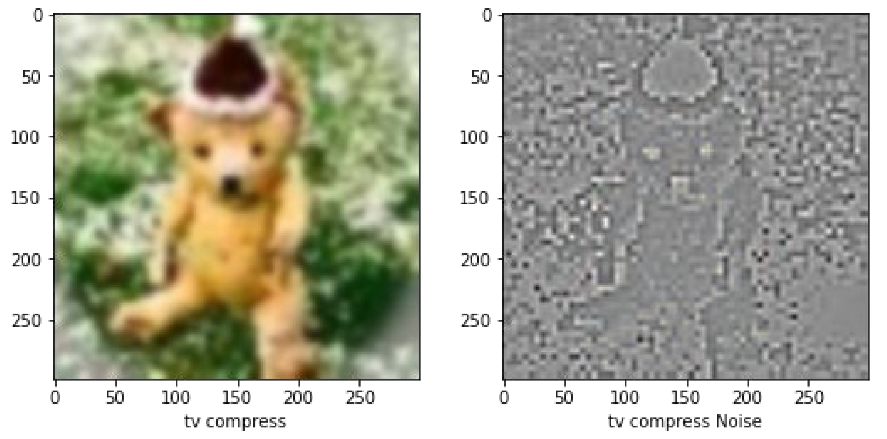
1. Processed image after we added noise



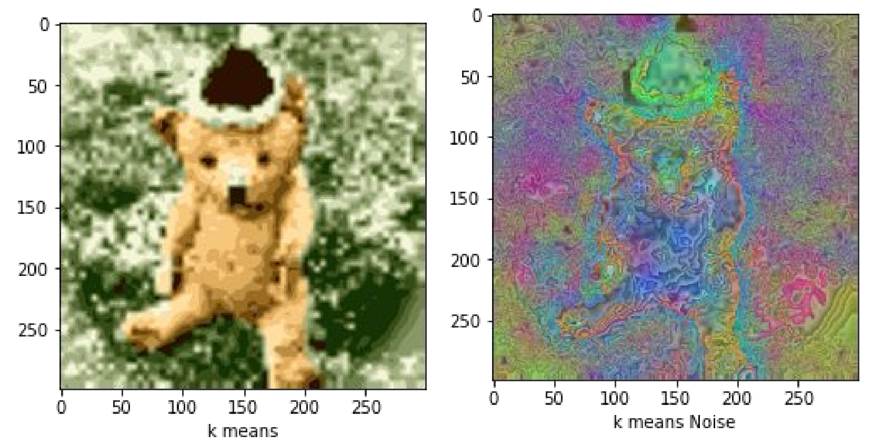
2. The left shows final image after we apply defense methods. The right shows its difference from the adversarial attack image.



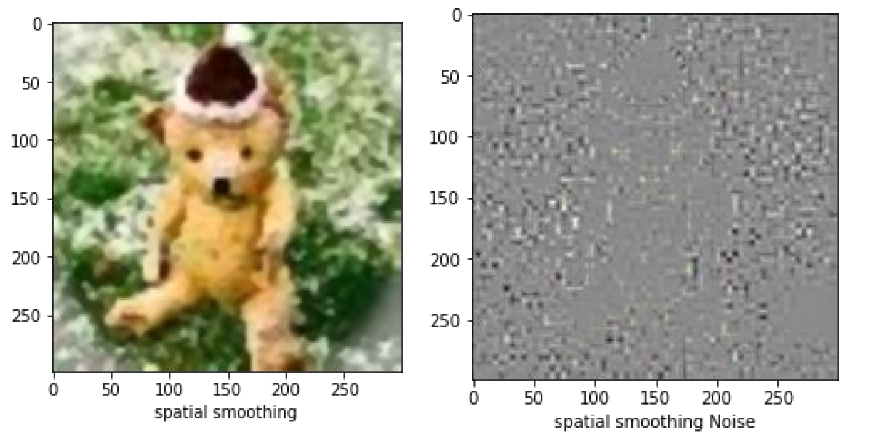
Depth-color squeezing



Total variance minimization



K-means



Spatial smoothing

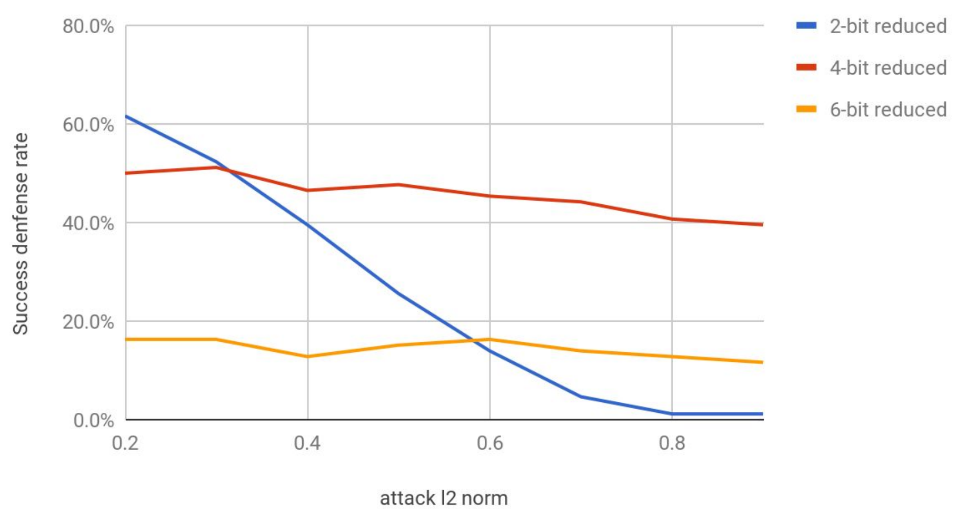
Discussion

From the result graphs, we have the following conclusions

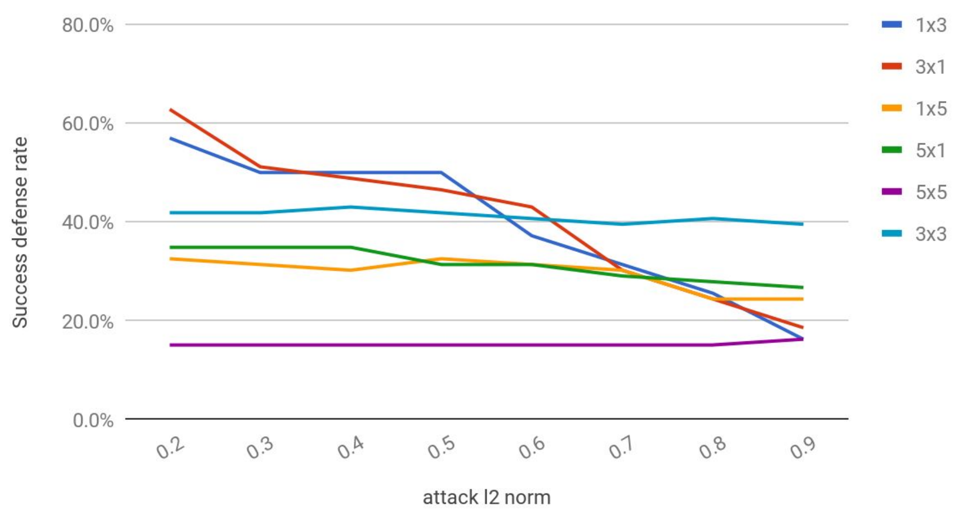
- Deepfool poses stronger attack than FGSM wrt attack at same L2 norm.
- Depth-color-squeezing: The more compressed image, the more robust to attack.
- For spatial smoothing, we found using 3 pixels as the sliding window is better than 5 because it preserves more local features. Using square(3*3) is better using stride because we remove more attack features which are more likely be in square shapes.
- Total variance minimization outperforms K-means, due to it minimizes local variance, while K-means minimize global variance.

Result

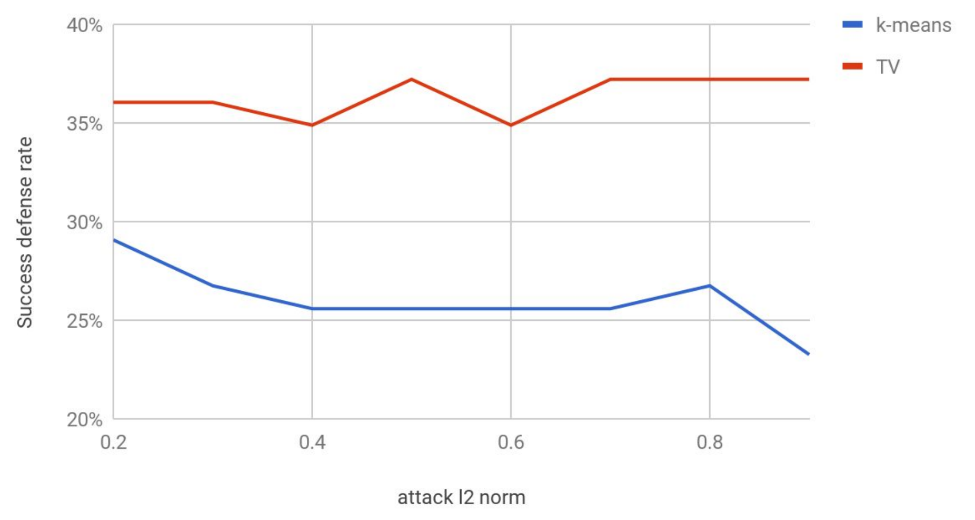
Depth-color-squeezing on FGSM attack



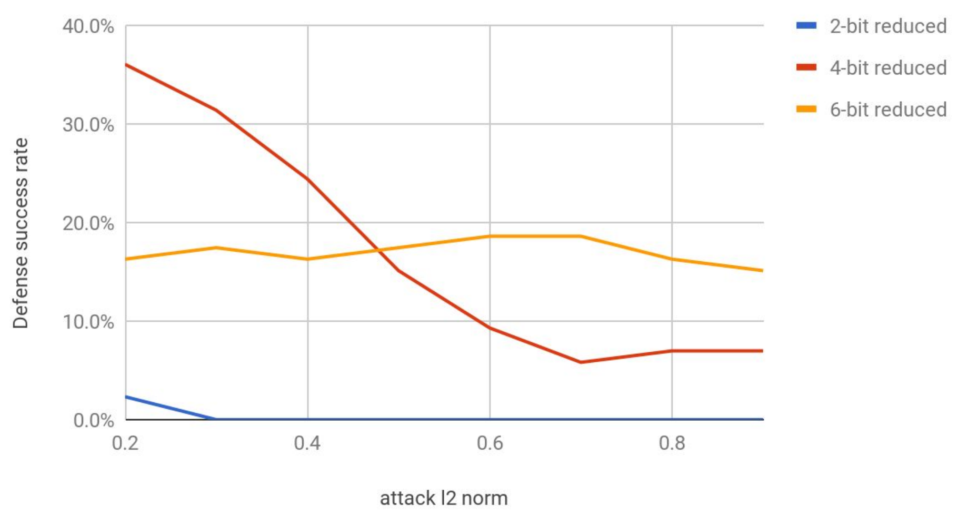
Spatial smoothing on FGSM attack



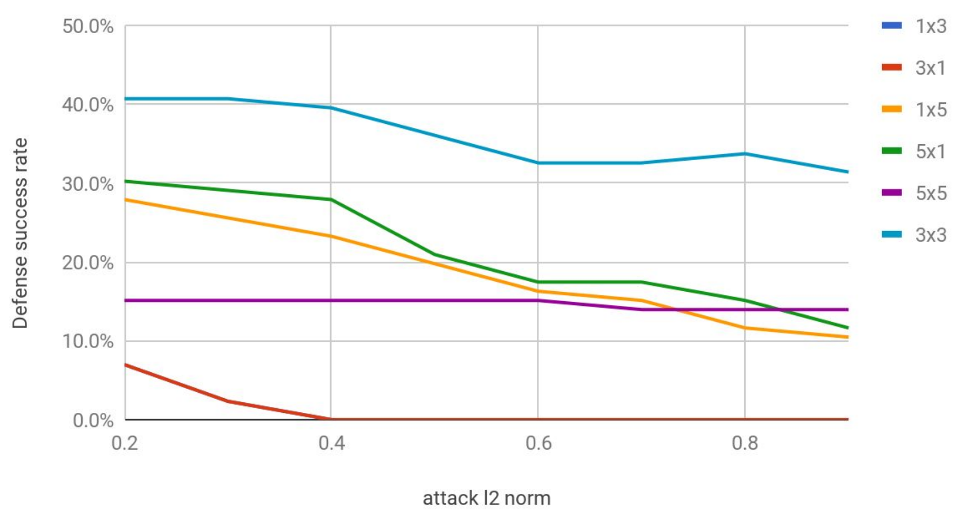
k-means and total variance on FGSM attack



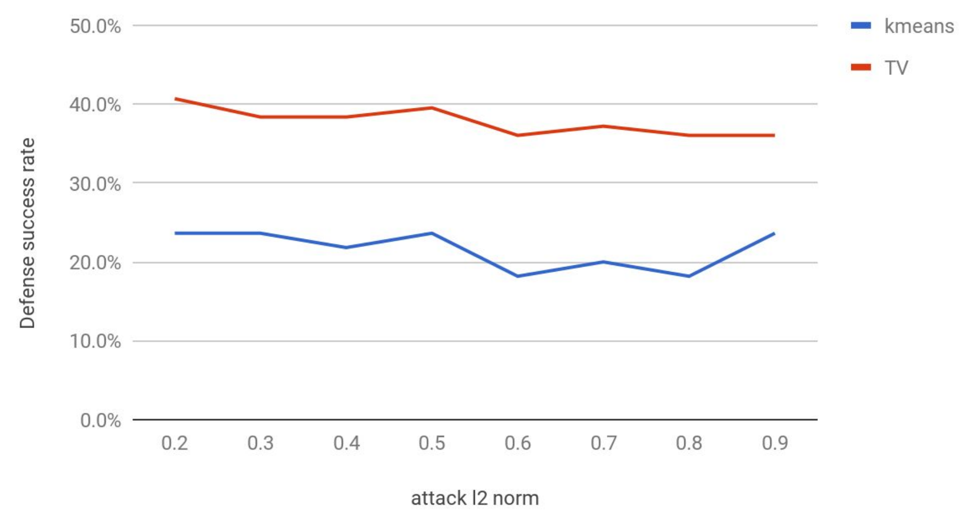
Depth-color-squeezing on Deepfool attack



Spatial smoothing on Deepfool attack



TV and kmeans on Deepfool attack



Stanford
University