Feature Denoising for Improving Adversarial Robustness

Saeed Rahaman & Quoc Le
Outline

- Problem Description
- Paper Solution
- Experiments & Results
- Conclusion
- Pros & Cons
- Q&A
What is the Problem?

- Perturbations add noise to an image which lead to noise in the feature map.
- Changes the semantically important regions.
Proposed Solution

1. Denoise the feature map by adding denoising blocks in between layers

1. Train networks on adversarial examples to reduce feature-map noise
Denoising Block

- Input is feature maps from a previous convolutional layer
- Denoising operation is performed to remove noise
- 1x1 convolution to normalize the denoised feature map
- The denoised feature map is combined with the original feature map
Denoising Block Design

- Removing 1x1 convolution reduces accuracy significantly

- Denoising by itself is not impactful

- Removing the residual connection makes the network unstable

- Combining the input features with denoised features is imperative for increased accuracy
Effect of Denoising Blocks

- After adding 4 denoising blocks to ResNet-152
- Noise in feature map significantly reduced
Denoising Operations - Non-local Means

\[ y_i = \frac{1}{C(x)} \sum_{j \in L} f(x_i, x_j) \cdot x_j \]

- $C(x)$ is a normalization function
- $f(x_i, x_j)$ is a feature weighting function
- $i$ and $j$ refers to spatial location on the feature map
- $L$ is all spatial locations in the image
- $y_i$ is the feature map output for a specific location
Gaussian Implementation

- The weight function is

\[ f(x_i, x_j) = e^{\frac{1}{\sqrt{d}} \theta(x_i)^T \phi(x_j)} \]

- The normalization function is

\[ C = \sum_{j \in \mathcal{L}} f(x_i, x_j) \]

- \( \Theta(x) \) and \( \Phi(x) \) are embedded versions of \( x \) obtained through 1x1 convolutions
  - Where \( \Theta(x_i) = W_\Theta x_i \) and \( \Phi(x_j) = W_\Phi x_j \)

- \( d \) = number of channels in the feature map

- \( f/C \) is similar to the softmax functions
Dot Product Implementation

- The weight function is

\[ f(x_i, x_j) = x_i^T x_j \]

- Normalization function is \( C(x) = N \)
  - \( N \) is the number of pixels in \( x \)

- Requires less parameters than the Gaussian Version
Denoising Operations - Bilateral Filter

\[ y_i = \frac{1}{C(x)} \sum_{\forall j \in \Omega(i)} f(x_i, x_j) \cdot x_j \]

- Is same as Non-Local Mean except looks at the “Local Regions” \( \Omega(i) \)
- Dot product and Gaussian implementation can be used here as well
Denoising Operations - Mean Filter

- Use average pooling with stride of 1
- Simplest form of denoising
- Reduces the noise image but will smoothes the structures of the image
Denoising Operations - Median Filter

\[ y_i = \text{median}\{\forall j \in \Omega(i) : x_j\}, \]

- Perform on the local region \( \Omega(i) \)
- Perform on each channel separately
- Good at removing salt-and-pepper noise and other similar type of outliers
Adversarial Training

- Generated adversarial examples with PGD
- For each mini-batch of images
  - PGD attack is performed
  - SGD is performed on perturbated images and weights are updated
  - Drastically increases runtime (30 iterations per image)
- ResNet-101 and ResNet-152 trained as baseline models
- 4 denoising blocks are added to ResNet-152 and trained
  - Used non-local means with Gaussian
Experiments

- PGD used on ImageNet to attack models
  - Iterations range from 10-2000
Denoising Blocks in Non-Adversarial Networks

- Denoising blocks do not affect accuracy against non-adversarial images
- Adversarial training effects accuracy on non-adversarial images

<table>
<thead>
<tr>
<th>Model (testing on clean images)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-152 (baseline) (no A.T.)</td>
<td>78.91%</td>
</tr>
<tr>
<td>ResNet-152 (denoised) (no A.T.)</td>
<td>79.08%</td>
</tr>
<tr>
<td>ResNet-152 (baseline) (A.T.)</td>
<td>62.32%</td>
</tr>
<tr>
<td>ResNet-152 (denoised) (A.T.)</td>
<td>65.30%</td>
</tr>
</tbody>
</table>
Denoising Operations in Non-Adversarial Networks

- Different denoising operations have similar performance on clean images

<table>
<thead>
<tr>
<th>model</th>
<th>accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-152 baseline</td>
<td>78.91</td>
</tr>
<tr>
<td>R-152 baseline, run 2</td>
<td>+0.05</td>
</tr>
<tr>
<td>R-152 baseline, run 3</td>
<td>-0.04</td>
</tr>
<tr>
<td>+4 bottleneck (R-164)</td>
<td>+0.13</td>
</tr>
<tr>
<td>+4 denoise: null (1×1 only)</td>
<td>+0.15</td>
</tr>
<tr>
<td>+4 denoise: 3×3 mean filter</td>
<td>+0.01</td>
</tr>
<tr>
<td>+4 denoise: 3×3 median filter</td>
<td>-0.12</td>
</tr>
<tr>
<td>+4 denoise: bilateral, Gaussian</td>
<td>+0.15</td>
</tr>
<tr>
<td>+4 denoise: non-local, Gaussian</td>
<td>+0.17</td>
</tr>
</tbody>
</table>
Comparison of Denoising Operators

- Compare Denoising block against the adversarial trained ResNet-152, Null Denoising Block, and Bottleneck Block
  - Null denoising block only had the 1 x 1 normalization block
  - Bottleneck block refers bottleneck block designed by He et al. [1]

- Used 4 blocks were added to ResNet-152

Result - Comparison of Denoising Operators
Black-Box Attack Result

- Evaluation criteria was to poll all of black-box attack they used to determine if a misclassification occurred
- Used the 5 best attackers of the IPS 2017 CAAD competitions

<table>
<thead>
<tr>
<th>model</th>
<th>accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAAD 2017 winner</td>
<td>0.04</td>
</tr>
<tr>
<td>CAAD 2017 winner, under 3 attackers</td>
<td>13.4</td>
</tr>
<tr>
<td>ours, R-152 baseline</td>
<td>43.1</td>
</tr>
<tr>
<td>+4 denoise: null (1 x 1 only)</td>
<td>44.1</td>
</tr>
<tr>
<td>+4 denoise: non-local, dot product</td>
<td>46.2</td>
</tr>
<tr>
<td>+4 denoise: non-local, Gaussian</td>
<td>46.4</td>
</tr>
<tr>
<td>+all denoise: non-local, Gaussian</td>
<td>49.5</td>
</tr>
</tbody>
</table>
Conclusion

- Showed that denoising the feature map improves the robustness of the model
- Proposed a new architecture style that includes a denoising block to improve robustness of the model
- Their proposed model had a significant improvement on accuracy against adversarially attacks
Strengths

- Shows a significant improvement against adversarial images
- Implements a new architecture style to improve the robustness of networks
- Robust against a 2000-iteration PGD attack
- Denoising blocks do not reduce performance against clean images
Weaknesses

- Did not have metric on the loss of feature detail on the feature map
- Does not explore why denoising blocks are effective
- Does not test against other white-box attacks
- Only looked at one architecture
Questions?