Error Diffusion Halftoning Against Adversarial Examples

ICIP 2021

Shao-Yuan Lo and Vishal M. Patel
Johns Hopkins University
Recall: Adversarial Examples

\[ x_{\text{adv}} = x + \delta \]

\[ f(x_{\text{adv}}) \neq y \]
Recall: Adversarial Examples

• Deep networks are **vulnerable** to adversarial examples.

Goodfellow et al. Explaining and Harnessing Adversarial Examples. ICLR’15.
Recall: Adversarial Examples

- White-box attack
- Black-box attack
- Gray-box attack
Defense Methods

• **Adversarial training**: Enhance the robustness of networks itself.

\[
\theta^* = \arg \min_{\theta} \mathbb{E}_{(x,y) \sim D} \left[ \max_{\delta \in \mathcal{S}} L(x + \delta, y; \theta) \right]
\]

• **Image transformation**: Remove perturbations from input images.

\[
C(x_{adv}) \neq y.
\]

\[
C(T(x_{adv})) = y.
\]

Madry et al. Towards deep learning models resistant to adversarial attacks. ICLR’18.
Image Transformation-based Defenses

• JPEG compression
• Bit-depth reduction
• Image denoising
  • Gaussian blur
  • Mean/median filter
  • Non-local means
• ...etc

Raff et al. Barrage of random transforms for adversarially robust defense. CVPR’19.
Image Transformation-based Defenses

• Most existing image transformation-based defenses are **NOT** robust against **white-box attacks**.

| Defense                      | Dataset    | Distance     | Accuracy |
|------------------------------|------------|--------------|----------|
| Buckman et al. (2018)        | CIFAR      | $0.031 (\ell_\infty)$ | 0%*      |
| Ma et al. (2018)             | CIFAR      | $0.031 (\ell_\infty)$ | 5%       |
| Guo et al. (2018)            | ImageNet   | $0.005 (\ell_2)$  | 0%*      |
| Dhillon et al. (2018)        | CIFAR      | $0.031 (\ell_\infty)$ | 0%       |
| Xie et al. (2018)            | ImageNet   | $0.031 (\ell_\infty)$ | 0%*      |
| Song et al. (2018)           | CIFAR      | $0.031 (\ell_\infty)$ | 9%*      |
| Samangouei et al. (2018)     | MNIST      | $0.005 (\ell_2)$  | 55%**    |

Athalye et al. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. ICML’18.
Proposed Method: Error Diffusion Halftoning

• Quantize each pixel in the raster order one-by-one, and spread the quantization error to the neighboring pixels.

\[
\hat{I}(i, j) = I(i, j) + \sum_{m, n \in S} h(m, n)e(i - m, j - n)
\]

\[
Q(i, j) = u(\hat{I}(i, j) - \theta) \quad e(i, j) = \hat{I}(i, j) - Q(i, j)
\]

Floyd and Steinberg. An adaptive algorithm for spatial grey scale. Proceedings of the Society of Information Display, 1976.
Proposed Method: Error Diffusion Halftoning

- The **quantization operation** invalid the adversarial variations.

- **Updating the values of the neighboring pixels repeatedly** makes the adaptive attacks hard to identify the mapping between the original image and the corresponding halftone.

- **Spreading quantization errors produces** better halftoning quality and tends to enhance edges and object boundary in an image.

- Take **both** adversarial robustness and clean data performance.

- Complementary to adversarial training.
Experimental Results

- Dataset: CIFAR-10
- Attacks (white-box): PGD [Madry et al.] and Mult [Lo and Patel]

| Method                     | Training                  | Clean | PGD-$\ell_\infty$ | PGD-$\ell_2$ | Mult-$\ell_\infty$ | Mult-$\ell_2$ | Avg$_{adv}$ | Avg$_{all}$ |
|-----------------------------|----------------------------|-------|-------------------|--------------|-------------------|--------------|-------------|-------------|
| Vanilla                     |                            | 94.03 | 0.01              | 0.20         | 0.05              | 0.01         | 0.07        | 18.86       |
| Gaussian blur               | Standard training          | 90.17 | 0.20              | 1.34         | 0.17              | 0.05         | 0.44        | 18.39       |
| Non-local means             |                            | 88.66 | 0.02              | 0.49         | 0.03              | 0.00         | 0.14        | 17.84       |
| JPEG compression            |                            | 90.06 | 2.97              | 4.82         | 1.81              | 0.22         | 2.46        | 19.98       |
| Bit-depth reduction         |                            | 78.87 | 15.26             | 10.84        | 10.79             | 4.52         | 10.35       | 24.06       |
| Halftoning (ours)           |                            | 88.57 | 9.53              | 11.98        | 5.54              | 1.07         | 7.03        | 23.34       |
| Vanilla                     | Adversarial training       | 83.31 | 51.15             | 50.68        | 54.10             | 40.29        | 49.06       | 55.91       |
| Gaussian blur               |                            | 75.96 | 44.59             | 47.12        | 45.07             | 32.48        | 42.32       | 49.04       |
| Non-local means             |                            | 75.47 | 44.67             | 45.29        | 16.59             | 14.53        | 30.27       | 39.31       |
| JPEG compression            |                            | 24.97 | 38.99             | 43.72        | 59.15             | 44.72        | 46.65       | 42.31       |
| Bit-depth reduction         |                            | 71.66 | 47.34             | 42.40        | 48.50             | 41.63        | 44.97       | 50.31       |
| Halftoning (ours)           |                            | 84.37 | 60.01             | 56.56        | 67.37             | 88.44        | 68.10       | 71.35       |
Feature Visualization

| vanilla blur | Gaussian blur | Non-local means | JPEG | Bit-depth reduction | Halftone |
|-------------|---------------|-----------------|------|---------------------|----------|
| Transformed image | Transformed image | Feature at the last conv layer | Feature at the last conv layer | Transformed image | Transformed image | Feature at the last conv layer | Feature at the last conv layer |
Feature Analysis

• Mean square differences between the features of clean images and the features of adversarial examples.
Conclusion

• Propose a new image transformation-based defense method using error diffusion halftoning.

• Remove adversarial perturbations and weaken adaptive attacks.

• Robust against white-box attacks.

• Produce high quality halftones and thus guarantee good clean data performance.