Defense against Universal Adversarial Perturbations (Supplementary material)

Naveed Akhtar*, Jian Liu*, Ajmal Mian
*The authors contributed equally.

1. Synthetic perturbation generation

The algorithm below summarizes the procedure of generating the synthetic image-agnostic perturbations constrained by their ℓ_2-norm. The algorithm is analogous to Algorithm (1) in Section 4.2 of the paper.

Algorithm 1 ℓ_2-norm synthetic perturbation generation

Input: Pre-generated perturbation samples $\mathcal{P} \subseteq \mathbb{R}^d$, number of samples to be generated η, threshold ξ.
Output: Synthetic perturbations $\mathcal{P}_s \subseteq \mathbb{R}^d$

set $\mathcal{P}_s = \{\}$; $\mathcal{P}_n = \mathcal{P}$ with ℓ_2-normalized elements.
while $|\mathcal{P}_s| < \eta$ **do**
  set $\boldsymbol{\rho}_s = \mathbf{0}$
  while $||\boldsymbol{\rho}_s||_2 < \xi$ **do**
    $z \sim \text{unif}(0,1) \odot (\xi - ||\boldsymbol{\rho}_s||_2)$
    $\boldsymbol{\rho}_s = \boldsymbol{\rho}_s + (z \odot \overset{\text{rand}}{\sim} \mathcal{P}_n)$
  end while
  $\mathcal{P}_s = \mathcal{P}_s \bigcup \boldsymbol{\rho}_s$
end while
return

Whereas generating a single universal adversarial perturbation [4] for a network, e.g. GoogLeNet [5] takes several hours using the tensorflow implementations, the average time to generate 230 synthetic image-agnostic perturbations was recorded around 1 minute for the ℓ_2-type perturbations, and nearly 10 minutes for the ℓ_∞-type perturbations. For the latter, the computation takes longer because of the additional constraint over the ℓ_2-norm of the perturbations.

2. DCT for rectifying perturbations

Dziugaite et al. [2] studied JPG compression to mitigate the fooling caused by image-specific perturbations, and suggested analyzing Discrete Cosine Transform (DCT) to reduce the effectiveness of quasi-imperceptible perturbations. We performed experiments to evaluate DCT compression as a potential solution for defending the networks against the universal adversarial perturbations. In Fig. 1, results of a representative experiment are shown. In this experiment, we perturbed all the available $10,000$ images with the 5 test perturbations (ℓ_2-type) for the GoogLeNet and tested the accuracy of the network on these images by rectifying them using the DCT. The plot in the figure clearly suggests that the performance of a simple DCT-based rectification is far from satisfactory as a defense against the universal perturbations. In comparison to the network's performance on the clean images (i.e. 69.30%) the best result using the DCT rectified images is significantly low (i.e. 47.91%). On the other hand, the accuracy of the network on the PRN rectified version of the same $10,000$ perturbed images is 65.97%. For visualization, we also show the rectification of an example perturbed image with DCT and PRN in Fig. 2.

A 'significant' gain of PRN over DCT-based rectification was consistently observed in our experiments with all the targeted networks. Based on our experiments we can safely conclude that whereas a simple DCT-based rectification can mitigate the effects of the universal adversarial perturbations, it is insufficient to use DCT compression alone as the defense against these perturbations - leaving alone the question of how much compression is required for the most effective rectification.

3. Cross-network results for Prot-B

Table 1. ℓ_2**-type cross-network defense** (Prot-B): Testing is done using the perturbations generated on the networks in the left-most column. The networks to generate the training perturbations are indicated in the second row.

PRN-restoration (%)			
	VGG-F	CaffeNet	GoogLeNet
VGG-F [1]	86.2	87.0	70.4
CaffeNet [3]	85.5	89.9	65.7
GoogLeNet [5]	83.5	80.2	92.4
Defense rate (%)			
	VGG-F	CaffeNet	GoogLeNet
VGG-F [1]	91.6	89.8	75.0
CaffeNet [3]	91.4	93.6	72.6
GoogLeNet [5]	87.5	84.0	94.8

The Tables 1 and 2 report the results of our experiments for the cross-network defense using the proposed frame-

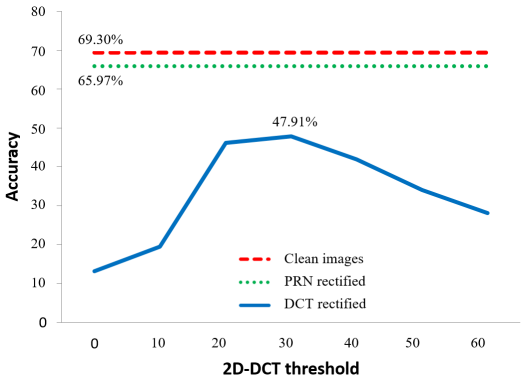

Figure 1. Performance of GoogLeNet on the perturbed images rectified by removing the 2D-DCT components of the images. The x-axis shows the threshold of the magnitude below which the components were removed. The accuracy of the network on the same perturbed images rectified by the proposed PRN (only) is 65.97%. The accuracy of GoogLeNet is 69.30% on the clean version of the same images.

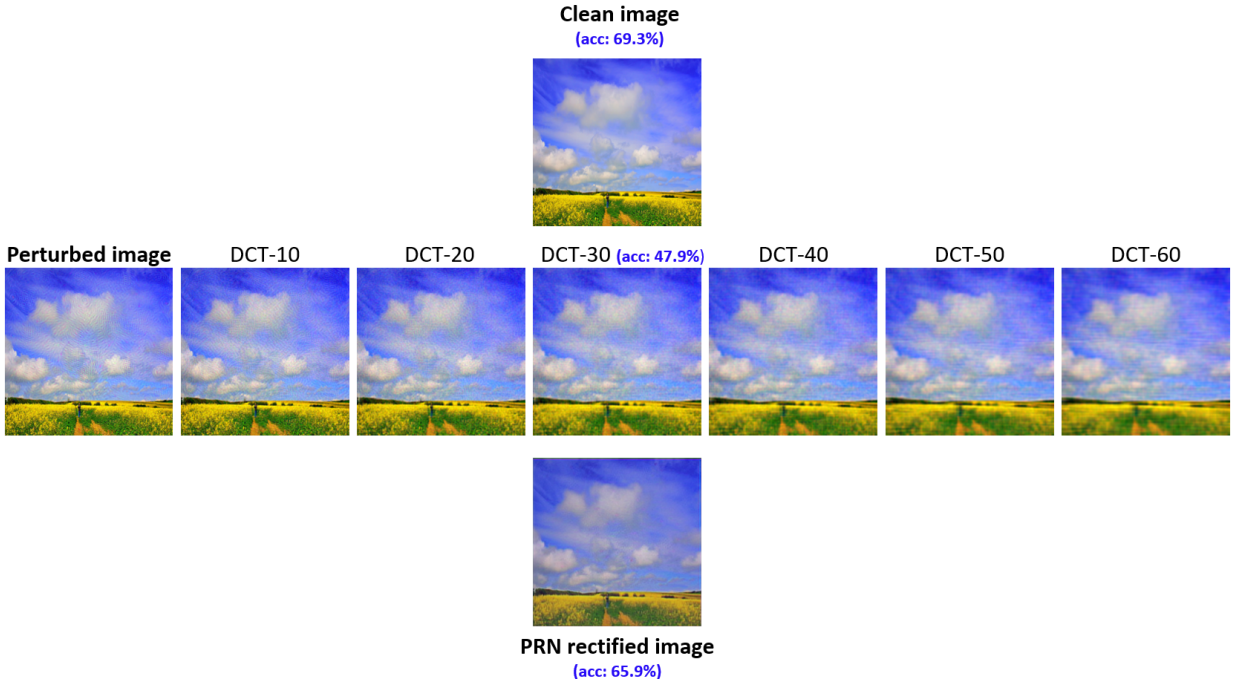

Figure 2. Rectification with 2D-DCT vs PRN: In 'DCT-X', 'X' refers to the threshold magnitude below which the DCT components were ignored in the image reconstruction. Whereas DCT compression is able to reduce the noise patterns in the images, it also results in unnecessary blur which is detrimental to the network performance. The shown accuracies of GoogLeNet are for the complete test dataset.

Table 2. ℓ_∞**-type cross-network defense** summary (Prot-B).

PRN-restoration (%)			
	VGG-F	CaffeNet	GoogLeNet
VGG-F [1]	84.4	84.6	63.5
CaffeNet [3]	83.2	88.7	62.9
GoogLeNet [5]	83.5	77.4	91.3
Defense rate (%)			
	VGG-F	CaffeNet	GoogLeNet
VGG-F [1]	90.1	87.7	68.2
CaffeNet [3]	87.9	92.5	69.2
GoogLeNet [5]	86.8	81.4	93.7

work under the Protocol-B mentioned in Section 5 of the paper. In contrast to Protocol-A, Protocol-B uses the testing images created by perturbing only those images that were necessarily correctly classified by the targeted network in their clean form. For the results in the tables, both 'detector' and 'rectifier' components of our framework were trained using the perturbations generated for the same (targeted) network.

4. Examples of rectified images

Representative examples are provided in figures 3 to 6.

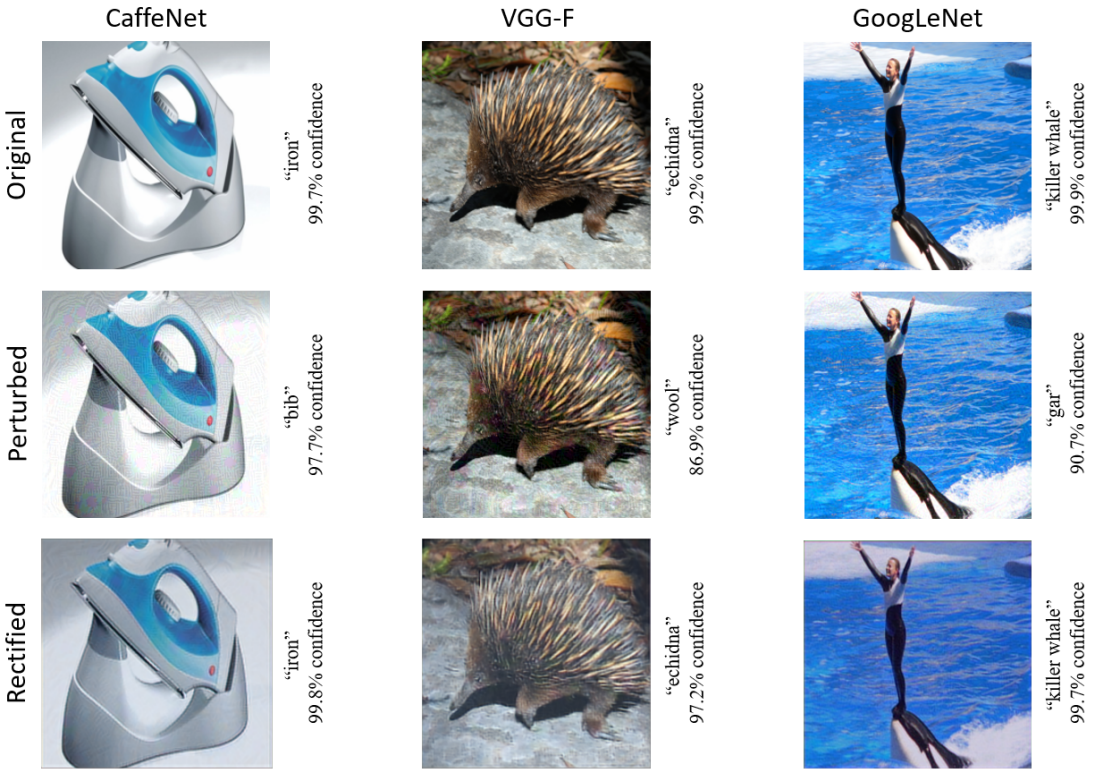

Figure 3. Representative examples of the original, perturbed and rectified images: The images are provided for the ℓ_∞-type perturbations.

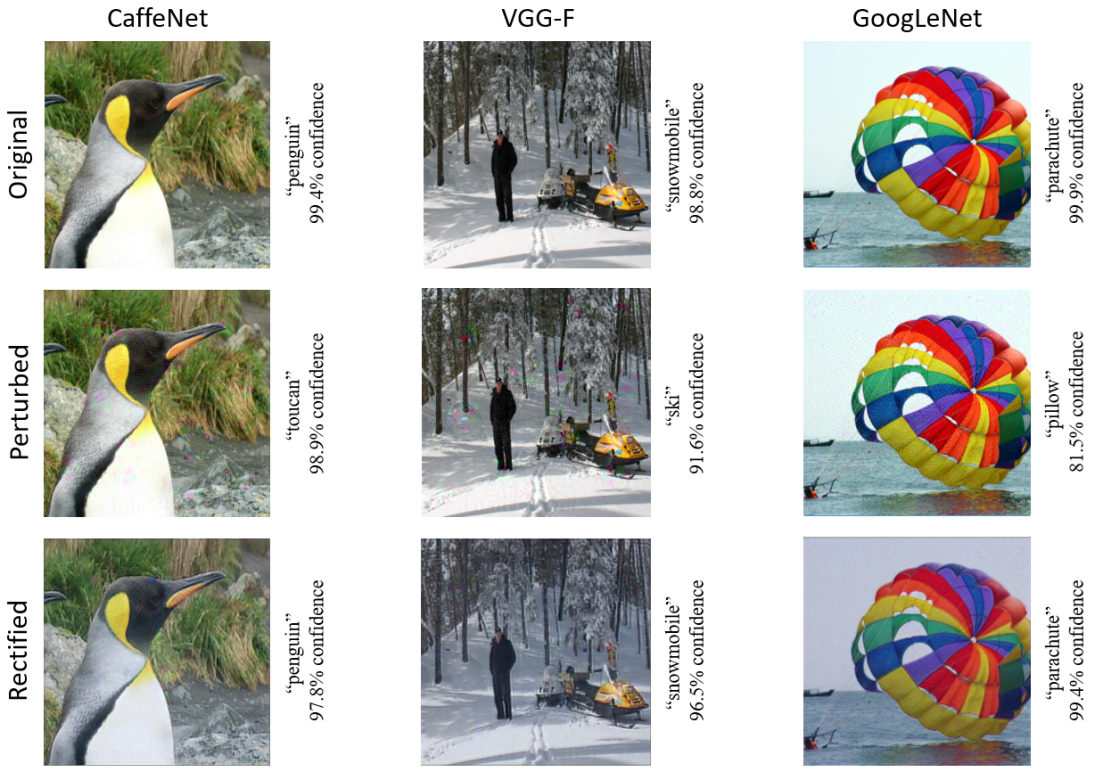

Figure 4. These examples are provided for the ℓ_2-type perturbations.

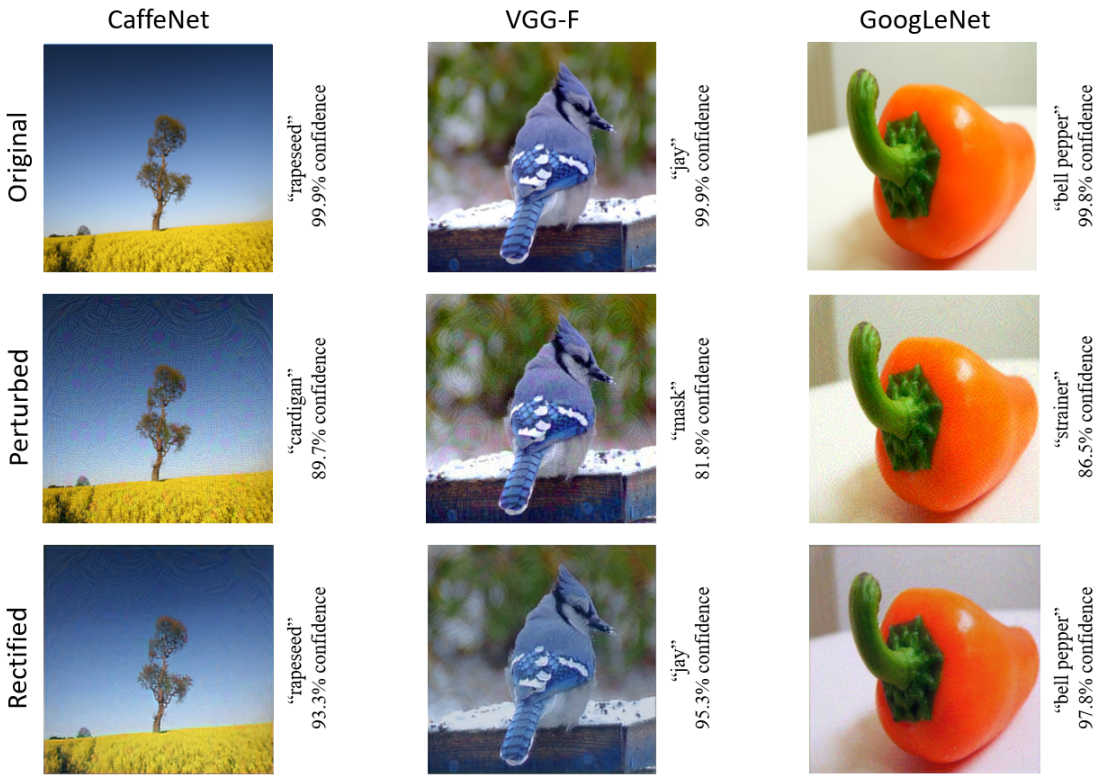

Figure 5. Further examples for the ℓ_∞-type perturbations.

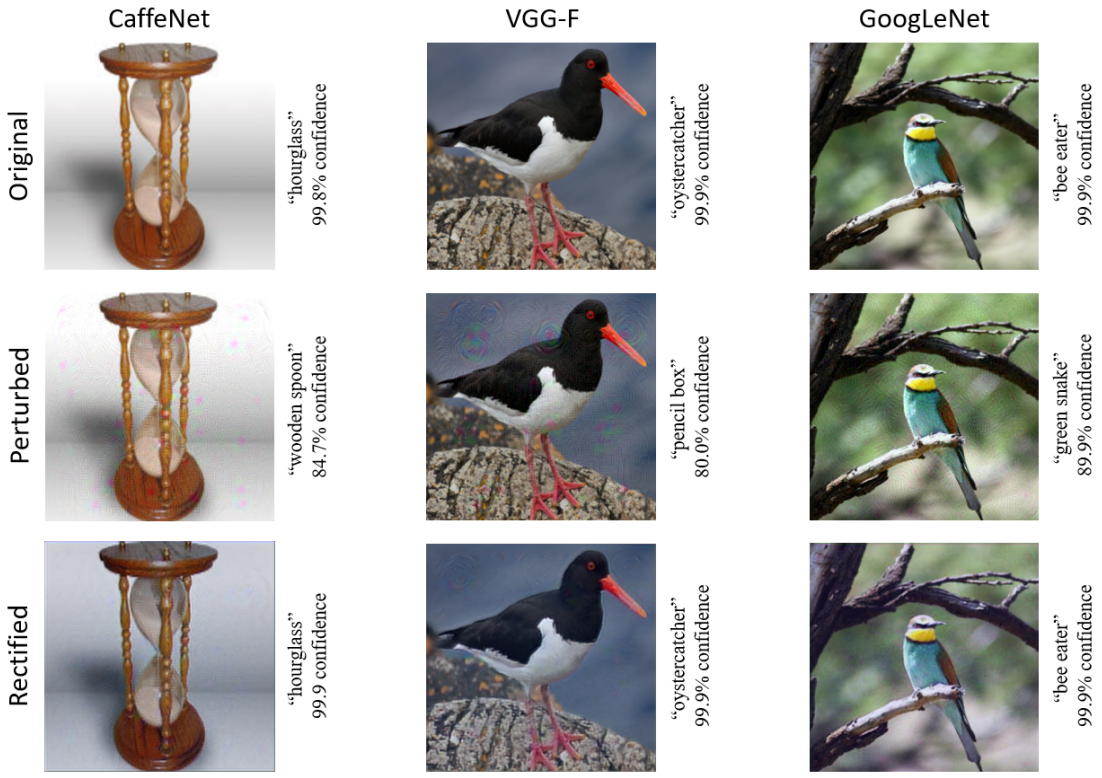

Figure 6. Further examples for the ℓ_2-type perturbations.

References

[1] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[2] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[4] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. *CVPR*, 2017.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.