Defending Against Universal Attacks Through Selective Feature Regeneration (Supplementary Material)

Tejas Borkar[1] Felix Heide[2,3] Lina Karam[1,4]
[1]Arizona State University [2]Princeton University [3]Algolux [4]Lebanese American University
{tsborkar,karam}@asu.edu fheide@princeton.edu

A. Maximum Adversarial Perturbation

We show in Section 4.1 of the main paper that the maximum possible adversarial perturbation in a convolutional filter activation map is proportional to the ℓ_1-norm of its corresponding filter kernel weights. Here, we provide a proof for Equation 2 in the main paper. For simplicity but without loss of generality, let A be a single-channel $n \times n$ input to a $k \times k$ convolutional filter with kernel W. For illustration, consider a 3×3 input A and a 2×2 kernel W as shown below:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \text{ and } W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

Assuming the origin for the kernel W is at the top-left corner and no padding for A (same proof applies also if padding is applied), then the vectorized convolutional output

$$\mathbf{e} = vec(A * W) = \begin{bmatrix} w_{11} \cdot a_1 + w_{12} \cdot a_2 + w_{21} \cdot a_4 + w_{22} \cdot a_5 \\ w_{11} \cdot a_2 + w_{12} \cdot a_3 + w_{21} \cdot a_5 + w_{22} \cdot a_6 \\ w_{11} \cdot a_4 + w_{12} \cdot a_5 + w_{21} \cdot a_7 + w_{22} \cdot a_8 \\ w_{11} \cdot a_5 + w_{12} \cdot a_6 + w_{21} \cdot a_8 + w_{22} \cdot a_9 \end{bmatrix}$$

can be expressed as a matrix-vector product as follows:

$$\mathbf{e} = vec(A * W) = M\mathbf{r} \tag{1}$$

$$M = \begin{bmatrix} w_{11} & w_{12} & 0 & w_{21} & w_{22} & 0 & 0 & 0 & 0 \\ 0 & w_{11} & w_{12} & 0 & w_{21} & w_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & w_{11} & w_{12} & 0 & w_{21} & w_{22} & 0 \\ 0 & 0 & 0 & 0 & w_{11} & w_{12} & 0 & w_{21} & w_{22} \end{bmatrix} \tag{2}$$

$$\mathbf{r}^T = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 \end{bmatrix} \tag{3}$$

where $vec(\cdot)$ unrolls all elements of the input matrix with N_1 rows and N_2 columns into an output column vector of size $N_1 N_2$, M is a circulant convolution matrix formed using the elements of W and $\mathbf{r} = vec(A)$.

Similarly, for $A \in \mathbb{R}^{n \times n}$ and $W \in \mathbb{R}^{k \times k}$ such that w_{ij} is an element in row i and column j of W, we have $M \in \mathbb{R}^{(n-k+1)^2 \times n^2}$, and $e \in \mathbb{R}^{(n-k+1)^2}$ is given by:

$$e = M\mathbf{r} \tag{4}$$

$$\|e\|_\infty = \|M\mathbf{r}\|_\infty = \max_{1 \le i \le (n-k+1)^2} |\sum_{j=1}^{n^2} m_{ij} r_j| \tag{5}$$

$$\le \max_{1 \le i \le (n-k+1)^2} \sum_{j=1}^{n^2} |m_{ij}||r_j|$$

$$\le \left(\max_{1 \le i \le (n-k+1)^2} \sum_{j=1}^{n^2} |m_{ij}| \right) \max_{1 \le j \le n^2} |r_j|$$

$$\le \left(\max_{1 \le i \le (n-k+1)^2} \sum_{j=1}^{n^2} |m_{ij}| \right) \|\mathbf{r}\|_\infty \tag{6}$$

where $\mathbf{r} = vec(A), \mathbf{r} \in \mathbb{R}^{n^2}$ such that r_j is the j^{th} element in the vector $\mathbf{r}$ and m_{ij} is the element in row i and column j of the matrix M.

From Equation 2, $\sum_{j=1}^{n^2} |m_{ij}|$ is always equal to the ℓ_1-norm of the filter kernel weights $\|W\|_1 = \sum_{i'=1}^{k} \sum_{j'=1}^{k} |w_{i'j'}|$ for any row i, $1 \le i \le (n-k+1)^2$. Equation 6, can now be rewritten as:

$$\|\mathbf{e}\|_\infty \le \|W\|_1 \|\mathbf{r}\|_\infty \tag{7}$$

Since $\|\cdot\|_\infty \le \|\cdot\|_1$ and $\|\cdot\|_\infty \le \|\cdot\|_2$, we have the following inequality:

$$\|\mathbf{e}\|_\infty \le \|W\|_1 \|\mathbf{r}\|_p \tag{8}$$

where $p = 1, 2, \infty$.

B. Masking Perturbations in Other Layers

In Section 4.1 of the main paper (Figure 3 in the main paper), we evaluate the effect of masking ℓ_∞-norm adversarial perturbations in a ranked subset (using ℓ_1-norm ranking) of convolutional filter activation maps of the first convolutional layer of a DNN. Here, in Figure 1, we evaluate

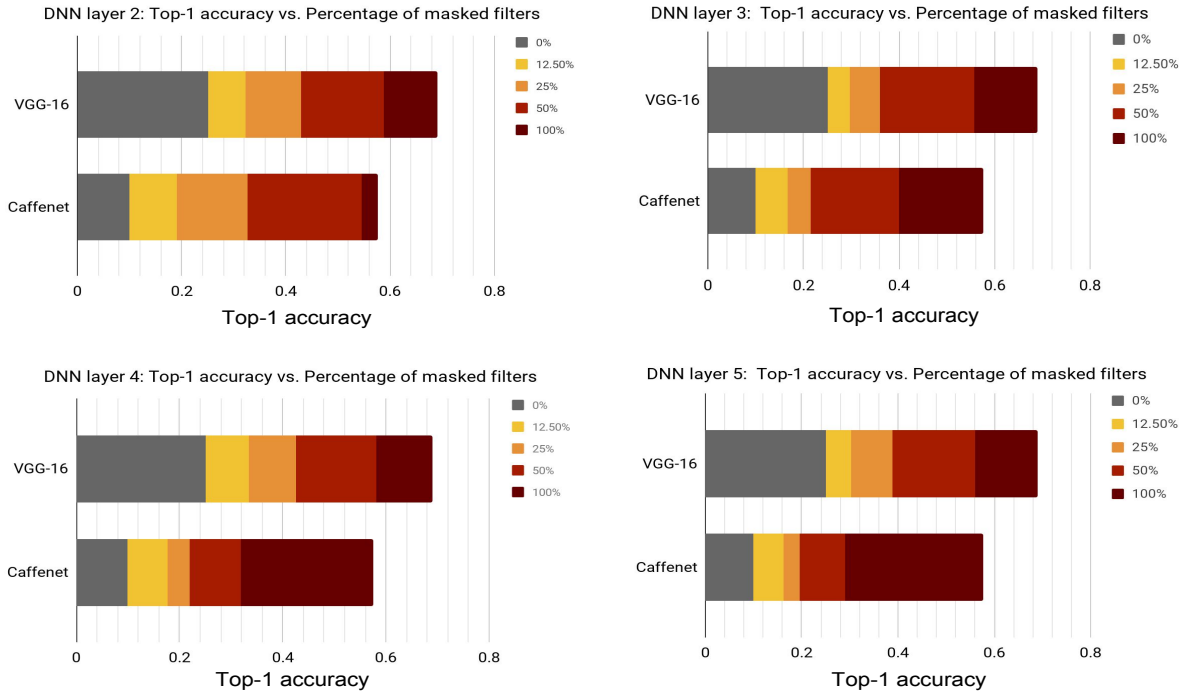

Figure 1. Effect of masking ℓ_∞-norm universal adversarial noise in ranked convolutional filter activations of CaffeNet [3] and VGG-16 [8], evaluated on a 1000-image subset of the ImageNet [1] training set. Top-1 accuracies for perturbation-free images are 0.58, 0.69 for CaffeNet and VGG-16, respectively. Similarly, top-1 accuracies for adversarially perturbed images with no noise masking are 0.1 and 0.25 for CaffeNet and VGG-16, respectively. For VGG-16, masking the noise in just 50% of the ranked filter activations restores more than $\approx$ 80% of the baseline accuracy on perturbation-free images.

the effect of masking ℓ_∞-norm adversarial perturbations in ranked filter activation maps of the convolutional layers 2, 3, 4 and 5 of CaffeNet [3] and VGG-16 [8]. We use the same evaluation setup as in Section 4.1 of the main paper (i.e., 1000 image random subset of the ImageNet [1] training set). The top-1 accuracy for perturbation-free images of the subset are 0.58 and 0.69 for CaffeNet and VGG-16, respectively. Similarly, the top-1 accuracies for adversarially perturbed images in the subset are 0.10 and 0.25 for CaffeNet and VGG-16, respectively. Similar to our observations in Section 4.1 of the main paper, for most DNN layers, masking the adversarial perturbations in just the top 50% most susceptible filter activation maps (identified by using the ℓ_1-norm ranking measure, Section 4.1 of the paper), is able to recover most of the accuracy lost by the baseline DNN (Figure 1). Specifically, masking the adversarial perturbations in the top 50% ranked filters of VGG-16 is able to restore at least 84% of the baseline accuracy on perturbation-free images.

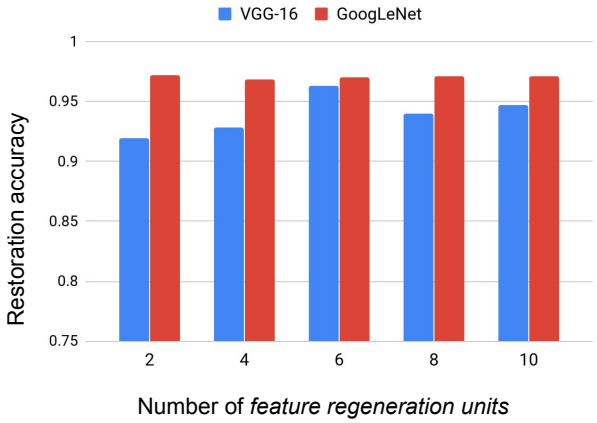

Figure 2. Effect of adding *feature regeneration units* on the restoration accuracy of our proposed defense. Adding just two *feature regeneration units* in GoogLeNet [9] achieves a restoration accuracy of 97% and adding more *feature regeneration units* to the DNN does not improve results any further. For VGG-16 [8], adding 6 *feature regeneration units* provides best results.

C. *Feature Regeneration Units*: An Ablation Study

In general, *feature regeneration units* can be added at the output of each convolutional layer in a DNN. However, this may come at the cost of increased computations, due to an increase in the number of DNN parameters. As mentioned in Section 5.1 of the main paper, we constrain the number of *feature regeneration units* added to the DNN, in order to avoid drastically increasing the training and inference cost for larger DNNs (i.e., VGG-16, GoogLeNet and ResNet-152). Here, we perform an ablation study to identify the least number of *feature regeneration units* needed to at least achieve a 95% restoration accuracy across most DNNs. Specifically, we use VGG-16 [8] and GoogLeNet [9] for this analysis. We evaluate the restoration accuracy on the ImageNet [1] validation set (ILSVRC2012) by adding an increasing number of *feature regeneration units*, starting from a minimum value of 2 towards a maximum value of 10 in steps of 2. Starting from the first convolutional layer in a DNN, each additional *feature regeneration unit* is added at the output of every second convolutional layer. In Figure 2, we report the results of this ablation study and observe that for GoogLeNet, adding just two *feature regeneration units* achieves a restoration accuracy of 97% and adding any more *feature regeneration units* does not have any significant impact on the restoration accuracy. However, for VGG-16, adding only 2 *feature regeneration units* achieves a restoration accuracy of only 91%. For VGG-16, adding more *feature regeneration units* improves the performance with the best restoration accuracy of 96.2% achieved with 6 *feature regeneration units*. Adding more than 6 *feature regeneration units* resulted in a minor drop in restoration accuracy and this may be due to data over-fitting. As a result, we restrict the number of *feature regeneration units* deployed for any DNN to $\min(\#\text{DNN layers}, 6)$.

D. Attacks using Surrogate Defense DNNs

In this section, we evaluate if it is possible for an attacker/adversary to construct a surrogate defense network if it was known that our defense was adopted. In situations where exact defense (*feature regeneration units* + baseline DNN) is typically hidden from the attacker (*oracle*), a DNN predicting output labels similar to our defense (*surrogate*), can be effective only if an attack generated using the *surrogate* is transferable to the *oracle*. UAP [4] attacks are transferable across baseline DNNs (Table 1 in main paper), i.e., adversarial perturbation computed for a DNN whose model weights and architecture are known (surrogate) can also effectively fool another target DNN that has a similar prediction accuracy, but whose model weights and architecture are not known to the attacker (oracle). Assuming that our defense (*feature regeneration units* + baseline DNN) for CaffeNet [3] and Res152 [2] is available publicly as a *surrogate*, universal attack examples computed from these DNNs may be used to attack our defenses for other DNNs, e.g. VGG-F or VGG-16 as an *oracle*. We show in Table 1 that our defense mechanism successfully breaks attack transferability and is not susceptible to attacks from *surrogate* DNNs based on our defense.

Table 1. Defense restoration accuracy for oracle DNNs equipped with our defense for an ℓ_∞-norm UAP [4] attack ($\xi = 10$) using surrogate defense DNNs equipped with our defense.

Surrogate	Oracle		
	VGG-F + defense	GoogLeNet + defense	VGG-16 + defense
CaffeNet + defense	0.906	0.963	0.942
Res152 + defense	0.889	0.925	0.925

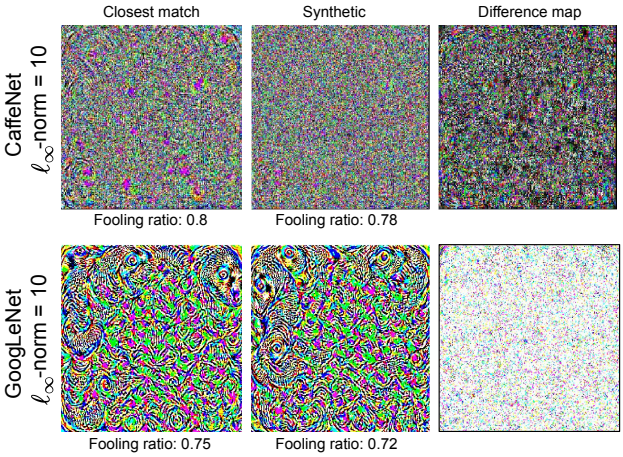

Figure 3. Visualization of synthetic perturbations (center) computed for CaffeNet [3] and GoogLeNet [9] along with their closest match in the set of original perturbations (left) and a per pixel difference map between the two (right).

E. Examples of Synthetic Perturbations

Sample visualizations of synthetic adversarial perturbations generated using our algorithm proposed in Section 4.3 (Algorithm 1) of the main paper are provided in Figure 3.

F. Examples of Feature Regeneration

Additional visualizations of DNN feature maps before and after *feature regeneration* using our proposed defense in Section 4.2 of the main paper are provided in Figure 4.

G. Examples of Universal Attack Perturbations

Sample visualizations of ℓ_∞-norm and ℓ_2-norm UAP [4] attack perturbations are shown in Figure 5.

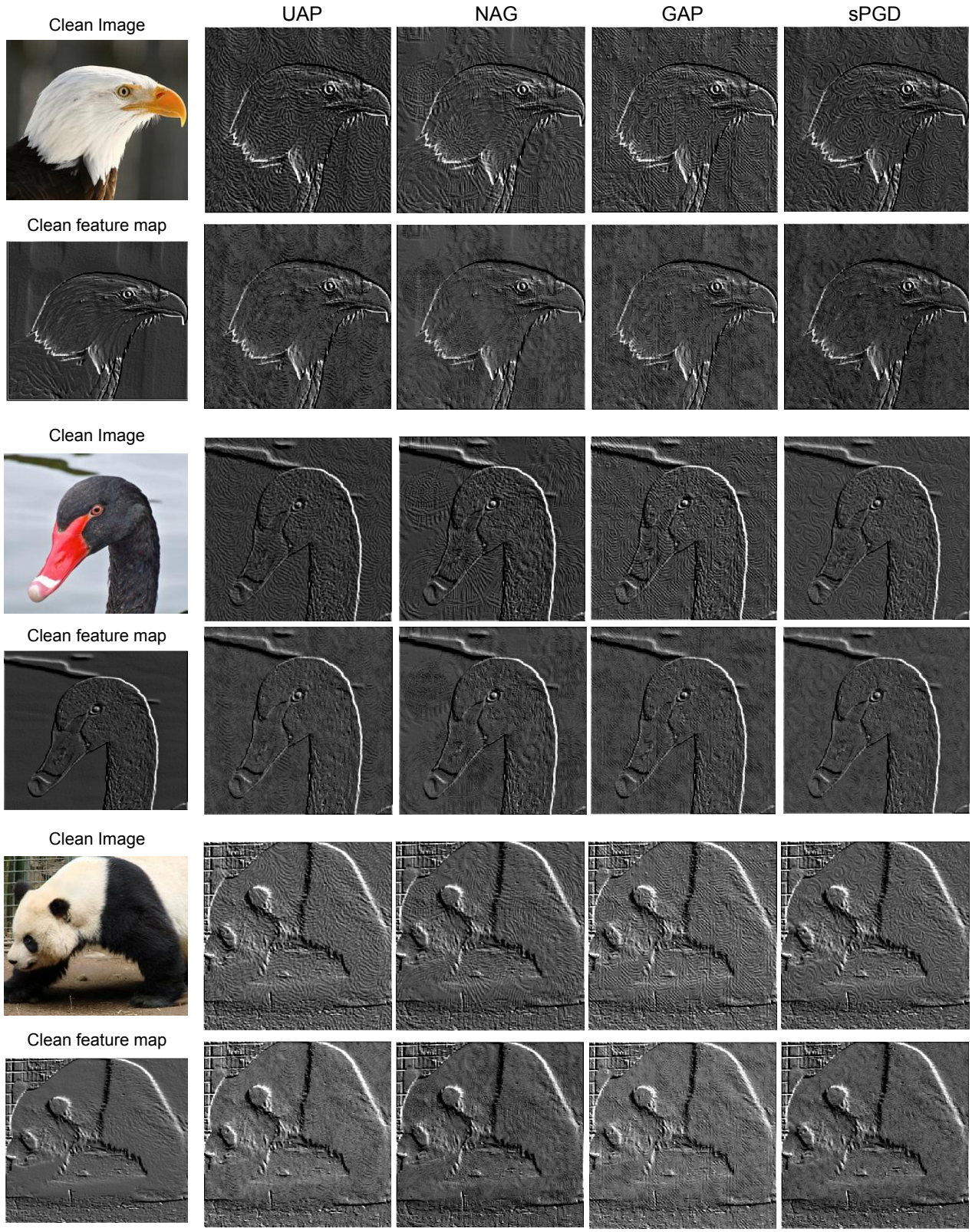

Figure 4. Visual examples of DNN feature maps before and after *feature regeneration* using our proposed method, for images perturbed by universal perturbations (UAP [4], NAG [5], GAP [7] and sPGD [6]). Perturbation-free feature map (clean feature map), different adversarially perturbed feature maps (Rows 1, 3 and 5) and corresponding feature maps regenerated by *feature regeneration units* (Rows 2, 4 and 6) are obtained for a single filter channel in conv1_1 layer of VGG-16 [8]. Our *Feature regeneration units* are only trained on UAP [4] attack examples.

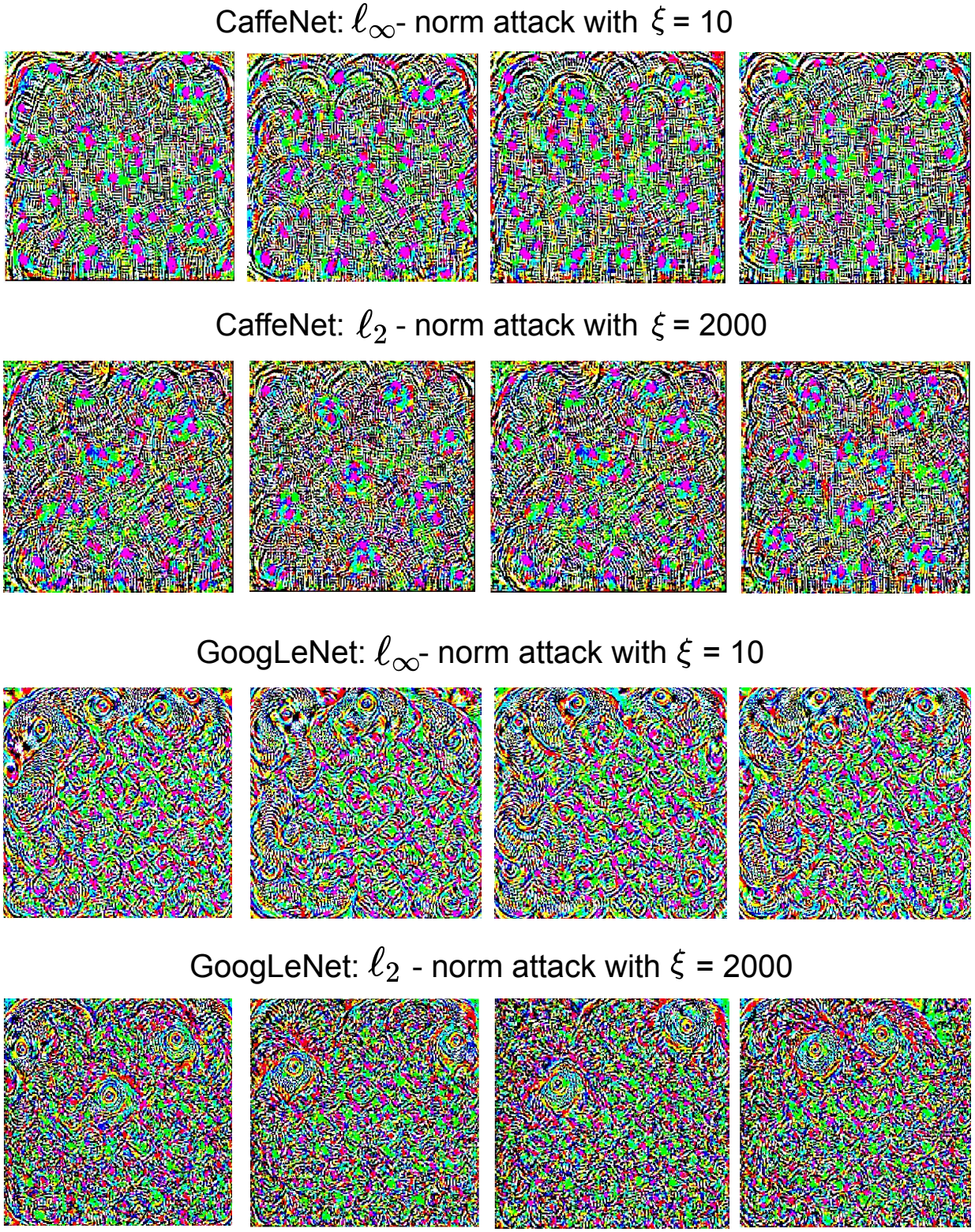

Figure 5. Visual examples of ℓ_∞-norm and ℓ_2-norm UAP [4] attack test perturbations for CaffeNet [3] and GoogLeNet [9].

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 3

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 2, 3, 5

[4] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 86–94, 2017. 3, 4, 5

[5] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. NAG: Network for adversary generation. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[6] Chaithanya Kumar Mummadi, Thomas Brox, and Jan Hendrik Metzen. Defending against universal perturbations with shared adversarial training. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 4

[7] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014. 2, 3, 4

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2, 3, 5