Defending Against Universal Attacks Through Selective Feature Regeneration

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

Abstract

Deep neural network (DNN) predictions have been shown to be vulnerable to carefully crafted adversarial perturbations. Specifically, image-agnostic (universal adversarial) perturbations added to any image can fool a target network into making erroneous predictions. Departing from existing defense strategies that work mostly in the image domain, we present a novel defense which operates in the DNN feature domain and effectively defends against such universal perturbations. Our approach identifies pre-trained convolutional features that are most vulnerable to adversarial noise and deploys trainable feature regeneration units which transform these DNN filter activations into resilient features that are robust to universal perturbations. Regenerating only the top 50% adversarially susceptible activations in at most 6 DNN layers and leaving all remaining DNN activations unchanged, we outperform existing defense strategies across different network architectures by more than 10% in restored accuracy. We show that without any additional modification, our defense trained on ImageNet with one type of universal attack examples effectively defends against other types of unseen universal attacks.

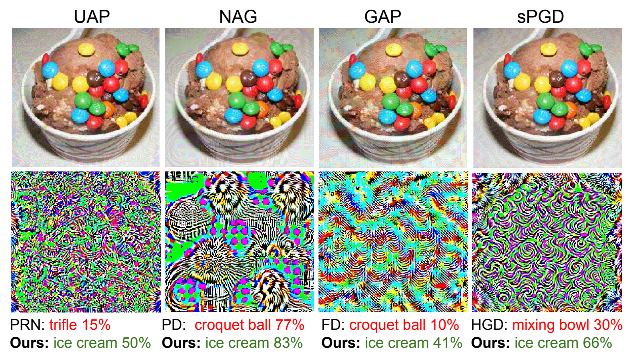

Figure 1. Effectiveness of proposed defense against various universal perturbations: row 1 shows an image (class label: "ice cream") perturbed by different universal attacks (UAP [38], NAG [44], GAP [51] and sPGD [45]) and the second row shows the variability across different universal perturbations. Row 3 shows predictions and confidence score for the proposed defense and the next best defense (PRN [1], PD [52], FD [66] and HGD [31]). Our method effectively defends against each universal attack by correctly classifying (green) the image with high confidence while all the other defenses misclassify it (red).

1. Introduction

Despite the continued success and widespread use of DNNs in computer vision tasks [25, 59, 62, 18, 55, 54, 58, 68], these networks make erroneous predictions when a small magnitude, carefully crafted perturbation (adversarial noise) almost visually imperceptible to humans is added to an input image [63, 15, 35, 6, 24, 41, 48, 26, 49]. Furthermore, such perturbations have been successfully placed in a real-world scene via physical adversarial objects [3, 12, 26], thus posing a security risk.

Most existing adversarial attacks use target network gradients to construct an image-dependent adversarial example [63, 15, 26, 41, 49, 6] that has limited transferability to other networks or images [63, 32, 47]. Other methods to generate image-dependent adversarial samples include accessing only the network predictions [20, 46, 61], using surrogate networks [48] and gradient approximation [2]. Although there is significant prior work on adversarial defenses such as adversarial training [63, 15, 35, 66], ensemble training [64], randomized image transformations and denoising [16, 52, 10, 40, 52, 60, 10, 33, 31] and adversarial sample rejection [29, 34, 67, 36, 37], a DNN is still vulnerable to adversarial perturbations added to a non-negligible portion of the input [2, 65]. These defenses mostly focus on making a DNN robust to image-dependent adversarial perturbations which are less likely to be encountered in realistic vision applications [1, 45].

Our proposed work focuses on defending against universal adversarial attacks. Unlike the aforementioned image-dependent adversarial attacks, universal adversarial attacks [38, 44, 43, 51, 23, 45, 53, 42, 30] construct a single *image-agnostic* perturbation that when added to any unseen image fools DNNs into making erroneous predictions with very high confidence. These universal perturbations are also not unique and many adversarial directions may exist in a

DNN's feature space (Figure 1, row 2) [39, 14, 13]. Furthermore, universal perturbations generated for one DNN can transfer to other DNNs, making them *doubly universal* [38]. Such *image-agnostic* perturbations pose a strong realistic threat model [45] for many vision applications as perturbations can easily be pre-computed and then inserted in real-time (in the form of a printed adversarial patch or sticker) into any scene [28, 5]. For example, while performing semantic segmentation, such *image-agnostic* perturbations can completely hide a target class (i.e., pedestrian) in the resulting segmented scene output and adversely affect the braking action of an autonomous car [19].

This work proposes a novel defense against a universal adversarial threat model [38, 43, 44, 51, 23, 45] through the following contributions:

- We show the existence of a set of vulnerable convolutional filters, that are largely responsible for erroneous predictions made by a DNN in an adversarial setting and the ℓ_1-norm of the convolutional filter weights can be used to identify such filters.
- Unlike, existing image-domain defenses, our proposed DNN feature space-based defense uses trainable *feature regeneration units*, which regenerate activations of the aforementioned vulnerable convolutional filters into resilient features (adversarial noise masking).
- A fast method is proposed to generate strong synthetic adversarial perturbations for training.
- We extensively evaluate the proposed defense on a variety of DNN architectures and show that our proposed defense outperforms all other existing defenses [1, 52, 66, 31, 35, 45] (Figure 1).
- Without any additional attack-specific training, our defense trained on one type of universal attack [38] effectively defends against other different unseen universal attacks [44, 43, 51, 45, 23, 42] (Figure 1) and we are the first to show such broad generalization across different universal attacks.

2. Related Work

Adversarial training (Adv. tr.) [63, 15, 35] has been shown to improve DNN robustness to image-dependent adversarial attacks through augmentation, in the training stage, with adversarial attack examples, which are computed on-the-fly for each mini-batch using gradient-ascent to maximize the DNN's loss. The robustness of adversarial training to black-box attacks can be improved by using perturbations computed against different target DNNs that are chosen from an ensemble of DNNs [64]. Kannan *et al.* [22] scale adversarial training to ImageNet [9] by encouraging the adversarial loss to match logits for pairs of adversarial and perturbation-free images (logit pairing) but this latter method fails against stronger iterative attacks [11]. In addition to adversarially training the baseline DNN, prior works ([66], [27]) further improved DNN robustness to image-dependent attacks by denoising intermediate DNN feature maps, either through a non-local mean denoiser (feature denoising [66]) or a denoising autoencoder (fortified nets [27]). Although Xie *et al.* report effective robustness against a strong PGD attack [35] evaluated on ImageNet [9], the additional non-local mean denoisers only add a 4% improvement over a DNN trained using standard adversarial training. Compared to feature denoising (FD) [66], the proposed feature regeneration approach has the following differences: (1) our *feature regeneration units* are not restricted to only perform denoising, but consists of stacks of trainable convolutional layers that provide our defense the flexibility to learn an appropriate feature-restoration transform that effectively defends against universal attacks, unlike the non-local mean denoiser used in FD; (2) in a selected DNN layer, only a subset of feature maps which are the most susceptible to adversarial noise (identified by our ranking metric) are regenerated leaving all other feature maps unchanged, whereas FD denoises all feature maps, which can result in over-correction or introduce unwanted artifacts in feature maps that admit very low magnitude noise; (3) instead of adversarially training all the parameters of the baseline DNN as in FD, we only train the parameters in the *feature renegeration units* (up to 90% less parameters than a baseline DNN) and leave all parameters in the baseline DNN unchanged, which can speed up training and reduce the risk of over-fitting.

Image-domain defenses mitigate the impact of adversarial perturbations by utilizing non-differentiable transformations of the input such as image compression [10, 8, 33], frequency domain denoising [60] and image quilting and reconstruction [16, 40] etc. However, such approaches introduce unnecessary artifacts in clean images resulting in accuracy loss [1][52]. Prakash *et al.* [52] propose a two-step defense that first performs random local pixel redistribution, followed by a wavelet denoising. Liao *et al.* [31] append a denoising autoencoder at the input of the baseline DNN and train it using a reconstruction loss that minimizes the error between higher layer representations of the DNN for an input pair of clean and denoised adversarial images (high level guided denoiser). Another popular line of defenses explores the idea of first detecting an adversarially perturbed input and then either abstaining from making a prediction or further pre-processing adversarial input for reliable predictions [29, 34, 67, 36, 37].

All of the aforementioned defenses are geared towards image-specific gradient-based attacks and none of them has, as of yet, been shown to defend against *image-agnostic* attacks. Initial attempts at improving robustness to universal attacks involved modelling the distribution of such perturbations [38, 17, 50], followed by model fine-tuning over this distribution of universal perturbations. However, the robustness offered by these methods has been unsatisfactory [45, 38] as the retrained network ends up overfitting to the small set of perturbations used. Extending adversarial training for image-dependent attacks to universal attacks has been attempted in [45] and [57]. Ruan and Dai [56] use additional shadow classifiers to identify and reject images perturbed by universal perturbations. Akhtar *et al.* [1] propose a defense against the universal adversarial perturbations attack (UAP) [38], using a detector which identifies adversarial images and then denoises them using a learnable *Perturbation Rectifying Network* (PRN).

3. Universal threat model

Let μ_c represent the distribution of clean (unperturbed) images in $\mathbb{R}^d$, $\mathscr{F}(\cdot)$ be a classifier that predicts a class label $\mathscr{F}(x)$ for an image $x \in \mathbb{R}^d$. The universal adversarial perturbation attack seeks a perturbation vector $v \in \mathbb{R}^d$ under the following constraints [38]:

$$\underset{x\sim\mu_c}{P}\Big(\mathscr{F}(x+v) \neq \mathscr{F}(x)\Big) \geq (1-\delta) \;\; \text{s.t.} \;\; \|v\|_p \leq \xi \quad (1)$$

where $P(\cdot)$ denotes probability, $\|\cdot\|_p$ is the ℓ_p-norm with $p \in [1,\infty)$, $(1-\delta)$ is the target *fooling ratio* with $\delta \in [0,1)$ (i.e., the fraction of samples in μ_c that change labels when perturbed by an adversary), and ξ controls the magnitude of adversarial perturbations.

4. Feature-Domain Adversarial Defense

4.1. Stability of Convolutional Filters

In this work, we assess the vulnerability of individual convolutional filters and show that, for each layer, certain filter activations are significantly more disrupted than others, especially in the early layers of a DNN.

For a given layer, let $\phi_m(u)$ be the output (activation map) of the m^{th} convolutional filter with kernel weights W_m for an input u. Let $e_m = \phi_m(u+r) - \phi_m(u)$ be the additive noise (perturbation) that is caused in the output activation map $\phi_m(u)$ as a result of applying an additive perturbation r to the input u. It can be shown (refer to Supplementary Material) that e_m is bounded as follows:

$$\|e_m\|_\infty \leq \|W_m\|_1 \|r\|_p \quad (2)$$

where as before $\|\cdot\|_p$ is the ℓ_p-norm with $p \in [1,\infty)$. Equation 2 shows that the ℓ_1-norm of the filter weights can be used to identify and rank convolutional filter activations in terms of their ability to restrict perturbation in their activation maps. For example, filters with a small weight ℓ_1-norm would result in insignificant small perturbations in their output when their input is perturbed, and are thus considered to be less vulnerable to perturbations in the input. For an

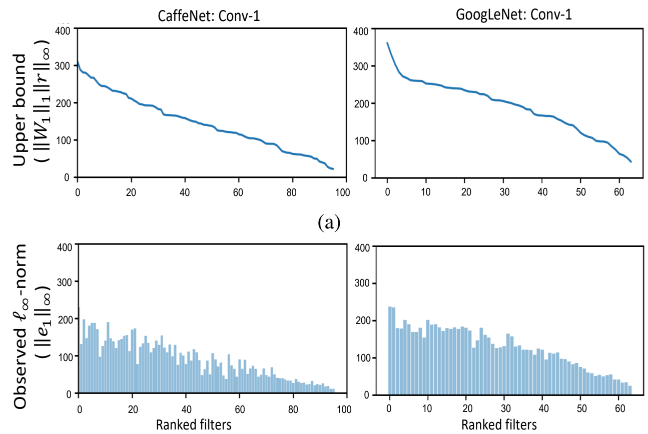

Figure 2. Observed ℓ_∞-norm for universal adversarial noise in the activation maps of ranked convolutional filters (ordered using our ℓ_1-norm ranking measure, from most to least vulnerable) of the first layer of CaffeNet [25] and GoogLeNet [62]. The ℓ_∞-norm attack is used with $\xi \leq 10$, i.e. $\|\mathbf{r}\|_\infty \leq 10$. (a) Adversarial noise upper-bound (Equation 2) in ranked conv-1 filter activations of DNNs. (b) Observed ℓ_∞-norm for adversarial noise in ranked conv-1 filter activations of DNNs.

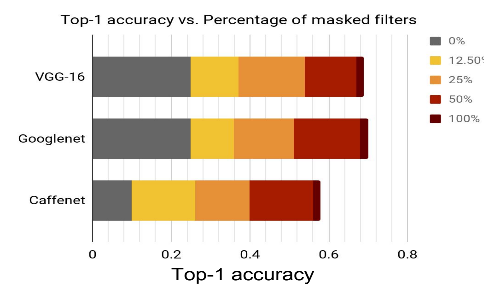

Figure 3. Effect of masking ℓ_∞-norm universal adversarial noise in ranked convolutional filter activations of the first layer in CaffeNet [25], GoogLeNet [62] and VGG-16 [59], evaluated on a 1000-image subset of the ImageNet [9] training set. Top-1 accuracies for perturbation-free images are 0.58, 0.70 and 0.69 for CaffeNet, GoogLeNet and VGG-16, respectively. Similarly, top-1 accuracies for adversarially perturbed images with no noise masking are 0.1, 0.25 and 0.25 for CaffeNet, GoogLeNet and VGG-16, respectively. Masking the noise in just 50% of the ranked filter activations restores most of the lost accuracy for all three DNNs.

ℓ_∞-norm universal adversarial input, Figure 2a shows the upper-bound on the adversarial noise in ranked (using the proposed ℓ_1-norm ranking) conv-1 filter activations of CaffeNet [25] and GoogLeNet [62], while Figure 2b shows the corresponding observed ℓ_∞-norm for adversarial noise in the respective DNN filter activations. We can see that our $\|W\|_1$-based ranking correlates well with the degree of perturbation (maximum magnitude of the noise perturbation) that is induced in the filter outputs. Similar observations can be made for other convolutional layers in the network.

In Figure 3, we evaluate the impact of masking the ad-

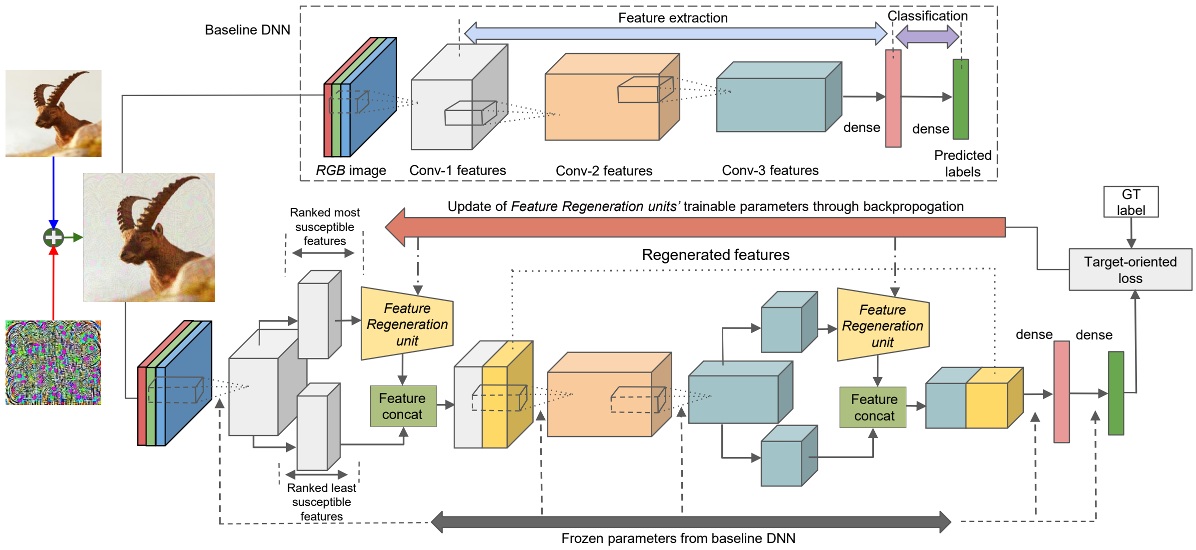

Figure 4. **Resilient Feature Regeneration Defense**: Convolutional filter activations in the baseline DNN (top) are first sorted in order of vulnerability to adversarial noise using their respective filter weight norms (Section 4.1). For each considered layer, we use a *feature regeneration unit*, consisting of a residual block with a single skip connection (4 layers), to regenerate only the most adversarially susceptible activations into resilient features that restore the lost accuracy of the baseline DNN, while leaving the remaining filter activations unchanged. We train these units on both clean and perturbed images in every mini-batch using the same target loss as the baseline DNN such that all parameters of the baseline DNN are left unchanged during training.

versarial noise in such ranked filters on the overall top-1 accuracy of CaffeNet [25], VGG-16 [59] and GoogLeNet [62]. Specifically, we randomly choose a subset of 1000 images (1 image per class) from the ImageNet [9] training set and generate adversarially perturbed images by adding an ℓ_∞-norm universal adversarial perturbation [38]. The top-1 accuracies for perturbation-free images are 0.58, 0.70 and 0.69 for CaffeNet, GoogLeNet and VGG-16, respectively. Similarly, the top-1 accuracies for adversarially perturbed images of the same subset are 0.10, 0.25 and 0.25 for CaffeNet, GoogLeNet and VGG-16, respectively. Masking the adversarial perturbations in 50% of the most vulnerable filter activations significantly improves DNN performance, resulting in top-1 accuracies of 0.56, 0.68 and 0.67 for CaffeNet, GoogLeNet and VGG-16, respectively, and validates our proposed selective feature regeneration scheme. See Figure 1 in Supplementary Material for similar experiments for higher layers.

4.2. Resilient Feature Regeneration Defense

Our proposed defense is illustrated in Figure 4. We learn a task-driven feature restoration transform (i.e., *feature regeneration unit*) for convolutional filter activations severely disrupted by adversarial input. Our *feature regeneration unit* does not modify the remaining activations of the baseline DNN. A similar approach of learning corrective transforms for making networks more resilient to image blur and additive white Gaussian noise has been explored in [4].

Let S_l represent a set consisting of indices for convolutional filters in the l^{th} layer of a DNN. Furthermore, let $S_{l_{reg}}$ be the set of indices for filters we wish to regenerate (Section 4.1) and let $S_{l_{adv}}$ be the set of indices for filters whose activations are not regenerated (i.e., $S_l = S_{l_{reg}} \cup S_{l_{adv}}$). If $\Phi_{S_{l_{reg}}}$ represents the convolutional filter outputs to be regenerated in the l^{th} layer, then our *feature regeneration unit* in layer l performs a feature regeneration transform $\mathscr{D}_l(\cdot)$ under the following conditions:

$$\mathscr{D}_l(\Phi_{S_{l_{reg}}}(u+r)) \approx \Phi_{S_{l_{reg}}}(u) \tag{3}$$

and

$$\mathscr{D}_l(\Phi_{S_{l_{reg}}}(u)) \approx \Phi_{S_{l_{reg}}}(u) \tag{4}$$

where u is the unperturbed input to the l^{th} layer of convolutional filters and r is an additive perturbation that acts on u. In Equations 3 and 4, $\approx$ denotes similarity based on classification accuracy in the sense that features are restored to regain the classification accuracy of the original perturbation-free activation map. Equation 3 forces $\mathscr{D}_l(\cdot)$ to pursue task-driven feature regeneration that restores lost accuracy of the DNN while Equation 4 ensures that prediction accuracy on unperturbed activations is not decreased, without any additional adversarial perturbation detector. We implement $\mathscr{D}_l(\cdot)$ (i.e., *feature regeneration unit*) as a shallow *residual block* [18], consisting of two stacked 3×3 convolutional layers sandwiched between a couple of 1×1 convolutional layers and a single skip connection. $\mathscr{D}_l(\cdot)$ is estimated using a target loss from the baseline network, through backpropagation, see Figure 4, but with significantly fewer trainable parameters compared to the baseline network.

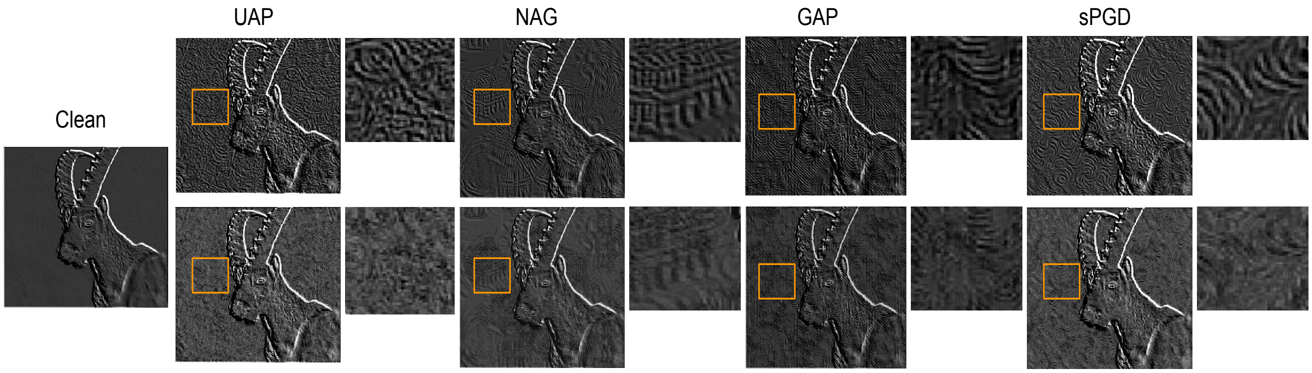

Figure 5. Effectiveness of *feature regeneration units* at masking adversarial perturbations in DNN feature maps for images perturbed by universal perturbations (UAP [38], NAG [44], GAP [51] and sPGD [45]). Perturbation-free feature map (clean), different adversarially perturbed feature maps (Row 1) and corresponding feature maps regenerated by *feature regeneration units* (Row 2) are obtained for a single filter channel in conv1_1 layer of VGG-16 [59], along with an enlarged view of a small region in the feature map (yellow box). *Feature regeneration units* are only trained on UAP [38] attack examples but are very effective at suppressing adversarial artifacts generated by unseen attacks (e.g., NAG [44], GAP [51] and sPGD [45]).

Given an L layered DNN Φ, pre-trained for an image classification task, Φ can be represented as a function that maps network input x to an N-dimensional output label vector $\Phi(x)$ as follows:

$$\Phi = \Phi_L \circ \Phi_{L-1} \circ \ldots \Phi_2 \circ \Phi_1 \tag{5}$$

where Φ_l is a mapping function (set of convolutional filters, typically followed by a non-linearity) representing the l^{th} DNN layer and N is the dimensionality of the DNN's output (i.e., number of classes). Without any loss of generality, the resulting DNN after deploying a *feature regeneration unit* that operates on the set of filters represented by $S_{l_{reg}}$ in layer l is given by:

$$\Phi_{reg} = \Phi_L \circ \Phi_{L-1} \circ \ldots \Phi_{l_{reg}} \ldots \Phi_2 \circ \Phi_1 \tag{6}$$

where $\Phi_{l_{reg}}$ represents the new mapping function for layer l, such that $\mathscr{D}_l(\cdot)$ regenerates only activations of the filter subset $\Phi_{S_{l_{reg}}}$ and all the remaining filter activations (i.e., $\Phi_{S_{l_{adv}}}$) are left unchanged. If $\mathscr{D}_l(\cdot)$ is parameterized by θ_l, then the *feature regeneration unit* can be trained by minimizing:

$$\mathcal{J}(\theta_l) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}(y_k, \Phi_{reg}(x_k)) \tag{7}$$

where $\mathcal{L}$ is the same target loss function of the baseline DNN (e.g., cross-entropy classification loss), y_k is the target output label for the k^{th} input image x_k, K represents the total number of images in the training set consisting of both clean and perturbed images. As we use both clean and perturbed images during training, x_k in Equation 7, represents a clean or an adversarially perturbed image.

In Figure 5, we visualize DNN feature maps perturbed by various universal perturbations and the corresponding feature maps regenerated by our *feature regeneration units*, which are only trained on UAP [38] attack examples. Compared to the perturbation-free feature map (clean), corresponding feature maps for adversarially perturbed images (Row 1) have distinctly visible artifacts that reflect the universal perturbation pattern in major parts of the image. In comparison, feature maps regenerated by our *feature regeneration units* (Row 2) effectively suppress these adversarial perturbations, preserve the object discriminative attributes of the clean feature map and are also robust to unseen attacks (e.g, NAG [44], GAP [51] and sPGD [45]), as illustrated in Figure 5 and Table 5.

4.3. Generating Synthetic Perturbations

Training-based approaches are susceptible to data overfitting, especially when the training data is scarce or does not have adequate diversity. Generating a diverse set of adversarial perturbations (≥ 100) using existing attack algorithms (e.g., [38, 44, 51, 45]), in order to avoid overfitting, can be computationally prohibitive. We propose a fast method (Algorithm 1) to construct synthetic universal adversarial perturbations from a small set of adversarial perturbations, $V \subseteq \mathbb{R}^d$, that is computed using any existing universal attack generation method ([38, 44, 51, 45]). Starting with the synthetic perturbation v_{syn} set to zero, we iteratively select a random perturbation $v_{new} \in V$ and a random scale factor $\alpha \in [0, 1]$ and update v_{syn} as follows:

$$v_{syn}(t) = \alpha v_{new} + (1-\alpha) v_{syn}(t-1) \tag{8}$$

where t is the iteration number. This process is repeated until the ℓ_2-norm of v_{syn} exceeds a threshold η. We set the threshold η to be the minimum ℓ_2-norm of perturbations in the set V.

Unlike the approach of Akhtar *et al.* [1], which uses an iterative random walk along pre-computed adversarial directions, the proposed algorithm has two distinct advantages:

Algorithm 1 Generating Synthetic Adversarial Perturbation

Input: Set of pre-computed perturbations $V \subseteq \mathbb{R}^d$ such that $v_i \in V$ is the i^{th} perturbation; threshold η

Output: Synthetic perturbation $v_{syn} \in \mathbb{R}^d$

1: $v_{syn} = 0$
2: **while** $\|\mathbf{v}_{syn}\|_2 \leq \eta$ **do**
3: $\quad \alpha \sim \text{uniform}(0, 1)$
4: $\quad v_{new} \overset{\text{rand}}{\sim} V$
5: $\quad v_{syn} = \alpha v_{new} + (1 - \alpha) v_{syn}$
6: **end while**
7: **return** v_{syn}

1) the same algorithm can be used for different types of attack norms without any modification, and 2) Equation 8 (Step 5 in Algorithm 1) automatically ensures that the ℓ_∞-norm of the perturbation does not violate the constraint for an ℓ_∞-norm attack (i.e., ℓ_∞-norm $\leq \xi$) and, therefore, no additional steps, like computing a separate perturbation unit vector and ensuring that the resultant perturbation strength is less than ξ, are needed.

5. Assessment

We use the ImageNet validation set (ILSVRC2012) [9] with all 50000 images and a single crop evaluation (unless specified otherwise) in our experiments. All our experiments are implemented using Caffe [21] and for each tested attack we use publicly provided code. We report our results in terms of top-1 accuracy and the restoration accuracy proposed by Akhtar *et al.* [1]. Given a set I_c containing clean images and a set $I_{p/c}$ containing clean and perturbed images in equal numbers, the restoration accuracy is given by:

$$\text{Restoration accuracy} = \frac{\text{acc}(I_{p/c})}{\text{acc}(I_c)} \tag{9}$$

where acc($\cdot$) is the top-1 accuracy. We use the universal adversarial perturbation (UAP) attack [38] for evaluation (unless specified otherwise) and compute 5 independent universal adversarial test perturbations per network using a set of 10000 held out images randomly chosen from the ImageNet training set with the fooling ratio for each perturbation lower-bounded to 0.8 on the held out images and the maximum normalized inner product between any two perturbations for the same DNN upper-bounded to 0.15.

5.1. Defense Training Methodology

In our proposed defense (Figure 4), only the parameters for *feature regeneration units* have to be trained and these parameters are updated to minimize the cost function given by Equation 7. Although we expect the prediction performance of defended models to improve with higher regeneration ratios (i.e., fraction of convolutional filter activations regenerated), we only regenerate 50% of the convolutional filter activations in a layer and limit the number of deployed *feature regeneration units* (1 per layer) as $\min(\#\text{DNN layers}, 6)$[1]. Using Algorithm 1, we generate 2000 synthetic perturbations from a set V of 25 original perturbations [38] and train *feature regeneration units* on a single Nvidia Titan-X using a standard SGD optimizer, momentum of 0.9 and a weight decay of 0.0005 for 4 epochs of the ImageNet training set [9]. The learning rate is dropped by a factor of 10 after each epoch with an initial learning rate of 0.1. After a defense model has been trained as outlined above, we can further iterate through the training of our defense with additional adversarial perturbations computed against our defense, which ensures robustness to secondary attacks against our defense (Section 5.2.5).

5.2. Analysis and Comparisons

5.2.1 Robustness across DNN Architectures

Top-1 accuracy of adversarially perturbed test images for various DNNs (no defense) and our proposed defense for respective DNNs is reported in Table 1 under both white-box (same network used to generate and test attack) and black-box (tested network is different from network used to generate attack) settings. As universal adversarial perturbations can be *doubly universal*, under a black-box setting, we evaluate a target DNN defense (defense is trained for attacks on target DNN) against a perturbation generated for a different network. Top-1 accuracy for baseline DNNs is severely affected by both white-box and black-box attacks, whereas our proposed defense is not only able to effectively thwart the white-box attacks but is also able to generalize to attacks constructed for other networks without further training (Table 1). Since different DNNs can share common adversarial directions in their feature space, our *feature regeneration units* learn to regularize such directions against unseen data and, consequently, to defend against black-box attacks.

5.2.2 Robustness across Attack Norms

Here, we evaluate defense robustness against both ℓ_∞-norm and ℓ_2-norm UAP [38] attacks. Since an effective defense must not only recover the DNN accuracy against adversarial images but must also maintain a high accuracy on clean images, we use restoration accuracy (Equation 9) to measure adversarial defense robustness (Tables 2 and 3). While Akhtar *et al.* [1] (PRN and PRN+det) only report defense results on the UAP [38] attack, we also compare results with pixel-domain defenses such as Pixel Deflection (PD [52]) and High Level Guided Denoiser (HGD [31]), defenses that use JPEG compression (JPEG comp. [10]) or DNN-based compression like Feature Distillation (Feat. Distill. [33]),

[1]From Figure 3 (main paper) and Figure 1 in Supplementary Material, we observe that an empirical regeneration ratio of 50% works well. Similarly, although *feature regeneration units* can be deployed for each layer in a DNN, from Figure 2 in Supplementary Material, we observe that regenerating features in at most 6 layers in a DNN effectively recovers lost prediction performance.

Table 1. **Cross-DNN evaluation on ILSVRC2012**: Top-1 accuracy against a ℓ_∞-norm UAP [38] attack with $\xi = 10$ and target fooling ratio of 0.8. DNNs in column one are tested with attacks generated for DNNs in row one.

	CaffeNet	VGG-F	GoogleNet	VGG-16	Res152
CaffeNet [25], orginal accuracy 56.4%					
CaffeNet	0.109	0.298	0.456	0.431	0.405
Ours	**0.542**	**0.524**	**0.510**	**0.457**	**0.470**
VGG-F [7], original accuracy 58.4%					
VGG-F	0.299	0.150	0.461	0.417	0.426
Ours	**0.556**	**0.550**	**0.548**	**0.492**	**0.513**
GoogLeNet [62], original accuracy 68.6%					
GoogLeNet	0.519	0.539	0.260	0.472	0.473
Ours	**0.651**	**0.653**	**0.653**	**0.637**	**0.642**
VGG-16 [59], original accuracy 68.4%					
VGG-16	0.549	0.559	0.519	0.240	0.484
Ours	**0.615**	**0.622**	**0.646**	**0.655**	**0.631**
Res152 [18], original accuracy 79%					
Res152	0.720	0.726	0.692	0.626	0.270
Ours	**0.764**	**0.769**	**0.769**	**0.763**	**0.761**

Table 2. **Same-norm evaluation on ILSVRC2012**: Restoration accuracy of DNNs and defenses against an ℓ_∞-norm UAP [38] attack with $\xi = 10$.

Methods	CaffeNet	VGG-F	GoogLeNet	VGG-16	Res152
ℓ_∞-norm attack, $\xi = 10$					
Baseline	0.596	0.628	0.691	0.681	0.670
PRN [1]	0.936	0.903	0.956	0.690	0.834
PRN+det [1]	0.952	0.922	0.964	0.690	0.834
PD [52]	0.873	0.813	0.884	0.818	0.845
JPEG comp. [10]	0.554	0.697	0.830	0.693	0.670
Feat. Distill. [33]	0.671	0.689	0.851	0.717	0.676
HGD [31]	n/a	n/a	n/a	n/a	0.739
Adv. tr. [35]	n/a	n/a	n/a	n/a	0.778
FD [66]	n/a	n/a	n/a	n/a	0.819
Ours	**0.976**	**0.967**	**0.970**	**0.963**	**0.982**

Table 3. **Cross-norm evaluation on ILSVRC2012**: Restoration accuracy against an ℓ_2-norm UAP [38] attack. Our defense, as well as the other defense models, are trained only on ℓ_∞-norm attack examples with $\xi = 10$.

Methods	CaffeNet	VGG-F	GoogLeNet	VGG-16	Res152
ℓ_2-norm attack, ξ = 2000					
Baseline	0.677	0.671	0.682	0.697	0.709
PRN [1]	0.922	0.880	0.971	0.834	0.868
PRN+det [1]	0.936	0.900	**0.975**	0.835	0.868
PD [52]	0.782	0.784	0.857	0.809	0.840
HGD [31]	n/a	n/a	n/a	n/a	0.730
Adv. tr. [35]	n/a	n/a	n/a	n/a	0.778
FD [66]	n/a	n/a	n/a	n/a	0.818
Ours	**0.964**	**0.961**	0.912	**0.876**	**0.926**

Table 4. Restoration accuracy on ILSVRC2012 for ℓ_∞-norm UAP [38] attack with stronger perturbation strengths (ξ) against CaffeNet. Our defense, as well as the other defense models, are trained only on ℓ_∞-norm attack examples with ξ=10.

Method	$\xi = 10$	$\xi = 15$	$\xi = 20$	$\xi = 25$
Baseline	0.596	0.543	0.525	0.519
PRN [1]	0.936	0.603	0.555	0.526
PRN+det [1]	0.952	0.604	0.555	0.526
PD [52]	0.873	0.616	0.549	0.524
Ours	**0.976**	**0.952**	**0.896**	**0.854**

defenses that use some variation of adversarial training like Feature Denoising (FD [66]) and standard Adversarial training (Adv. tr. [35]).

In Table 2, we report results for an ℓ_∞-norm UAP attack [38] against various DNNs and show that our proposed defense outperforms all the other defenses[2] for all networks with the highest restoration accuracy (98.2%) being achieved for Res152 [18]. Our *feature regeneration units* are trained on ℓ_∞-norm attack examples (same-norm evaluation). Even without a perturbation detector, our defense outperforms the existing defense with a perturbation detector (PRN+det) of Akhtar *et al.* [1] for all networks. Similarly, for Res152 [18], we outperform adversarially trained defenses (FD [66], Adv. tr. [35]) and pixel denoisers (PD [52], HGD [31]) by more than 10%. In Table 3, we also evaluate how well our defense trained on an ℓ_∞-norm attack defends against an ℓ_2-norm attack (cross-norm evaluation). Our *feature regeneration units* are able to effectively generalize to even cross-norm attacks and outperform all other defenses for most DNNs.

5.2.3 Stronger Attack Perturbations ($\xi > 10$)

Although we use an attack perturbation strength ξ = 10 during training, in Table 4, we evaluate the robustness of our defense when the adversary violates the attack threat model using a higher perturbation strength. Compared to the baseline DNN (no defense) as well as PRN [1] and PD [52], our proposed defense is much more effective at defending against stronger perturbations, outperforming other defenses by almost 30% even when the attack strength is more than double the value used to train our defense. Although defense robustness decreases for unseen higher perturbation strengths, our defense handles this drop-off much more gracefully and shows much better generalization across attack perturbation strengths, as compared to existing defenses. We also note that adversarial perturbations are no longer visually imperceptible at $\xi = 25$.

5.2.4 Generalization to Unseen Universal Attacks

Although the proposed method effectively defends against UAP [38] attacks (Tables1-4), we also assess its robustness to other unseen universal attacks without additional attack-specific training. Note that [1] and [45] do not cover this experimental setting. Since existing attacks in the literature are tailored to specific DNNs, we use CaffeNet [25] and Res152 [18] DNNs for covering a variety of universal attacks like Fast Feature Fool (FFF) [43], Network for adversary generation (NAG) [44], Singular fool (S.Fool) [23], Generative adversarial perturbation (GAP) [51], Generalizable data-free universal adversarial perturbation (G-UAP) [42], and stochastic PGD (sPGD) [45].

[2] FD [66], HGD [31] and Adv. tr. [35] defenses publicly provide trained defense models only for Res152 [18]) among the evaluated DNNs; we report results using only the DNN models provided by the respective authors.

Table 5. **Robustness to unseen attacks**: Restoration accuracy evaluated on ILSVRC2012, against other unseen universal attacks using our defense trained on just ℓ_∞-norm UAP [38] attack examples with a fooling ratio and ℓ_∞-norm of 0.8 and 10, respectively. Results for all other defenses are reported using publicly provided defense models. Attacks are constructed for the baseline DNN.

	CaffeNet			Res152		
Methods	**FFF** [43]	**NAG** [44]	**S.Fool** [23]	**GAP** [51]	**G-UAP** [42]	**sPGD** [45]
Baseline	0.645	0.670	0.815	0.640	0.726	0.671
PRN [1]	0.729	0.660	0.732	0.774	0.777	0.823
PD [52]	0.847	0.767	0.871	0.784	0.807	0.890
HGD [31]	n/a	n/a	n/a	0.663	0.782	0.932
Adv. tr [35]	n/a	n/a	n/a	0.776	0.777	0.775
FD [66]	n/a	n/a	n/a	0.815	0.813	0.815
Ours	**0.941**	**0.840**	**0.914**	**0.922**	**0.914**	**0.976**

Our defense trained on just UAP [38] attack examples is able to effectively defend against all other universal attacks and outperforms all other existing defenses (Table 5). Even against stronger universal attacks like NAG [44] and GAP [51], we outperform all other defenses including PRN [1], which is also trained on similar UAP [38] attack examples, by almost 10%. From our results in Table 5, we show that our *feature regeneration units* learn transformations that generalize effectively across perturbation patterns (Figure 5). Note that we are the first to show such broad generalization across universal attacks.

5.2.5 Robustness to Secondary White-Box Attacks

Although in practical situations, an attacker may not have full or even partial knowledge of a defense, for completeness, we also evaluate our proposed defense against a white-box attack on the defense (secondary attacks), i.e., adversary has full access to the gradient information of our *feature regeneration units*. We use the UAP [38] (on CaffeNet) and sPGD [45] (on Res152) attacks for evaluation.

Figure 6 shows the robustness of our defense to such a secondary UAP [38] attack seeking to achieve a target fooling ratio of 0.85 on our defense for the CaffeNet [25] DNN. Such an attack can easily converge (achieve target fooling ratio) against a baseline DNN in less than 2 attack epochs, eventually achieving a final fooling ratio of 0.9. Similarly, we observe that even PRN [1] is susceptible to a secondary UAP [38] attack, achieving a fooling ratio of 0.87, when the adversary can access gradient information for its *Perturbation Rectifying Network*. In comparison, using our defense model with iterative adversarial example training (as described in Section 5.1), the white-box adversary can achieve a maximum fooling ratio of only 0.42, which is 48% lower than the fooling ratio achieved against PRN [1], even after attacking our defense for 600 attack epochs. Similarly, in Table 6, using the same attack setup outlined in [45], we evaluate white-box sPGD [45] attacks computed by utilizing gradient-information of both the defense and the baseline DNNs, for Res152 [18]. As shown in Table 6, our defense trained using sPGD attack examples computed against

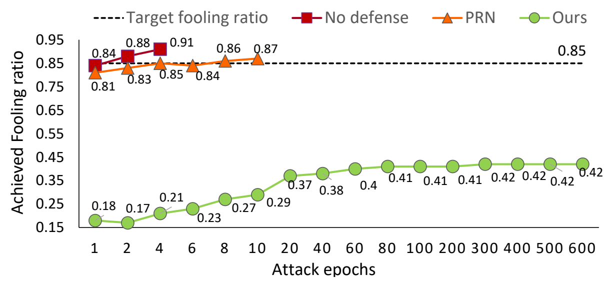

Figure 6. **Robustness to white-box attacks against defense (secondary attacks)**: Achieved fooling ratio by attacker vs. attack epochs for an ℓ_∞-norm UAP [38] attack ($\xi = 10$) against CaffeNet [25], where the attacker has full knowledge of the baseline DNN and also the defense. The target fooling ratio for attack is set to 0.85.

Table 6. Top-1 accuracy for white-box ℓ_∞-norm sPGD [45] attack against Res152-based ℓ_∞-norm defenses ($\xi = 10$), evaluated on ILSVRC2012. Top-1 accuracy for Res152 on clean images is 0.79.

Baseline	**Ours**	**FD** [66]	**Adv. tr.** [35]	**HGD** [31]	**Shared tr.** [45][3]
0.270	**0.731**	0.641	0.635	0.689	0.727

both the baseline DNN and our defense, is robust to subsequent sPGD white-box attacks.

6. Conclusion

We show that masking adversarial noise in a few select DNN activations significantly improves their adversarial robustness. To this end, we propose a novel selective feature regeneration approach that effectively defends against universal perturbations, unlike existing adversarial defenses which either pre-process the input image to remove adversarial noise and/or retrain the entire baseline DNN through adversarial training. We show that the ℓ_1-norm of the convolutional filter kernel weights can be effectively used to rank convolutional filters in terms of their susceptibility to adversarial perturbations. Regenerating only the top 50% ranked adversarially susceptible features in a few DNN layers is enough to restore DNN robustness and outperform all existing defenses. We validate the proposed method by comparing against existing state-of-the-art defenses and show better generalization across different DNNs, attack norms and even unseen attack perturbation strengths. In contrast to existing approaches, our defense trained solely on one type of universal adversarial attack examples effectively defends against other unseen universal attacks, without additional attack-specific training. We hope this work encourages researchers to design adversarially robust DNN architectures and training methods which produce convolutional filter kernels that have a small ℓ_1-norm.

[3] As an implementation of Shared Adversarial Training (Shared tr. [45]) was not publicly available, we report results published by the authors in [45] and which were only provided for white-box attacks computed against the defense, whereas results for white-box attacks against the baseline DNN were not provided.

References

[1] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3389–3398, June 2018. 1, 2, 3, 5, 6, 7, 8

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the International Conference on Machine Learning, (ICML)*, 2018. 1

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 284–293, 2018. 1

[4] Tejas S. Borkar and Lina J. Karam. DeepCorrect: Correcting DNN models against image distortions. *CoRR*, abs/1705.02406, 2017. 4

[5] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017. 2

[6] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017. 1

[7] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 7

[8] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E. Kounavis, and Duen Horng Chau. SHIELD: Fast, practical defense and vaccination for deep learning using JPEG compression. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 196–204, NY, USA, 2018. ACM. 2

[9] 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, 4, 6

[10] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. A study of the effect of JPG compression on adversarial images. *CoRR*, abs/1608.00853, 2016. 1, 2, 6, 7

[11] Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *CoRR*, abs/1807.10272, 2018. 2

[12] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models. *CoRR*, abs/1707.08945, 2017. 1

[13] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. The robustness of deep networks: A geometrical perspective. *IEEE Signal Process. Mag.*, 34(6):50–62, 2017. 2

[14] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Classification regions of deep neural networks. *CoRR*, abs/1705.09552, 2017. 2

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[16] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. *CoRR*, abs/1711.00117, 2017. 1, 2

[17] Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. *CoRR*, abs/1708.05207, 2017. 3

[18] 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. 1, 4, 7, 8

[19] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *IEEE International Conference on Computer Vision*, pages 2774–2783, Oct 2017. 2

[20] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *CoRR*, abs/1804.08598, 2018. 1

[21] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 6

[22] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018. 2

[23] Valentin Khrulkov and Ivan Oseledets. Art of singular vectors and universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8562 – 8570, June 2018. 1, 2, 7, 8

[24] Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. *CoRR*, abs/1702.06832, 2017. 1

[25] 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. 1, 3, 4, 7, 8

[26] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016. 1

[27] Alex Lamb, Jonathan Binas, Anirudh Goyal, Dmitriy Serdyuk, Sandeep Subramanian, Ioannis Mitliagkas, and Yoshua Bengio. Fortified networks: Improving the robustness of deep networks by modeling the manifold of hidden representations. *CoRR*, abs/1804.02485, 2018. 2

[28] Juncheng Li, Frank R. Schmidt, and J. Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. *CoRR*, abs/1904.00759, 2019. 2

[29] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *IEEE International Conference on Computer Vision*, pages 5775 – 5783, Oct 2017. 1, 2

[30] Yingwei Li, Song Bai, Cihang Xie, Zhenyu Liao, Xiaohui Shen, and Alan L. Yuille. Regional homogeneity: Towards learning transferable universal adversarial perturbations against defenses. *CoRR*, abs/1904.00979, 2019. 1

[31] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial at-

tacks using high-level representation guided denoiser. *CoRR*, abs/1712.02976, 2017. 1, 2, 6, 7, 8

[32] Y Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial samples and black-box attacks. *CoRR*, abs/1611.02770, 2016. 1

[33] Zihao Liu, Qi Liu, Tao Liu, Yanzhi Wang, and Wujie Wen. Feature Distillation: DNN-oriented JPEG compression against adversarial examples. *International Joint Conference on Artificial Intelligence*, 2018. 1, 2, 6, 7

[34] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *IEEE International Conference on Computer Vision*, pages 446–454, Oct 2017. 1, 2

[35] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017. 1, 2, 7, 8

[36] Dongyu Meng and Hao Chen. MagNet: A two-pronged defense against adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147, NY, USA, 2017. ACM. 1, 2

[37] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *International Conference on Learning Representations*, 2017. 1, 2

[38] 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. 1, 2, 3, 4, 5, 6, 7, 8

[39] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto. Analysis of universal adversarial perturbations. *CoRR*, abs/1705.09554, 2017. 2

[40] Seyed-Mohsen Moosavi-Dezfooli, Ashish Shrivastava, and Oncel Tuzel. Divide, denoise, and defend against adversarial attacks. *CoRR*, abs/1802.06806, 2018. 1, 2

[41] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, June 2016. 1

[42] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 41, pages 2452–2465, 2019. 1, 2, 7, 8

[43] Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1–12, 2017. 1, 2, 7, 8

[44] 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. 1, 2, 5, 7, 8

[45] 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. 1, 2, 3, 5, 7, 8

[46] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *CoRR*, abs/1612.06299, 2016. 1

[47] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1607.02533, 2016. 1

[48] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM Asia Conference on Computer and Communications Security*, pages 506–519, 2017. 1

[49] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay. Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy*, pages 372–387, 2016. 1

[50] Julien Perolat, Mateusz Malinowski, Bilal Piot, and Olivier Pietquin. Playing the game of universal adversarial perturbations. *CoRR*, abs/1809.07802, 2018. 3

[51] 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. 1, 2, 5, 7, 8

[52] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8571–8580, June 2018. 1, 2, 6, 7, 8

[53] Konda Reddy Mopuri, Phani Krishna Uppala, and R. Venkatesh Babu. Ask, acquire, and attack: Data-free uap generation using class impressions. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

[54] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 1

[55] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 1

[56] Yibin Ruan and Jiazhu Dai. TwinNet: A double sub-network framework for detecting universal adversarial perturbations. In *Future Internet*, volume 10, pages 1–13, 2018. 3

[57] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S. Davis, and Tom Goldstein. Universal adversarial training. *CoRR*, abs/1811.11304, 2018. 3

[58] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, Apr. 2017. 1

[59] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014. 1, 3, 4, 5, 7

[60] Sibo Song, Yueru Chen, Ngai-Man Cheung, and C.-C. Jay Kuo. Defense against adversarial attacks with Saak transform. *CoRR*, abs/1808.01785, 2018. 1, 2

[61] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *CoRR*, abs/1710.08864, 2017. 1

[62] 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. 1, 3, 4, 7

[63] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2

[64] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. 1, 2

[65] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *CoRR*, abs/1802.05666, 2018. 1

[66] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 7, 8

[67] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Network and Distributed System Security Symposium*, 2018. 1, 2

[68] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2015. 1