

# A Self-supervised Approach for Adversarial Robustness

Muzammal Naseer<sup>\*†‡</sup>, Salman Khan<sup>†</sup>, Munawar Hayat<sup>†</sup>, Fahad Shahbaz Khan<sup>†§</sup>, Fatih Porikli<sup>\*</sup>

<sup>\*</sup>Australian National University, Australia, <sup>†</sup>Data61-CSIRO, Australia

<sup>‡</sup>Inception Institute of Artificial Intelligence, UAE, <sup>§</sup>CVL, Linköping University, Sweden

{muzammal.naseer, fatih.porikli}@anu.edu.au

{salman.khan, munawar.hayat, fahad.khan}@inceptioniai.org

## Abstract

Adversarial examples can cause catastrophic mistakes in Deep Neural Network (DNNs) based vision systems e.g., for classification, segmentation and object detection. The vulnerability of DNNs against such attacks can prove a major roadblock towards their real-world deployment. Transferability of adversarial examples demand generalizable defenses that can provide cross-task protection. Adversarial training that enhances robustness by modifying target model’s parameters lacks such generalizability. On the other hand, different input processing based defenses fall short in the face of continuously evolving attacks. In this paper, we take the first step to combine the benefits of both approaches and propose a self-supervised adversarial training mechanism in the input space. By design, our defense is a generalizable approach and provides significant robustness against the **unseen** adversarial attacks (e.g. by reducing the success rate of translation-invariant **ensemble** attack from 82.6% to 31.9% in comparison to previous state-of-the-art). It can be deployed as a plug-and-play solution to protect a variety of vision systems, as we demonstrate for the case of classification, segmentation and detection. Code is available at: <https://github.com/Muzammal-Naseer/NRP>.

## 1. Introduction

Adversarial training (AT) has shown great potential to safeguard neural networks from adversarial attacks [29, 35]. So far in literature, AT is performed in the model space i.e., a model’s parameters are modified by minimizing empirical risk for a given data distribution as well as the perturbed images. Such AT strategy results in the following challenges. **(a) Task dependency:** AT is task-dependent e.g. robust classification models cannot directly be incorporated into an object detection or a segmentation pipeline, since the overall system would still require further training

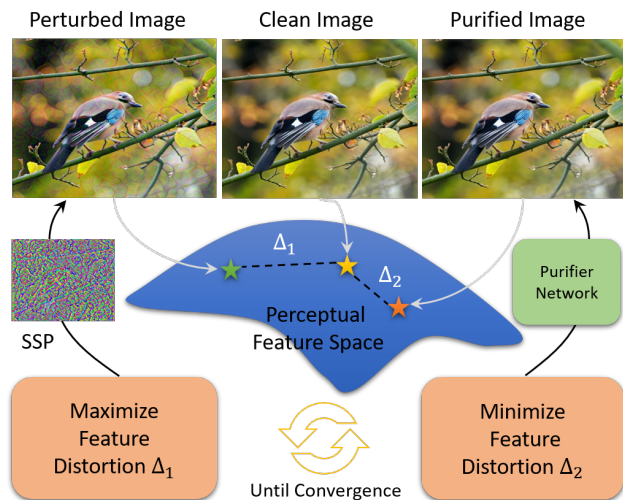


Figure 1: Our main idea is to train a Purifier Network in a self-supervised manner. We generate perturbed images using our proposed *Self-supervised Perturbation* (SSP) attack that disrupts the deep perceptual features. The *Purifier Network* projects back the perturbed images close to the perceptual space of clean images. This creates a training loop independent of the task or label space.

with modified task-dependant loss functions. **(b) Computational cost:** AT is computationally expensive [29] which restricts its applicability to high-dimensional and large-scale datasets such as ImageNet [34]. **(c) Accuracy drop:** models trained with AT lose significant accuracy on the original distribution e.g. ResNet50 [17] accuracy on ImageNet validation set drops from 76% to 64% when robustified against PGD attack [29] at a perturbation budget of only  $\epsilon \leq 2$  (i.e. maximum change in each pixel can be  $2/255$ ). **(d) Label leakage:** supervised AT suffers from label leakage [23] which allows the model to overfit on perturbations thus affecting model generalization to unseen adversaries [50].

In comparison to AT, input processing methods [14, 45] for adversarial defense are scalable and can work across different tasks. However, they have been broken in white-box

settings [2] and shown to be least effective in black-box settings. For example, [10] successfully transfer their attack against multiple input processing based defenses even when the backbone architecture is adversarially trained using [43]. Furthermore, input transformations (e.g., Gaussian smoothing and JPEG compression) can maximize the attack strength instead of minimizing it [32, 10].

Motivated by the complementary strengths of AT and input processing methods, we propose a self-supervised AT mechanism in the input space. Our approach (Fig. 1) uses a min-max (saddle point) formulation to learn an optimal input processing function that enhances model robustness. In this way, our optimization rule implicitly performs AT. The main advantage of our approach is its generalization ability, once trained on a dataset, it can be applied off-the-shelf to safeguard a completely different model. This makes it a more attractive solution compared to popular AT approaches that are computationally expensive (and thus less scalable to large-scale datasets). Furthermore, in comparison to previous pre-processing based defenses that are found to be vulnerable towards recent attacks, our defense demonstrates better robustness. Our main contributions are:

- **Task Generalizability:** To ensure a task independent AT mechanism, we propose to adversarially train a purifying model named Neural Representation Purifier (NRP). Once trained, NRP can be deployed to safeguard across different tasks, e.g., classification, detection and segmentation, without any additional training (Sec. 3).
- **Self-Supervision:** The supervisory signal used for AT should be self-supervised to make it independent of label space. To this end, we propose an algorithm to train NRP on adversaries found in the feature space in random directions to avoid any label leakage (Sec. 3.1).
- **Defense against strong perturbations:** Attacks are continuously evolving. In order for NRP to generalize, it should be trained on worst-case perturbations that are transferable across different tasks. We propose to find highly transferable perceptual adversaries (Sec. 4.3).
- **Maintaining Accuracy:** A strong defense must concurrently maintain accuracy on the original data distribution. We propose to train the NRP with an additional discriminator to bring adversarial examples close to original samples by recovering the fine texture details (Sec. 4.2).

## 2. Related Work

**Defenses:** A major class of adversarial defenses processes the input images to achieve robustness against adversarial patterns. For example, [14] used JPEG compression to remove high-frequency components that are less important to human vision using discrete cosine transform. A compressed sensing approach called Total Variation Minimization (TVM) was proposed in [14] to remove the small localized changes caused by adversarial perturbations. Xie

*et al.* [46] introduced the process of Random Resizing and Padding (R&P) as a pre-processing step to mitigate the adversarial effect. A High-level representation Guided Denoiser (HGD) [26] framework was used as a pre-processing step to remove perturbations. NeurIPS 2017 Defense Competition Rank-3 (NeurIPS-r3) approach [42] introduced a two step pre-processing pipeline where the images first undergo a series of transformations (JPEG, rotation, zoom, shift and shear) and then passed through an ensemble of adversarially trained models to obtain the weighted output response as a prediction. [36] proposed to recover adversaries using GAN and [31] super-resolve images to minimize adversarial effect. As compared to the above defenses, we design an input processing model that derives a self-supervised signal from the deep feature space to adversarially train the defense model. Our results show significantly superior performance to all so-far developed input processing based defenses.

**Attacks:** The self-supervised perturbation signal obtained to adversarially train our proposed approach can also be used as an adversarial attack. Since the seminal work of Szegedy *et al.* [41], many adversarial attack algorithms [12, 13, 3, 9] have been proposed to show the vulnerability of neural networks against imperceptible changes to inputs. A single-step attack, called Fast Gradient Sign Method (FGSM), was proposed in [12]. In a follow-up work, Kurakin *et al.* [13] proposed a robust multi-step attack, called Iterative Fast Gradient Sign Method (I-FGSM) that iteratively searches the loss surface of a network under a given metric norm. To improve transferability, a variant of I-FGSM, called momentum iterative fast gradient sign method (MI-FGSM), was introduced [9], which significantly enhanced the transferability of untargeted attacks on ImageNet dataset [34] under a  $l_\infty$  norm budget. More recently, [47] proposed a data augmentation technique named input diversity method (DIM) to further boost the transferability of these attack methods. In contrast to our self-supervised attack approach, all of these methods are supervised adversarial attacks that rely on cross-entropy loss to find the deceptive gradient direction.

## 3. Neural Representation Purifier

Our defense aims to combine the benefits of adversarial training and input processing methods in a single framework that is computationally efficient, generalizable across different tasks and retains the clean image accuracy. The basic intuition behind our defense mechanism is to effectively use information contained in the feature space of deep networks to obtain an automatic supervisory signal. To this end, we design a Neural Representation Purifier (NRP) model that learns to clean adversarially perturbed images based on the automatically derived (self) supervision.

The objective is to recover the original benign image  $x$

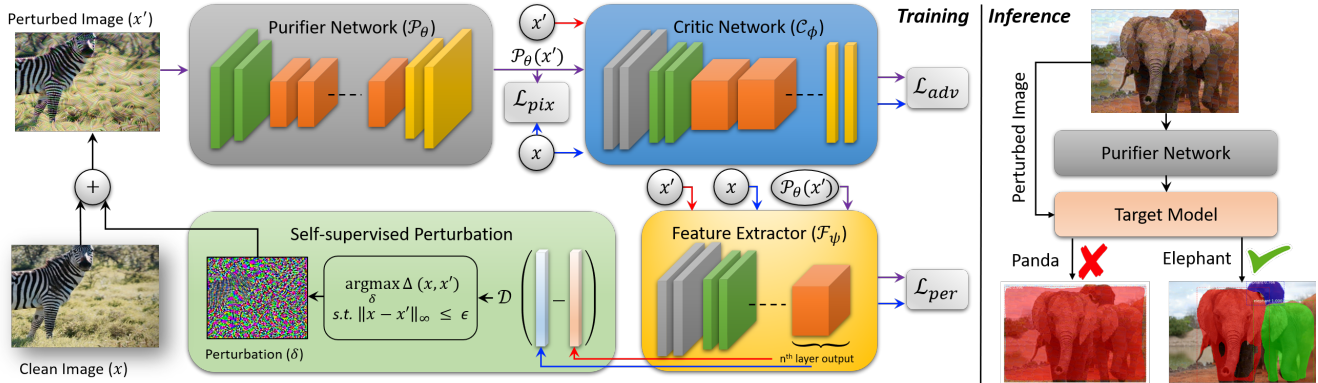


Figure 2: *Neural Representation Purifier*. Using a self-supervision signal, the proposed defense learns to purify perturbed images, such that their corresponding perceptual representation in deep feature space becomes close to clean natural images.

given an input adversarial image  $x'$ . We wish to remove the adversarial patterns by training a neural network  $\mathcal{P}_\theta$  parameterized by  $\theta$ , which we refer as the purifier network. The main objective is to be independent of the task-specific objective function, such that once trained, the proposed defense is transferable to other models (even across tasks). Towards this end, the network  $\mathcal{P}_\theta$  is trained in an adversarial manner by playing a game with the critic network  $\mathcal{C}_\phi$ , and a feature extractor  $\mathcal{F}_\psi$  (see Fig. 2). The function of the purifier and critic networks is similar to generator and discriminator in a traditional Generative Adversarial Network (GAN) framework, with the key difference that in our case,  $\mathcal{P}_\theta$  performs image restoration instead of image generation. The feature extractor,  $\mathcal{F}_\psi$ , is pretrained on ImageNet and remains fixed, while the other two networks are optimized during training. Adversarial examples  $x'$  are created by maximizing the  $\mathcal{F}_\psi$ 's response in random directions defined by a distance measure (Algorithm 1), while at minimization step,  $\mathcal{P}_\theta$  tries to recover the original sample  $x$  by minimizing the same distance (Algorithm 2).

### 3.1. Self-Supervision

The automatic supervision signal to train NRP defense is obtained via a loss-agnostic attack approach. Below, we first outline why such a Self-Supervised Perturbation (SSP) is needed and then describe our approach.

**Motivation:** Strong white-box attacks [13, 6], that are generally used for AT, consider already-known network parameters  $\theta$  and perturb the inputs to create  $x'$ , such that they are misclassified by the target model, *i.e.*  $\mathcal{T}(x'; \theta) \neq y$ . Since the perturbations are calculated using gradient directions specific to  $\theta$ , the resulting perturbed images  $x'$  do not generalize well to other networks [9, 38, 9, 47, 52]. This dependency limits these attacks to a specific network and task. In contrast, our goal is to design a self-supervised perturbation mechanism that can generalize across networks and tasks, thus enabling a transferable defense approach.

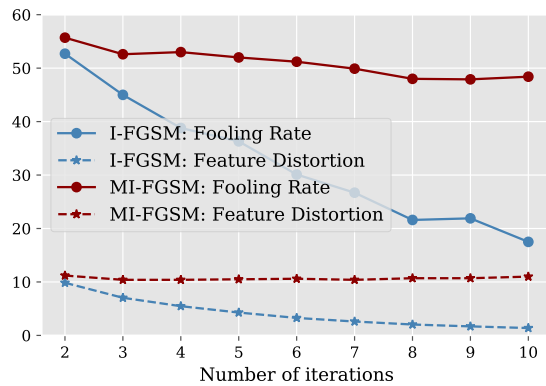


Figure 3: Fooling rate of Inc-v4 and average feature distortion is shown for adversaries generated on Inc-v3 (black-box setting) by I-FGSM and MI-FGSM. As the number of iterations increases, fooling rate of I-FGSM decreases along with its feature distortion while MI-FGSM maintains its distortion as iterations increase.

The self-supervised perturbation is based on the concept of ‘feature distortion’, introduced next.

**Feature Distortion:** Given a clean image  $x$  and its perturbed counterpart  $x'$  that is crafted to fool the target model  $\mathcal{T}(\cdot)$ , the feature distortion refers to the change that  $x'$  causes to the internal representations of a neural network  $\mathcal{F}(\cdot)$  relative to  $x$ . This can be represented by,

$$\Delta(x, x') = d(\mathcal{F}(x; \theta)|_n, \mathcal{F}(x'; \theta)|_n), \quad (1)$$

where,  $\mathcal{F}(x; \theta)|_n$  denotes the internal representation obtained from the  $n^{\text{th}}$  layer of a pretrained deep network  $\mathcal{F}(\cdot)$  and  $d(\cdot)$  is a distance metric which can be  $\ell_p$  [12], Wasserstein distance [1] or cosine similarity between the features of the original and perturbed sample.

The reason why we base our self-supervised perturbation on feature distortion is its direct impact on the perturbation transferability. To show this, we conduct a proof-of-concept experiment by generating adversarial examples

---

**Algorithm 1** SSP: Self-Supervised Perturbation

---

**Require:** A feature extractor  $\mathcal{F}_\psi$ , batch of clean samples  $\mathbf{x}$ , input transformation  $\mathcal{R}$ , perturbation budget  $\epsilon$ , step-size  $\kappa$ , and number of iterations  $T$ .

**Ensure:** Perturbed sample  $\mathbf{x}'$  with  $\|\mathbf{x}' - \mathbf{x}\|_\infty \leq \epsilon$ .

- 1:  $\mathbf{g}_0 = 0; \mathbf{x}' = \mathcal{R}(\mathbf{x});$
- 2: **for**  $t = 1$  to  $T$  **do**
- 3:   Forward pass  $\mathbf{x}'_t$  to  $\mathcal{F}_\psi$  and compute  $\Delta$  using Eq. 1;
- 4:   Compute gradients  $\mathbf{g}_t = \nabla_{\mathbf{x}} \Delta(\mathbf{x}_t, \mathbf{x}')$ ;
- 5:   Generate adversaries using;  
$$\mathbf{x}'_{t+1} = \mathbf{x}'_t + \kappa \cdot \text{sign}(\mathbf{g}_t); \quad (2)$$

- 6:   Project adversaries in the vicinity of  $\mathbf{x}$   
$$\mathbf{x}'_{t+1} = \text{clip}(\mathbf{x}'_{t+1}, \mathbf{x} - \epsilon, \mathbf{x} + \epsilon); \quad (3)$$

7: **end for**

8: **return**  $\mathbf{x}' = \mathbf{x}'_T$ .

---

on ImageNet-NeurIPS [7]. We consider two popular attack methods, MI-FGSM [9] and I-FGSM [13], among which MI-FGSM has higher transferability compared to I-FGSM. Interestingly, feature distortion strength of I-FGSM decreases as the number of attack iterations increases, compared to MI-FGSM (Fig. 3). MI-FGSM maintains its perturbation strength with increasing number of iterations. This indicates that feature distortion has a direct impact on transferability and therefore maximizing the objective in Eq. 1 (signifying feature-space distortion) can boost the transferability of adversarial examples without using any decision boundary information. Based on this observation, our proposed perturbation generation approach directly maximizes the distortion in deep feature space to create strong, highly generalizable and task-independent adversarial examples.

**Self-supervised Perturbation:** Conventional black-box attacks operate in the logit-space of deep networks. The objective of ‘logit-based’ adversarial attacks is to change the target model’s prediction for a clean image  $\mathcal{T}(\mathbf{x}) \neq \mathcal{T}(\mathbf{x}')$  such that  $\mathbf{x}'$  is bounded:  $\|\mathbf{x} - \mathbf{x}'\| \leq \epsilon$ . In contrast to these methods, we propose to find adversaries by maximizing the feature loss (Sec. 3.2) of neural networks. Our approach does not rely on decision-boundary information since our ‘representation-based’ attack directly perturbs the feature space by solving the following optimization problem:

$$\max_{\mathbf{x}'} \Delta(\mathbf{x}, \mathbf{x}') \text{ subject to: } \|\mathbf{x} - \mathbf{x}'\|_\infty \leq \epsilon, \quad (4)$$

Our proposed method to maximize feature distortion for a given input sample is summarized in Algorithm 1. We apply a transformation  $\mathcal{R}$  to input  $\mathbf{x}$  at the first iteration (Algorithm 1) to create a neural representation difference between an adversarial and benign example and then maximize the difference within a given perturbation budget. There can be different choices for  $\mathcal{R}$  but in this work,  $\mathcal{R}$  simply adds random noise to the input sample, *i.e.* our algorithm takes a random step at the first iteration.

---

**Algorithm 2** NRP: Neural Representation Purification via Self-Supervised Adversarial Training

---

**Require:** Training data  $\mathcal{D}$ , Purifier  $\mathcal{P}_\theta$ , feature extractor  $\mathcal{F}_\psi$ , critic network  $\mathcal{C}_\phi$ , perturbation budget  $\epsilon$  and loss criteria  $\mathcal{L}$ .

**Ensure:** Randomly initialize  $\mathcal{P}_\theta$  and  $\mathcal{C}_\phi$ .

- 1: **repeat**
  - 2:   Sample mini-batch of data,  $\mathbf{x}$ , from the training set.
  - 3:   Find adversaries,  $\mathbf{x}'$ , at a given perturbation budget  $\epsilon$  by maximizing distance,  $\Delta$  (Eq. 1), using Algorithm 1.
  - 4:   Forward-pass  $\mathbf{x}'$  through  $\mathcal{P}_\theta$  and calculate  $\mathcal{L}_{\mathcal{P}_\theta}$  (Eq. 8).
  - 5:   Back-pass and update  $\theta$  to minimize  $\mathcal{L}_{\mathcal{P}_\theta}$  (Eq. 8).
  - 6:   Update  $\mathcal{C}_\phi$  to classify  $\mathbf{x}$  from  $\mathcal{P}_\theta(\mathbf{x}')$ .
  - 7: **until**  $\mathcal{P}_\theta$  converges.
- 

### 3.2. NRP Loss functions

We propose a hybrid loss function that is used to train the purifier network (see Algorithm 2). This loss function consists of three terms that we explain below:

**Feature loss:** The Self-supervised Perturbation (SSP) generated by Algorithm 1 is the direct result of increasing the feature loss function,  $\Delta$ , defined on the feature extractor  $\mathcal{F}_\psi$ . In order to learn the purifier network, we must decrease this distance as follows:

$$\mathcal{L}_{feat} = \Delta(\mathcal{F}_\psi(\mathbf{x}), \mathcal{F}_\psi(\mathcal{P}_\theta(\mathbf{x}'))), \quad (5)$$

where,  $\Delta$  is formally defined in Eq. 1, and the distance measure used to compute  $\Delta$  is the mean absolute error (MAE). We empirically observe that removing  $\mathcal{L}_{feat}$  loss leads to a network that does not converge to a meaningful state and produces weaker defense (see Fig. 5).

**Pixel loss:** Smoothing images can help in mitigating the adversarial effect since the perturbation patterns resemble to that of noise. Therefore, in order to encourage smoothness, we apply  $l_2$  loss in the image pixel space,

$$\mathcal{L}_{img} = \|\mathcal{P}_\theta(\mathbf{x}') - \mathbf{x}\|_2. \quad (6)$$

**Adversarial loss:** Instead of using vanilla GAN objective, we use relativistic average GAN which has shown better convergence properties [20, 32]. For a given batch of original,  $\mathbf{x}$ , and adversarial examples,  $\mathbf{x}'$ , the relativistic loss for the purifier network  $\mathcal{P}_\theta$  is given as:

$$\mathcal{L}_{adv} = -\log(\sigma(\mathcal{C}_\phi(\mathcal{P}_\theta(\mathbf{x}')) - \mathcal{C}_\phi(\mathbf{x}))), \quad (7)$$

where  $\sigma$  represents the sigmoid layer. The overall loss objective for  $\mathcal{P}_\theta$  is the combination of losses defined on pixel and feature spaces as well as the relativistic loss:

$$\mathcal{L}_{\mathcal{P}_\theta} = \underbrace{\alpha \cdot \mathcal{L}_{adv}}_{\text{Adversarial loss}} + \underbrace{\gamma \cdot \mathcal{L}_{img}}_{\text{Pixel loss}} + \underbrace{\lambda \cdot \mathcal{L}_{feat}}_{\text{Feature loss}}. \quad (8)$$

The pixel and feature losses focus on restoring image content and style, while adversarial loss restores texture details.

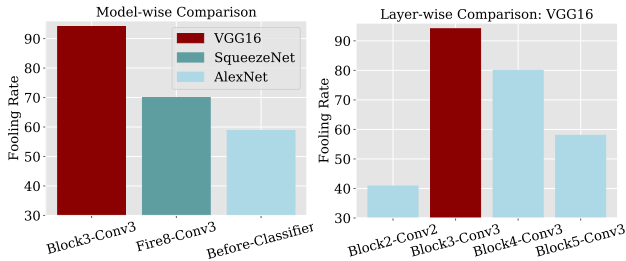


Figure 4: Fooling rate for Inc-v3 [23] on ImageNet-NeurIPS dataset. Adversaries are created by applying SSP (Algorithm 1) at different layers and best results for each model is selected. Perceptual adversaries found in VGG space has the highest transferability (further analysis is in supplementary material).

### 3.3. NRP Architecture

Here, we outline the architecture of generator, feature extractor and discriminator blocks. **Generator** ( $\mathcal{P}_\theta$ ): Our generator architecture is inspired by [24, 44]. It consists of a convolution layer followed by multiple “basic blocks”. Each basic block is composed of 3 “dense blocks” and each dense block contains five convolutional layers followed by leaky-relu [48] and finally a convolutional layer that has output with same dimension as input. Generally, adding a skip connection from input to generator’s output helps in restoration tasks e.g., image super resolution [24] and deblurring [22]. However, in our case an important design criteria is to avoid such skip connection since our objective is to remove adversarial noise and a direct skip connection can potentially reintroduce harmful noise patterns. **Feature Extractor** ( $\mathcal{F}_\psi$ ): It is a VGG [37] network pretrained on ImageNet. During training,  $\mathcal{F}_\psi$  remains fixed while its response is maximized in random directions (adversary generation process) and minimized (purification process) using a predefined distance metric. In our experiments, we demonstrate the effectiveness of VGG space for creating strong adversaries as compared to other deep architectures. **Discriminator** ( $\mathcal{C}_\phi$ ): Our discriminator architecture is also based on VGG network [37]. It consists of five convolutional blocks containing convolutional layers followed by batch-norm and leaky-relu and then a fully connected layer.

### 3.4. On Suitable Perceptual Adversaries

The intuition to train NRP on boundary-agnostic perceptual adversaries is based on the extensive study [51] that found correlation of deep features with human perception. Specifically, [51] compares three models *i.e.* VGG [37], AlexNet [21] and SqueezeNet [19]. Following [51], we study these models from adversarial perspective by applying feature distortion at different layers in Fig. 4. Our findings are as follows: (a) VGG’s perceptual adversaries are more transferable than AlexNet and SqueezeNet (a detailed transferability analysis on seen/unseen perturbations of VGG is in supplementary material), (b) under same fea-

ture distortion settings, adversaries found at different layers are not equally transferable *e.g.* conv3.3 (block 3, layer 3) features offer better adversarial transferability than the rest of the network. We believe this is because the initial VGG layers learn low-level features while the deeper ones become too specific to the label space. Further, we found that increasing the representation loss at multiple network layers does not notably increase attack success rate and adds a significant computational overhead. Since NRP training process is agnostic to the label-space of the source model *i.e.*, it neither depends on a particular task-specific loss function (e.g., cross entropy) nor on the ground-truth labels, this makes it a generic algorithm, which can defend a totally unseen model. Furthermore, we demonstrate that perturbations discovered with our SSP approach offer high transferability across models trained on different datasets and tasks.

## 4. Experiments

### 4.1. Training Details

Training is done on randomly selected 25k images from MS-COCO data set. These images are resized to  $480 \times 480 \times 3$ . Adversaries created using SSP are fed as inputs to NRP with their corresponding clean images used as target labels. During training, we randomly crop images of  $128 \times 128 \times 3$ . Batch size is set to 16 and training is done on four Tesla v100 GPUs. Learning rates for generator and discriminator are set to  $10^{-4}$ , with the value of  $\alpha = 5 \times 10^{-3}$ ,  $\gamma = 1 \times 10^{-2}$  and  $\lambda = 1$ . We study eight models trained on the ImageNet [34]. Five of these models are naturally trained. These include Inceptionv3 (Inc-v3) [40], Inceptionv4 (Inc-v4), Inception Resnet v2 (IncRes-v2) [39], Resnet v2-152 (Res-152) [18] and VGG-19 [37]. The other three models including Adv-v3 [23], Inc-v3<sub>ens3</sub> and IncRes-v2<sub>ens</sub> [43] are adversarially trained. The specific details about these models can be found in [23, 43].

### 4.2. Defense Results and Insights

(a) **Generalizability Across Attacks:** Figs. 6, 7 & 8 demonstrate generalization ability of NRP to recover images from strong adversarial noise. Quantitative analysis in Table 1 shows that compared to previously broken defenses [10], NRP achieves strong robustness against state-of-the-art attacks [47, 10], bringing down the effectiveness of the **ensemble** translation-invariant attack with input diversity ( $\text{DIM}_{TI}$ ) [10] from 79.8% to 31.9%.

(b) **NRP as Cross-task Defense:** In order to measure the cross-task defense capabilities, we deploy NRP against cross-domain attack (CDA) [32], a state-of-the-art attack that generates diverse cross-domain adversarial perturbations. Results in Table 2 demonstrate that NRP successfully removes all unseen perturbations and proves a generic cross-task defense for classification, object detection and in-

Table 1: Robustness of different defense methods against state-of-the-art black-box attacks (*lower is better*). IncRes-v2<sub>ens</sub> is used as backbone model following [10]. NRP significantly reduces the attack success rate. Adversaries ( $\epsilon \leq 16$ ) are created against Inc-v3, Inc-v4, IncRes-v2, Res-v2-152 and Ensemble.

	Defenses	Attacks					
		FGSM	FGSM <sub>T/I</sub>	MIFGSM	MIFGSM <sub>T/I</sub>	DIM	DIM <sub>T/I</sub>
Inc-v3	JPEG [15]	19.9	25.5	20.3	28.2	30.7	37.0
	TVM [15]	18.8	30.7	19.4	34.9	24.4	44.2
	NIPS-r3 [42]	9.8	24.5	12.9	26.7	18.0	41.4
	R&P [45]	6.5	19.8	8.7	23.9	13.3	36.8
	HGD [25]	<b>2.1</b>	18.4	6.9	25.7	9.7	38.3
	APE-GAN [36]	19.6	28.0	17.9	30.4	23.6	38.6
	SR [31]	23.0	36.7	23.6	38.3	32.5	49.0
	NRP	3.2	<b>4.8</b>	<b>4.5</b>	<b>9.1</b>	<b>5.1</b>	<b>11.0</b>
Inc-v4	JPEG [15]	21.8	27.9	26.0	31.6	38.6	43.5
	TVM [15]	19.9	31.8	24.8	38.4	29.1	45.6
	NIPS-r3 [42]	11.5	24.6	15.6	29.5	14.1	41.9
	R&P [45]	7.9	21.6	12.1	28.0	17.2	39.3
	HGD [25]	<b>2.6</b>	18.1	9.6	27.8	32.4	58.7
	APE-GAN [36]	21.1	28.8	20.7	32.8	25.0	39.0
	SR [31]	25.3	34.1	29.2	42.3	39.3	52.3
	NRP	3.1	<b>4.4</b>	<b>4.8</b>	<b>10.3</b>	<b>5.2</b>	<b>12.5</b>
IncRes-v2	JPEG [15]	24.7	32.4	31.6	45.9	47.2	55.7
	TVM [15]	23.4	38.5	34.4	55.4	41.7	66.2
	NIPS-r3 [42]	13.3	31.4	22.7	46.2	37.6	61.5
	R&P [45]	9.9	28.1	18.6	45.2	30.2	61.4
	HGD [25]	3.9	25.4	19.6	45.1	32.4	58.7
	APE-GAN [36]	24.7	36.8	30.4	50.5	36.3	60.5
	SR [31]	27.6	42.4	42.6	62.1	54.3	72.2
	NRP	<b>3.5</b>	<b>6.9</b>	<b>7.6</b>	<b>18.7</b>	<b>7.5</b>	<b>20.8</b>
Res-v2-152	JPEG [15]	24.0	32.7	31.2	38.3	42.4	50.8
	TVM [15]	22.0	38.1	24.5	41.2	36.8	55.7
	NIPS-r3 [42]	12.5	30.1	18.0	34.4	34.4	52.9
	R&P [45]	8.6	27.4	14.6	31.1	26.4	50.4
	HGD [25]	3.6	24.4	15.1	31.8	32.6	51.8
	APE-GAN [36]	24.3	37.1	23.2	38.6	34.3	53.8
	SR [31]	26.3	41.8	30.2	49.2	48.4	63.9
	NRP	<b>3.4</b>	<b>6.5</b>	<b>5.8</b>	<b>11.9</b>	<b>6.3</b>	<b>17.8</b>
Ensemble	JPEG [15]	38.1	43.3	67.7	77.2	82.5	83.4
	TVM [15]	30.0	39.8	50.1	72.1	64.1	79.8
	NIPS-r3 [42]	19.8	33.9	43.9	71.4	63.7	83.1
	R&P [45]	13.8	31.2	32.8	68.3	51.7	81.4
	HGD [25]	4.9	29.9	38.6	73.3	57.7	82.6
	APE-GAN [36]	32.0	42.1	44.6	69.3	59.6	74.5
	SR [31]	38.1	45.8	65.2	79.9	79.3	84.9
	NRP	<b>3.7</b>	<b>7.9</b>	<b>10.1</b>	<b>27.8</b>	<b>11.4</b>	<b>31.9</b>

stance level segmentation against CDA.

(c) **Ablation:** Fig. 5 thoroughly investigates the impact of different training mechanisms in combination with our defense, and provides the following insights: (i) Relativistic GAN loss offers a more robust solution than vanilla GAN, (ii) NRP performance decreases slightly without pixel loss, (iii) NRP without feature loss loses supervisory signal defined by perceptual-space boundary, hence the generator

Table 2: NRP generalizability across different adversarial attacks. Classification model is defended against CDA trained against Inc-v3 while detection and segmentation models are defended against CDA trained against Res-v2-152 (*higher is better*). (q=quantity, w=weights, win=window size)

Classification: Defending IncRes-v2 <sub>ens</sub> [43] against CDA [32]							
Method	No Attack	ImageNet		Comics		Paintings	
		$l_\infty \leq 8$	$l_\infty \leq 16$	$l_\infty \leq 8$	$l_\infty \leq 16$	$l_\infty \leq 8$	$l_\infty \leq 16$
No Defense	<b>97.8</b>	83.0	30.9	94.0	56.6	71.6	23.7
JPEG (q=75)	97.6	74.9	18.6	90.1	42.6	68.0	18.0
JPEG (q=50)	96.2	74.2	19.0	90.1	43.4	66.0	19.2
JPEG (q=20)	94.1	73.4	21.7	87.0	51.3	62.7	18.8
TVM (w=10)	93.1	82.3	30.2	91.0	77.2	72.7	27.4
TVM (w=30)	96.0	81.1	27.3	93.4	66.4	70.6	24.1
MF (win=3)	95.4	77.3	27.7	92.4	66.8	65.0	22.1
NRP	95.6	<b>95.7</b>	<b>96.0</b>	<b>95.4</b>	<b>94.2</b>	<b>95.3</b>	<b>94.1</b>
Detection: Defending Mask-RCNN [16] against CDA [32]							
No Defense	<b>59.9</b>	35.2	8.1	40.5	16.8	41.7	14.8
JPEG (q=75)	57.6	41.3	11.9	41.6	19.4	44.5	18.3
JPEG (q=50)	54.6	41.7	14.5	39.5	18.5	47.7	19.9
JPEG (q=20)	39.7	30.7	15.1	28.2	14.7	30.5	15.3
TVM (w=10)	54.1	32.1	14.3	40.5	28.9	37.6	21.5
TVM (w=30)	58.0	39.9	10.1	46.8	21.0	45.4	17.2
MF (win=3)	54.7	32.1	9.0	41.1	20.4	37.6	15.2
NRP	54.4	<b>51.5</b>	<b>50.3</b>	<b>53.5</b>	<b>53.7</b>	<b>53.2</b>	<b>54.3</b>
Segmentation: Mask-RCNN [16] defense against CDA [32]							
No Defense	<b>56.8</b>	32.4	7.3	37.6	15.5	39.1	13.8
JPEG (q=75)	54.4	38.5	11	38.5	17.8	41.7	16.9
JPEG (q=50)	51.5	38.9	13.4	36.6	17.3	40	18.2
JPEG (q=20)	37.1	28.8	14.0	26.3	13.8	28.3	14.3
TVM (w=10)	50.8	29.8	13.2	37.6	26.6	34.9	19.8
TVM (w=30)	54.4	37.1	9.3	43.7	19.3	42.3	15.9
MF (win=3)	51.5	29.8	8.3	36.0	18.8	34.9	13.9
NRP	51.3	<b>48.4</b>	<b>47.3</b>	<b>50.3</b>	<b>50.8</b>	<b>50.2</b>	<b>51.4</b>

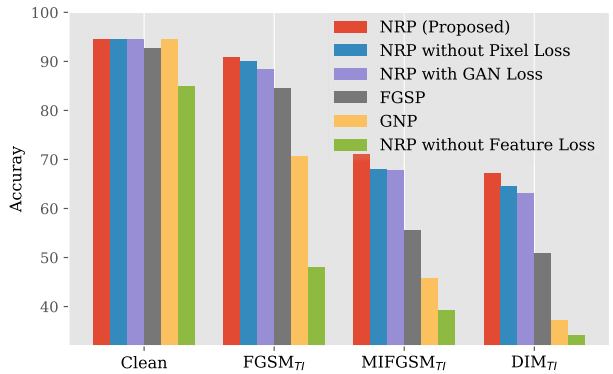


Figure 5: Ablation. Proposed NRP is able to recover input samples from the strong black-box ensemble attack [10] as compared to GNP and FGSP. NRP trained without  $\mathcal{L}_{feat}$  performs poorly indicating the importance of perceptual loss. Top-1 accuracy (*higher is better*) is reported for IncRes-v2<sub>ens</sub> [43] on ImageNet-NeurIPS.

does not converge to a meaningful state, (iv) Gaussian smoothing (Gaussian noise data augmentation) proves to be useful in reducing adversarial vulnerability of classifier [8, 49]. Training NRP as a Gaussian denoiser, named Gaus-

Table 3: Success rate (*lower is better*) of BPDA [6] and DIM<sub>TI</sub> [10] attacks against NRP. Res-v2-152 [18] is combined with other purifier networks (ResG [24], UNet [33]). Adversaries are then transferred to the naturally and adversarially trained models. NRP protects the backbone network even when the attacker tries to bypass using BPDA technique. (attack iterations: 10,  $\epsilon \leq 16$ )

Source	Attack	NRP	Inc-v3	Inc-v4	IncRes-v2	Adv-v3	Inc-v3 <sub>ens3</sub>	IncRes-v2 <sub>ens</sub>
Res-v2-152	DIM <sub>TI</sub>	✗	77.4	77.9	74.2	51.2	56.2	47.7
ResG $\oplus$ Res-v2-152	DIM <sub>TI</sub> $\oplus$ BPDA	✓	29.7	26.2	19.6	22.3	22.1	16.1
UNet $\oplus$ Res-v2-152	DIM <sub>TI</sub> $\oplus$ BPDA	✓	29.0	27.1	19.5	26.9	27.7	18.8

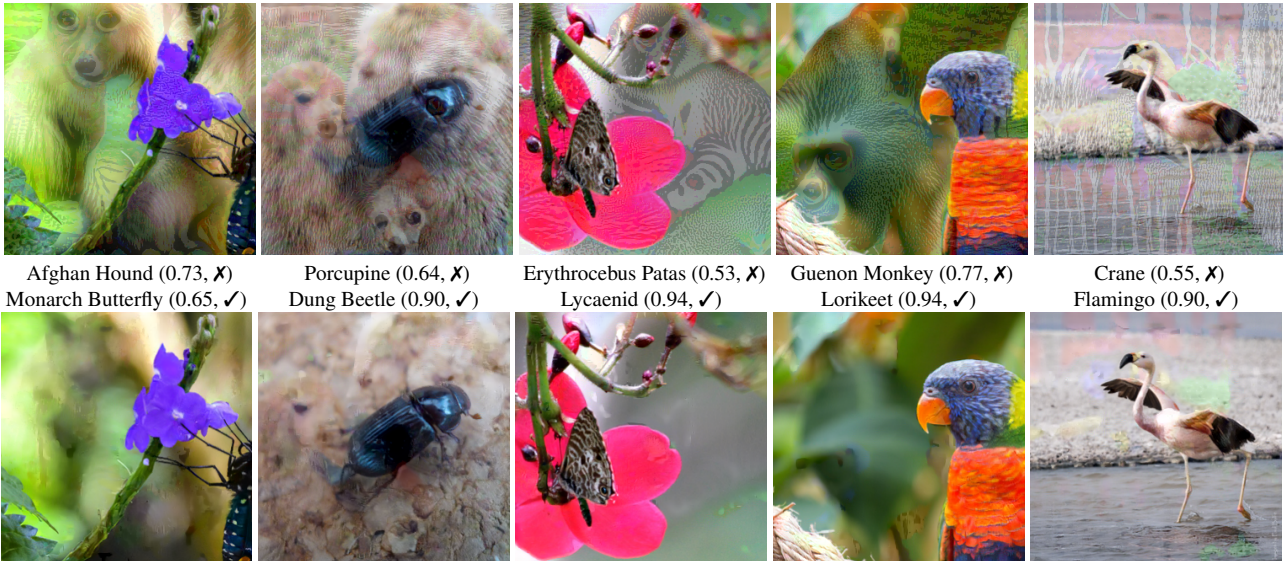


Figure 6: A visual illustration of NRP generalizability to different adversaries ( $\epsilon \leq 16$ ) (*top*: attacked; *bottom*: purified). Our method can clean challenging adversarial patterns resulting from SSP applied to adversarially robust model [11]. Previous denoising methods are not designed for this type of structured noise. IncRes-v2<sub>ens</sub> backbone is used here. (see supplementary material for more examples)

sian Noise Purifier (GNP) does not prove effective against translation-invariant attacks [10], and (v) Training NRP to stabilize FGSM adversaries (termed FGSP in Fig. 5) performs relatively better than GNP.

**(d) What if Attacker knows about the Defense:** We study this difficult scenario with the following criteria: **(i)** attacker knows that the defense is deployed and has access to its training data and training mechanism, and **(ii)** attacker trains a local defense similar to NRP, and then uses BPDA [6] to bypass the defense. To simulate this attack, we train residual generator (ResG) [24] and UNet [33] with the same training mechanism as described in Sec. 4.1. We then combine BPDA [2] with translation-invariant attack to bypass NRP. Under these challenging settings, NRP shows a relative gain of 74% and 66% respectively for IncRes-v2, IncRes-v2<sub>ens</sub> (see Table 3).

### 4.3. Self Supervised Perturbation as an Attack

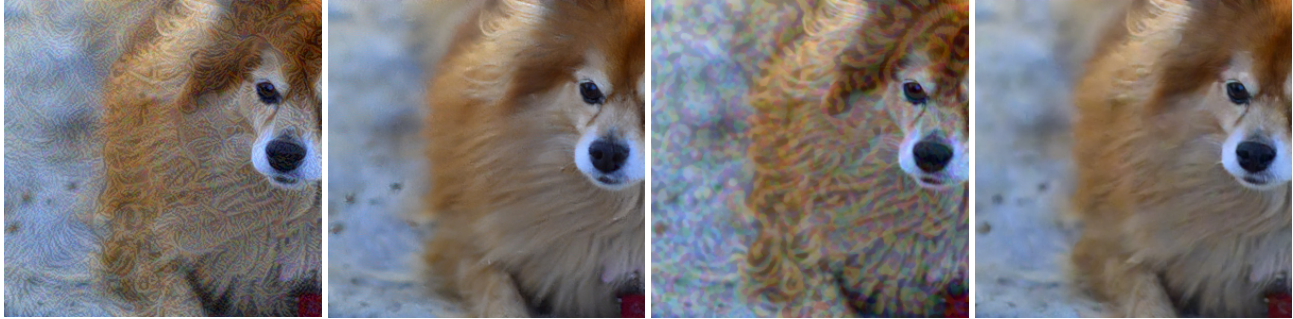
Next, we evaluate the strength of SSP as an attack for the tasks of classification, detection and segmentation.

**Classification:** Table 5 compares SSP with FGSM [12], R-FGSM [43], I-FGSM [13], MI-FGSM [9], TAP [52] and DIM [47] using their standard hyper-parameters (see sup-

Table 4: Cross-task SSP Attack: Pixel-level accuracy is shown for SegNet-Basic [4] on Camvid testset [5], while mAP (with IoU = 0.5) is reported for Mask-RCNN.

Problem	Method	No Attack	SSP ( $l_\infty \leq 8$ )	SSP ( $l_\infty \leq 16$ )
Semantic Seg.	SegNet [4]	79.70	52.48	32.59
Instance Seg.	Mask-RCNN [16]	56.8	29.4	8.8
Object Det.	RetinaNet [27]	53.78	22.75	5.16
	Mask-RCNN [16]	59.50	31.8	9.7

plementary material). The results in Table 5 provide the following insights. **(i)** SSP consistently demonstrates a strong black-box adversarial transferability on both naturally and adversarially trained models, bringing down top-1 accuracy of IncRes-v2 [39] from 100.0% to 14.1%, **(ii)** While MI-FGSM [9] and DIM [47] perform slightly better on adversarially trained ensemble models [43] in terms of top-1 accuracy, SSP shows comparable top-1 rate and surpasses in terms of top-5 accuracy, and **(iii)** These results indicate that decision-boundary based attacks flip the label of input sample to the near-by class category, while SSP being agnostic to decision-level information pushes the adversaries far from the original input category.



DIM [47]: Welsh Springer (0.52, ✗)    Purified: Pomeranian (0.88, ✓)    DIM<sub>TI</sub> [10]: Cocker (0.71, ✗)    Purified: Pomeranian (0.86, ✓)

Figure 7: NRP successfully recovers diverse patterns from strongest black-box attacks ( $l_\infty \leq 16$ ). IncRes-v2<sub>ens</sub> used as backbone.



CDA [32]: Adversarial    Prediction for Adversarial    Purified    Prediction for Purified

Figure 8: NRP successfully removes perturbation generated by CDA[32] ( $\epsilon \leq 16$ ) and stabilizes Mask-RCNN [16] predictions.

Table 5: SSP as an attack for Classification. Top-1 (T-1) and Top-5 (T-5) accuracies are reported under untargeted  $l_\infty$  adversarial attacks on ImageNet-NIPS with perturbation budget  $l_\infty \leq 16$ . ‘\*’ indicates white-box attacks.

Attack	Naturally Trained										Adv. Trained						
	Inc-v3		Inc-v4		Res-152		IncRes-v2		VGG-19		Adv-v3		Inc-v3 <sub>ens3</sub>		IncRes-v2 <sub>ens</sub>		
	T-1	T-5	T-1	T-5	T-1	T-5	T-1	T-5	T-1	T-5	T-1	T-5	T-1	T-5	T-1	T-5	
Res-152	FGSM [12]	55.1	81.1	62.6	85.1	18.9*	44.7*	65.0	86.5	43.9	70.4	64.6	85.8	76.9	93.5	87.9	98.2
	R-FGSM [43]	60.8	84.3	68.4	88.1	14.6*	40.3*	71.9	90.3	55.8	71.4	74.8	92.3	81.1	96.0	87.1	97.5
	I-FGSM [13]	80.9	96.7	85.3	97.8	0.9*	10.8*	93.1	98.8	75.9	94.8	89.2	99.2	90.5	97.9	94.6	99.5
	MI-FGSM [9]	38.9	72.7	44.8	76.5	0.6*	2.9*	47.7	79.6	42.1	71.8	67.0	89.9	69.4	93.3	81.5	96.4
	TAP [52]	48.2	-	55.7	-	7.6*	-	55.2	-	-	-	49.2	-	57.8	-	64.1	-
	DIM [47]	15.9	44.0	17.3	48.4	0.8*	3.0*	20.0	50.2	25.6	56.3	55.8	82.8	54.9	84.2	71.5	93.1
VGG16	FGSM [12]	32.6	58.6	38.4	62.6	38.5	66.3	44.5	68.5	8.8	25.1	51.7	75.3	54.9	81.7	70.8	90.7
	R-FGSM [43]	44.4	69.5	47.6	75.1	51.1	78.8	56.4	78.8	11.2	31.8	65.5	87.4	66.7	89.2	77.5	93.6
	I-FGSM [13]	69.2	93.0	75.2	93.7	79.0	96.2	85.6	96.8	14.4	49.3	83.5	97.7	83.9	96.7	92.1	98.8
	MI-FGSM [9]	20.4	45.0	19.7	43.2	25.2	53.8	26.8	53.8	1.5	12.1	43.0	70.9	42.0	72.7	62.0	86.8
	TAP [52]	23.9	-	28.1	-	23.9	-	32.3	-	-	-	38.8	-	41.9	-	63.8	-
	DIM [47]	14.7	38.8	16.6	39.0	21.0	48.0	21.5	45.7	<b>0.6</b>	7.6	35.8	65.8	<b>31.8</b>	60.8	<b>53.7</b>	79.5
	FFF [30]	61.7	80.7	60.8	78.7	72.8	90.1	76.1	90.1	44.0	68.0	79.6	93.1	83.1	93.1	92.8	98.5
	SSP	<b>5.3</b>	<b>11.0</b>	<b>5.9</b>	<b>11.9</b>	<b>16.5</b>	<b>29.5</b>	<b>14.1</b>	<b>25.5</b>	2.7	<b>6.8</b>	<b>25.9</b>	<b>43.2</b>	40.2	<b>58.3</b>	58.0	<b>75.0</b>

**Cross-task Adversarial Attack:** Since SSP is loss-agnostic, it enables attacks on altogether different tasks. Table 4 explores SSP for object detection and image segmentation. For Segmentation, the self-supervised perturbations created on CAMVID [5] in VGG-16 feature space are able to bring down the per pixel accuracy of Segnet-Basic by 47.11% within  $l_\infty \leq 16$ . For object detection, on MS-COCO validation set [28], mean Average Precision (mAP) with 0.5 intersection over union (IOU) of RetinaNet [27] and Mask-RCNN [16] drop from 53.78% to 5.16% and 59.5% to 9.7%, respectively, under  $l_\infty \leq 16$ .

## 5. Conclusion

We propose a novel defense approach that removes harmful perturbations using an adversarially trained purifier. Our defense does not require large training data and is independent of the label-space. It exhibits a high generalizability to the unseen state-of-the-art attacks and successfully defends a variety of tasks including classification, segmentation and object detection. Notably, our defense is able to remove structured noise patterns where an adversarial image is maliciously embedded into the original image.



## References

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. [3](#)
- [2] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018. [2](#), [7](#)
- [3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International Conference on Machine Learning (ICML)*, 2017. [2](#)
- [4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2481–2495, 2017. [7](#)
- [5] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. [7](#), [8](#)
- [6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. [3](#), [7](#)
- [7] NeurIPS Challenge. <https://www.kaggle.com/c/nips-2017-defense-against-adversarial-attack/data>. Kaggle, 2017. [4](#)
- [8] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019. [6](#)
- [9] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [2](#), [3](#), [4](#), [7](#), [8](#)
- [10] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. [2](#), [5](#), [6](#), [7](#), [8](#)
- [11] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations, 2019. [7](#)
- [12] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICRL)*, 2015. [2](#), [3](#), [7](#), [8](#)
- [13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICRL)*, 2017. [2](#), [3](#), [4](#), [7](#), [8](#)
- [14] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations (ICRL)*, 2017. [1](#), [2](#)
- [15] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. [6](#)
- [16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. [6](#), [7](#), [8](#)
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [1](#)
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. [5](#), [7](#)
- [19] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 1mb model size. *ArXiv*, abs/1602.07360, 2017. [5](#)
- [20] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018. [4](#)
- [21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60:84–90, 2012. [5](#)
- [22] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Ji Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [5](#)
- [23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. [1](#), [5](#)
- [24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. [5](#), [7](#)
- [25] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [6](#)
- [26] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Jun Zhu, and Xiaolin Hu. Defense against adversarial attacks using high-level representation guided denoiser. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2017. [2](#)
- [27] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. [7](#), [8](#)
- [28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. [8](#)
- [29] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learn-

- ing models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1
- [30] 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)*, 2017. 8
- [31] Aamir Mustafa, Salman H Khan, Munawar Hayat, Jianbing Shen, and Ling Shao. Image super-resolution as a defense against adversarial attacks. *arXiv preprint arXiv:1901.01677*, 2019. 2, 6
- [32] Muzammal Naseer, Salman H Khan, Harris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. *Advances in Neural Information Processing Systems*, 2019. 2, 4, 5, 6, 8
- [33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 7
- [34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 2, 5
- [35] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019. 1
- [36] Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. *arXiv preprint arXiv:1707.05474*, 2017. 2, 6
- [37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. 5
- [38] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy? – a comprehensive study on the robustness of 18 deep image classification models. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 644–661, Cham, 2018. Springer International Publishing. 3
- [39] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 5, 7
- [40] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 5
- [41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICRL)*, 2014. 2
- [42] Anil Thomas and Oguz Elibol. Defense against adversarial attacks-3rd place. <https://github.com/anlthms/nips-2017/blob/master/poster/defense.pdf>, 2017. 2, 6
- [43] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICRL)*, 2018. 2, 5, 6, 7, 8
- [44] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 5
- [45] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017. 1, 6
- [46] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 2
- [47] Cihang Xie, Zhishuai Zhang, Jianyu Wang, Yuyin Zhou, Zhou Ren, and Alan Yuille. Improving transferability of adversarial examples with input diversity. *arXiv preprint arXiv:1803.06978*, 2018. 2, 3, 5, 7, 8
- [48] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 5
- [49] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. *ArXiv*, abs/1707.06728, 2017. 6
- [50] Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. *arXiv preprint arXiv:1907.10764*, 2019. 1
- [51] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 5
- [52] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3, 7, 8