Identifying Attack-Specific Signatures in Adversarial Examples

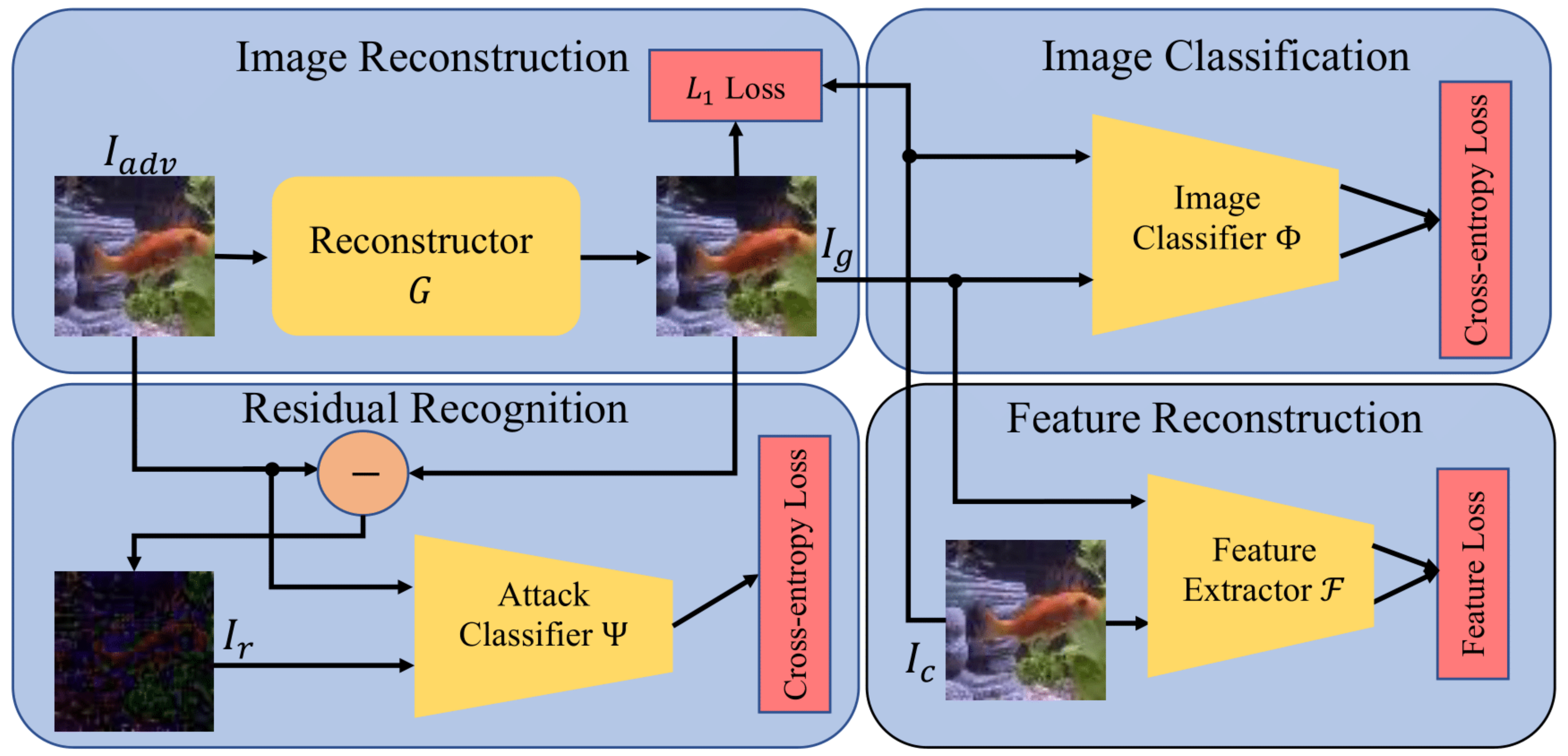
Hossein Souri*[1], Pirazh Khorramshahi*[1], Chun Pong Lau[1], Micah Goldblum[2], Rama Chellappa[1]

[1]Johns Hopkins University, [2]New York University

GitHub:

SCAN ME

Introduction

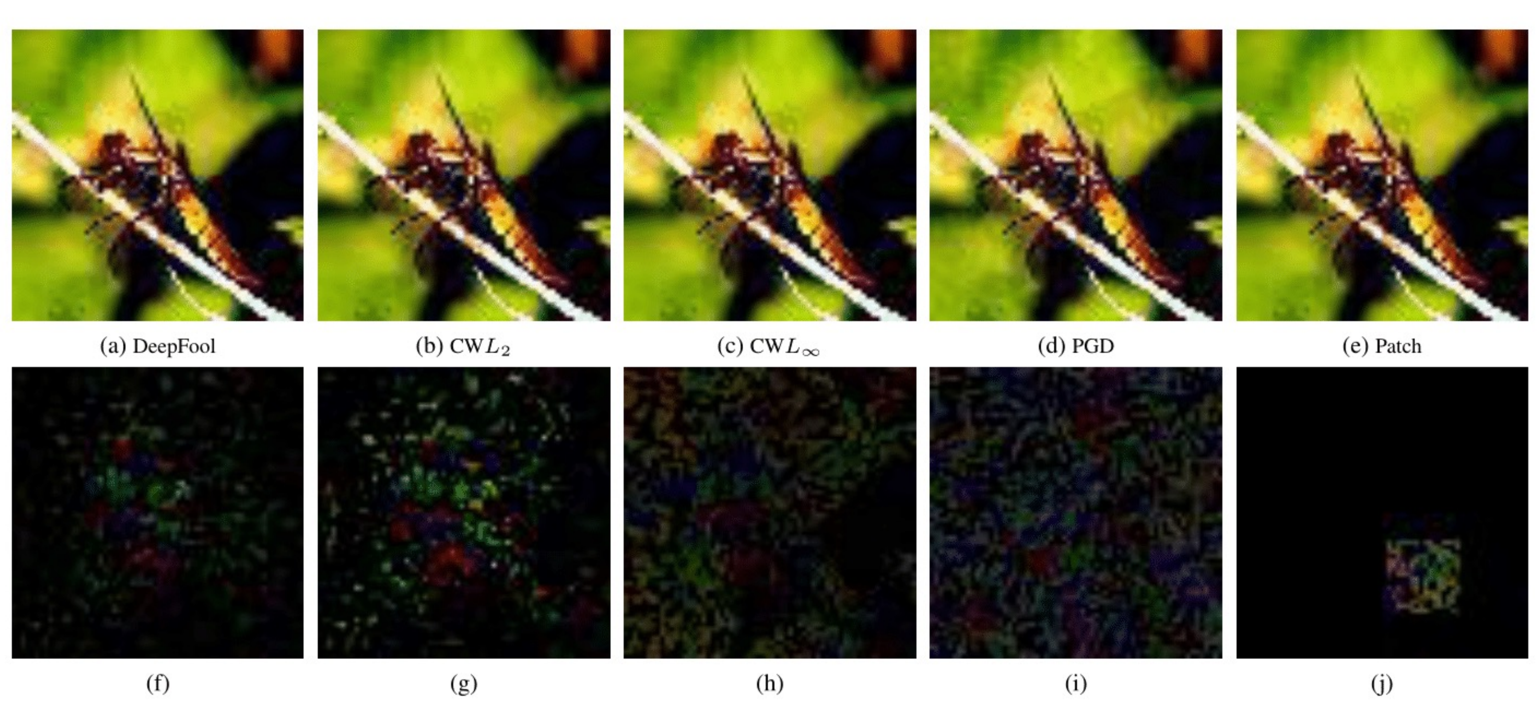


In this work, we examine the extent to which the precise attack algorithm used influences the adversarial examples it generates. To this end, we build a pipeline (***REDRL***) for classifying adversarial examples by the associated attack algorithm, finding that in fact different attacks generate unique examples.

Our contributions can be summarized as follows:

- We demonstrate that the perturbations generated by each attack algorithm have ***distinctive signatures***, facilitating the identification of the attack type.

- We propose an adversarial perturbation recovery framework*, **Reverse Engineering of Deceptions via Residual Learning (REDRL),*** to estimate the adversarial perturbations and to detect attack algorithm.



Adversarial samples (first row) and their respective perturbations (second row).

Method (REDRL)

- **Image Reconstruction:**

A reconstructed image I_g should lie close in pixel space to the clean image I_c that was used to generate the adversarial example:

$$\mathcal{L}_R(G) = \mathop{\mathbb{E}}_{I_c,\delta}\Big[|I_c - G(I_c+\delta)|_1\Big]$$

- **Feature Reconstruction:**

To encourage semantic similarity, the reconstructed image I_g should also lie close to the clean image I_c in feature space:

$$\mathcal{L}_F(G) = \mathop{\mathbb{E}}_{I_c,\delta}\Big[|\mathcal{F}(I_c) - \mathcal{F}(G(I_c+\delta))|_2\Big]$$

- **Image Classification:**

A pretrained image classifier Φ should yield similar classification scores on the reconstructed image I_g and the clean image I_c. This objective which can be framed in the context of Knowledge Distillation:

$$\mathcal{L}_{IC}(G) = \mathop{\mathbb{E}}_{I_c,\delta}\Big[-\log(\frac{e^{\Phi_i(G(I_c+\delta))}}{\sum_{j=1}^{C} e^{\Phi_j(G(I_c+\delta))}})\Big]$$

- **Residual Recognition:**

As an estimate of the adversarial perturbation, the residual image $I_r = I_{adv} - I_g$ along with the adversarial image I_{adv} is fed to the attack classification network Ψ to be classified into one of the adversarial attack algorithm classes.

$$\mathcal{L}_{AC}(G) = \mathop{\mathbb{E}}_{I_c,\delta}\Big[-\log(\frac{e^{\Psi_i(I_r,I_c+\delta)}}{\sum_{j=1}^{A} e^{\Psi_j(I_r,I_c+\delta)}})\Big]$$

- **End-To-End Training:**

The four stages of REDRL are trained simultaneously in an end-to-end fashion for the purpose of adversarial perturbation estimation and attack algorithm recognition:

$$\mathcal{L}_{total} = \min_G\Big[\mathcal{L}_{AC}(G) + \lambda_1\mathcal{L}_R(G) + \lambda_2\mathcal{L}_F(G) + \lambda_3\mathcal{L}_{IC}(G)\Big]$$

Experiments

Experimental Setup:

- In this study, we consider the CIFAR-10 and Tiny ImageNet datasets and the following candidate attacks: *PGD*, *DeepFool*, *CWL$_2$*, *CWL$_\infty$*, and *Adversarial Patch*. We use ResNet-50 , ResNeXt-50, DenseNet-121, and VGG-19 for image classifier Φ. For the attack classification network Ψ, we employ a ResNet-18 with label smoothing.

Attack Type	Configuration
DeepFool	Steps: 50
PGD	$\epsilon \in \{4, 8, 16\}$
	α : 0.01, Steps: 100
CWL_2	Steps: 1000, $c \in \{100, 1000\}$
	Learning Rate: 0.01, κ : 0
CWL_∞	Steps: 100, $\epsilon \in \{4, 8, 16\}$
	Learning Rate: 0.005, c : 5
Adversarial Patch	Steps: 100, $\epsilon \in \{4, 8, 16\}$
	Patch Size $\in \{4 \times 4, 8 \times 8, 16 \times 16\}$

Experimental Evaluation:

- Adversarial attack classification performance (%) based on adversarial images I_{adv}, ground-truth adversarial perturbations δ, and estimated residuals I_r, i.e., *REDRL*.

Class	Dataset					
	CIFAR-10 Input to Ψ			Tiny ImageNet Input to Ψ		
	I_{adv}	δ	I_r	I_{adv}	δ	I_r
Clean	12.0	100	100	62.5	99.9	99.7
PGD	73.5	99.9	99.9	88.7	99.7	99.9
DeepFool	56.2	99.9	97.4	53.2	64.0	75.3
CWL_2	73.4	98.6	96.6	28.0	96.4	66.3
CWL_∞	33.4	71.6	74.1	24.2	92.7	57.7
Patch	58.4	99.9	99.9	73.8	99.9	99.6
Total	57.5	94.2	94.2	59.4	95.7	85.5

Ablation Study:

A. We ignore FR and IC stages and only optimize network G for $L_R(G)$ and $L_{AC}(G)$
B. We add L_F so that network G is optimized on the $L_R(G)$, $L_F(G)$, and $L_{AC}(G)$ objectives.
C. We investigate the effect of image classification on the overall performance. Therefore, we optimize G on $L_R(G)$, $L_{IC}(G)$, and L_{AC}.

Class	Dataset							
	CIFAR-10				Tiny ImageNet			
	A	B	C	REDRL	A	B	C	REDRL
Clean	99.9	98.9	100	100	99.8	99.5	99.5	99.7
DeepFool	99.3	98.8	99.8	97.4	87.1	93.8	71.9	75.3
PGD	99.9	99.6	99.9	99.9	99.9	99.8	99.9	99.9
CWL_2	84.2	88.7	93.3	96.6	58.7	60.2	61.5	66.3
CWL_∞	63.3	70.8	71.6	74.1	42.9	43.0	53.8	57.7
Patch	99.7	99.8	99.9	99.9	98.6	98.9	99.2	99.6
Total	90.59	92.58	93.51	**94.28**	81.9	82.7	83.72	**85.57**