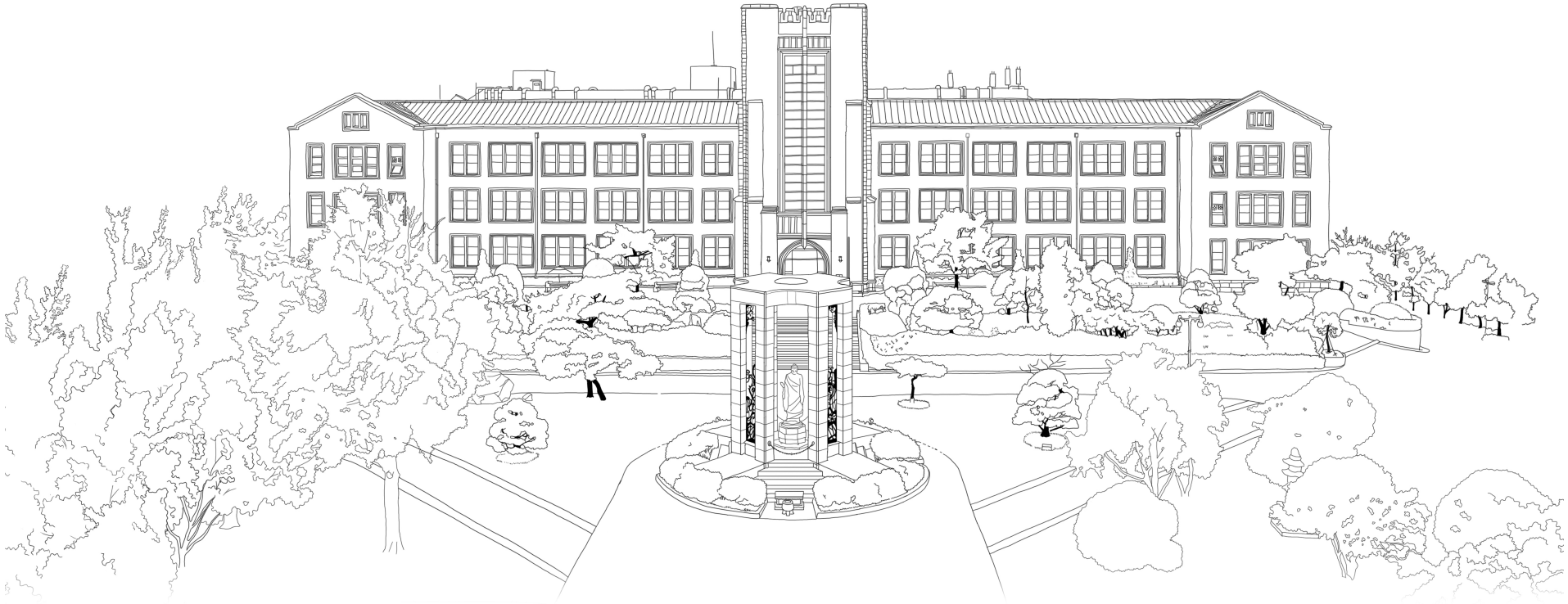


Feature Denoising for Improving Adversarial Robustness

C.Xie, Y.wu, L.maaten, AL.Yuille "Feature Denoising for Improving Adversarial Robustness" CVPR, 2019, pp. 501-509.

DongYeon Kim
Department of Multimedia Engineering
Dongguk University

BigDataLab
Dept. of Multimedia Engineering at Dongguk University

dongguk UNIVERSITY

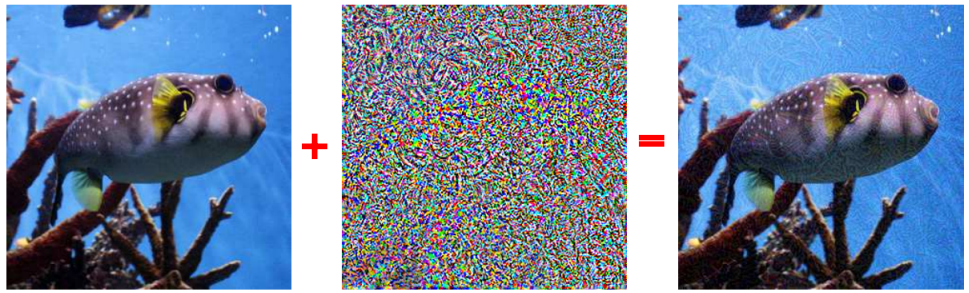
Contents

Introduction

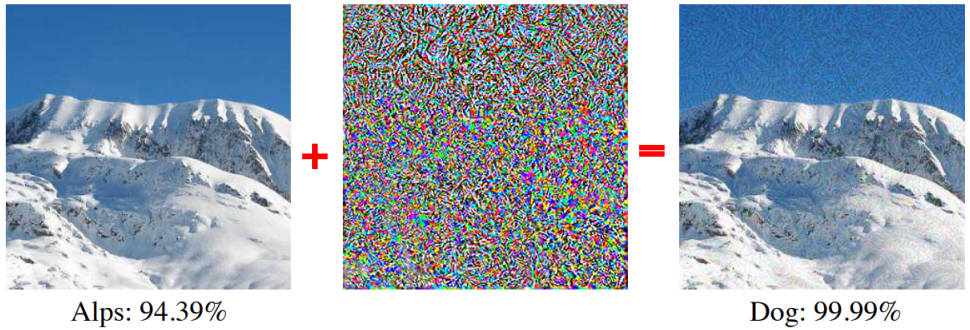
❖ Adversarial attack

- **Definition of adversarial attack**
  - ➢ add small perturbations to images that lead classification systems into making incorrect prediction
  - ➢ Perturbations are imperceptible or perceived as small noise in the image
  - ➢ effective against convolutional network

[1]



Original image perturbation Perturbed image



Original image perturbation Perturbed image

  - ➢ Demonstrate that networks perform computations that are dramatically different from those in human brain

[1] K. Dabo, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. TIP, 16(8):2080–2095, 2007..

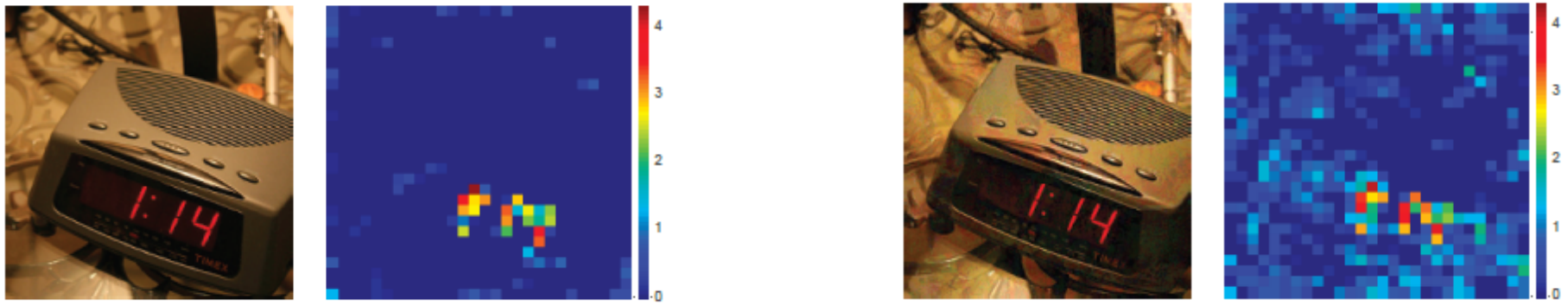
❖ Adversarial attack

- **Type of adversarial attack**
  - ➢ White-box attack
    - ✓ The attacker knows all the information about the network architecture (all parameters)
    - ✓ Attack the network based on the gradient of the network loss function (maximize loss)
    - ✓ Hard to defense
  - ➢ Black-box attack
    - ✓ The attacker attacks without knowing the information on the model
    - ✓ Only observe the network output on some inputs
    - ✓ Attack the network based on greedy local search on the actual gradient
    - ✓ Relatively easy to defense

Introduction

❖ Motivation

- Feature Map



➢ Randomly selected feature map of a ResNet architecture

➢ Clean image feature map (left image)

✓ features appear to focus primarily on semantically informative content in the image

➢ Adversarially perturbed counterpart (right image)

✓ Adversarial perturbations are small in the pixel space

✓ Perturbations are substantial noise in the feature maps of the network

➢ Features are activated across irrelevant regions

Introduction

❖ Proposed method

- Explore $\boldsymbol{feature\ denoising}$ approaches
  - ➢ To improve the robustness of convolutional networks
  - ➢ Defense from adversarial attacks

- Develop new convolutional network architectures
  - ➢ Add building blocks designed to denoise feature maps
  - ➢ Trained end-to-end on adversarially generated samples
  - ➢ Learn to reduce feature map perturbations

- For feature denoising, they use some denoising algorithms
  - ➢ Best performance is achieved by networks using $non-local\ means$ algorithm
  - ➢ Using $mean\ filters, median\ filters\ and\ bilateral\ filters$ also improves adversarial robustness

Related Works

❖ Adversarial training methods

- **Adversarial logit paring (ALP)** [2]
  - ➢ A type of adversarial training
  - ➢ Encourage the logit prediction of a network for a clean image
  - ➢ Weaken the logit prediction for the adversarial image

- **Use high-level features to guide the pixel denoiser** [3]
  - ➢ Denoising is applied directly on pixel

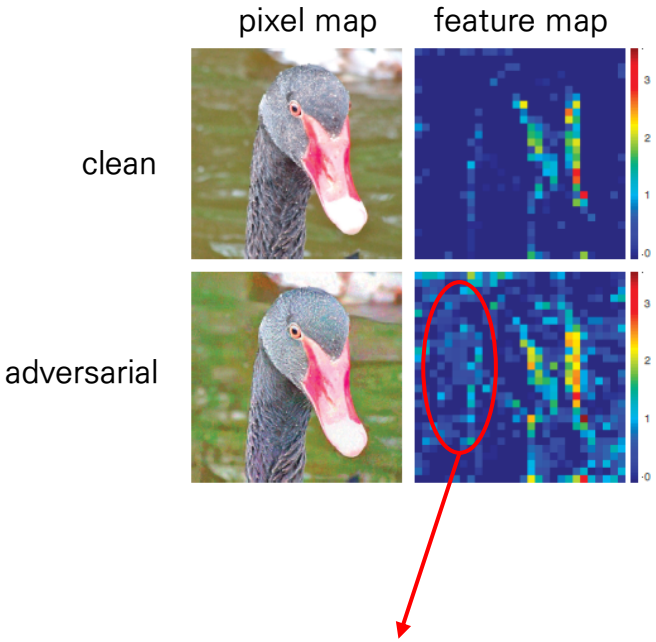
- **Transform the images via non-differentiable image preprocessing** [4]
  - ➢ using image quilting, total variance minimization, quantization

→ these defenses may be effective in black-box settings, can be circumvented in white-box settings
→ proposed models are able to improve adversarial robustness against strong white-box attacks

[2] H. Kannan, A. Kurakin, and I. Goodfellow. Adversarial logit pairing. arXiv preprint arXiv:1803.06373, 2018.
[3] F. Liao, M. Liang, Y. Dong, and T. Pang. Defense against adversarial attacks using high-level representation guided denoiser. In CVPR, 2018.
[4] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In ICLR, 2018.

❖ Feature Noise

- Adversarial image
  - ➢ definition
    - ✓ Noisy image that is created by adding perturbations to image
  - ➢ Perturbations are imperceptible by humans or perceived as very small noise



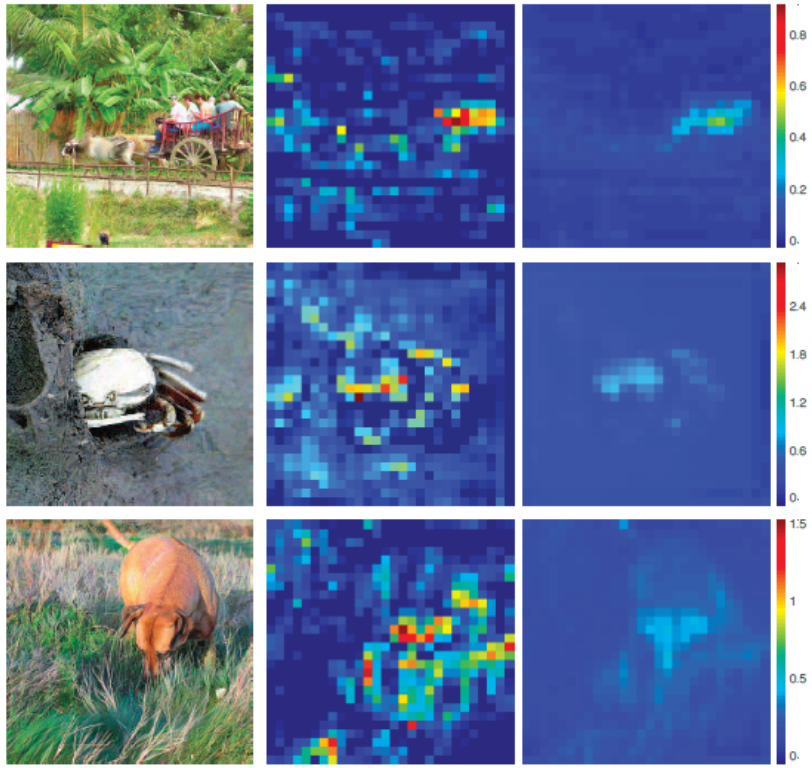
- ➢ Perturbations are constrained to be small at the *pixel level*
  - ✓ Can not recognize it visually
- ➢ Perturbations has a very important influence in feature map
  - ✓ Gradually increases as the image is propagated through the network
  - ✓ Non-existing activations in the feature maps are occurred

→ if the wrong activations overwhelm the true signal, the network makes wrong predictions

Proposed Method

❖ Feature Noise

- **Feature denoising**
  - ➢ The authors attempt to address this problem by feature denoising



- ➢ Using feature denoising operations
  - ✓ Successfully suppress the noise in the feature map
  - ✓ Make the responses focus on meaningful content
- ➢ Add a denoising block in the network + end-to-end train
  - ✓ Change the magnitudes and distributions of all features
  - ✓ Make the model reduce the noisr in feature map

Proposed Method

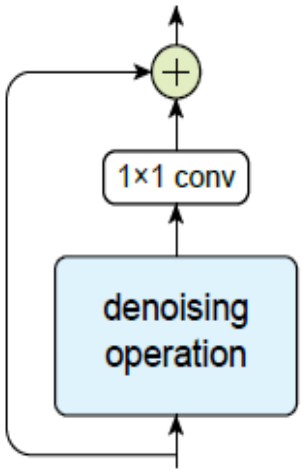
❖ Denoising Feature Maps

- Introduction
  - ➢ Add denoising blocks at intermediate layers of convolutional networks
    - ✓ Improve adversarial robustness
  - ➢ The denoising blocks are trained jointly with all layers of the network
    - ✓ in an end-to-end manner using adversarial training
    - ✓ Eliminate feature map noise generated by the attacker
  - ➢ The best-performed denoising blocks are inspired by non-local networks
    - ✓ Focus on the design of denoising blocks and denoising effect!

Proposed Method

❖ Denoising Feature Maps

- **Denoising Block**
  - ➢ The procedure in denoising block



A generic denoising block

The input to the block can be any feature layer in the convolutional network

⬇

The denoising block process the input features by a *denoising operation*

⬇

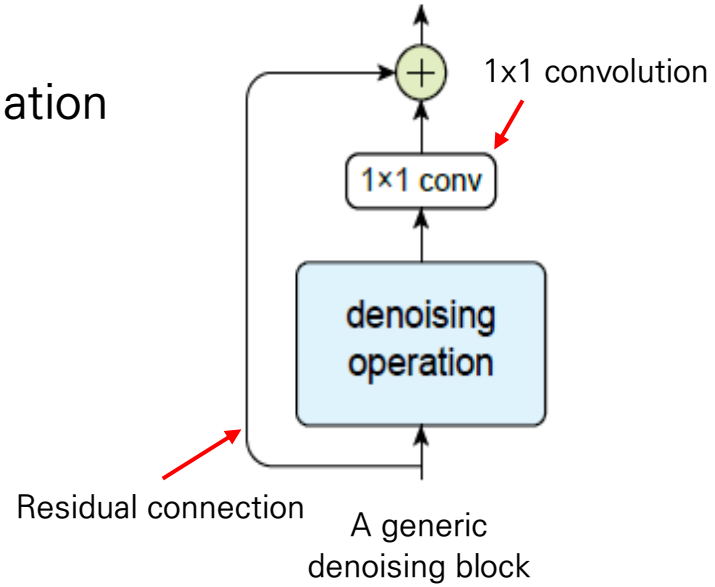
The denoised representation is first processed by 1x1 convolution

⬇

Added to the block's input via a residual connection

❖ Denoising Feature Maps

- **Denoising Block**
  - ➢ Denoising operation in denoising block is actually doing the denoising
  - ➢ 1x1 convolution & residual connection are mainly for feature combination
  - ➢ 1x1 convolution
    - ✓ Control the number of channels of denoised output
    - ✓ Adjust the tradeoff between removing noise and retaining signal
  - ➢ Residual connection
    - ✓ Various operations can suppress noise, but also impact signal
    - ✓ Add denoised output to the original input value
    - ✓ Can help the network to retain signals



❖ Denoising Feature Maps

- **Denoising Operations**
  - ➢ They experiment with four different instantiations of the denoising operation
  - ➢ Non-local means
    - ✓ Compute a denoised feature map y of an input feature map x by taking a weighted mean of features in all spatial location $\mathcal{L}$

$$y_i = \frac{1}{\mathcal{C}(x)} \sum_{\forall j \in \mathcal{L}} f(x_i, x_j) \cdot x_j,$$

- x_i : the center pixel where the calculation is computed
- x_j : other pixel in map x calculating the weight value with x_i
- $f(x_i, x_j)$: feature-dependent weighting function
- $\mathcal{C}(x)$: normalization function
- y_i : weighted mean value of pixel i

❖ Denoising Feature Maps

- **Denoising Operations**
  - ➢ Non-local means
    - ✓ Gaussian Weighting function : $f(x_i, x_j)$

$$y_i = \frac{1}{C(x)} \sum_{\forall j \in \mathcal{L}} f(x_i, x_j) \cdot x_j,$$

$$f(x_i, x_j) = e^{\frac{1}{\sqrt{d}} \theta(x_i)^T \phi(x_j)}$$

- d : the number of channels
- $\theta(x), \emptyset(x)$: the embedded versions of x
  - Each represent $W_\theta x, W_\emptyset x$
  - W : weight matrix
- $C(x)$: $\sum_{\forall j \in \mathcal{L}} f(x_i, x_j)$
- f/C : represent the $softmax$ function

→ equation $y_i = \frac{1}{C(x)} \sum_{\forall j \in \mathcal{L}} f(x_i, x_j) \cdot x_j,$ equal about $softmax\ function : softmax(x^T W_\theta^T W_\emptyset\ x)x$

$$S(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{j} e^{y_j}}$$

❖ Denoising Feature Maps

- **Denoising Operations**
  - ➢ Non-local means
    - ✓ Dot product Weighting function : $f(x_i, x_j)$

$$y_i = \frac{1}{C(x)} \sum_{\forall j \in \mathcal{L}} f(x_i, x_j) \cdot x_j,$$

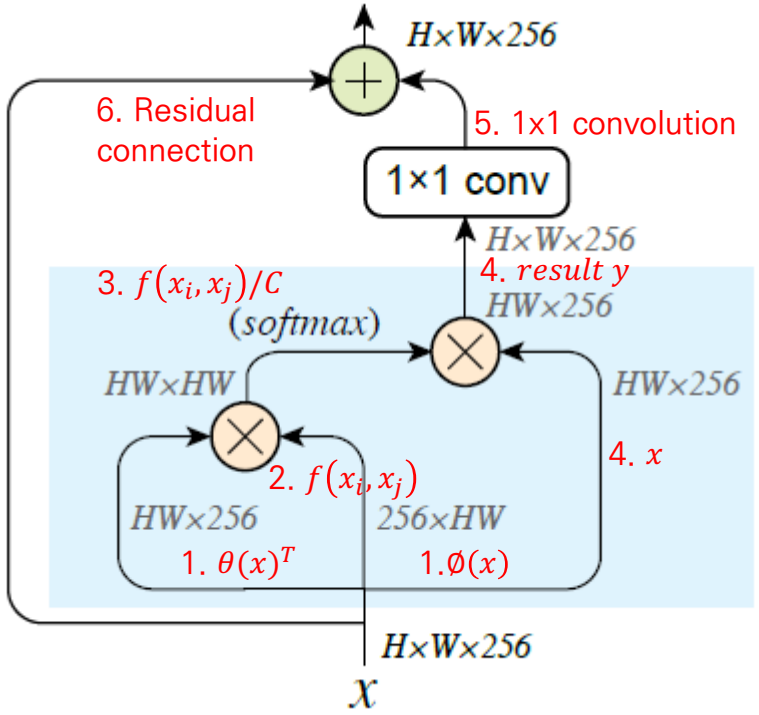
$$f(x_i, x_j) = x_i^{\mathrm{T}} x_j$$

- $C(x) : N$ – the number of pixels in x

- Unlike the Gaussian non-local means, the weights of the weighted mean don't sum up to 1
- Find that it is unnecessary to embed x in the dot-product version of non-local means
- No extra parameters are needed

❖ Denoising Feature Maps

- **Denoising Operations**
  - ➢ Non-local means : Revised denoising block
    - ✓ Use non-local means as a denoising operation



$$y_i = \frac{1}{\mathcal{C}(x)} \sum_{\forall j \in \mathcal{L}} f(x_i, x_j) \cdot x_j, \qquad f(x_i, x_j) = e^{\frac{1}{\sqrt{d}} \theta(x_i)^{\mathrm{T}} \phi(x_j)}$$

1) Compute the $\theta(x)$ and $\emptyset(x)$
2) Compute the weighting function $f(x_i, x_j) = e^{\frac{1}{\sqrt{d}} \theta(x_i)^{\mathrm{T}} \phi(x_j)}$
3) Compute f/C – represent $softmax$ function
4) Compute the result y
5) 1x1 convolution
6) Combine with original input based on residual connection

Proposed Method

❖ Denoising Feature Maps

- **Denoising Operations**
  - ➢ Bilateral filter
    - ✓ Change non-local means to local means

$$y_i = \frac{1}{C(x)} \sum_{\forall j \in \mathcal{L}} f(x_i, x_j) \cdot x_j, \quad \Rightarrow \quad y_i = \frac{1}{C(x)} \sum_{\forall j \in \Omega(i)} f(x_i, x_j) \cdot x_j.$$

    - ✓ It only change from all spatial location $\forall j \in \mathcal{L}$ to neighborhood(patch) $\Omega(i)$
    - ✓ $\Omega(i)$: local region around pixel i
  - ➢ Mean filter
    - ✓ Reduce noise but also smooth structure
    - ✓ Expect to perform worse than the weighted means
    - ✓ Still improve adversarial robustness
  - ➢ Median filter
    - ✓ Extract the median over a local region $\Omega(i)$

$$y_i = \text{median}\{\forall j \in \Omega(i) : x_j\},$$

    - ✓ Also can improve adversarial robustness

Proposed Method

❖ Adversarial Training

- introduction
  - ➢ Strong experimental results are driven by a successful implementation of $adversarial\ training$
  - ➢ The idea of adversarial training
    - ✓ Train the network on adversarially perturbed images
    - ✓ Perturbed images can be generated by a given white-box attacker -〉 know current parameters of the model
    - ✓ Use PGD attacker

- PGD attacker[5]
  - ➢ Access to the model gradients & has a copy of model's weights (white-box)
  - ➢ Attempt to find the perturbation that maximises the loss of a model on a particular input
    - ✓ Keep the maximum size of the perturbation (ϵ) – constraint expressed as L_∞ norm
    - ✓ Iterate to make perturbation
  - ➢ Using PGD in adversarial training, we can initialize the adversarial image by the clean image
  - ➢ Network is trained using clean image & perturbed image

[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In ICLR, 2018.

Experimental Results

❖ Introduction

- Evaluate feature denoising on the ImageNet classification dataset

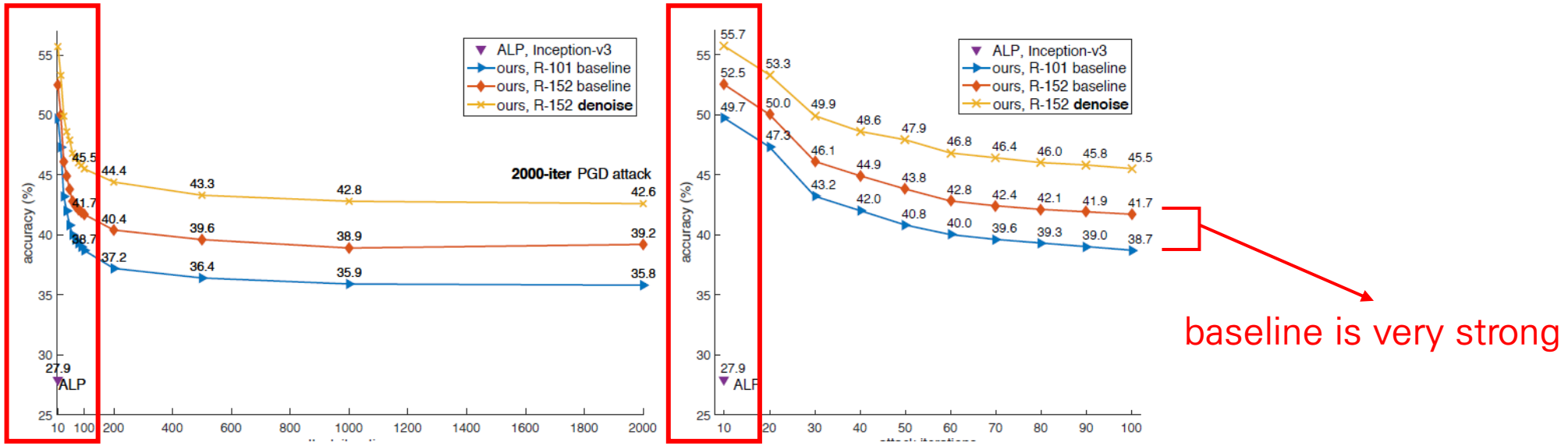
- Adversarial perturbation is considered under L_∞ norm
  - ✓ Allowed maximum value of ϵ to the each pixel

- Baselines are ResNet-101/152
  - ➢ Add 4 denoising blocks to a ResNet
  - ➢ Each is added after the last residual block of res_2, res_3, res_4, res_5 respectly
  - ➢ Models are trained using author's adversarial training implementation

Experimental Results

❖ Against White-box Attacks

- Conditions : Evaluate with $\epsilon = 16$, PGD attack iterations ranging from 10 to 2000
- Main result



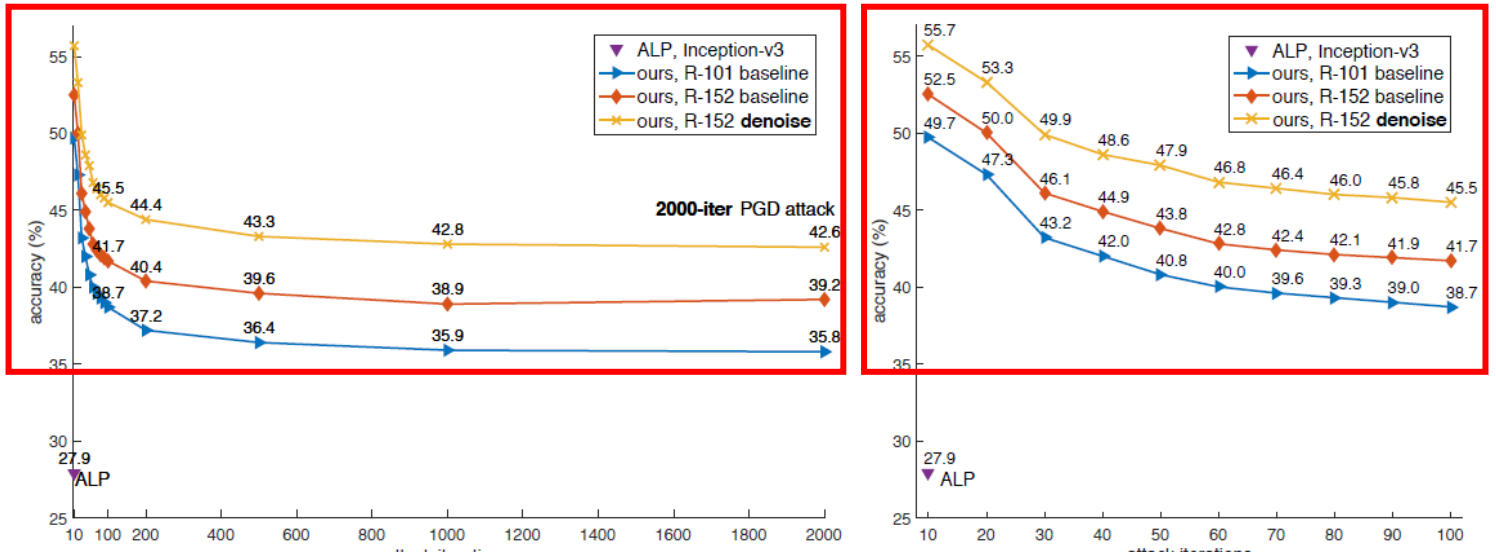
- ➢ Compare with previous state-of-the -art ALP
- ➢ Evaluated only under 10 iteration PGD attack
- ➢ Baseline models are considerably better than the ALP result

  → Shows that adversarial training system is solid

Experimental Results

❖ Against White-box Attacks

- Conditions : Evaluate with $\epsilon = 16$, PGD attack iterations ranging from 10 to 2000
- Main result

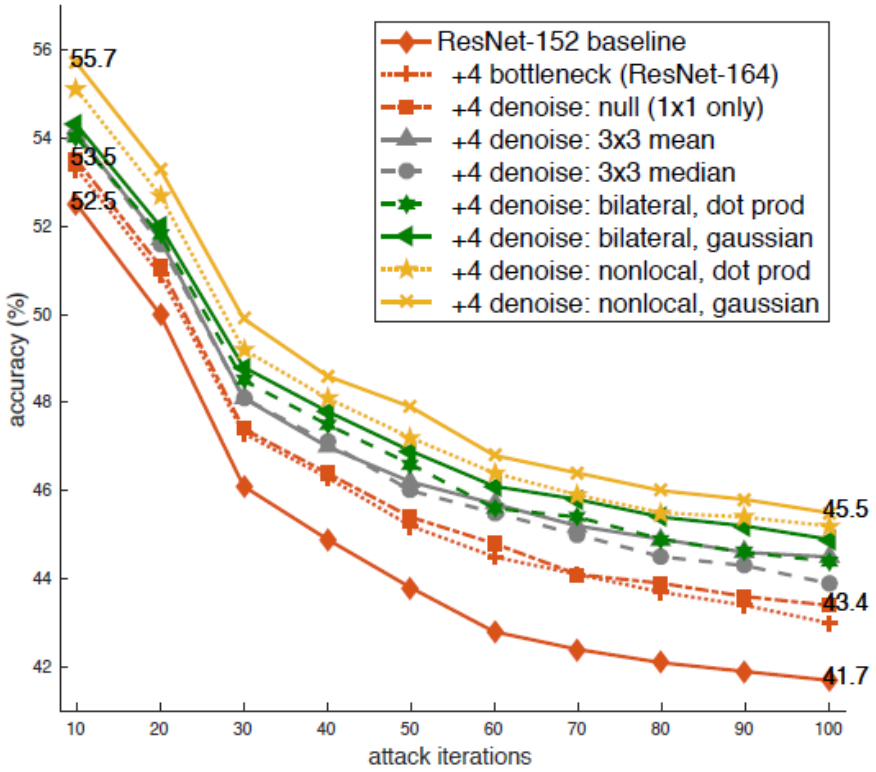


- ➢ Proposed methods are robust even under 2000-iteration PGD attacks
- ➢ The attacker performance diminished with 1000~2000 attack iterations
- ➢ Consistence performance improvement introduced by the denoising blocks

Experimental Results

❖ Against White-box Attacks

- Conditions : Evaluate with $\epsilon = 16$, PGD attack iterations ranging from 10 to 100
- Variants of denoising operations



- Add different kinds of blocks to baseline ResNet-152
  - ✓ $3\ x\ 3\ mean\ filtering$
  - ✓ $3\ x\ 3\ median\ filtering$
  - ✓ $3x3\ bilateral\ filtering$
  - ✓ $Non-local\ filtering$
- Add $null$ version of the denoising block
  - ✓ Residual block only contains a single 1x1 convolution

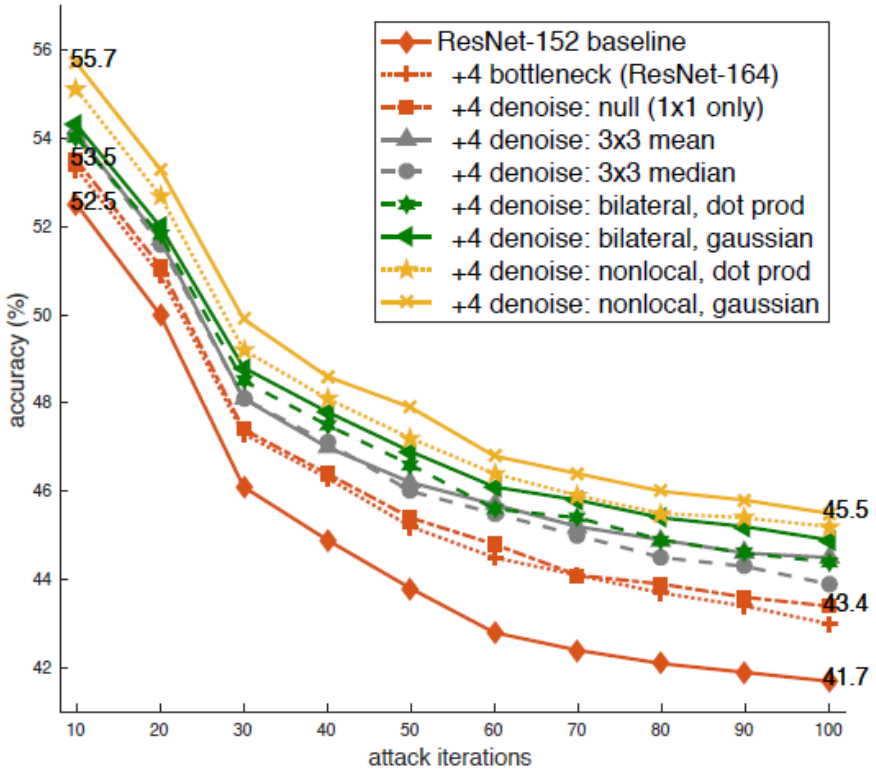
[6]

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.

Experimental Results

❖ Against White-box Attacks

- Conditions : Evaluate with $\epsilon = 16$, PGD attack iterations ranging from 10 to 100
- Variants of denoising operations

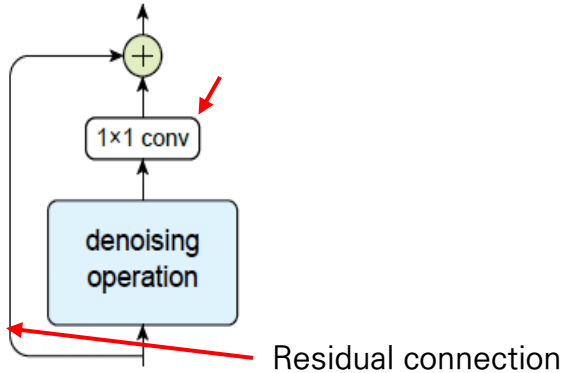


- All of the denoising operations have the better accuracy than 3 baseline model
  - ✓ Show that feature denoising appears to be a useful for adversarial robustness
- The best-performing model is the non-local(Gaussian) version

Experimental Results

❖ Against White-box Attacks

- Design decisions of the denoising block



➢ The denoising block has a 1x1 layer and a residual connection
  ✓ Both components do not perform denoising
  ✓ But important for the denoising blocks to work well

- The importance of 1x1 convolution & residual connection

attack iterations	10	100
non-local, Gaussian	55.7	45.5
removing 1×1	52.1	36.8
removing residual	NaN	NaN

➢ Removing the 1x1 convolution in the denoising block
  ✓ Accuracy drop considerably
➢ Removing the residual connection
  ✓ Makes training unstable
  ✓ Loss does not decrease in adversarial training

→ suggest that denoising features in itself is not sufficient
→ essential to properly combine the denoised features with the original features

Experimental Results

❖ Against Black-box Attacks

- Conditions
  - ➢ defense from the 5 $best\ attackers$ of the 2017 CAAD competition[1]
  - ➢ Maximum perturbation : $\epsilon = 32$, models are trained with $\epsilon = 16$
  - ➢ "All or nothing" : assume that correctly classified only if the model correctly classifies all adversarial image created by all attackers

model	accuracy (%)
CAAD 2017 winner	0.04
CAAD 2017 winner, under 3 attackers	13.4
ours, R-152 baseline	43.1
+4 denoise: null (1×1 only)	44.1
+4 denoise: non-local, dot product	46.2
+4 denoise: non-local, Gaussian	**46.4**
+all denoise: non-local, Gaussian	**49.5**

- ➢ CAAD winner has only 0.04% accuracy
- ➢ Suggest that a successful implementation of adversarial training is critical for adversarial robustness
- ➢ Add denoising blocks to all residual blocks

[1] K. Dabo, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. TIP, 16(8):2080-2095, 2007..

Experimental Results

❖ Denoising Blocks in Non-Adversarial Settings

- Non-adversarially trained models

model	accuracy (%)
R-152 baseline	78.91
R-152 baseline, run 2	+0.05
R-152 baseline, run 3	-0.04
+4 bottleneck (R-164)	+0.13
+4 denoise: null (1×1 only)	+0.15
+4 denoise: 3×3 mean filter	+0.01
+4 denoise: 3×3 median filter	-0.12
+4 denoise: bilateral, Gaussian	+0.15
+4 denoise: non-local, Gaussian	+0.17

- The networks are trained without adversarial training for the classification of clean images (R-152 baseline)
- Denoising blocks have no obvious advantage over the baseline R-152
  - ✓ All results are in the range of about ±0.2%

→ Denoising blocks could have special advantages in settings that require adversarial robustness

Conclusions

❖ Conclusions about this Paper

- Focus on the noisy appearance of feature maps from adversarial images
- Propose the potential of feature denoising for improving the adversarial robustness of convolutional networks
- They suggest the certain architecture designs (denoising blocks) that are particularly good for adversarial robustness
- When combined with adversarial training, these particular architecture designs are more appropriate for modeling the distribution of adversarial images

Thank You!