

Analyse Layer-wise feature of adversarial examples and their Texture bias to build defense

- ❖ Adversarial attack
- ❖ CNN is shape biased or texture biased
- ❖ Transferability of adversarial example
- ❖ Analyse Layer-wise feature of adversarial examples

Rated work : adversarial attack

❖ What is adversarial attack?

The simplest method——FGSM



x

“panda”

57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_x J(\theta, x, y))$

“nematode”

8.2% confidence

$=$



$x +$

$\epsilon \text{sign}(\nabla_x J(\theta, x, y))$

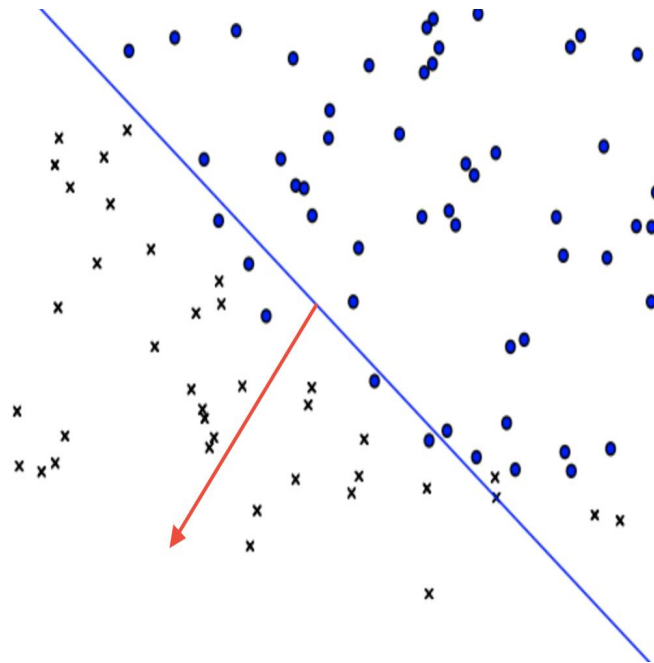
“gibbon”

99.3 % confidence

Rated work : adversarial attack

❖ Why lead prediction to a wrong class?

1. Getting parameter through SGD= minimize the loss, fit prediction to label
2. Then parameter of Neural network is fixed.
3. Add derivative to input, update input to maximize loss
4. loss increase = prediction mismatches labels



picture 3

Rated work : adversarial attack

❖ Target and Untarget attack

Untarget:

$$x^{adv} = x + \varepsilon \cdot \text{sign}(\nabla_x J(x, y_{true})),$$

where

x is the input (clean) image,

x^{adv} is the perturbed adversarial image,

J is the classification loss function,

y_{true} is true label for the input x .

Target:

$$x^{adv} = x - \varepsilon \cdot \text{sign}(\nabla_x J(x, y_{target})),$$

where

y_{target} is the target label for the adversarial attack.

Rated work : adversarial attack

❖ State-of-Art Attack Method:

- FGSM(Fast Gradient Sign Method)
- Iterative FGSM

$$x_0^{adv} = x, \quad x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(\nabla_x J(x_t^{adv}, y)).$$

- other optimization method to find the maxima of loss



picture2

Related work: CNN is base on texture or shape?



(a) Texture image

81.4%	Indian elephant
10.3%	indri
8.2%	black swan



(b) Content image

71.1%	tabby cat
17.3%	grey fox
3.3%	Siamese cat



(c) Texture-shape cue conflict

63.9%	Indian elephant
26.4%	indri
9.6%	black swan

Related work: CNN is base on texture or shape?

❖ Image trained CNN, but not human, exhibit a strong texture

human observers (red circles)

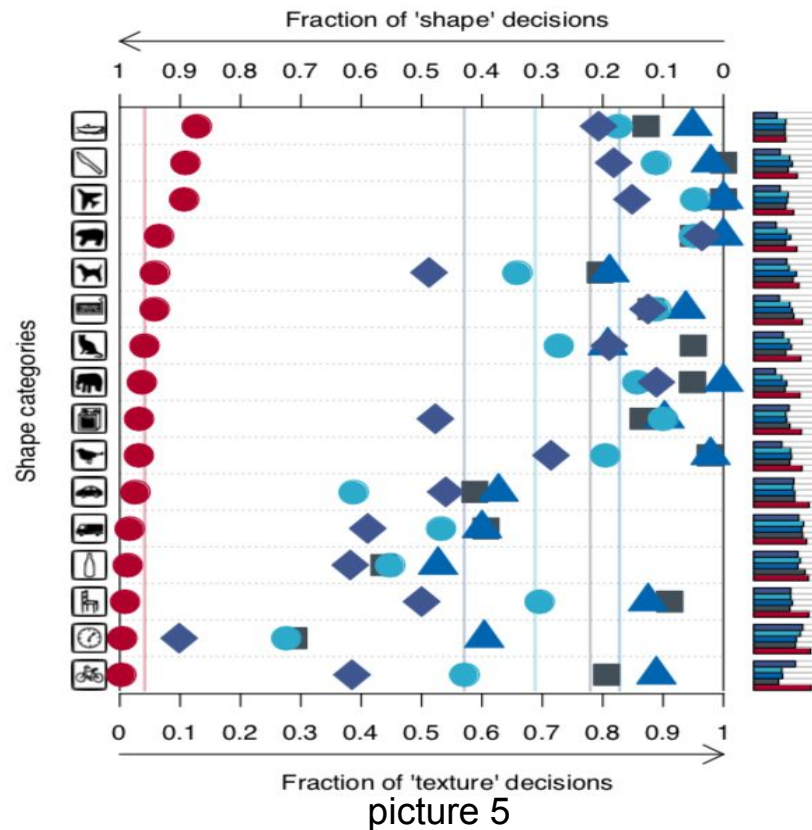
AlexNet (purple diamonds),

VGG-16 (blue triangles),

GoogLeNet (turquoise circles)

ResNet-50 (grey squares)

Notice that VGG-16 (blue triangles) is almost the most texture biased one!



Related work: the transferability of adversarial example

VGG get the best attack success rate

#iter = 10, untargeted, success rate

Source Model	AlexNet	1.0	0.8	0.7	0.8	0.6	0.8	0.8	0.8	0.7	1.0	1.0	0.9	0.7	0.7	0.7	0.8	1.0	0.9
	DenseNet-121-k32	0.8	1.0	1.0	1.0	0.6	0.9	0.8	0.7	0.7	0.9	0.9	0.9	0.6	0.6	0.7	0.7	0.9	0.9
	DenseNet-161-k48	0.8	1.0	1.0	1.0	0.6	0.9	0.8	0.8	0.8	0.8	0.9	0.9	0.7	0.7	0.6	0.7	0.9	0.9
	DenseNet-169-k32	0.8	1.0	1.0	1.0	0.6	0.9	0.8	0.7	0.7	0.9	0.9	0.8	0.6	0.6	0.6	0.7	0.9	0.9
	Inception-ResNet-v2	0.8	0.7	0.5	0.6	1.0	0.7	0.6	0.7	0.6	0.9	0.9	0.8	0.3	0.4	0.5	0.5	0.8	0.8
	Inception-v1	0.7	0.8	0.7	0.7	0.5	1.0	0.8	0.6	0.6	0.9	0.9	0.8	0.4	0.5	0.5	0.7	0.8	0.8
	Inception-v2	0.6	0.5	0.3	0.4	0.3	0.6	1.0	0.4	0.3	0.8	0.7	0.5	0.2	0.3	0.3	0.4	0.5	0.5
	Inception-v3	0.7	0.5	0.4	0.5	0.5	0.6	0.5	1.0	0.5	0.9	0.8	0.7	0.2	0.4	0.3	0.4	0.7	0.6
	Inception-v4	0.8	0.7	0.5	0.6	0.5	0.7	0.6	0.6	1.0	0.9	0.9	0.8	0.4	0.4	0.4	0.4	0.7	0.8
	MobileNet-0.25-128	0.8	0.4	0.2	0.3	0.3	0.5	0.4	0.4	0.3	1.0	1.0	0.6	0.2	0.4	0.3	0.5	0.7	0.7
	MobileNet-0.50-160	0.8	0.5	0.3	0.4	0.2	0.5	0.4	0.4	0.3	0.9	1.0	0.8	0.2	0.3	0.3	0.4	0.7	0.7
	MobileNet-1.0-224	0.8	0.7	0.5	0.6	0.5	0.8	0.6	0.6	0.5	0.9	1.0	1.0	0.4	0.5	0.5	0.6	0.8	0.8
	NASNet	0.9	0.8	0.6	0.7	0.6	0.8	0.7	0.7	0.7	0.9	0.9	0.8	1.0	0.6	0.6	0.7	0.9	0.9
	ResNet-v2-101	0.8	0.8	0.6	0.7	0.7	0.8	0.7	0.8	0.7	0.9	0.9	0.9	0.6	1.0	1.0	1.0	0.8	0.8
	ResNet-v2-152	0.8	0.8	0.6	0.7	0.6	0.8	0.7	0.7	0.6	0.9	0.9	0.9	0.5	1.0	1.0	1.0	0.8	0.8
	ResNet-v2-50	0.8	0.7	0.6	0.7	0.6	0.8	0.7	0.7	0.6	0.9	0.9	0.8	0.5	0.9	0.9	1.0	0.8	0.8
	VGG 16	1.0	1.0	1.0	1.0	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	1.0	1.0	1.0
	VGG 19	0.9	1.0	1.0	1.0	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	1.0	1.0

Hypothesis: adversarial example is texture biased

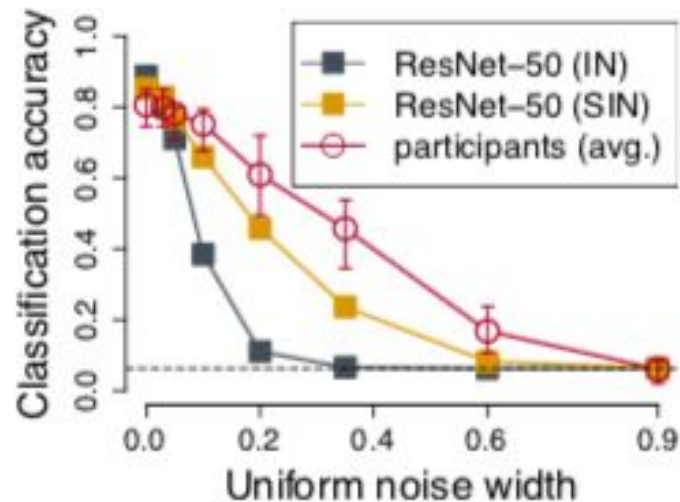
❖ How to prove?

1. shape biased CNN is more robust to noisy input
2. visionlizing CNN last conv layer feature

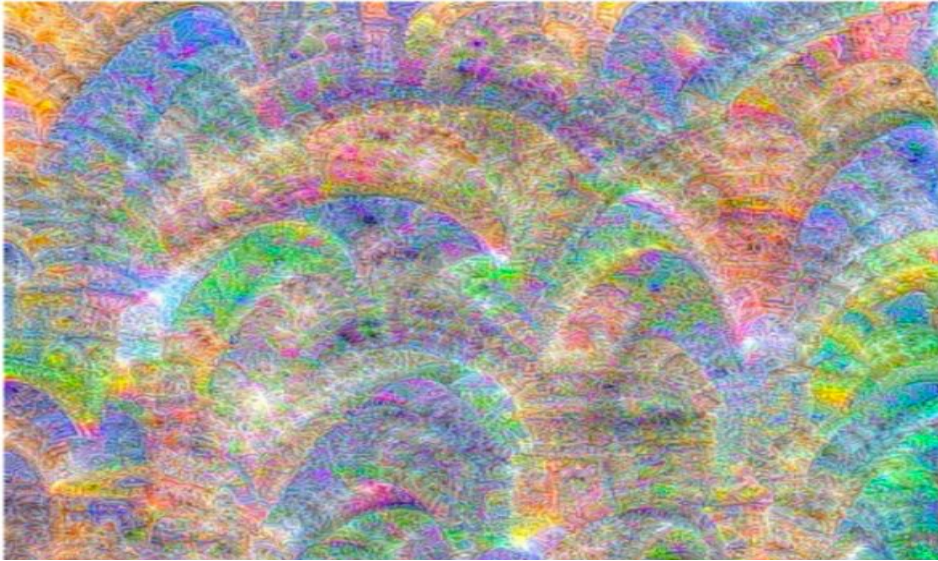
How to prove: 1. shape biased CNN is more robust to noisy input (previous others work)

SIN: texture transferred Imagenet (each class with diverse texture)

cue conflict (style transfer)

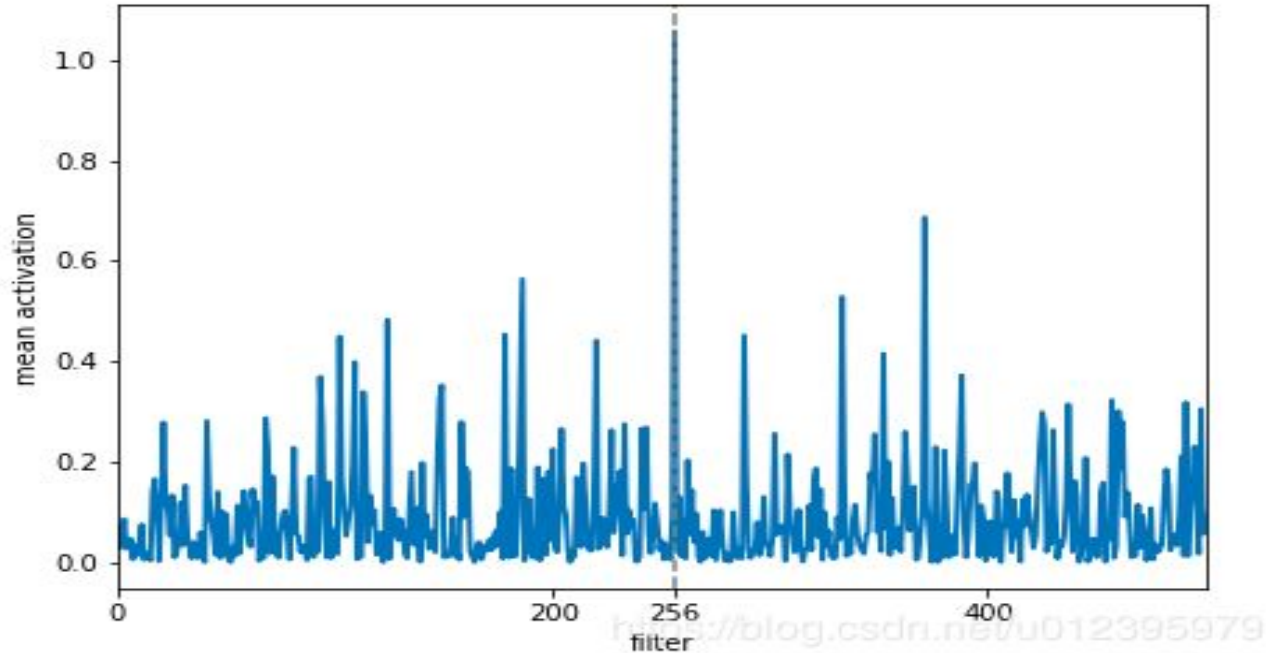


How to prove: 2.visionlizing CNN last conv layer
feature (previous others work)
visualizing the feature of 286 filter guess class



How to prove: 2.visionlizing CNN last conv layer feature

Activation
of filter



How to prove: 2. visionlizing CNN last conv layer feature

My Idea: (prospective experiment)

Visualizing the feature against adversarial examples

How does the misclassified high activation feature look like?

How to defend against adversarial example?

Idea:

1. Build more shape biased network.
2. detect and distinguish texture feature.
3. Election of the shape feature in inference

Citation

[1]Explaining and Harnessing Adversarial Examples

[2]ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness

[3] Is Robustness the Cost of Accuracy? – A Comprehensive Study on the Robustness of 18 Deep Image Classification Models