DOES ENHANCED SHAPE BIAS IMPROVE NEURAL NETWORK ROBUSTNESS TO COMMON CORRUPTIONS?

ICLR 2021

(Under Review)

이 정 수

DAVIAN

Data and Visual Analytics Lab

Previous work

IMAGENET-TRAINED CNNS ARE BIASED TOWARDS TEXTURE; INCREASING SHAPE BIAS IMPROVES ACCURACY AND ROBUSTNESS (ICLR 2019)

(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

- ImageNet-trained CNN has texture bias
- Increase shape bias via data augmentation with stylized images increase corruption robustness

[ImageNet-trained CNNs are biased towards texture; Increasing shape bias improves accuracy and robustness (ICLR2019)]

Previous work

Corruption robustness (ImageNet–C)

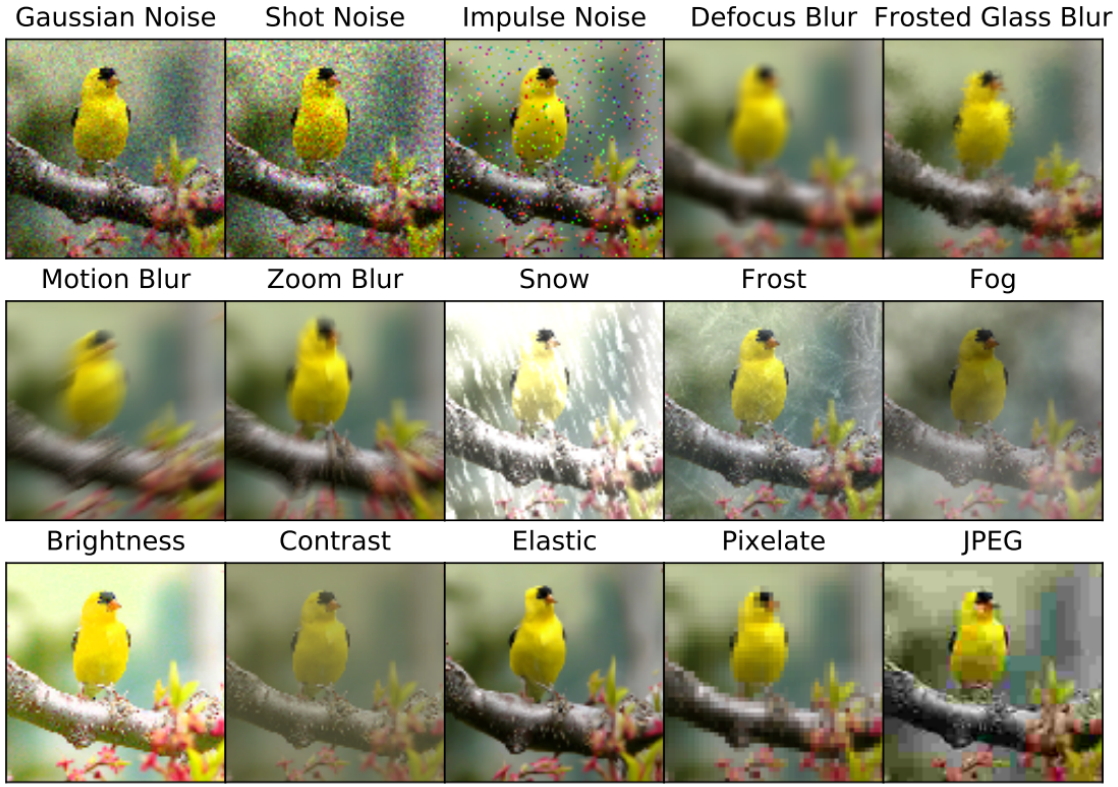


Figure 1: Our IMAGENET-C dataset consists of 15 types of algorithmically generated corruptions from noise, blur, weather, and digital categories. Each type of corruption has five levels of severity, resulting in 75 distinct corruptions. See different severity levels in Appendix B.

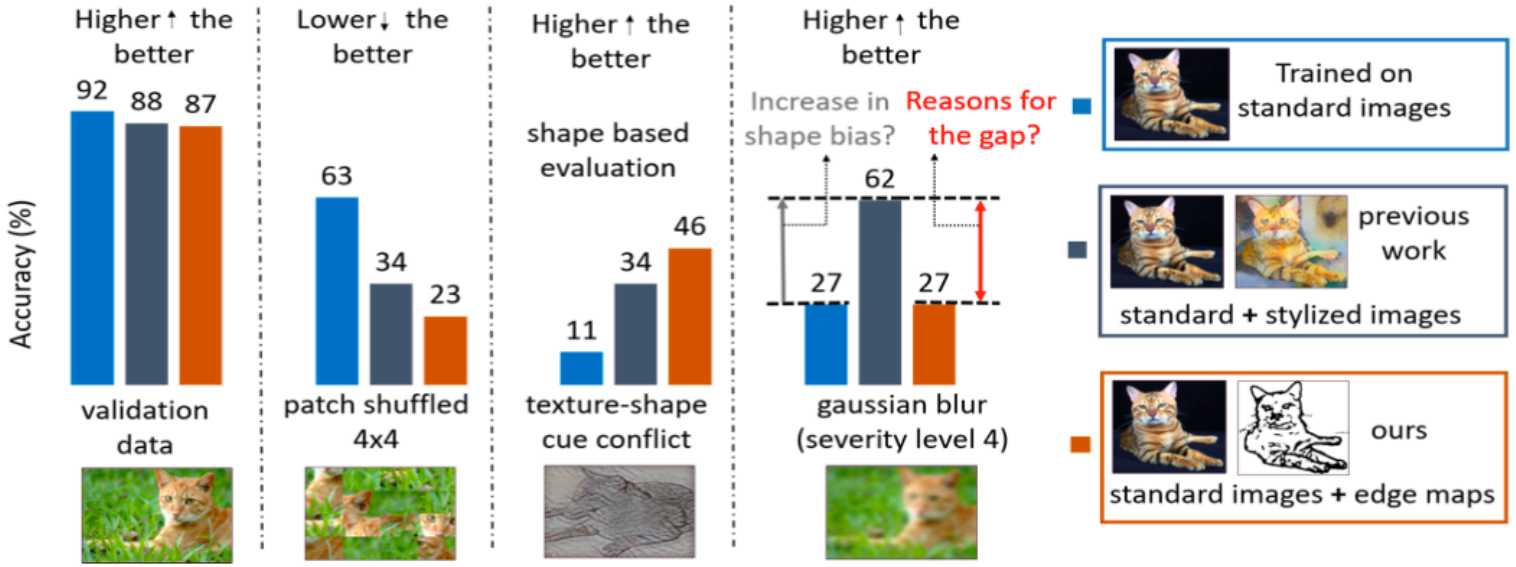
[Benchmarking neural network robustness to common corruptions and perturbations (ICLR2019)]

Claim

Increasing shape bias **does not help** in terms of corruption robustness

Previous work: data augmentation with stylized images -> increase shape bias -> increase corruption robustness
This work: data augmentation with stylized images ----------> increase corruption robustness

- Shape bias & increase robustness has correlation, not a causal relationship
- Increasing shape bias through a different method does not increase corruption robustness
- Image stylization encourages robust representation regardless whether the representations are shape-based or not



Method

2 methods to maximize shape bias

1. Style Randomization: inserted before every residual block (4 total)

$$\hat{X}_i := \hat{\sigma}_i * \left(\frac{X_i - \mu_i}{\sigma_i} \right) + \hat{\mu}_i$$

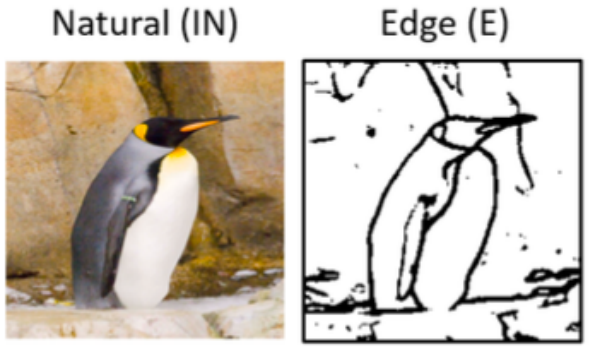
X_i : i[th] feature map of an intermediate layer

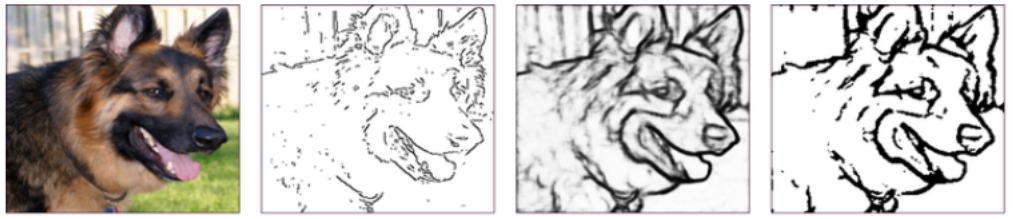
$\hat{\sigma}_i \sim \text{Uniform}(0.1, 1)$

$\hat{\mu}_i \sim \text{Uniform}(-1, 1)$

-> better generalize to out-of-distribution data

2. Edge map dataset





How edge map dataset is constructed

[Richer Convolutional Features for Edge Detection (CVPR2017)]

Dataset & Stylization Variants

IN:	ImageNet20
E:	Edge map of IN
SIN:	Style transfer (ADAIN) with “Painter by Numbers” as style image on IN
SE:	Style transfer (ADAIN) with “Painter by Numbers” as style image on Edge map
I-SIN:	Style transfer (ADAIN) with “IN” as style image on IN
I-SE:	Style transfer (ADAIN) with “IN” as style image on E
Superposition:	Interpolation of IN & SE

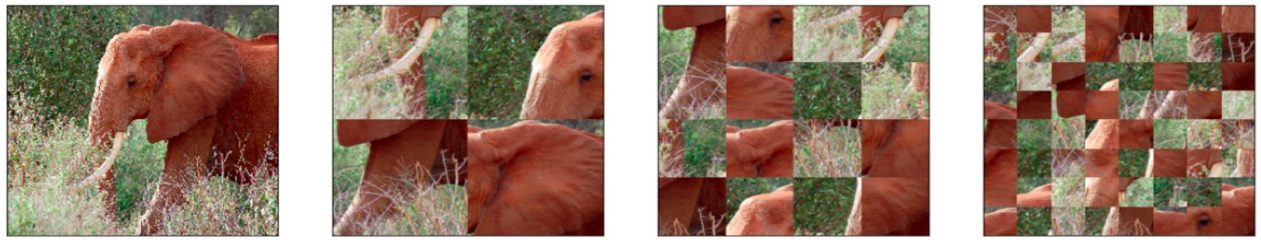
Training Details

1. IN & other methods
   - No fine-tuning: IN
   - Fine-tuning : other methods (E, SIN, SE, I-SIN, I-SE, Superposition)
     - Dataset variant (E) -> finetuned with dataset variant (E) + **IN**
     - Equal number of training samples from both datasets
     - Fine-tuning with different distribution degrade performance
     - Style Randomization (SR) only applied on IN
       - Purpose: reduce texture bias
     - SR applied on other variants showed no differences

2. ResNet-18 model
   - All methods have validation accuracy of 87% on IN
   - Supplementary: ResNet-50, DenseNet121, MobileNet V2

Results

Two different ways to evaluate shape bias



(a) The image is divided into $n \times n$ patches and the patches are randomly shuffled. The global object shape is increasingly perturbed with larger n.

(b) Images with conflicting shape and texture cues. The images are obtained by applying style transfer with a texture image as style source.

Shuffled image patches

Texture-shape cue conflict

Results

SR & Edge map increase shape bias

Network	shuffled image patches 4×4 acc(%)			shape based cue conflict #400		
	No styling	style blending	style randomization	No styling	style blending	style randomization
IN	67.22	51.34	41.97	63	82	86
SIN	38.46	36.96	34.95	144	155	156
E	34.11	33.95	**28.43**	155	166	**193**

Table 1: Comparison of different feature space style augmentation methods on 4×4 shuffled image patches and number of shape based predictions in texture-shape cue conflict images. Evaluation of shuffled patches is conducted on 598 correctly classified validation images by all the networks.

SR adopted for all IN

Network	shuffled image patches acc(%)			texture-shape cue conflict results		
	2×2	4×4	8×8	shape #400	shape #100	texture #100
IN	78.57	41.93	31.21	86	18	20
SIN	75.78	35.56	18.48	156	32	**2**
E	73.29	28.42	11.18	193	46	15
SE	**66.77**	28.73	12.89	224	55	6
E-SIN	71.12	**23.76**	**10.25**	**234**	**58**	6

Table 2: Comparison of models trained on different datasets on shuffled image patches and number of texture-shape cue conflict predictions based on shape and texture labels. Evaluation of shuffled image patches is conducted on 644 validation images that are correctly classified by all the networks.

- **Shuffled image patches**: Evaluated on ImageNet20 validation images that were correctly classified by all networks
- **Shape based cue conflict**:

shape label:	elephant	dog	cat	car
texture label:	cat	car	elephant	dog

-> predict shape labels

ImageNet20 400 images
- 300 images: only shape labels from ImageNet20
- 100 images: both shape & texture labels from ImageNet20

Results

- E / SE / E-SIN have stronger shape bias but show lower performance compared to SIN on both clean & corrupted images

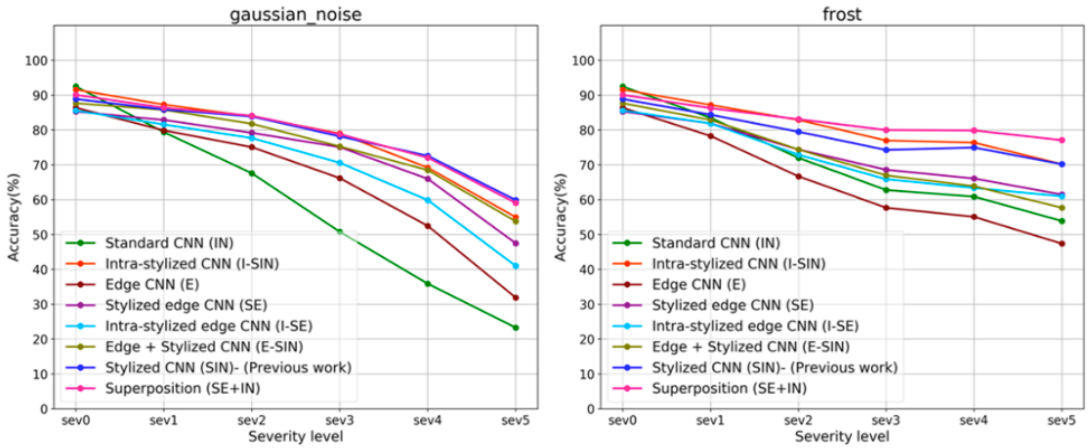


Figure 3: Classification accuracy of different networks on two corruptions across 5 severity levels. Severity 0 represents accuracy on clean validation data of IN. Severity levels 1 - 5 follow the corruption parameters from Hendrycks & Dietterich (2019) and represent increasingly strong corruptions.

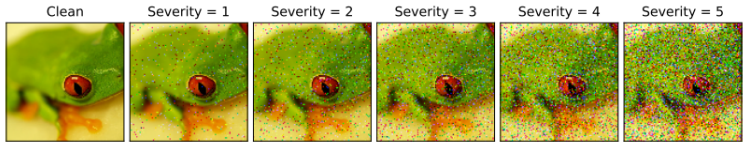


Figure 7: Impulse noise modestly to markedly corrupts a frog, showing our benchmark's varying severities.

Results

Stylization: strong augmentation method, not necessarily be shape-based

Network	Input image composition			Shape #100	Texture #100	Mean corruption acc(%)
	Natural image	Edge map	Style transfer			
IN	✓	✗	✗	11	39	64.69
SIN	✓	✗	✓	34	2	77.64
E	✗	✓	✗	46	15	62.01
SE	✗	✓	✓	55	6	71.81
E-SIN	✓	✓	✓	62	5	71.55
SE+IN	✓	✓	✓	22	13	**78.96**

shape bias ↓

robustness ↑

Table 3: Mean corruption accuracy (mCA) and texture/shape results on texture-shape cue conflict dataset of different networks. mCA is the mean accuracy over 15 ImageNet-C corruption and severities ranging from 1 to 5. Networks trained with style transfer augmentation perform better than those without and network trained on superpositioned images (SE+IN) yield best mCA.

mCA: mean accuracy over 15 ImageNet-C corruption and severities ranging from 1 to 5

Fine-tune affine parameters on target corruptions

- CNN encodes robust representation that can be leveraged when adapting affine parameters on a target domain

Network	Corruptions acc(%)				SIN val acc(%)	Cue Conflict	
	Speckle noise	Gaussian blur	Frost	Pixelate		shape #400	texture #100
IN	61.28	42.96	66.62	78.54	42.0	63	39
IN (fine-tuned)	82.7	77.3	81.02	87.02	68.0	130	13
E	67.76	44.48	61.04	70.94	62.3	193	15
E (fine-tuned)	80.18	71.74	73.7	74.78	72.4	222	9

Table 4: Mean corruption accuracy, SIN and cue conflict results of networks with & without additional fine-tuning of the affine parameters of normalization layers on the respective corruptions. Fine-tuned networks perform significantly better, despite only the normalization layers are updated.

*ICLR Review

Score: 6 / 7 / 9 / 6

1. Experiment results are limited to ImageNet20 (other datasets including CIFAR10)
    - Also conducted experiment on ImageNet 200
2. Style Randomization has lack of technical novelty
    - OOD / domain generalization -> purpose well aligns
3. Edge Dataset has limited amount of information compared to SIN
    - Fine-tuned with ImageNet / validation accuracy on IN: 87%

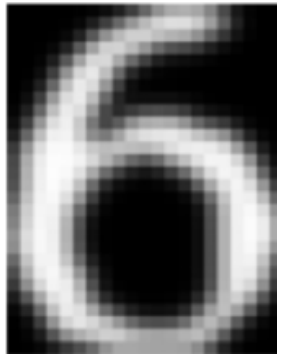
*Distortion

1. Pixelate (= mosaic)

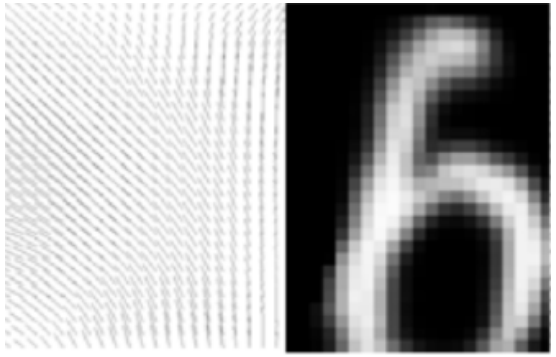
Original image

Pixelate

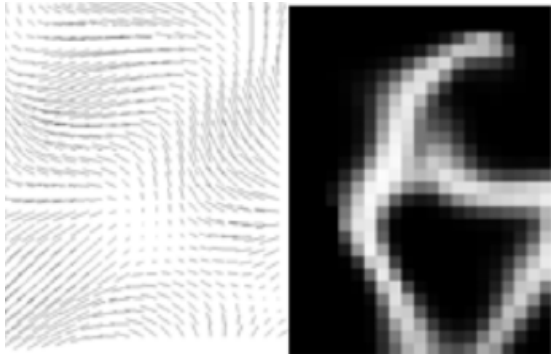
2. Elastic distortion



Original image



Affine transformation



Elastic distortion:
Different distortion per each pixel

THE END

THANK YOU!