XEP33VKR, **Filip Naiser** presents**:**

High-frequency Component Helps Explain the Generalization of Convolutional Neural Networks

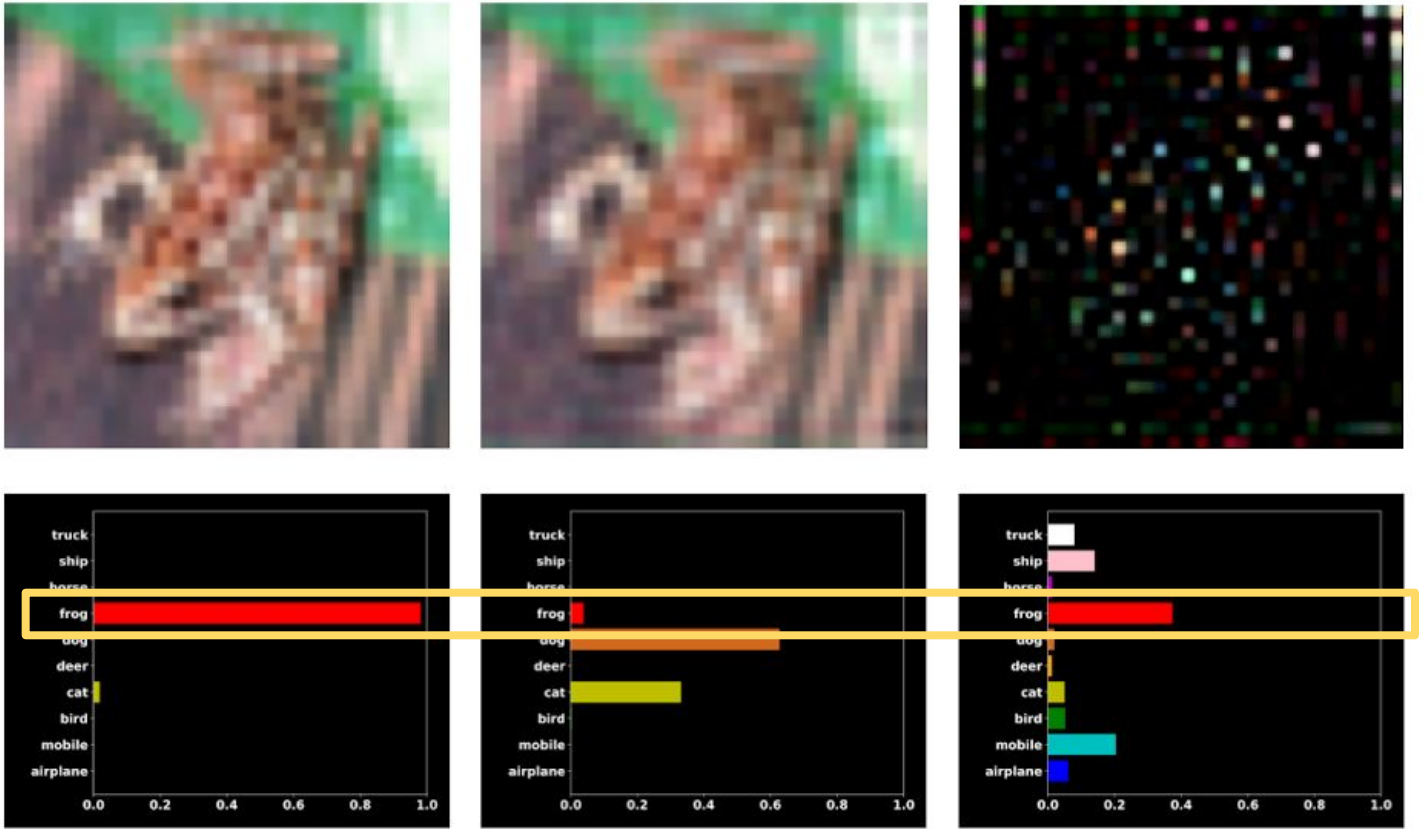
Haohan Wang, Xindi Wu, Zeyi Huang, Eric P. Xing School of Computer Science
Carnegie Mellon University

CVPR 2020

Outline

- Motivation, Introduction
- HFC, LFC, definition
- Heuristics (Dropout, mix-up, BatchNorm, Adversarial Training)
- Are HFC just noises?
- A hypothesis on Batch Normalization
- examples CIFAR, ImageNet, Detection
- Discussion

An C

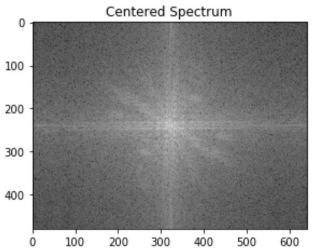


(a) A sample of frog

High/Low-Frequency Component (HFC, LFC)

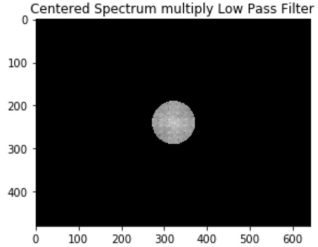
Fourier Transform

$$\mathbf{z} = \mathcal{F}(\mathbf{x})$$



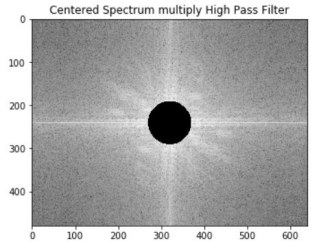
filter radius

$$\mathbf{z}_l, \mathbf{z}_h = t(\mathbf{z}; r)$$



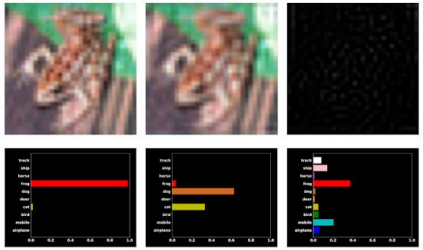
$$\mathbf{x}_h = \mathcal{F}^{-1}(\mathbf{z}_h)$$

LFC

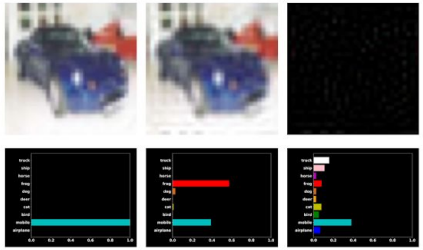


$$\mathbf{x}_l = \mathcal{F}^{-1}(\mathbf{z}_l)$$

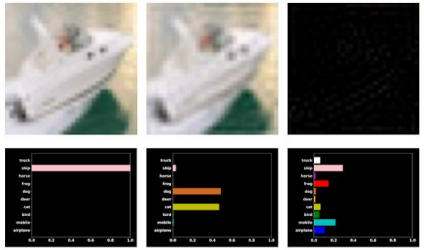
HFC



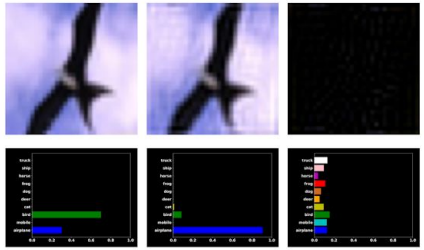
(a) A sample of frog



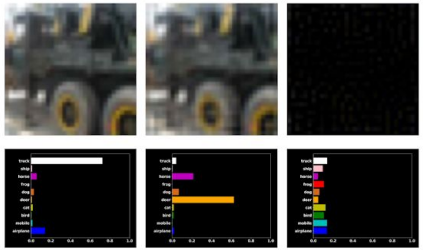
(b) A sample of mobile



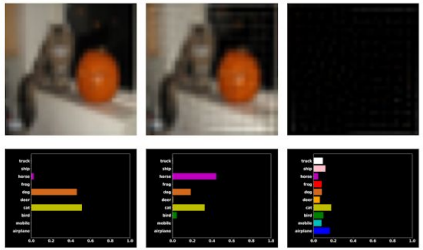
(c) A sample of ship



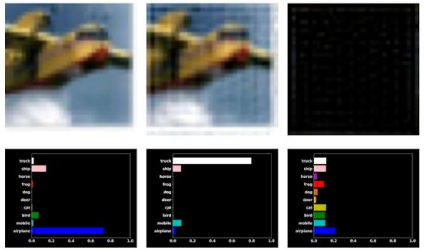
(d) A sample of bird



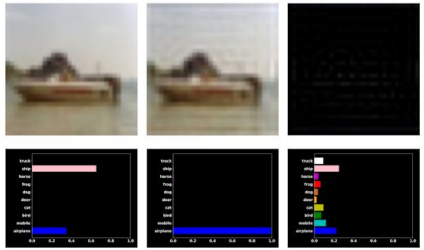
(e) A sample of truck



(f) A sample of cat



(g) A sample of airplane



(h) A sample of ship

Figure 2. Eight testing samples selected from CIFAR10 that help explain that CNN can capture the high-frequency image: the model (ResNet18) correctly predicts the original image (1st column in each panel) and the high-frequency reconstructed image (3rd column in each panel), but incorrectly predict the low-frequency reconstructed image (2nd column in each panel). The prediction confidences are also shown. The frequency components are split with $r = 12$. Details of the experiment will be introduced later.

What is it about

- Accuracy vs. Robustness (adversarial examples)
- Generalization vs. Overfitting
- Shuffled label memorization paradox (overfitting)

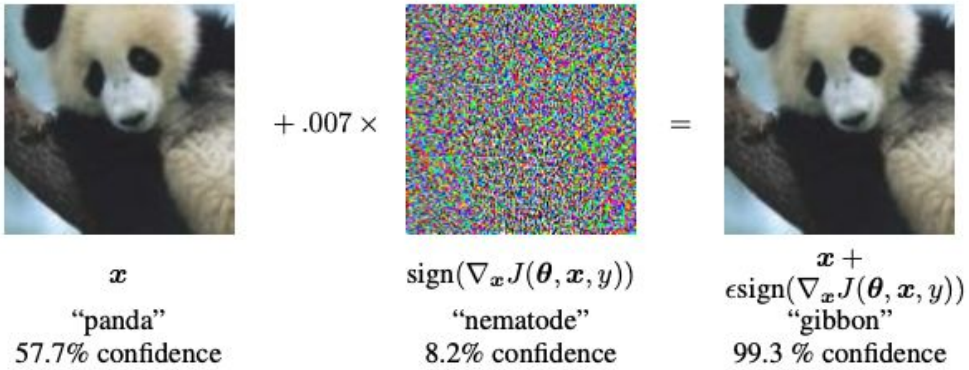
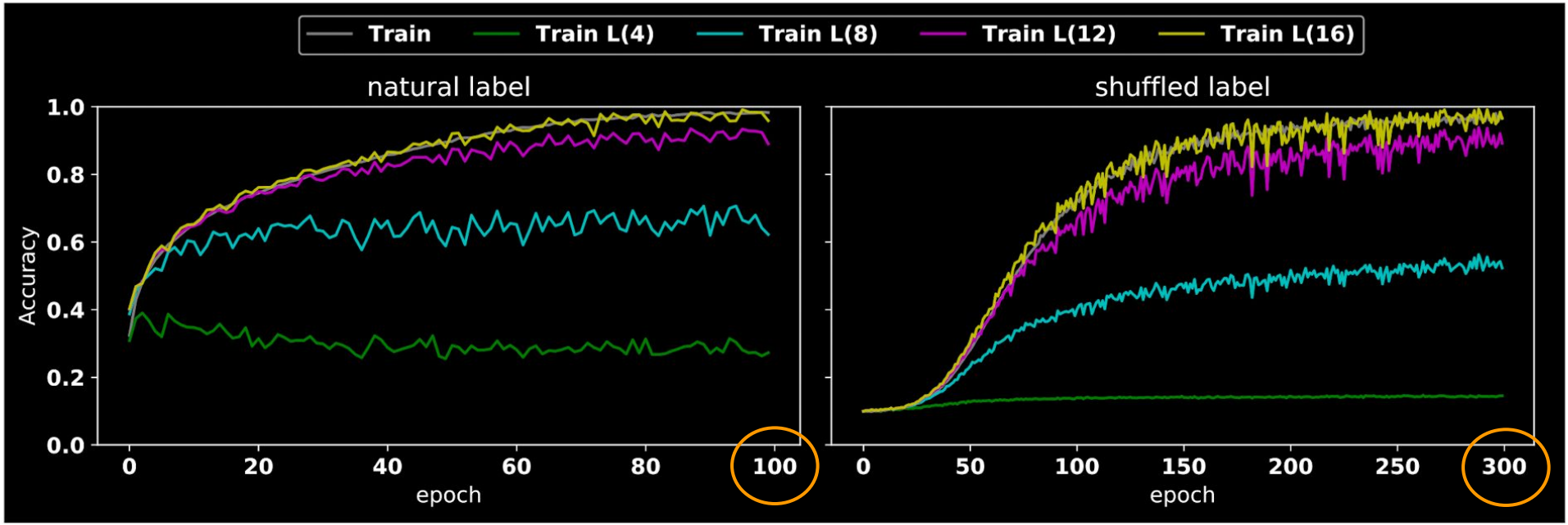


Image Credit: Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

Figure 3. Training curves of the original label case (100 epoches) and shuffled label case (300 epoches), together plotted with the low-frequent counterpart of the images. All curves in this figure are from train samples.



Generalization power of LFC and HFC

- CIFAR 10, ResNet-18, ADAM, LR=1e-4, batch size=100, epochs=100

Table 1. We test the generalization power of LFC and HFC by training the model with $\mathbf{x}_l$ or $\mathbf{x}_h$ and test on the original test set.

LFC			HFC		
r	train acc.	test acc.	r	train acc.	test acc.
4	0.9668	0.6167	4	0.9885	0.2002
8	0.9786	0.7154	8	0.9768	0.092
12	0.9786	0.7516	12	0.9797	0.0997
16	0.9839	0.7714	16	0.9384	0.1281

High/Low-Frequency Component (HFC, LFC)

An image is decomposed $\mathbf{x} = \{\mathbf{x}_l, \mathbf{x}_h\}$

$$\mathbf{z} = \mathcal{F}(\mathbf{x}), \qquad \mathbf{z}_l, \mathbf{z}_h = t(\mathbf{z}; r),$$

$$\mathbf{x}_l = \mathcal{F}^{-1}(\mathbf{z}_l), \qquad \mathbf{x}_h = \mathcal{F}^{-1}(\mathbf{z}_h),$$

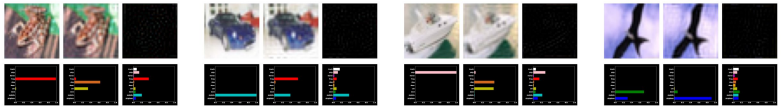
$$\mathbf{z}_l(i,j) = \begin{cases} \mathbf{z}(i,j), & \text{if } d((i,j),(c_i,c_j)) \leq r \\ 0, & \text{otherwise} \end{cases},$$

$$\mathbf{z}_h(i,j) = \begin{cases} 0, & \text{if } d((i,j),(c_i,c_j)) \leq r \\ \mathbf{z}(i,j), & \text{otherwise} \end{cases}$$

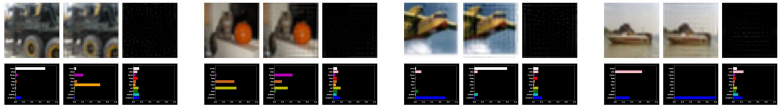
- this process is channel-wise
- imaginary part of $\mathcal{F}^{-1}(\cdot)$ is discarded.

Are HFC just Noise?

- An Experiment:
  - “LFC and HFC separation” is done using SVD decomposition
  - The first image is reconstructed based on dominant singular values
  - The second is reconstructed with trailing singular values
- With this set-up, much fewer images supporting the story in Figure 2. (where CNN is classifying HFC image correctly) was observed.

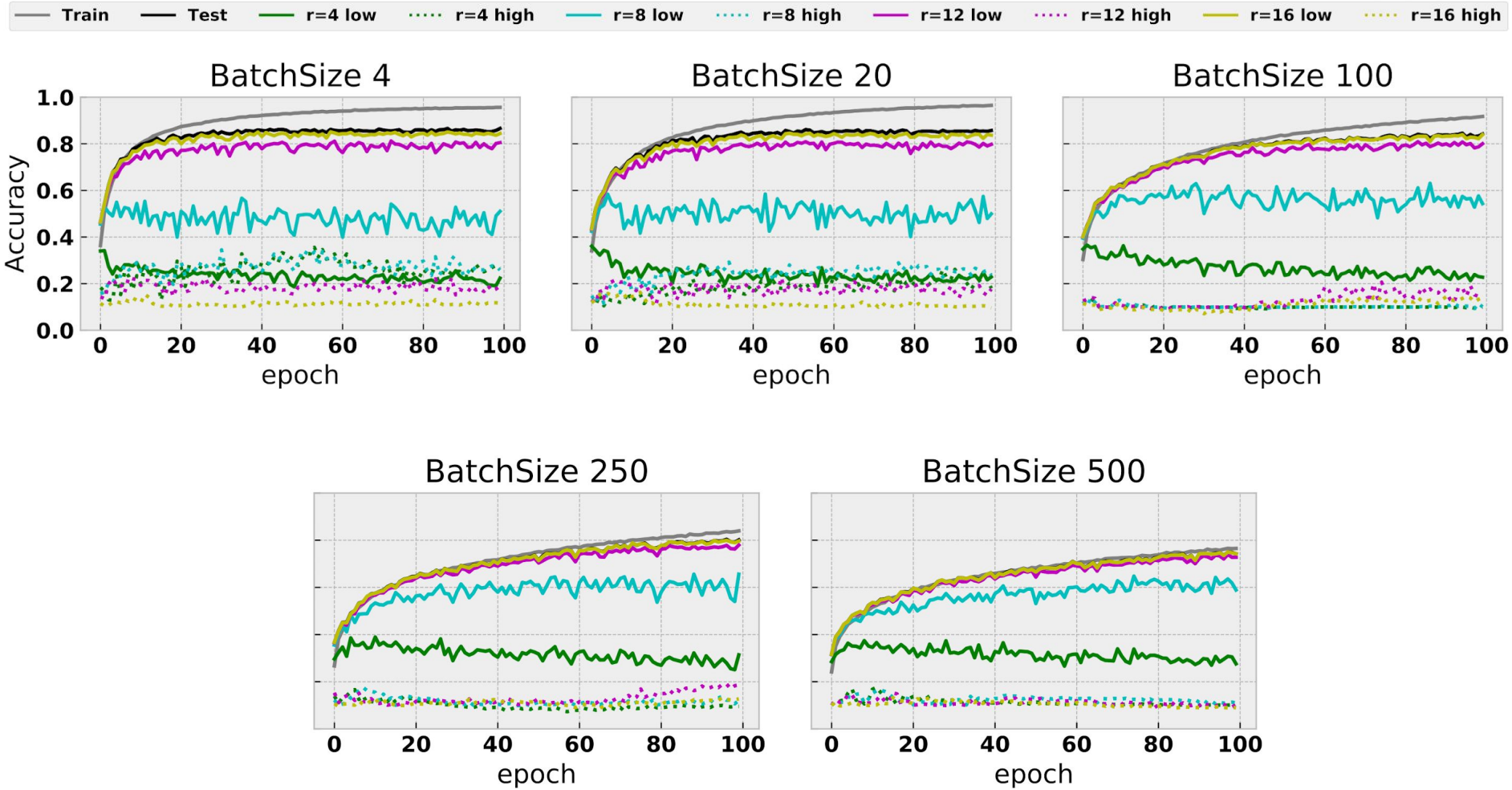


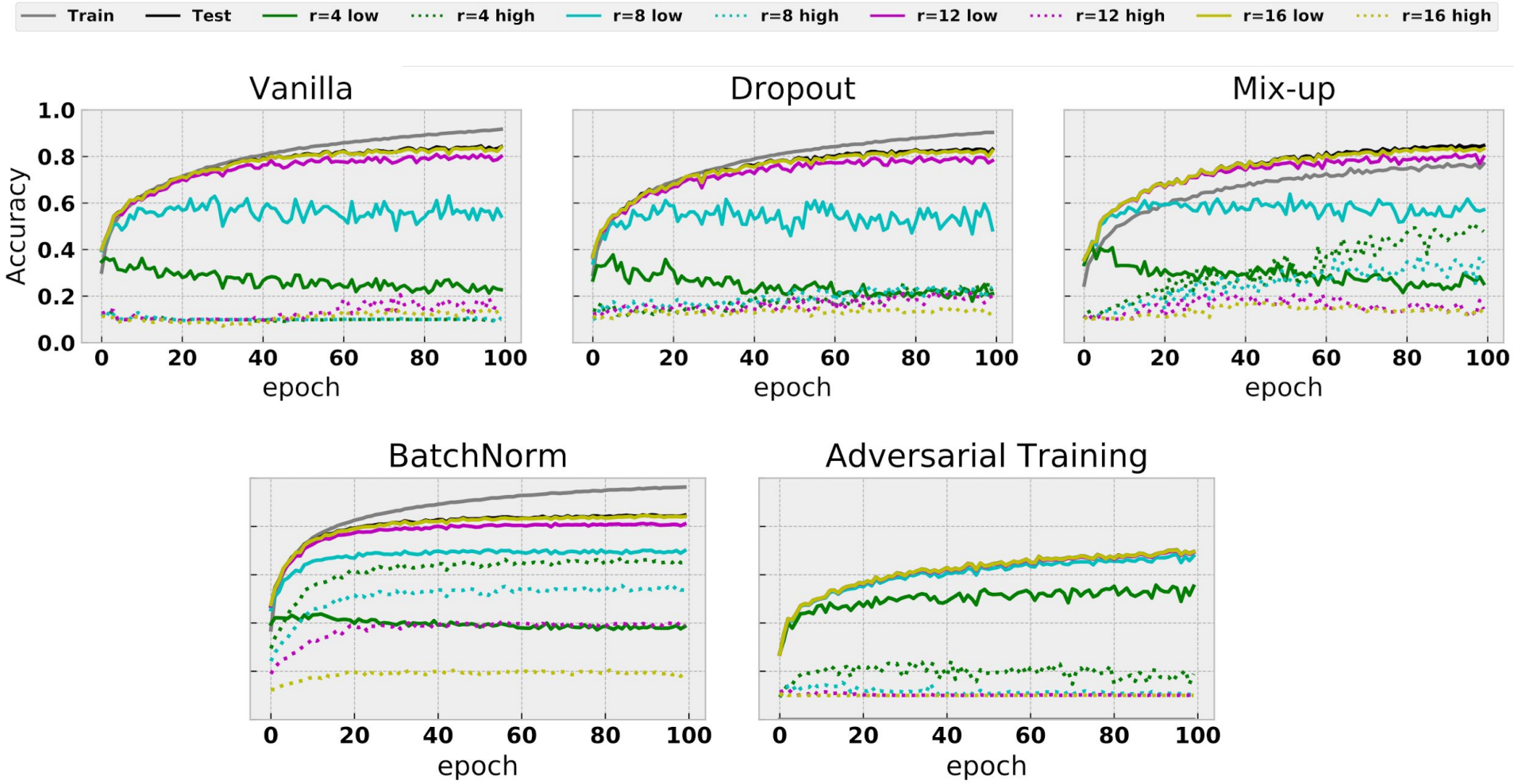
(a) A sample of frog (b) A sample of mobile (c) A sample of ship (d) A sample of bird

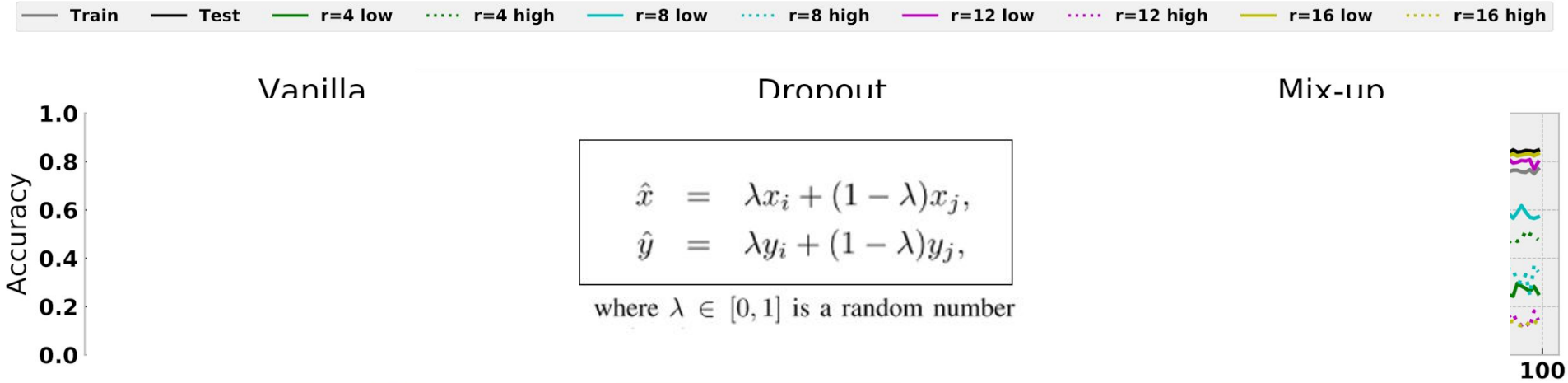


(e) A sample of truck (f) A sample of cat (g) A sample of airplane (h) A sample of ship

Figure 2. Eight testing samples selected from CIFAR10 that help explain that CNN can capture the high-frequency image: the model (ResNet18) correctly predicts the original image (1st column in each panel) and the high-frequency reconstructed image (3rd column in each panel), but incorrectly predict the low-frequency reconstructed image (2nd column in each panel). The prediction confidences are also shown. The frequency components are split with $r = 12$. Details of the experiment will be introduced later.


Train
Test
r=4 low
r=4 high
r=8 low
r=8 high
r=12 low
r=12 high
r=16 low
r=16 high
BatchSize 4
BatchSize 20
BatchSize 100
BatchSize 250
BatchSize 500
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
epoch


Train
Test
r=4 low
r=4 high
r=8 low
r=8 high
r=12 low
r=12 high
r=16 low
r=16 high
Vanilla
Dropout
Mix-up
BatchNorm
Adversarial Training
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
epoch



$$\hat{x} = \lambda x_i + (1-\lambda) x_j,$$
$$\hat{y} = \lambda y_i + (1-\lambda) y_j,$$

where $\lambda \in [0, 1]$ is a random number

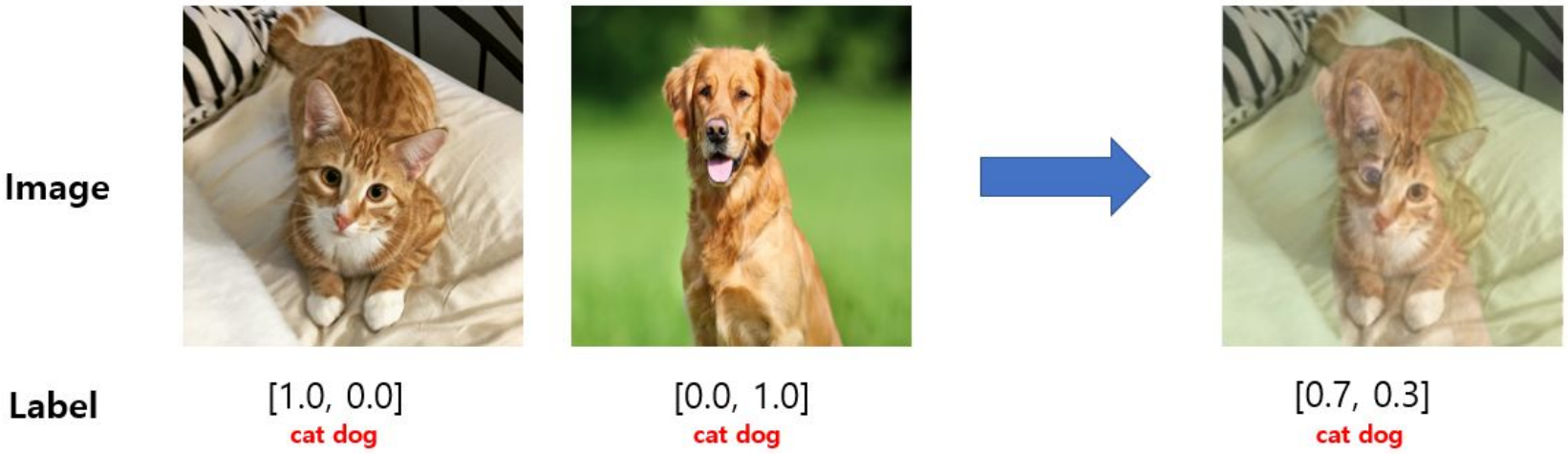


image credit: https://www.kaggle.com/kaushal2896/data-augmentation-tutorial-basic-cutout-mixup

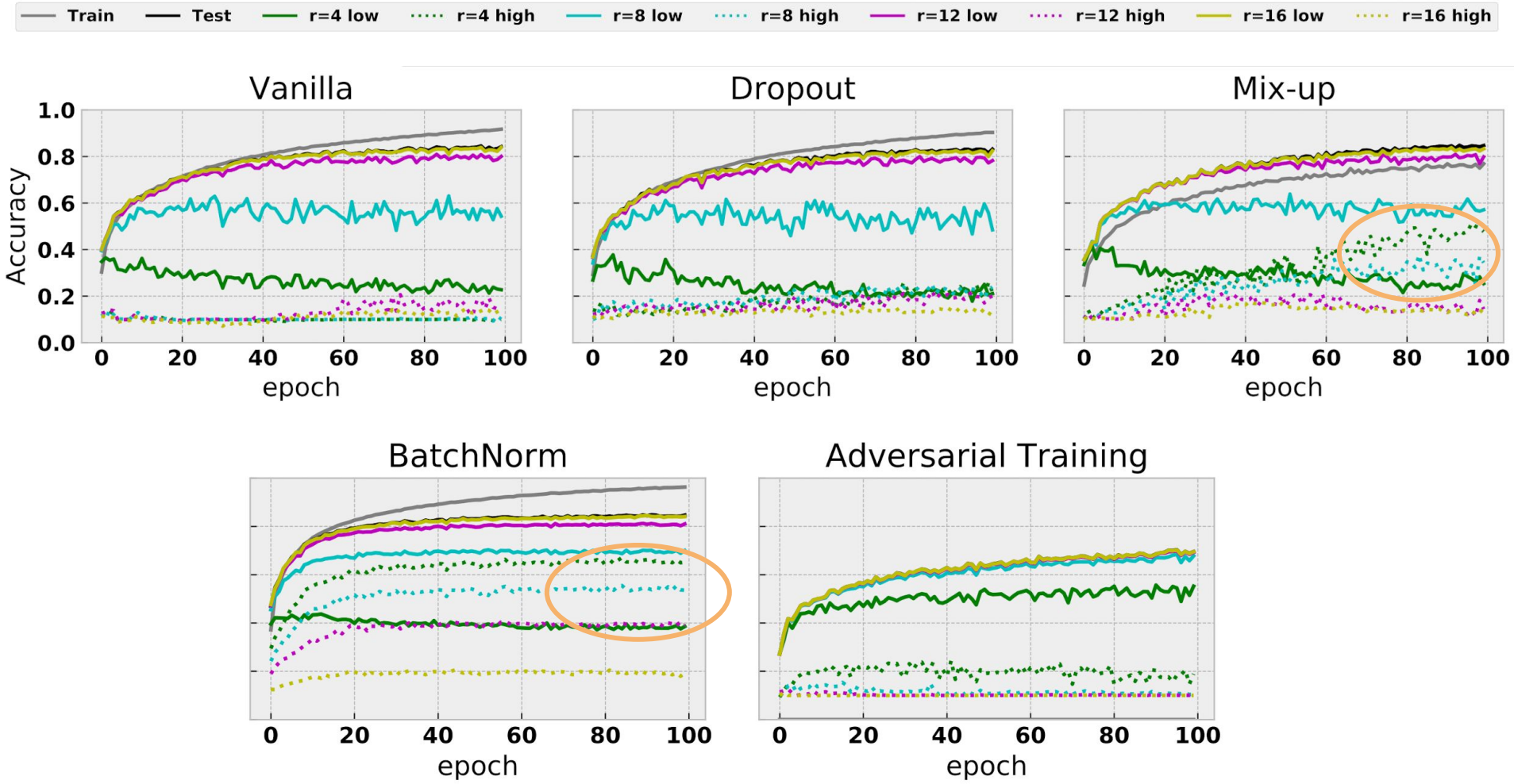
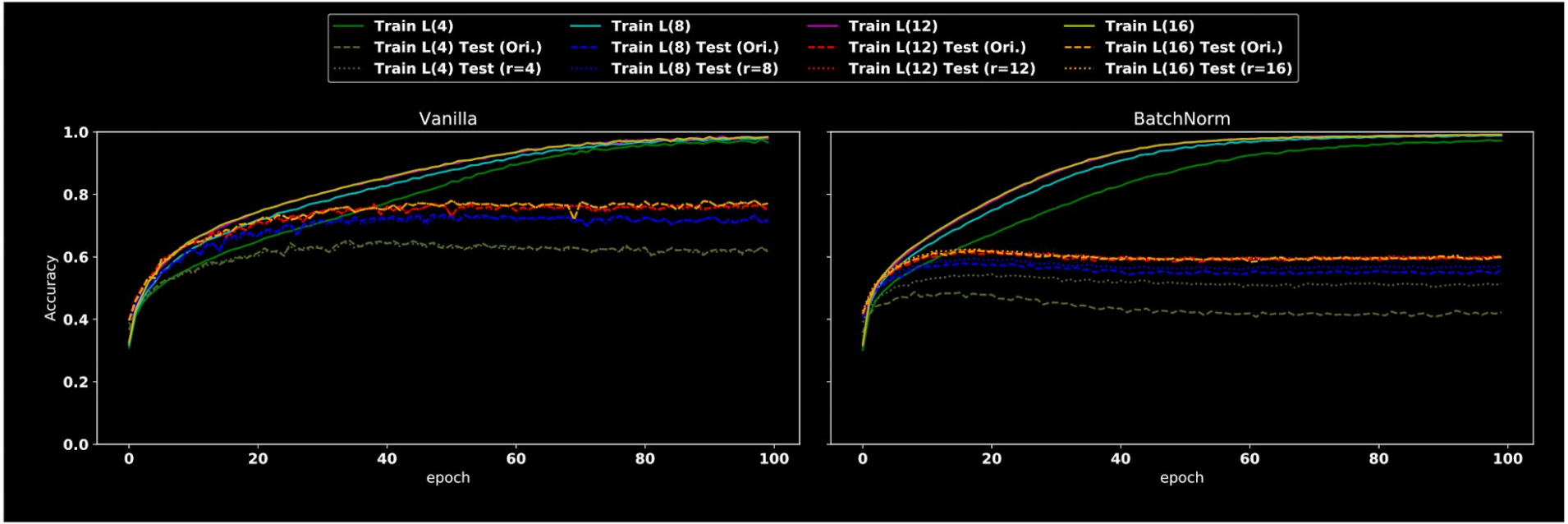

Train
Test
r=4 low
r=4 high
r=8 low
r=8 high
r=12 low
r=12 high
r=16 low
r=16 high
Vanilla
Dropout
Mix-up
BatchNorm
Adversarial Training
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
epoch

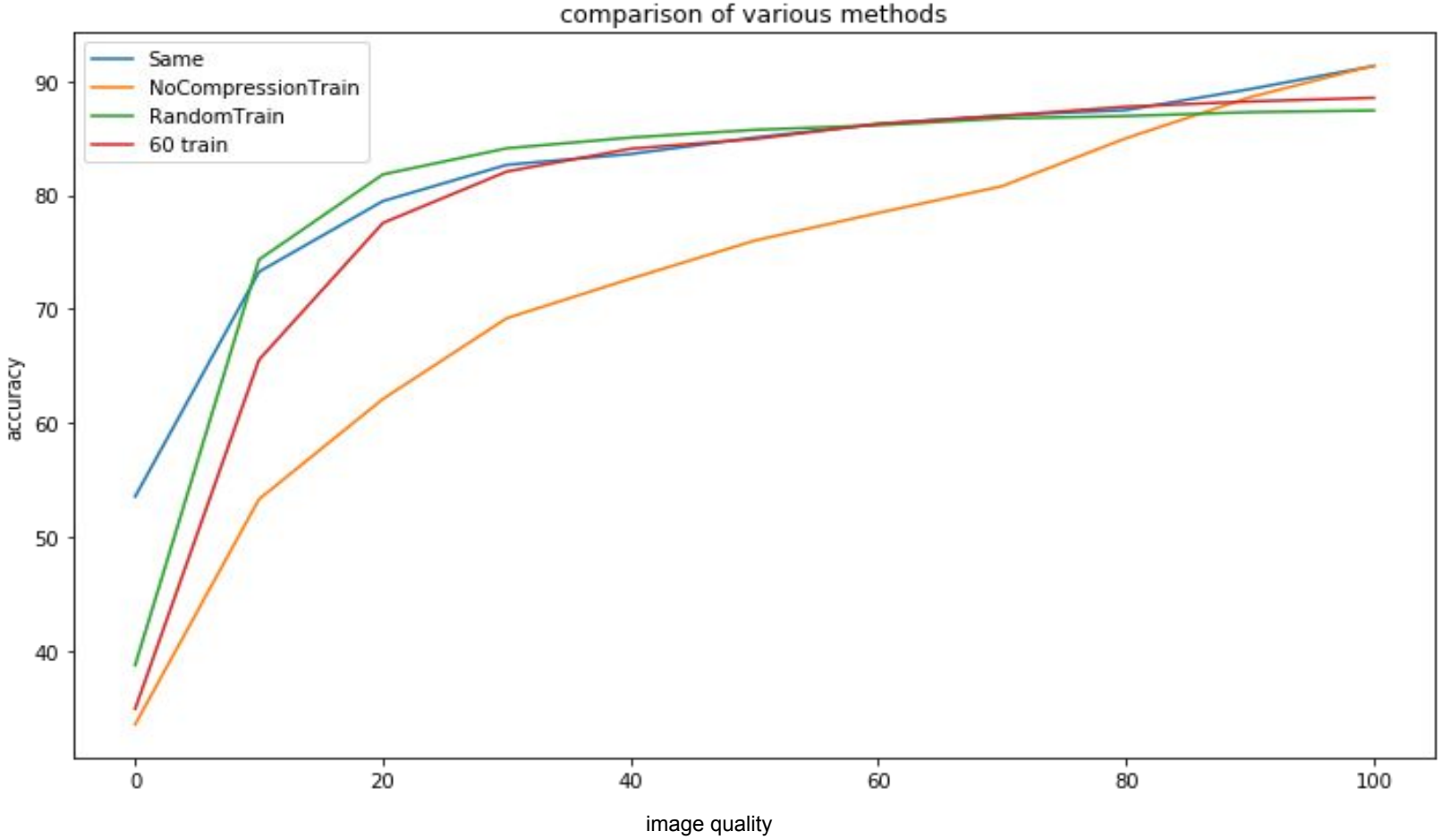
Figure 6. Comparison of models with vs. without BatchNorm trained with LFC data.



Bonus: JPEG compression

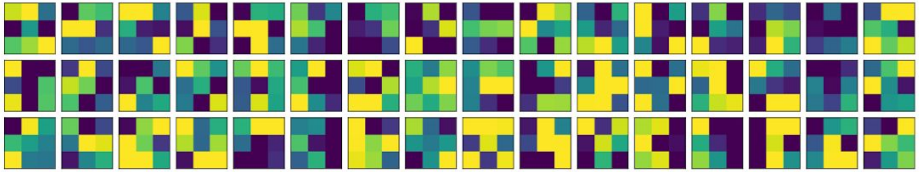
- CIFAR 100, ResNet-18
- We investigated effect of JPEG compression on the inference accuracy.
- We tried to use JPEG compression as an augmentation
-
- Same = NN was trained and evaluated with the same compression
- RandomTrain = random compression was picked during training
- 60Train = image quality was fixed during training to 60
- NoCompressionTrain = the data for training was taken as provided in CIFAR-100

Credits: Jan Pokorný, Milan Šulc, Lukáš Picek

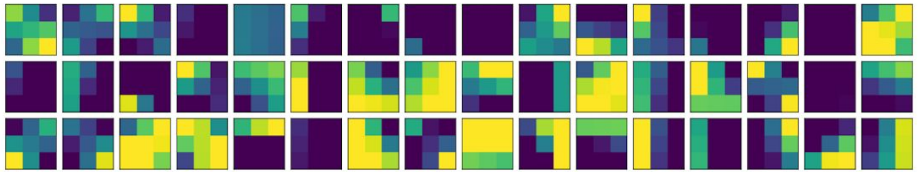


Credits: Jan Pokorný, Milan Šulc, Lukáš Picek

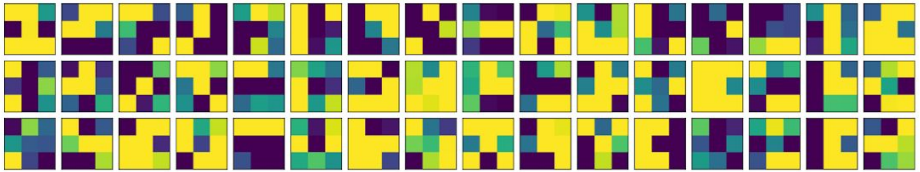
Robust Models Have Smooth Kernels



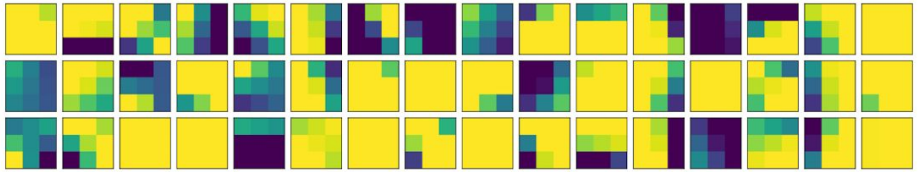
(a) convoluational kernels of $M_{natural}$



(c) convoluational kernels of $M_{natural}(\rho=1.0)$



(b) convoluational kernels of $M_{adversarial}$



(d) convoluational kernels of $M_{adversarial}(\rho=1.0)$

Figure 7. Visualization of convolutional kernels (16 kernels each channel $\times$ 3 channels at the first layer) of models.

Robust Models Have Smooth Kernels

	Clean	FGSM			PGD		
		$\epsilon = 0.03$	$\epsilon = 0.06$	$\epsilon = 0.09$	$\epsilon = 0.03$	$\epsilon = 0.06$	$\epsilon = 0.09$
$\mathsf{M}_{\mathsf{natural}}$	**0.856**	0.107	0.069	0.044	0.003	0.002	0.002
$\mathsf{M}_{\mathsf{natural}}(\rho = 0.10)$	0.815	0.149	0.105	0.073	0.009	0.002	0.001
$\mathsf{M}_{\mathsf{natural}}(\rho = 0.25)$	0.743	0.16	0.11	0.079	0.021	0.005	0.005
$\mathsf{M}_{\mathsf{natural}}(\rho = 0.50)$	0.674	0.17	0.11	0.083	0.031	0.016	0.014
$\mathsf{M}_{\mathsf{natural}}(\rho = 1.0)$	0.631	**0.171**	**0.14**	**0.127**	**0.086**	**0.078**	**0.078**
$\mathsf{M}_{\mathsf{adversarial}}$	**0.707**	**0.435**	**0.232**	0.137	**0.403**	0.138	0.038
$\mathsf{M}_{\mathsf{adversarial}}(\rho = 0.10)$	0.691	0.412	0.192	0.109	0.379	0.13	0.047
$\mathsf{M}_{\mathsf{adversarial}}(\rho = 0.25)$	0.667	0.385	0.176	0.097	0.352	0.116	0.04
$\mathsf{M}_{\mathsf{adversarial}}(\rho = 0.50)$	0.653	0.365	0.18	0.106	0.334	0.121	0.062
$\mathsf{M}_{\mathsf{adversarial}}(\rho = 1.0)$	0.638	0.356	0.223	**0.186**	0.337	**0.175**	**0.131**

Table 2. Prediction performance of models against different adversarial attacks with different ϵ.

Beyond Image Classification

- RetinaNet ResNet50 + FPN on MS COCO -> mAP 35.6%
- Use r=128 and map imgs to LFC and HFC and test the same model
  - LFC 27.5% mAP
  - HFC 10.7% mAP
- 1684 where LFC actually performs better (Fig 9)

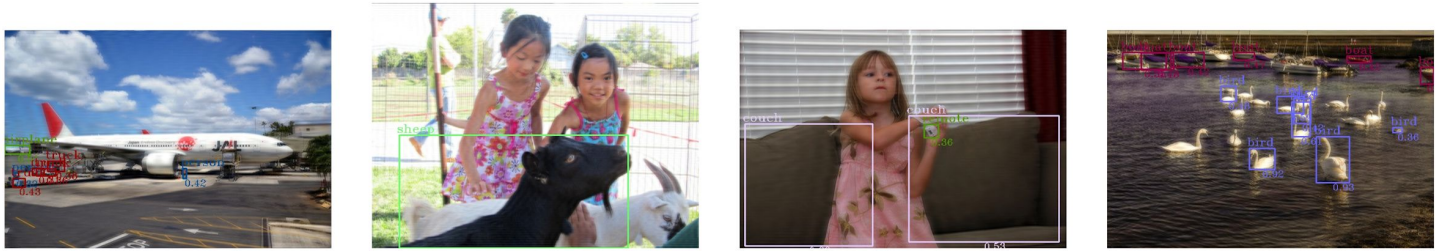


Figure 8. Some objects are recognized worse (lower MAP scores) when the experiments are repeated with low-frequent images. Marked objects are the ones that induce differences.

Figure 9. Some objects are recognized better (higher MAP scores) when the experiments are repeated with low-frequent images. Marked objects are the ones that induce differences.

Critique

- Hard-to-read plots.
- Not justified relation between human labeled data and LHF (e.g. what if the task is to detect cracks in a metal plate).
- Missing appendix even though it is mentioned multiple times.
- They do not mention how many times they run experiments.
- A table with values accompanying Figure 5 would be nice.

Trade-off between Robustness and Accuracy

A1: "only $\mathbf{x}_l$ is perceivable to human, but both $\mathbf{x}_l$ and $\mathbf{x}_h$ are perceivable to a CNN," we have:

$$\mathbf{y} := f(\mathbf{x}; \mathcal{H}) = f(\mathbf{x}_l; \mathcal{H}),$$

but when a CNN is trained with

$$\arg\min_{\theta} l(f(\mathbf{x}; \theta), \mathbf{y}),$$

which is equivalent to

$$\arg\min_{\theta} l(f(\{\mathbf{x}_l, \mathbf{x}_h\}; \theta), \mathbf{y}),$$

CNN may learn to exploit $\mathbf{x}_h$ to minimize the loss. As a result, CNN's generalization behavior appears unintuitive to a human. □

Trade-off between Robustness and Accuracy

the accuracy of θ

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})}\alpha(f(\mathbf{x};\theta),\mathbf{y}) \tag{1}$$

and the adversarial robustness of θ as in *e.g.*, [8]:

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})} \min_{\mathbf{x}':d(\mathbf{x}',\mathbf{x})\leq\epsilon} \alpha(f(\mathbf{x}';\theta),\mathbf{y}) \tag{2}$$

where ϵ is the upper bound of the perturbation allowed.

Trade-off between Robustness and Accuracy

A2: "for model θ, there exists a sample $\langle \mathbf{x}, \mathbf{y} \rangle$ such that:

$$f(\mathbf{x}; \theta) \neq f(\mathbf{x}_l; \theta),"$$

Corollary 1. *With assumptions A1 and A2, there exists a sample $\langle \mathbf{x}, \mathbf{y} \rangle$ that the model θ cannot predict both accurately (evaluated to be 1.0 by Equation 1) and robustly (evaluated to be 1.0 by Equation 2) under any distance metric $d(\cdot, \cdot)$ and bound ϵ as long as $\epsilon \geq d(\mathbf{x}, \mathbf{x}_l)$.*

Critique

If a model can exploit multiple different sets of signals, then why $M_{natural}$ prefers to learn LFC that happens to align well with the human perceptual preference? While there are explanations suggesting neural networks' tendency towards simpler functions [48], we conjecture that this is simply because, since the data sets are organized and annotated by human, the LFC-label association is more "generalizable" than the one of HFC: picking up LFC-label association will lead to the steepest descent of the loss surface, especially at the early stage of the training.

Conclusion

- CNN may capture HFC that are misaligned with human visual preference.
- Heuristics like Mix-up and BatchNorm may encourage capturing HFC
- Bonus: If you can’t control input quality during inference, add JPEG compression as an augmentation.

Thanks Giorgos for comments and organization!
Thank you for your attention!

Discussion