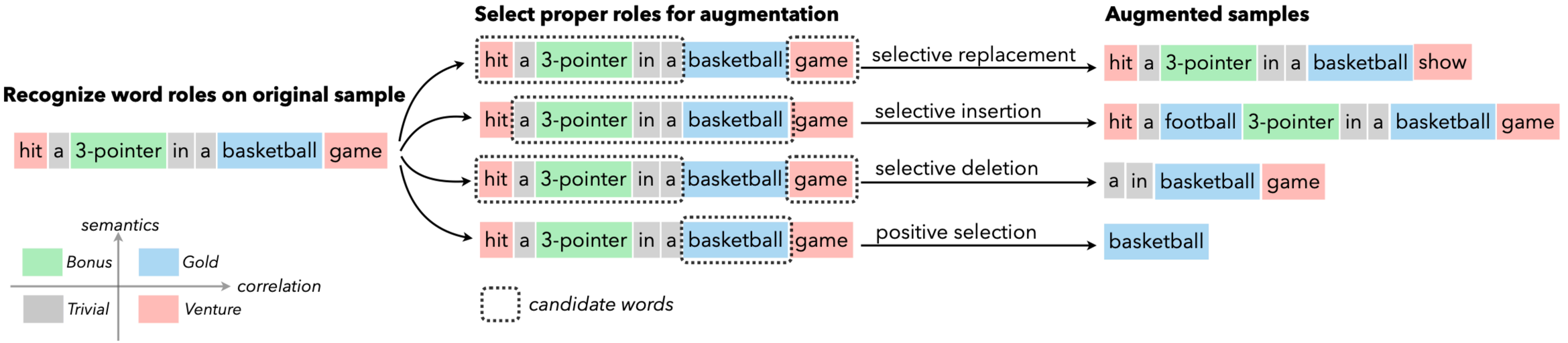
Data Augmentation

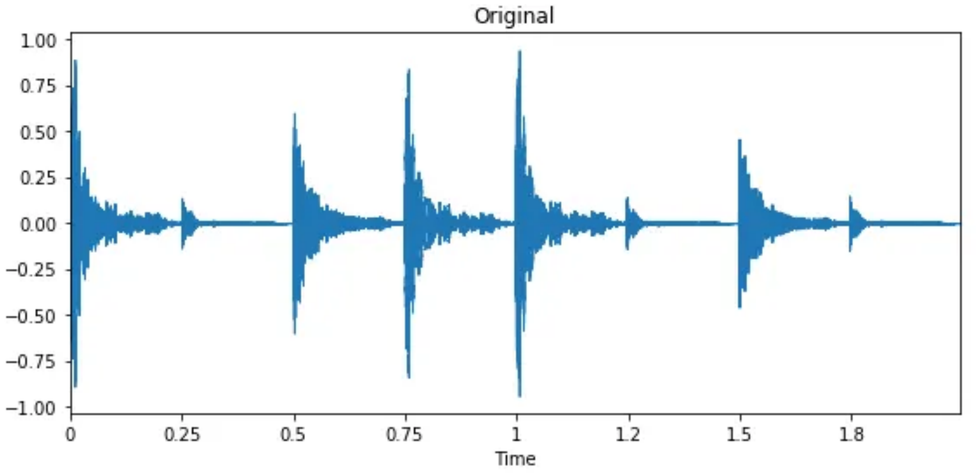
Nguyen Phuc Tue

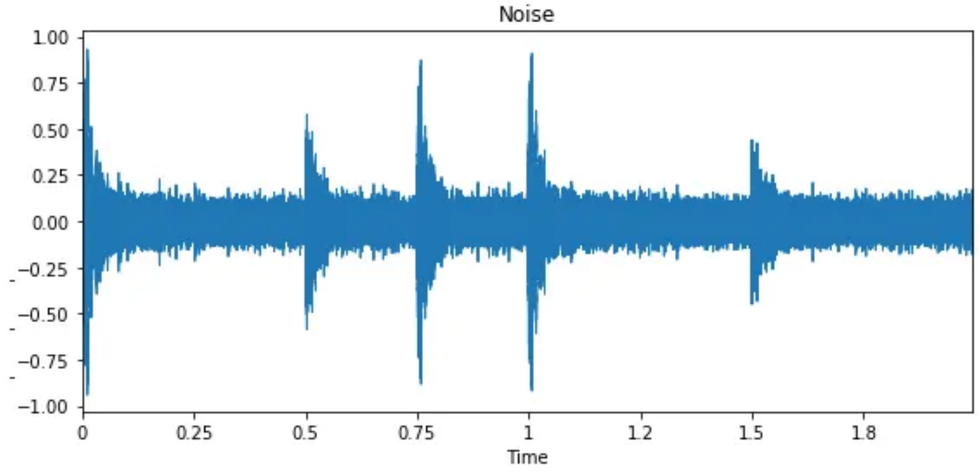
- Data augmentation
- Augmentation during learning
- Test-time augmentation
- Cutout
- Mixup
- Cutmix
- Augmix

Data is expensive!



Text data augmentation





Voice data augmentation

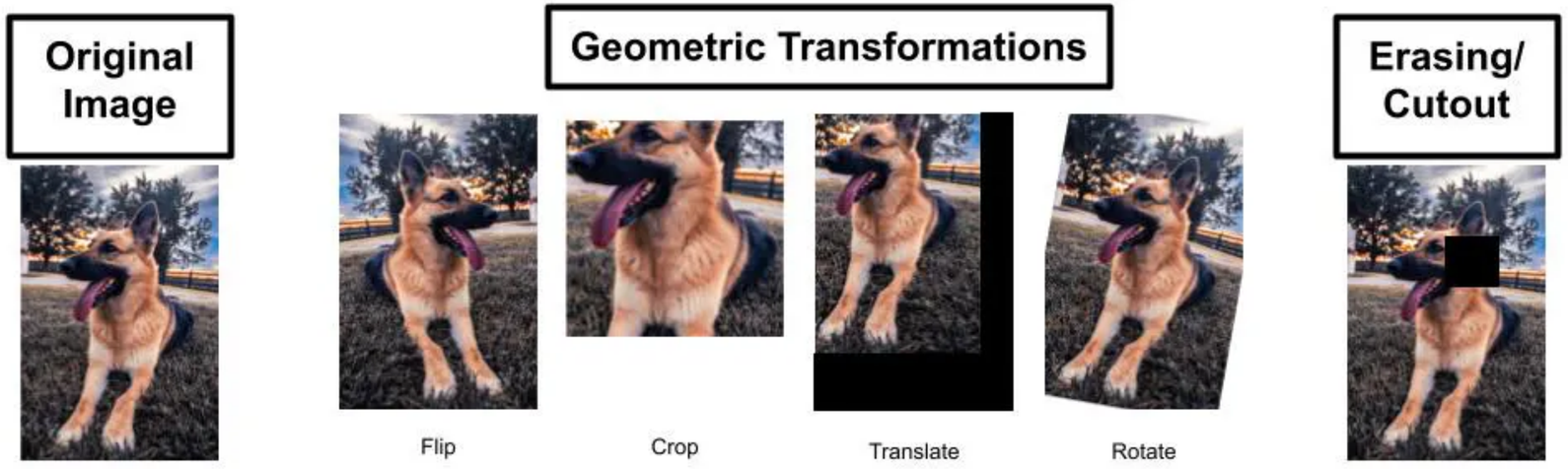
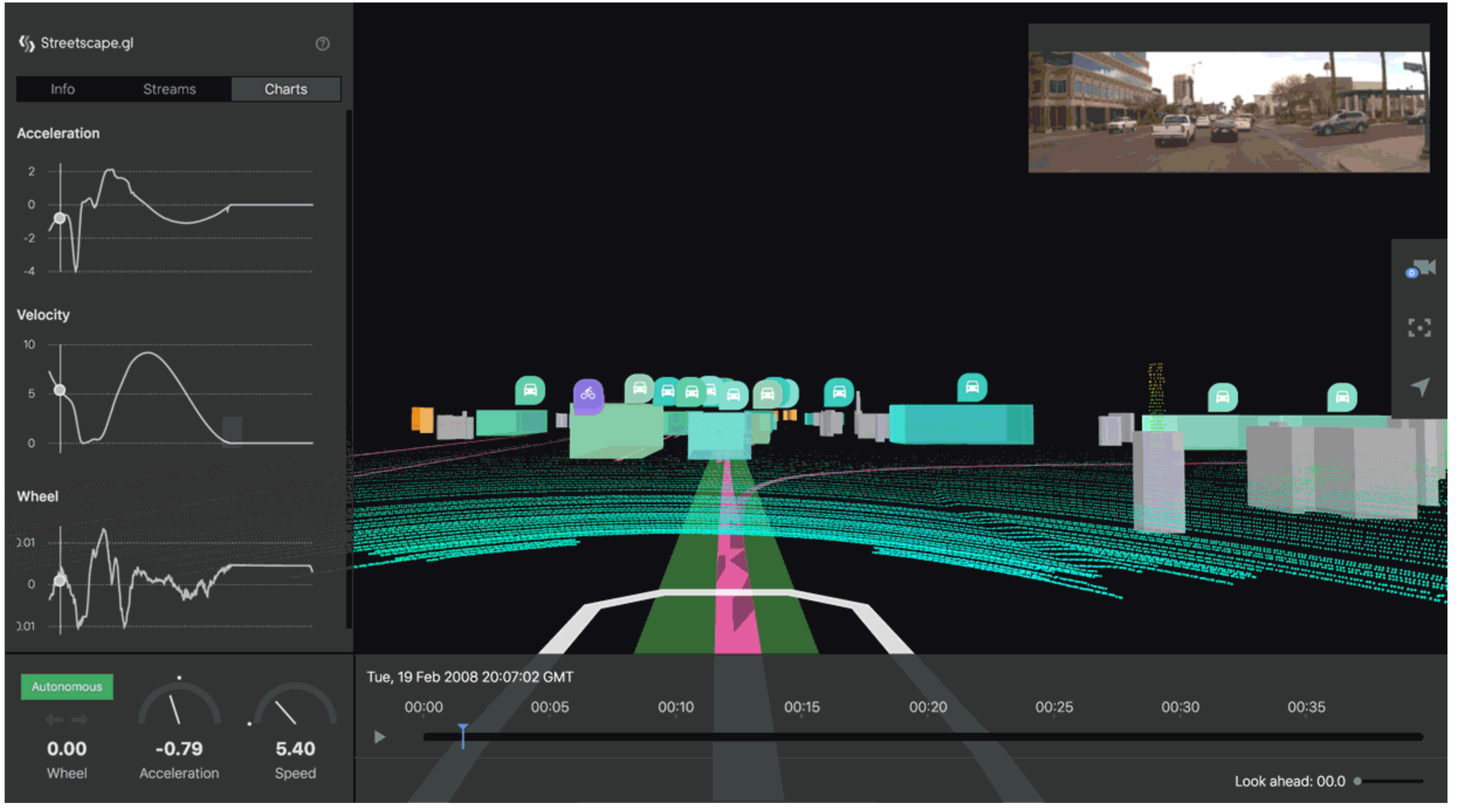
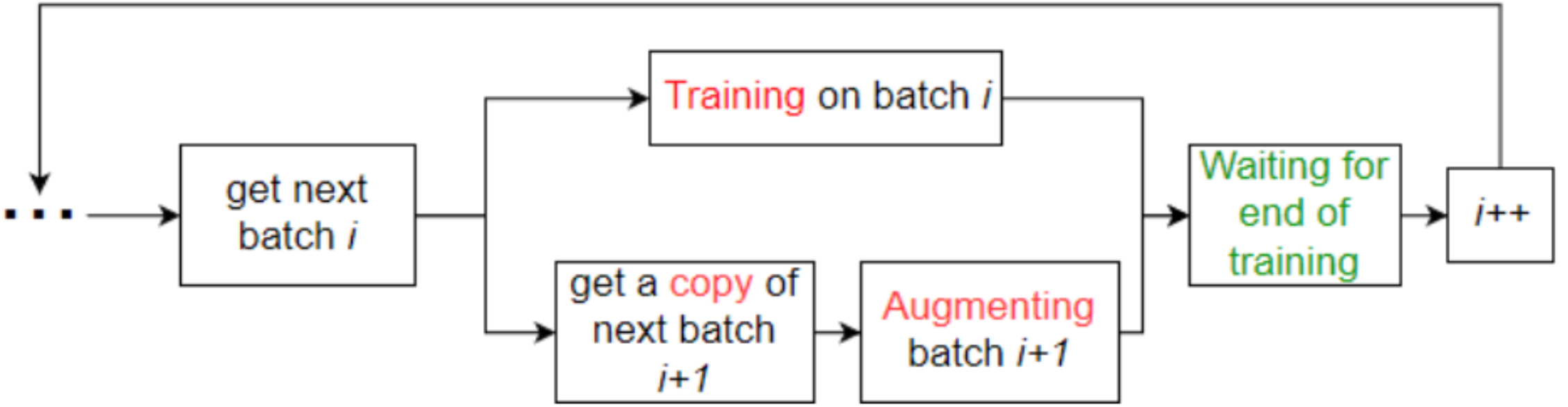


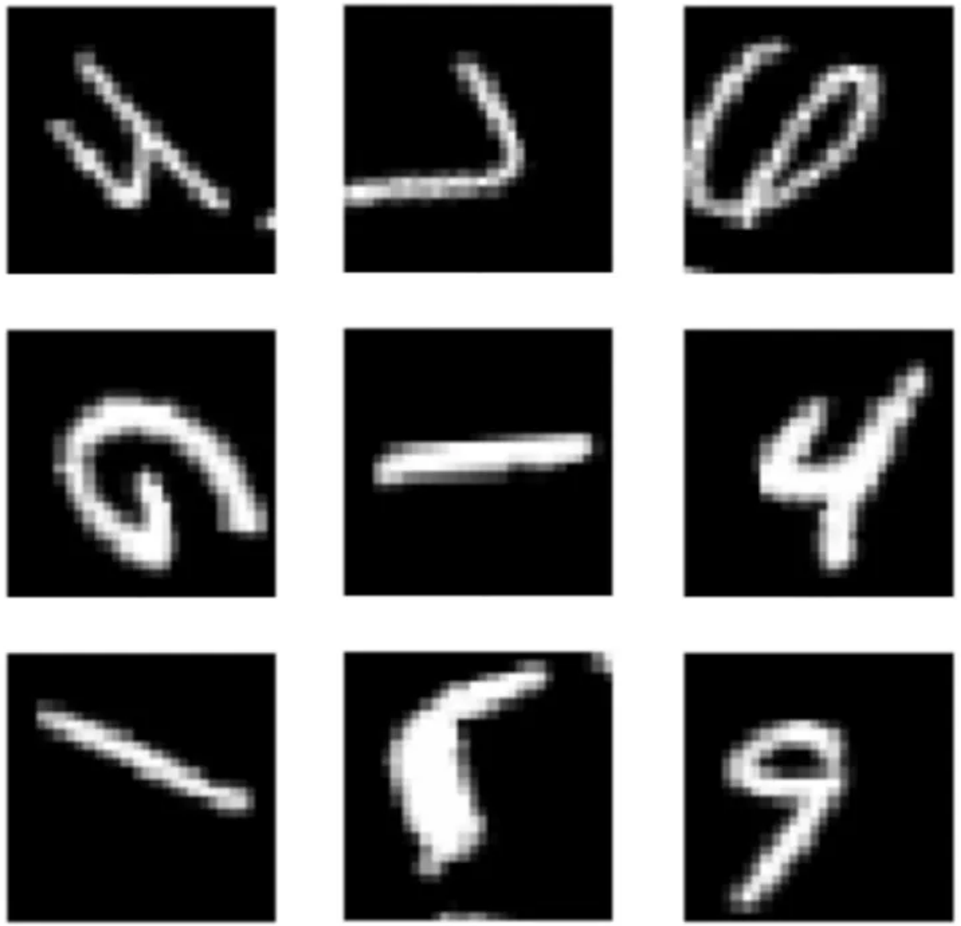
Image augmentation



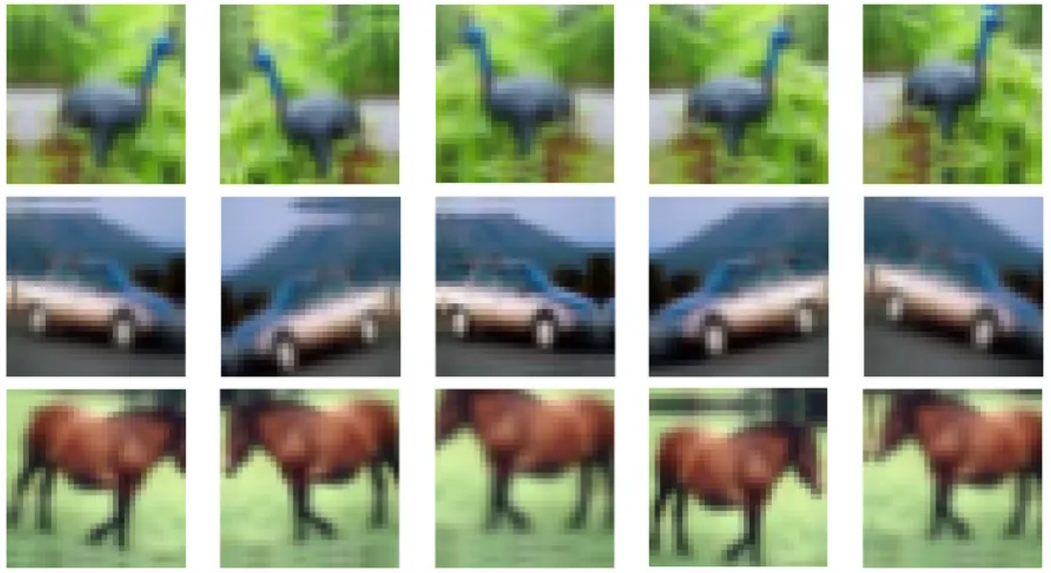
Simulated data augmentation



Augmentation during learning



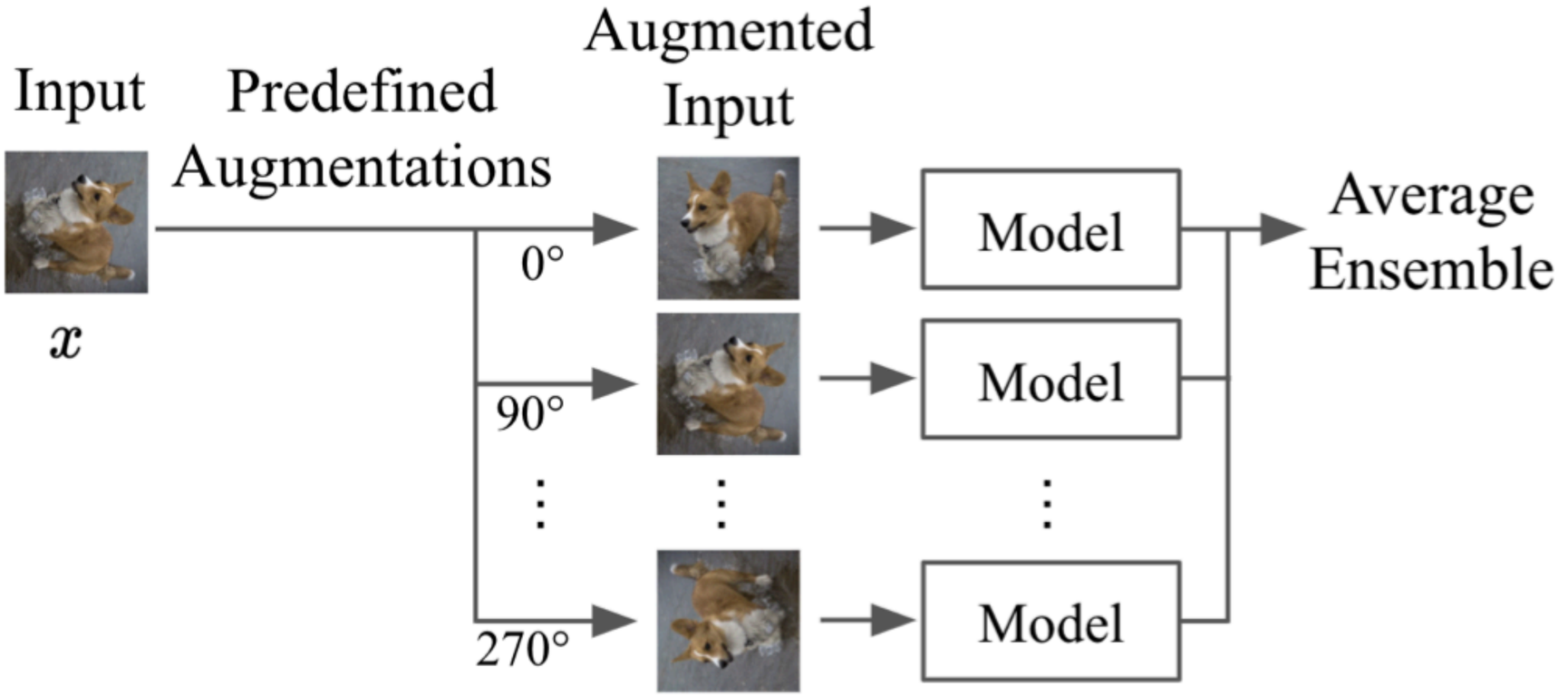
Bad augmentation



Good augmentation

Augmentation simplifies neural network

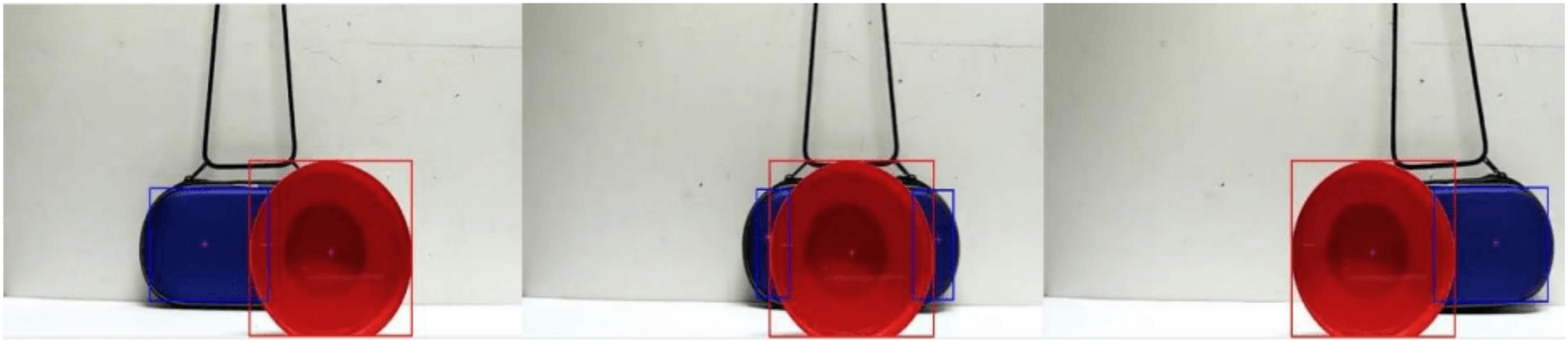
Архитектура w: число коэффициентов mul: число умножений	Error rate, без аугментации	Error rate, с аугментацией на лету
ResNet, w: 40 тыс., mul: 1.63 млн.	**5.68%**	5.06%
ResNet, w: 18 тыс., mul: 730 тыс.	6.71%	**5.73%**



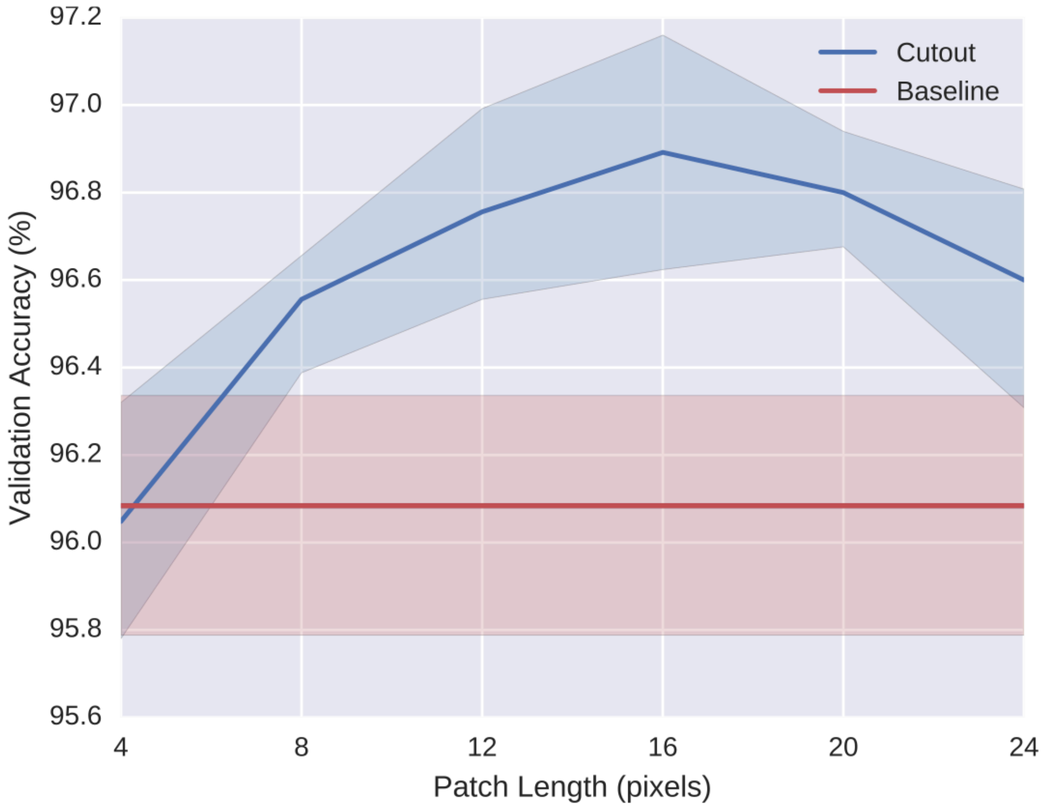
Test time augmentation

Early version of cutout

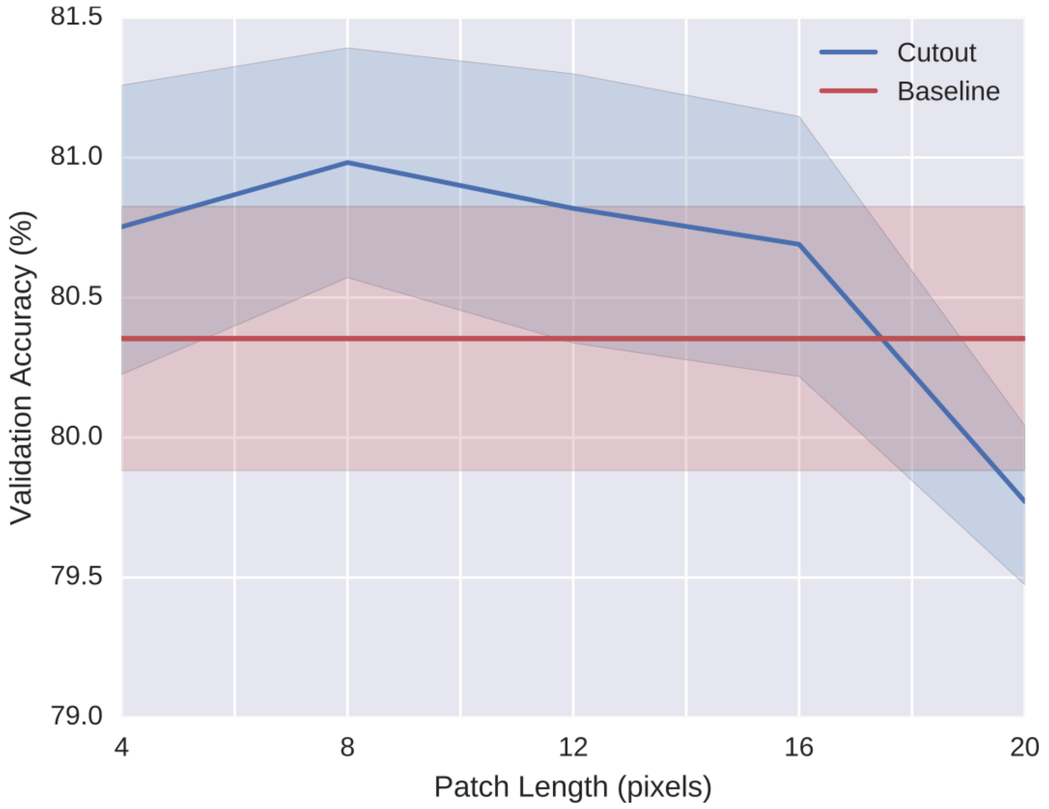
Cutout



Object occlusion

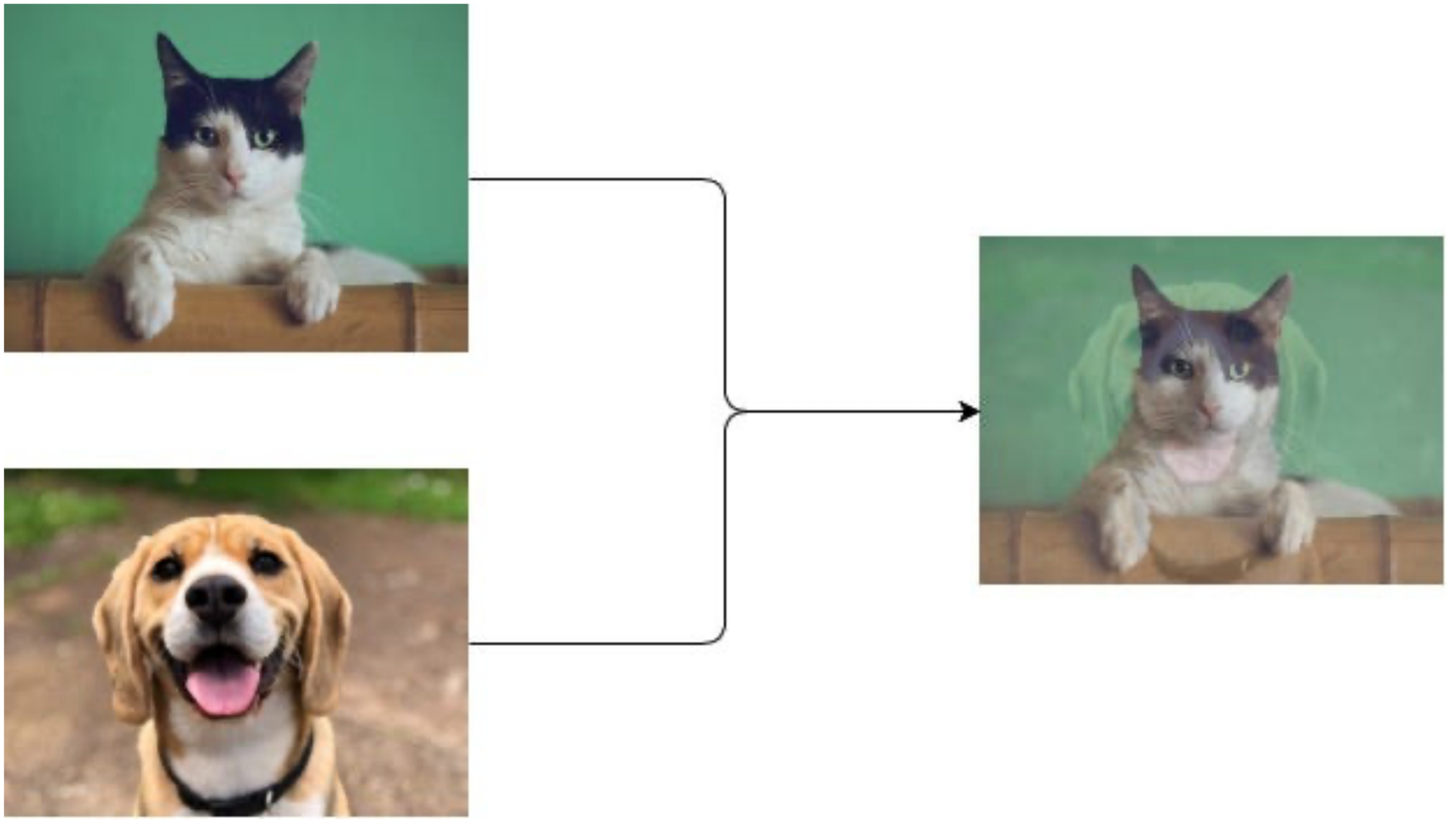


(a) CIFAR-10



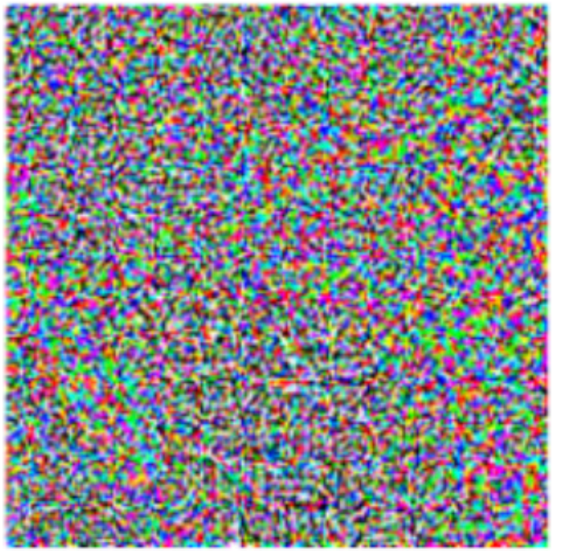
(b) CIFAR-100

Cutout performance on different datasets



Mixup

$+ .007 \times$



$=$

$\boldsymbol{x}$

"panda"
57.7% confidence

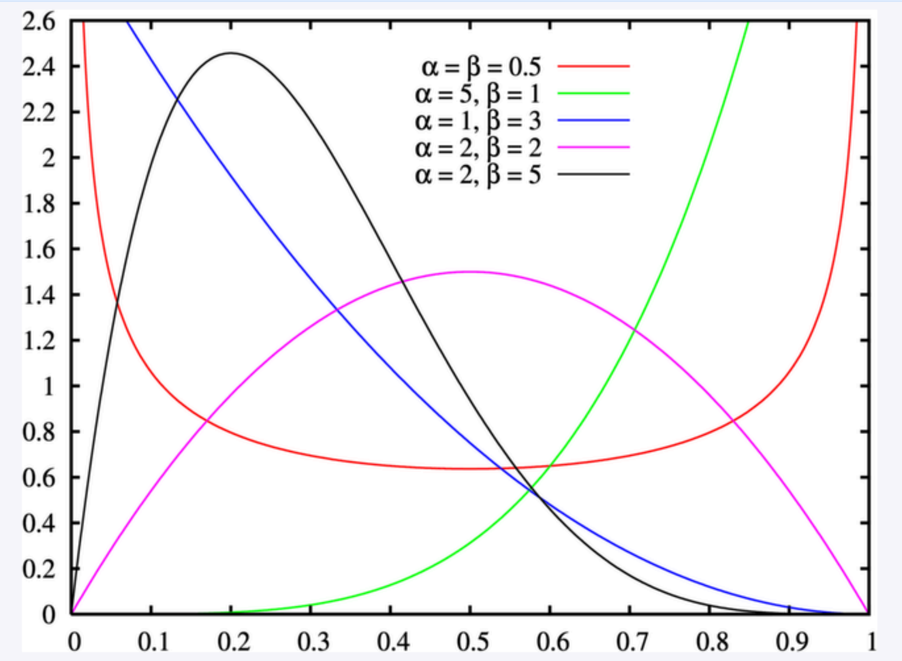
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

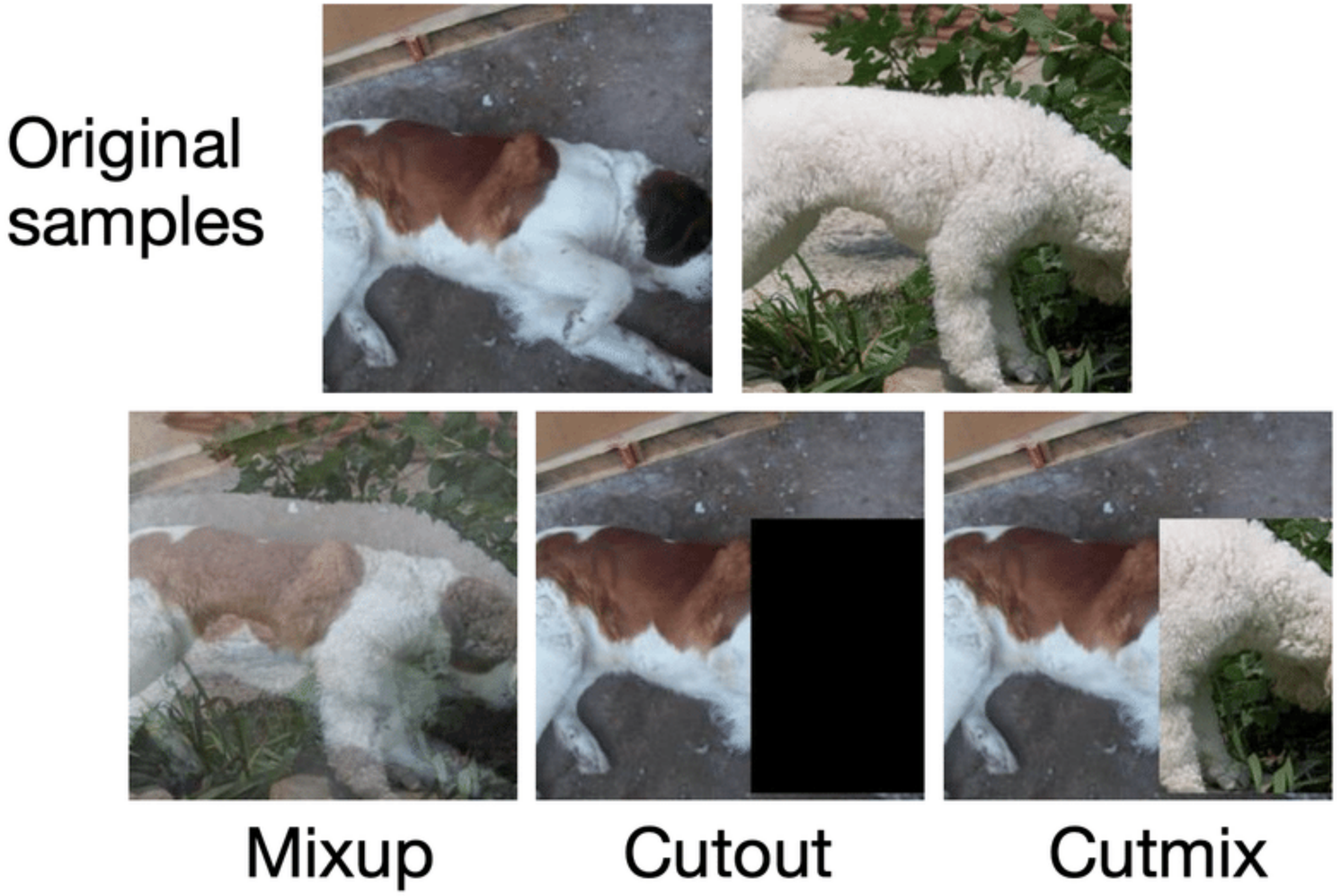
$\boldsymbol{x} +$
$\epsilon \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Adversial data

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j, \qquad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j, \qquad \text{where } y_i, y_j \text{ are one-hot label encodings}$$



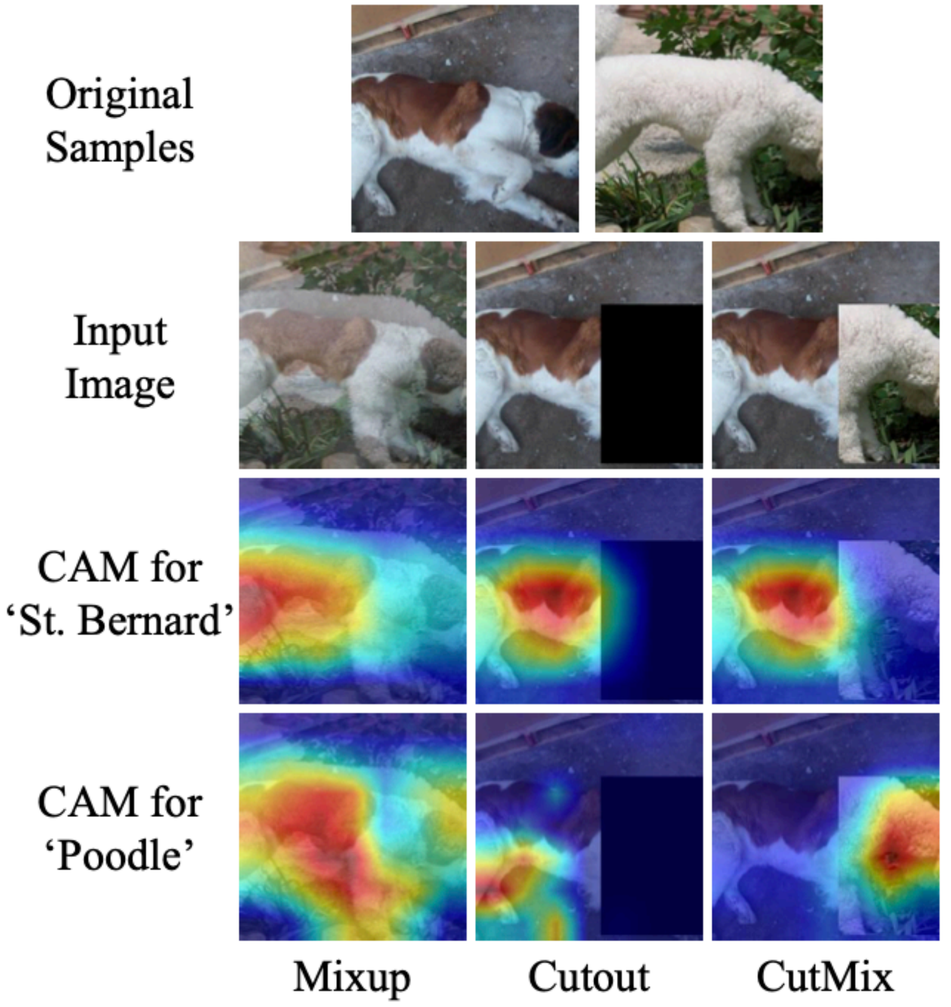
Implementation of Mixup



Difference between Mixup, Cutout and Cutmix

	Mixup	Cutout	CutMix
Usage of full image region	✔	✘	✔
Regional dropout	✘	✔	✔
Mixed image & label	✔	✘	✔

Comparison of Mixup, Cutout and Cutmix



Class Activation Mapping visualizations on Mixup, Cutout and Cutmix

$$\tilde{x} = \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B$$
$$\tilde{y} = \lambda y_A + (1 - \lambda) y_B,$$

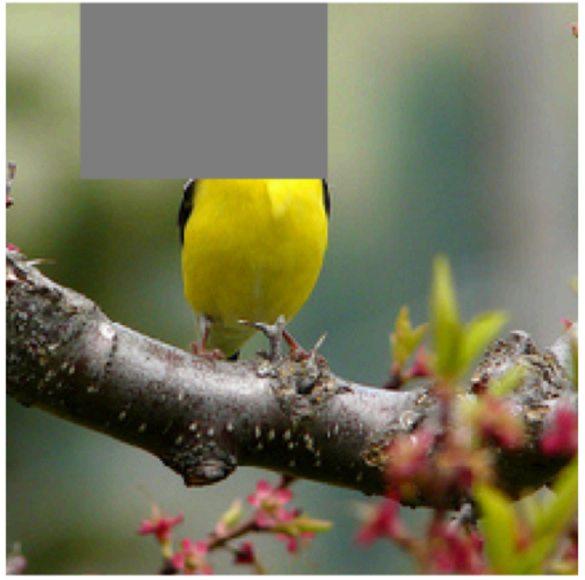
$$r_x \sim \text{Unif}\ (0, W)\,, \quad r_w = W\sqrt{1-\lambda},$$
$$r_y \sim \text{Unif}\ (0, H)\,, \quad r_h = H\sqrt{1-\lambda}$$

Implementation of Cutmix

	ResNet-50	Mixup [48]	Cutout [3]	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4
ImageNet Cls (%)	76.3 (+0.0)	77.4 (+1.1)	77.1 (+0.8)	**78.6** **(+2.3)**
ImageNet Loc (%)	46.3 (+0.0)	45.8 (-0.5)	46.7 (+0.4)	**47.3** **(+1.0)**
Pascal VOC Det (mAP)	75.6 (+0.0)	73.9 (-1.7)	75.1 (-0.5)	**76.7** **(+1.1)**

Accuracy of methods in different models

CutOut

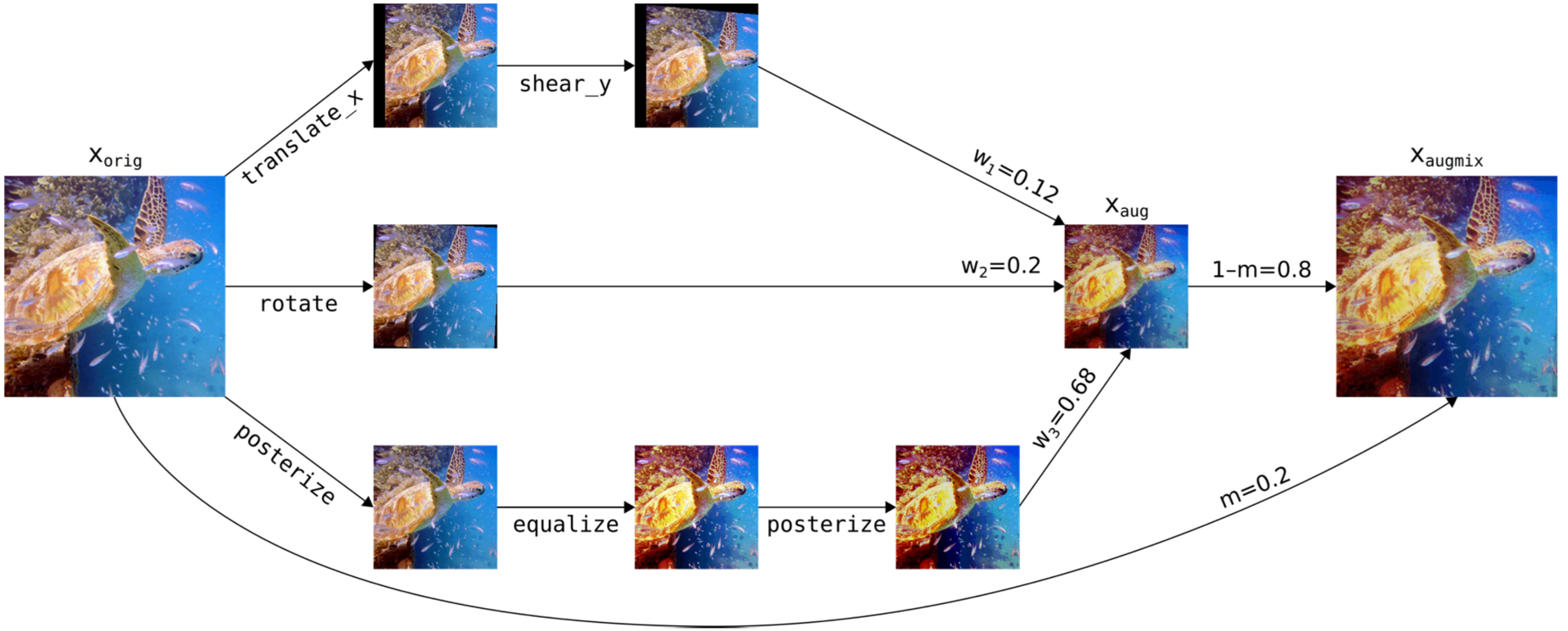


MixUp

CutMix

AugMix

Visual difference between Mixup, Cutout, Cutmix, Augmix



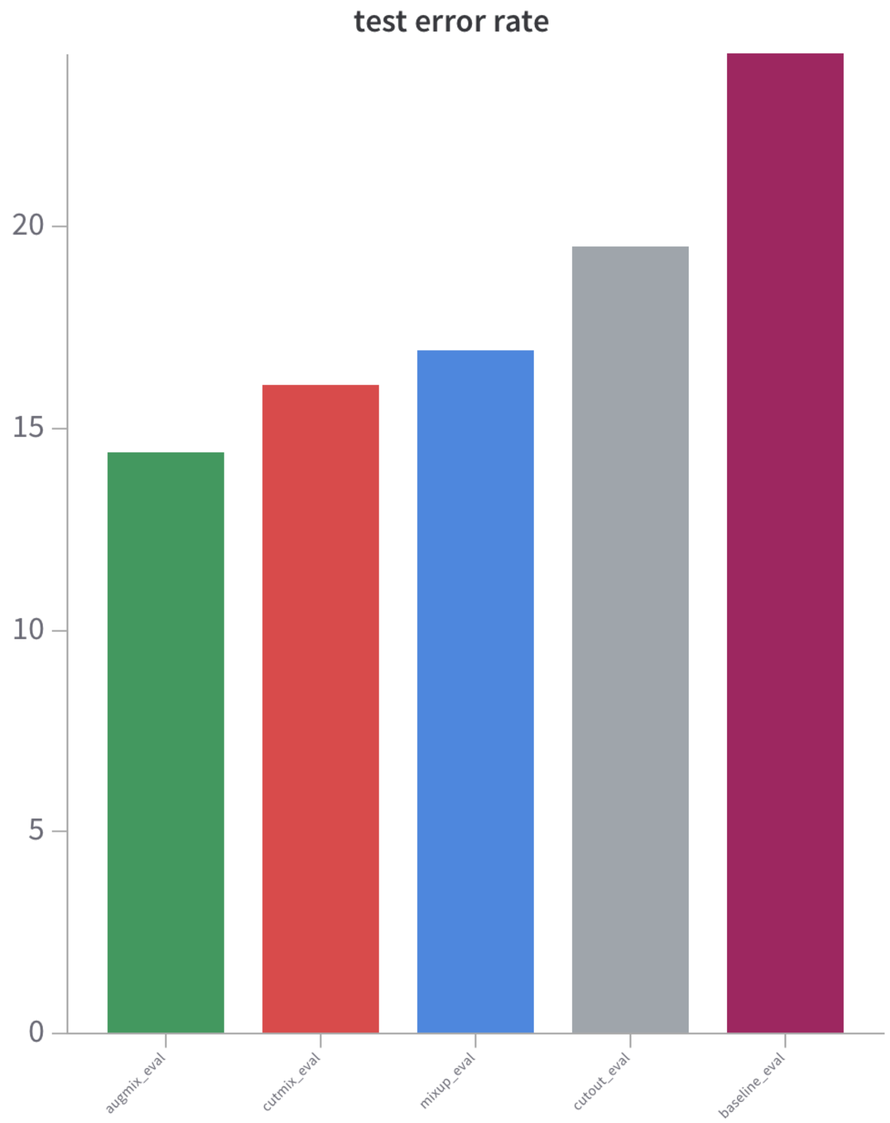
Implementation of Augmix includes augmentation, mixing and Jensen-Shannon Divergence Consistency Loss

$$\mathcal{L}(p_{\text{orig}}, y) + \lambda\, \mathbf{JS}(p_{\text{orig}}; p_{\text{augmix1}}; p_{\text{augmix2}})$$

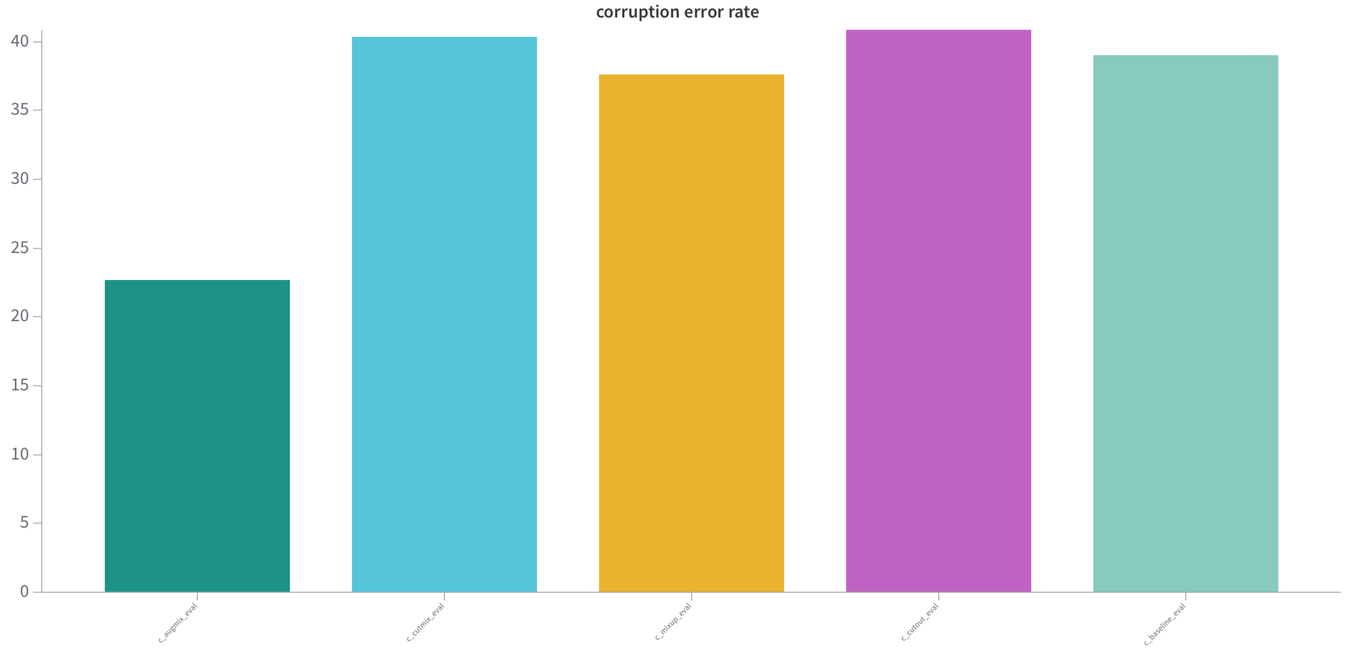
$$M = (p_{\text{orig}} + p_{\text{augmix1}} + p_{\text{augmix2}})/3$$

$$\mathbf{JS}(p_{\text{orig}}; p_{\text{augmix1}}; p_{\text{augmix2}}) = \frac{1}{3}\Big(\mathbf{KL}[p_{\text{orig}} \| M] + \mathbf{KL}[p_{\text{augmix1}} \| M] + \mathbf{KL}[p_{\text{augmix2}} \| M]\Big)$$

Jensen-Shannon Divergence Consistency Loss



Error rate of different methods on CIFAR-10 dataset



Error rate of different methods on corrupted CIFAR-10 dataset

References

https://arxiv.org/pdf/1708.04552v2.pdf

https://arxiv.org/pdf/1905.04899.pdf

https://arxiv.org/pdf/1710.09412.pdf

https://arxiv.org/pdf/1912.02781.pdf

https://encord.com/blog/data-augmentation-guide/

https://blog.pangeanic.com/audio-data-augmentation-techniques-and-methods

https://www.datacamp.com/tutorial/complete-guide-data-augmentation

https://habr.com/ru/companies/smartengines/articles/499566/

https://towardsdatascience.com/test-time-augmentation-tta-and-how-to-perform-it-with-keras-4ac19b67fb4d