

Аугментации данных

Зачем нужны аугментации

- Повышают размер тренировочной выборки
- Борются с переобучением
- “Сглаживают” модель
- Позволяют получить принципиально другие данные

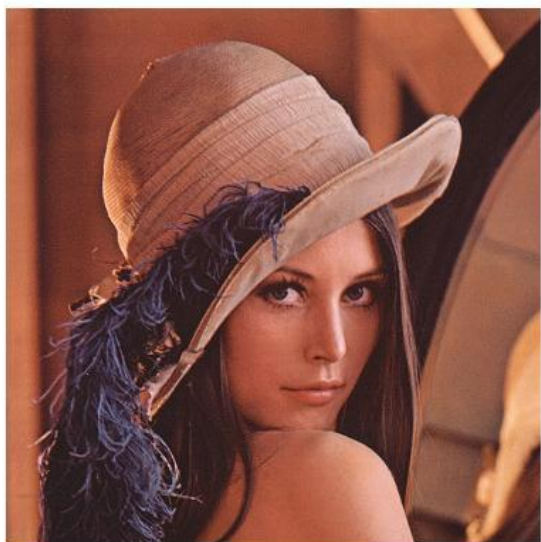
Flip augmentation



Rotation with borders



Rotation with crop/interpolation(without borders)



Input

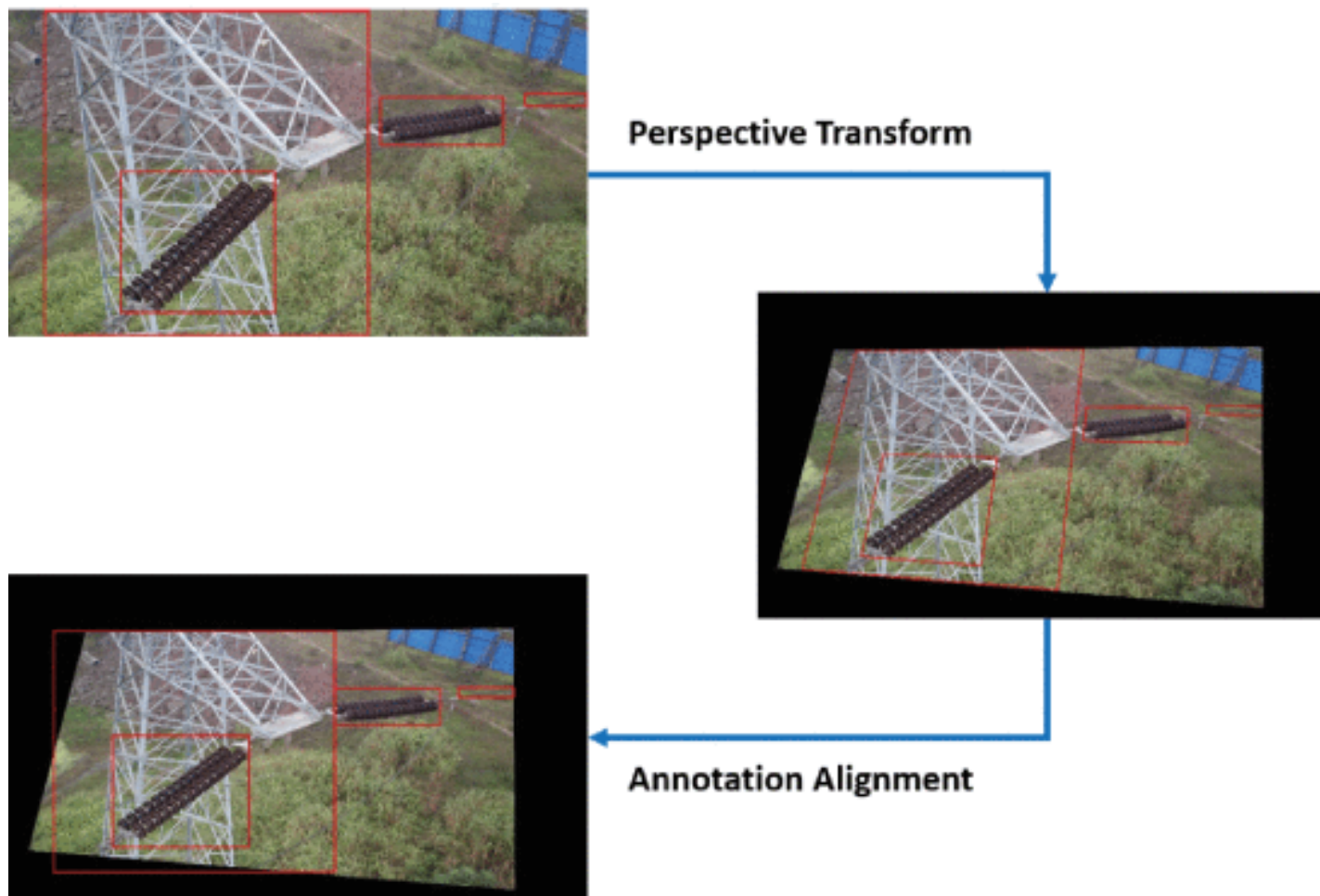


A



B

Perspective augmentation



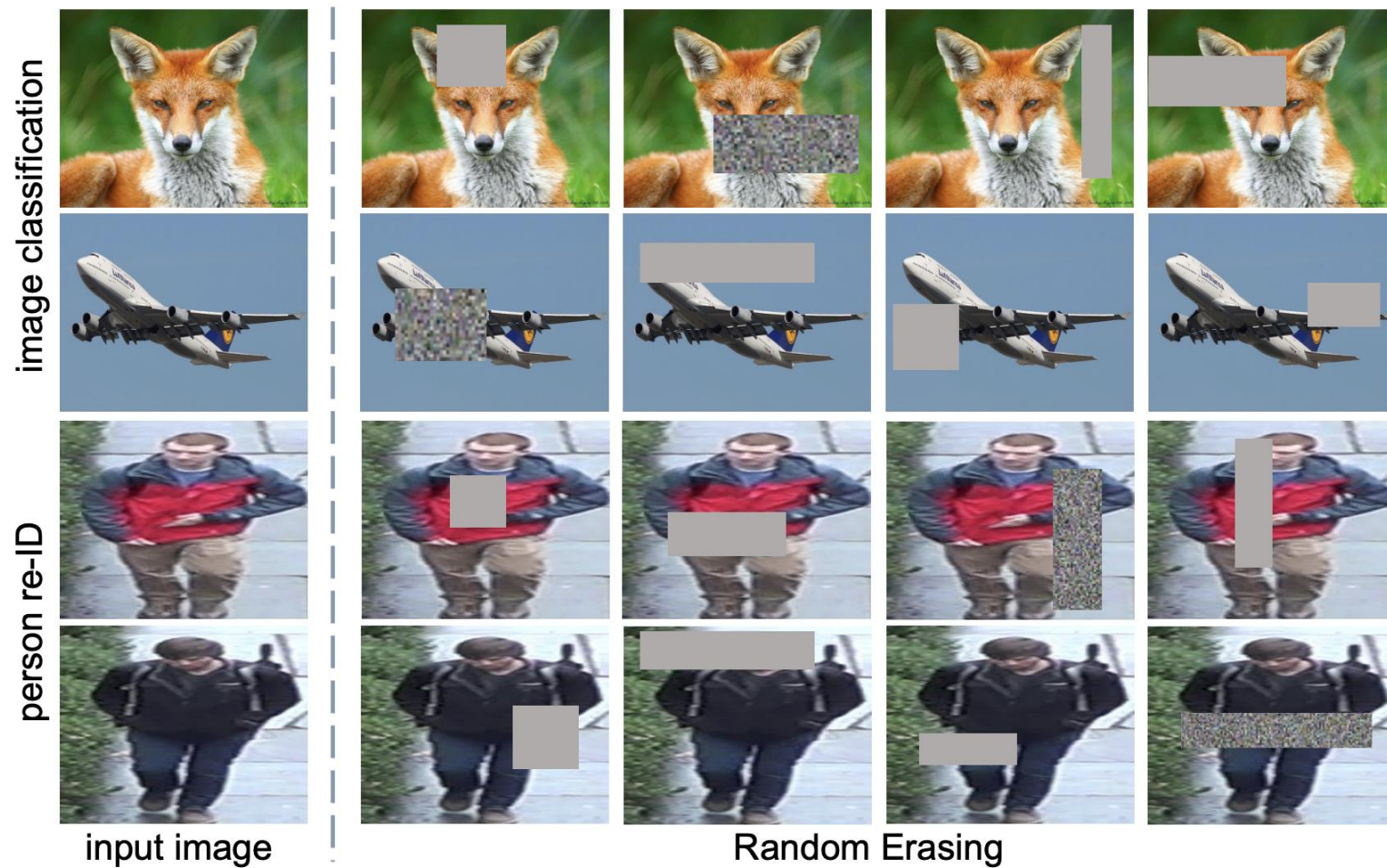
Crop augmentation



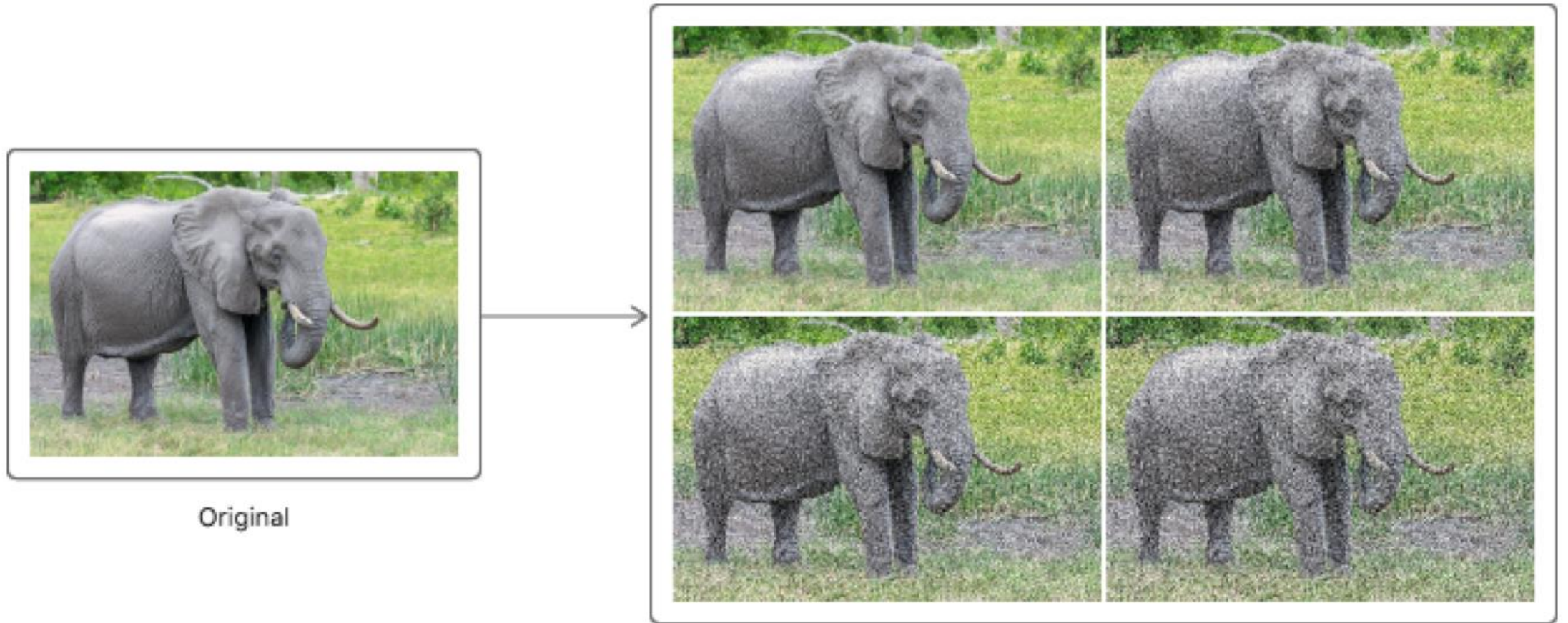
Original



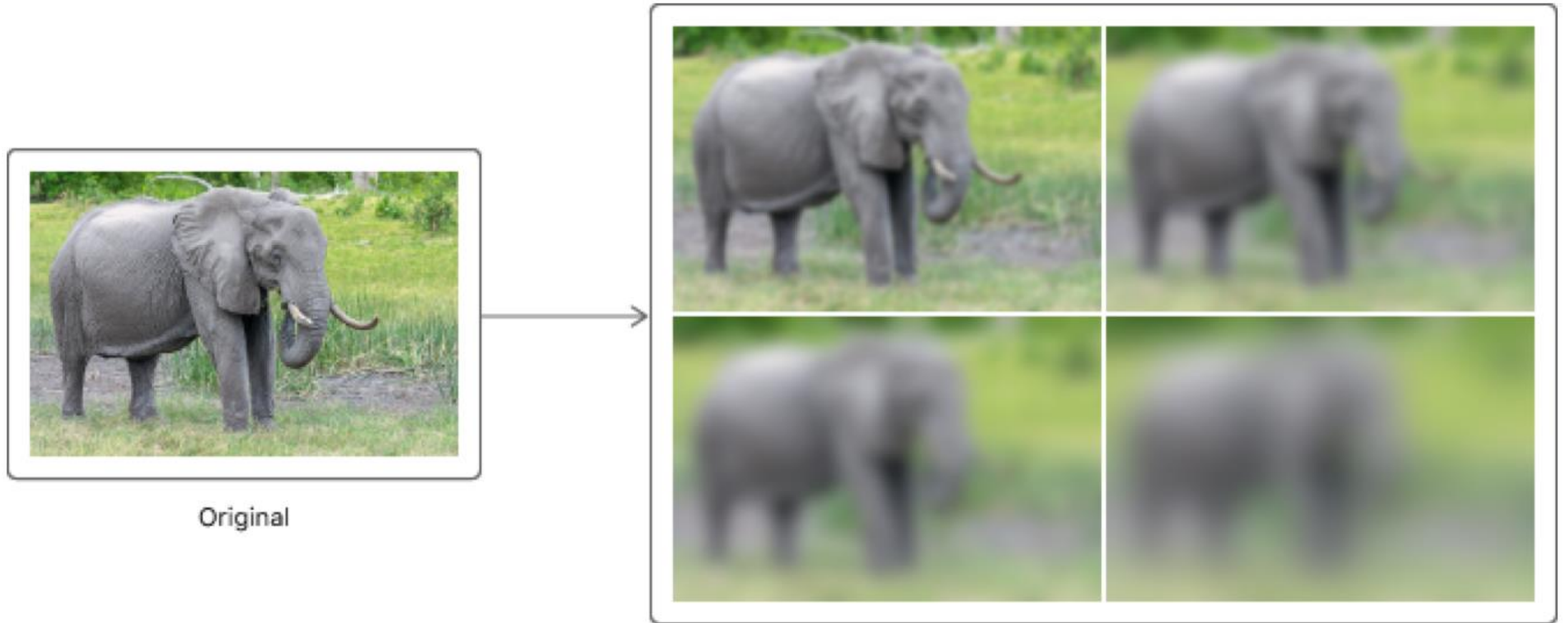
Cutout(erase) augmentation



Noise augmentation



Blur augmentation



ColorJitter augmentation



Не реклама

- Иногда нужно делать аугментации сохраняя bounding boxes или другие точки на изображении
- По моим наблюдениям ещё и работает быстрее чем `torch.transforms`
- Ну и просто удобно работать
- [alumentations](#)



Alumentations

Mixup augmentation

- Модели очень любят выучивать обучающую выборку, вместо её обобщения
- Получаем, что в данных из обучающей выборки работает хорошо, а уже даже в небольшой окрестности плохо
- Давайте заполним пустоту ~~в сердце~~ между точками обучающей выборки, чтобы модель вела себя более “гладко” в этой области

Mixup augmentation

- Давайте возьмём 2 точки из обучающей выборки и добавим третью:

$$\hat{x} = \lambda x_i + (1 - \lambda)x_j$$

$$\hat{y} = \lambda y_i + (1 - \lambda)y_j$$

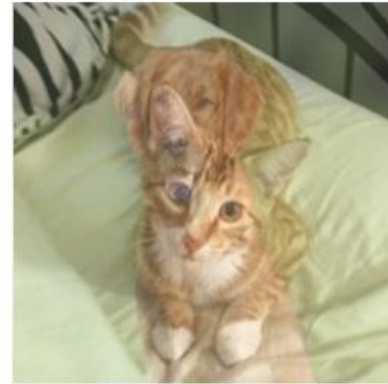
- Лямбду стоит брать как случайную величину Beta(alpha=0.2)

Міхур візуально

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

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

Image



Label

[1.0, 0.0]
cat dog

[0.0, 1.0]
cat dog

[0.7, 0.3]
cat dog

Міхир немого математики

$$R(f) = \int \ell(f(x), y) dP(x, y).$$

$$P_\delta(x, y) = \frac{1}{n} \sum_{i=1}^n \delta(x = x_i, y = y_i),$$

$$R_\delta(f) = \int \ell(f(x), y) dP_\delta(x, y) = \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i).$$

Міхир немого математики

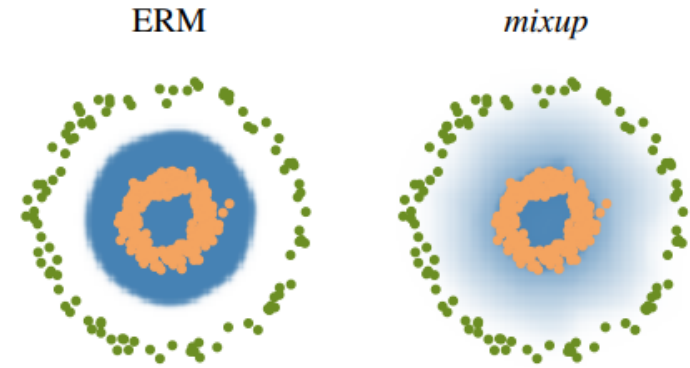
$$P_\nu(\tilde{x}, \tilde{y}) = \frac{1}{n} \sum_{i=1}^n \nu(\tilde{x}, \tilde{y} | x_i, y_i), \quad \nu(\tilde{x}, \tilde{y} | x_i, y_i) = \mathcal{N}(\tilde{x} - x_i, \sigma^2) \delta(\tilde{y} - y_i)$$

$$R_\nu(f) = \frac{1}{m} \sum_{i=1}^m \ell(f(\tilde{x}_i), \tilde{y}_i).$$

Міхур пример реализации

```
# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    optimizer.zero_grad()
    loss(net(x), y).backward()
    optimizer.step()
```

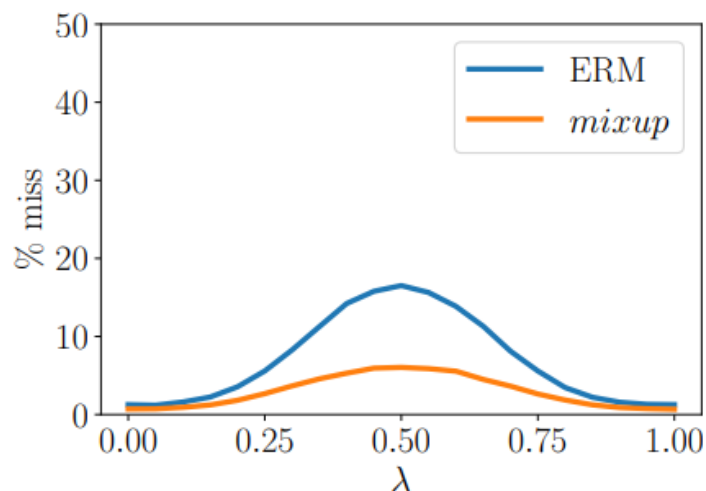
(a) One epoch of *mixup* training in PyTorch.



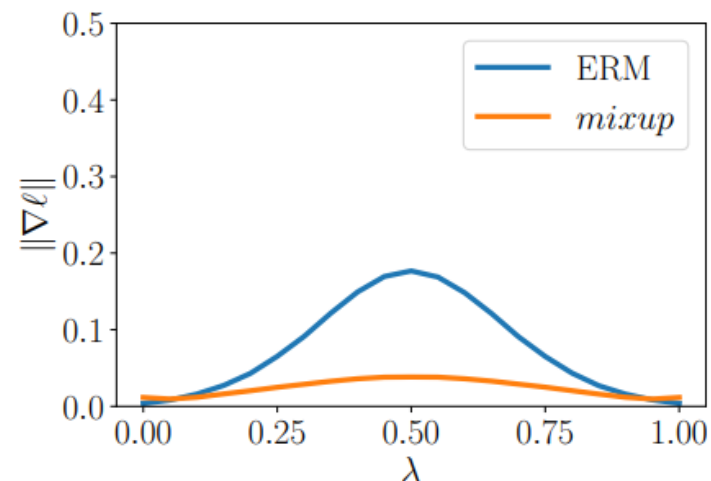
(b) Effect of *mixup* ($\alpha = 1$) on a toy problem. Green: Class 0. Orange: Class 1. Blue shading indicates $p(y = 1|x)$.

Figure 1: Illustration of *mixup*, which converges to ERM as $\alpha \rightarrow 0$.

Міхур результати



(a) Prediction errors in-between training data. Evaluated at $x = \lambda x_i + (1 - \lambda)x_j$, a prediction is counted as a “miss” if it does not belong to $\{y_i, y_j\}$. The model trained with *mixup* has fewer misses.



(b) Norm of the gradients of the model w.r.t. input in-between training data, evaluated at $x = \lambda x_i + (1 - \lambda)x_j$. The model trained with *mixup* has smaller gradient norms.

Mixup результаты

Model	Method	Epochs	Top-1 Error	Top-5 Error
ResNet-50	ERM (Goyal et al., 2017)	90	23.5	-
	<i>mixup</i> $\alpha = 0.2$	90	23.3	6.6
ResNet-101	ERM (Goyal et al., 2017)	90	22.1	-
	<i>mixup</i> $\alpha = 0.2$	90	21.5	5.6
ResNeXt-101 32*4d	ERM (Xie et al., 2016)	100	21.2	-
	ERM	90	21.2	5.6
	<i>mixup</i> $\alpha = 0.4$	90	20.7	5.3
ResNeXt-101 64*4d	ERM (Xie et al., 2016)	100	20.4	5.3
	<i>mixup</i> $\alpha = 0.4$	90	19.8	4.9
ResNet-50	ERM	200	23.6	7.0
	<i>mixup</i> $\alpha = 0.2$	200	22.1	6.1
ResNet-101	ERM	200	22.0	6.1
	<i>mixup</i> $\alpha = 0.2$	200	20.8	5.4
ResNeXt-101 32*4d	ERM	200	21.3	5.9
	<i>mixup</i> $\alpha = 0.4$	200	20.1	5.0

Table 1: Validation errors for ERM and *mixup* on the development set of ImageNet-2012.

Cutmix augmentation

Original
samples



Mixup



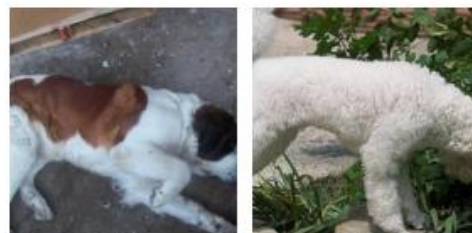
Cutout



Cutmix

Cutmix почему лучше

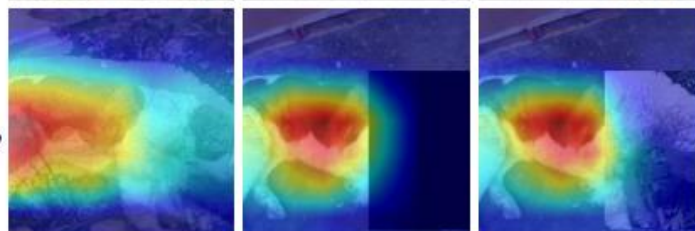
Original
Samples



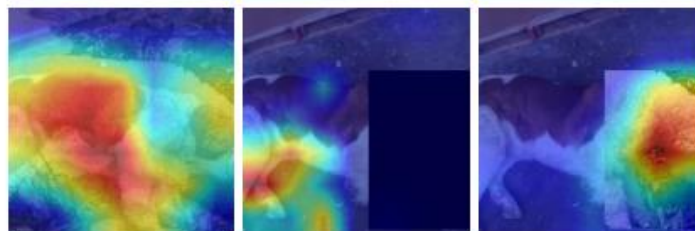
Input
Image



CAM for
'St. Bernard'



CAM for
'Poodle'

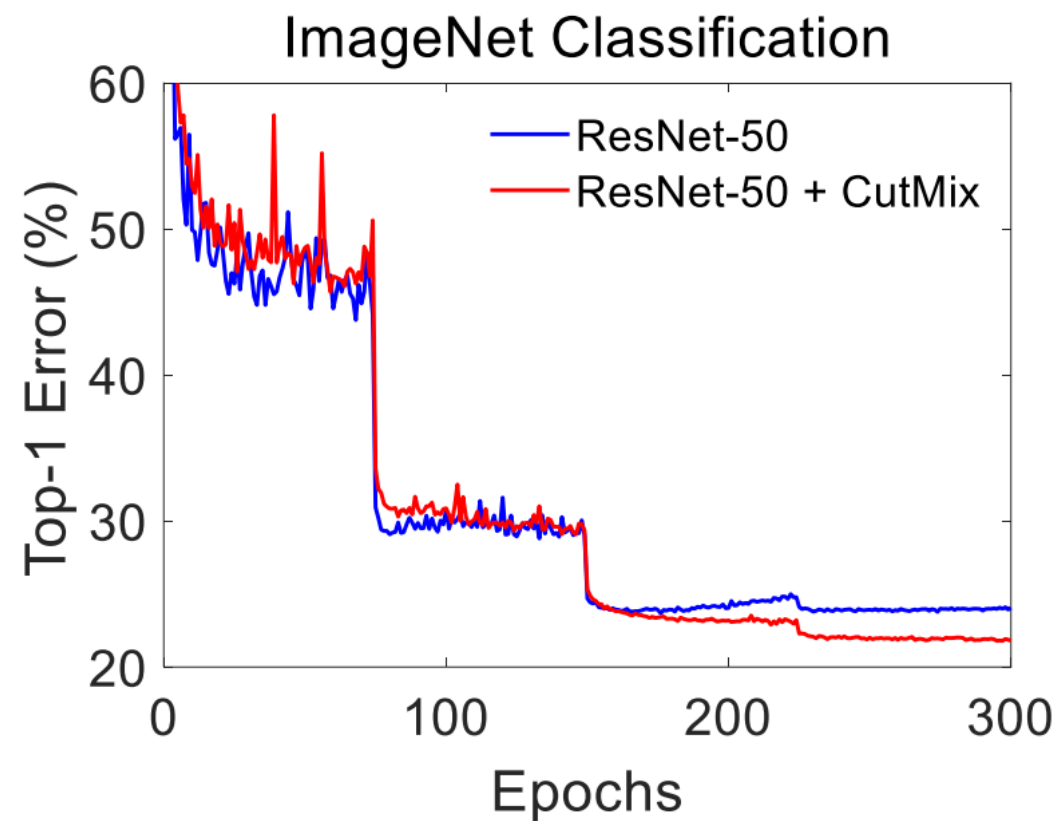
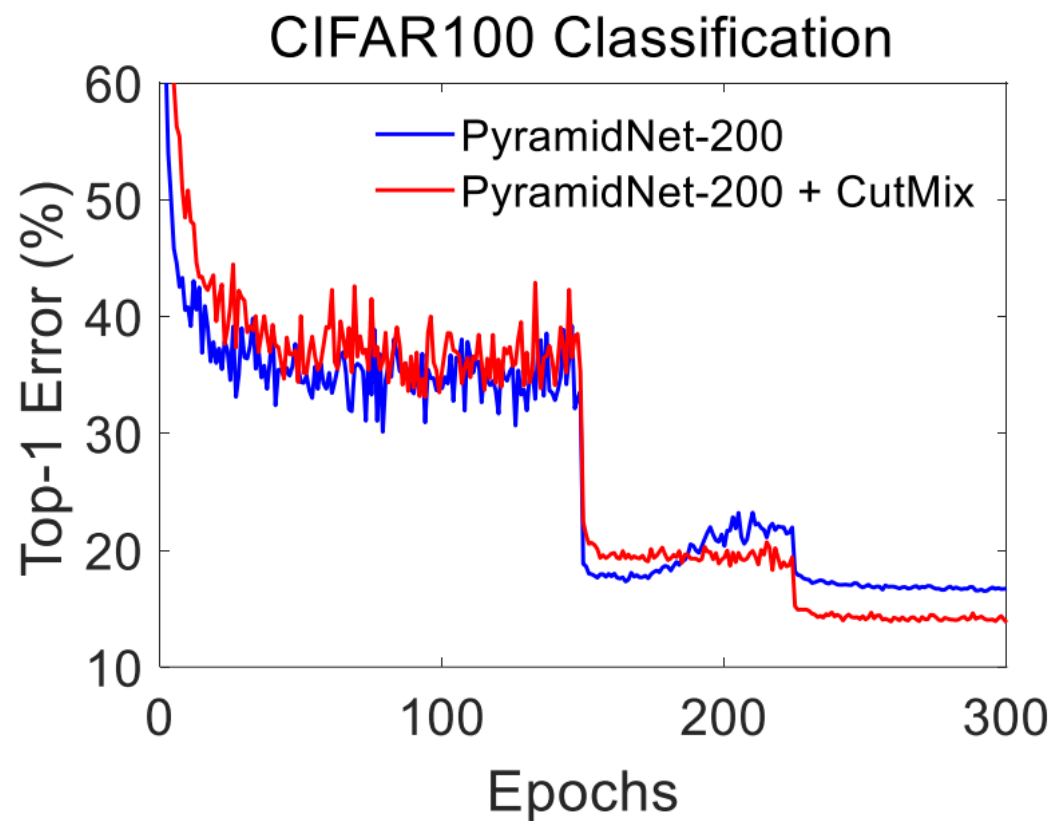


Mixup

Cutout

CutMix

Cutmix результаты



Cutmix результаты

Model	# Params	Top-1 Err (%)	Top-5 Err (%)
ResNet-152*	60.3 M	21.69	5.94
ResNet-101 + SE Layer* [15]	49.4 M	20.94	5.50
ResNet-101 + GE Layer* [14]	58.4 M	20.74	5.29
ResNet-50 + SE Layer* [15]	28.1 M	22.12	5.99
ResNet-50 + GE Layer* [14]	33.7 M	21.88	5.80
ResNet-50 (Baseline)	25.6 M	23.68	7.05
ResNet-50 + Cutout [3]	25.6 M	22.93	6.66
ResNet-50 + StochDepth [17]	25.6 M	22.46	6.27
ResNet-50 + Mixup [48]	25.6 M	22.58	6.40
ResNet-50 + Manifold Mixup [42]	25.6 M	22.50	6.21
ResNet-50 + DropBlock* [8]	25.6 M	21.87	5.98
ResNet-50 + Feature CutMix	25.6 M	21.80	6.06
ResNet-50 + CutMix	25.6 M	21.40	5.92

Table 3: ImageNet classification results based on ResNet-50 model. ‘*’ denotes results reported in the original papers.

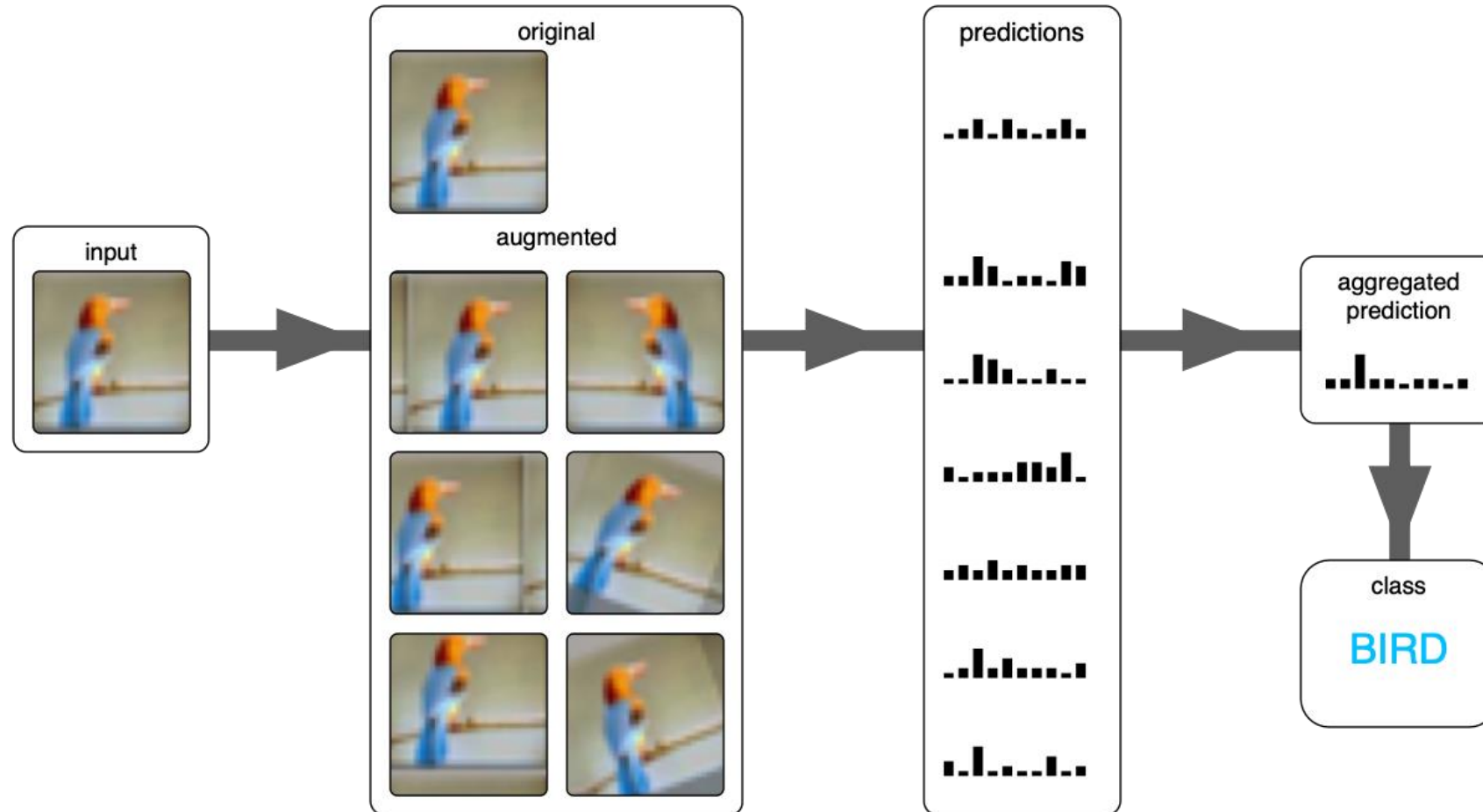
Model	# Params	Top-1 Err (%)	Top-5 Err (%)
ResNet-101 (Baseline) [12]	44.6 M	21.87	6.29
ResNet-101 + Cutout [3]	44.6 M	20.72	5.51
ResNet-101 + Mixup [48]	44.6 M	20.52	5.28
ResNet-101 + CutMix	44.6 M	20.17	5.24
ResNeXt-101 (Baseline) [45]	44.1 M	21.18	5.57
ResNeXt-101 + CutMix	44.1 M	19.47	5.03

Table 4: Impact of CutMix on ImageNet classification for ResNet-101 and ResNext-101.

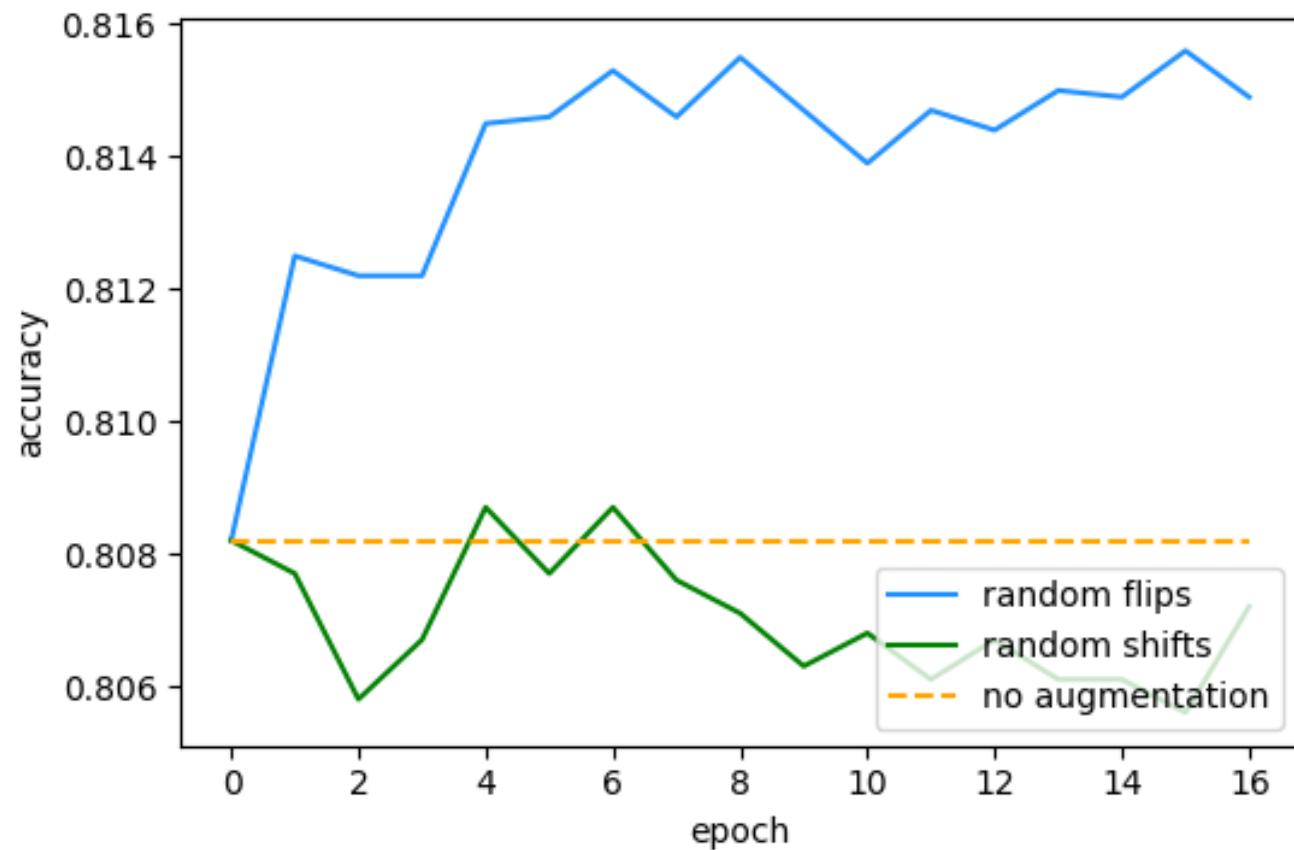
Cutmix/Mixup что же лучше

- Если выбирать что-то одно, то лучший скор показывает Cutmix
- Но никто не запрещает совмещать
- Поэтому обычно делают связку Cutmix+Mixup и получают качество ещё лучше

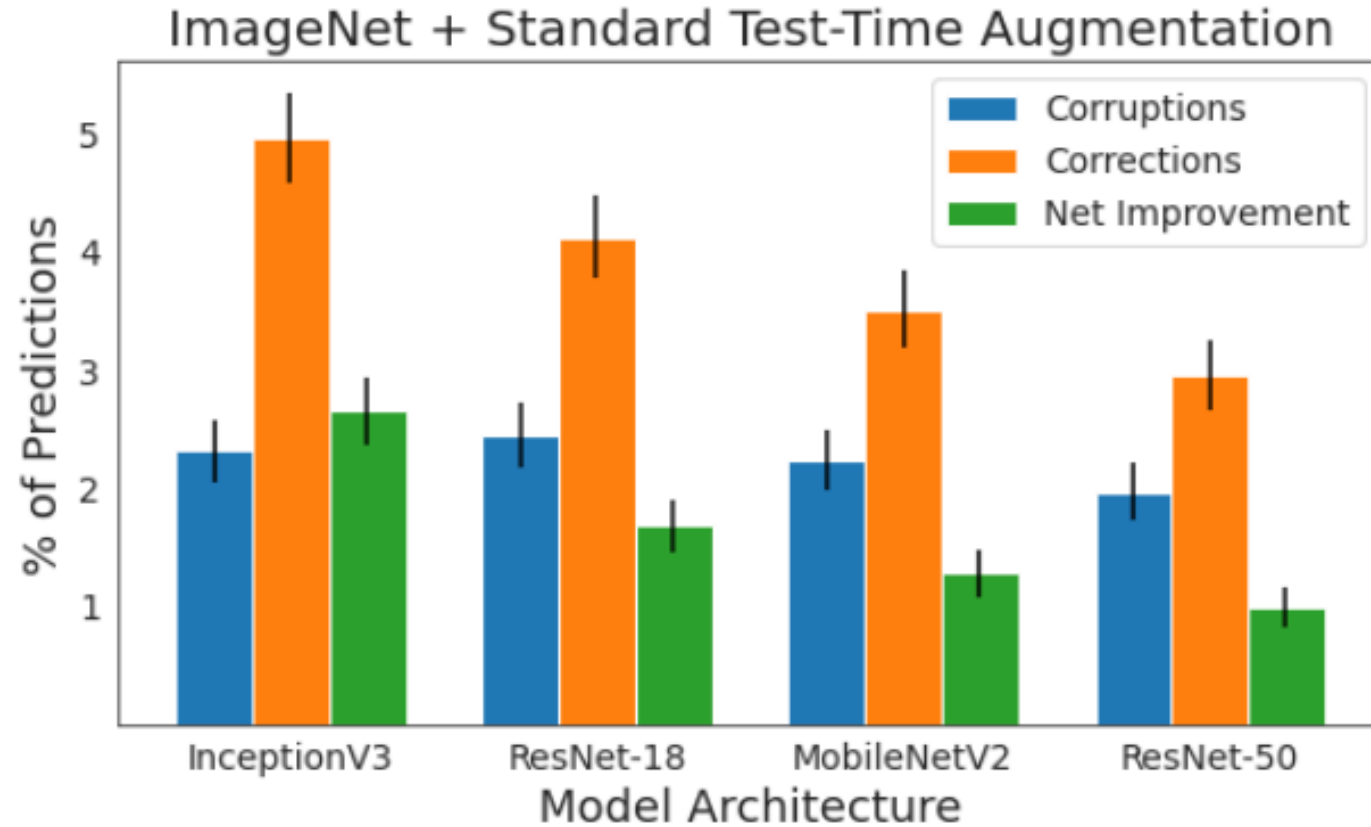
Test-time augmentation



Test-time augmentation подводные камни



Test-time augmentation результаты



Заключение

- Аугментации помогают расширить тренировочную выборку: как просто сделать её больше, так и добавить туда принципиально новые примеры
- Аугментации позволяют сглаживать функции и не давать модели переобучаться
- Применение аугментаций хоть и замедляет обучение в начале, но позволяет добиться лучших результатов в конце
- Применяя аугментации к тестовым данным и усредняя результат можем получать улучшенные предсказания