Mixup and Application

신욱수

shinoops@korea.ac.kr

발표자 소개

- 신욱수 (Wooksoo Shin)
  - Data Mining & Quality Analytics Lab
  - 박사과정 (2019.03 ~ present)

- Research Interest
  - Machine Learning Algorithm including Image Classification
  - Complexity Analysis
  - Scheduling & Allocation Algorithm

- Contact
  - E-mail : shinoops@korea.ac.kr

목차

Mixup

Published as a conference paper at ICLR 2018

mixup: BEYOND EMPIRICAL RISK MINIMIZATION

Hongyi Zhang
MIT

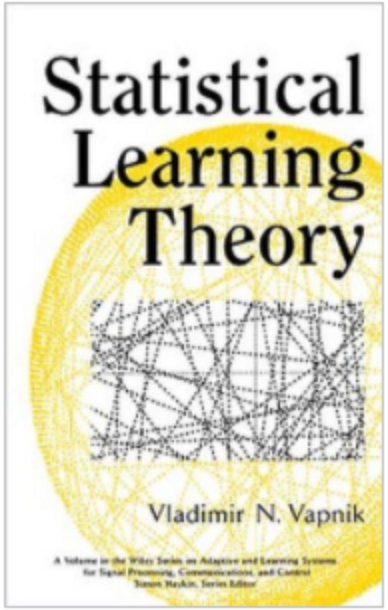
Moustapha Cisse, Yann N. Dauphin, David Lopez-Paz*
FAIR

ABSTRACT

Large deep neural networks are powerful, but exhibit undesirable behaviors such as memorization and sensitivity to adversarial examples. In this work, we propose *mixup*, a simple learning principle to alleviate these issues. In essence, *mixup* trains a neural network on convex combinations of pairs of examples and their labels. By doing so, *mixup* regularizes the neural network to favor simple linear behavior in-between training examples. Our experiments on the ImageNet-2012, CIFAR-10, CIFAR-100, Google commands and UCI datasets show that *mixup* improves the generalization of state-of-the-art neural network architectures. We also find that *mixup* reduces the memorization of corrupt labels, increases the robustness to adversarial examples, and stabilizes the training of generative adversarial networks.

* 2021.3.30 현재 1567회 인용

Empirical Risk Minimization ?



Risk $R(f) = E[L(f(x), y)] = \int L(f(x), y)dP(x, y)$

, where $(x, y) \sim P(X, Y)$, Loss function l, Risk R , $f \sim \mathcal{F}$

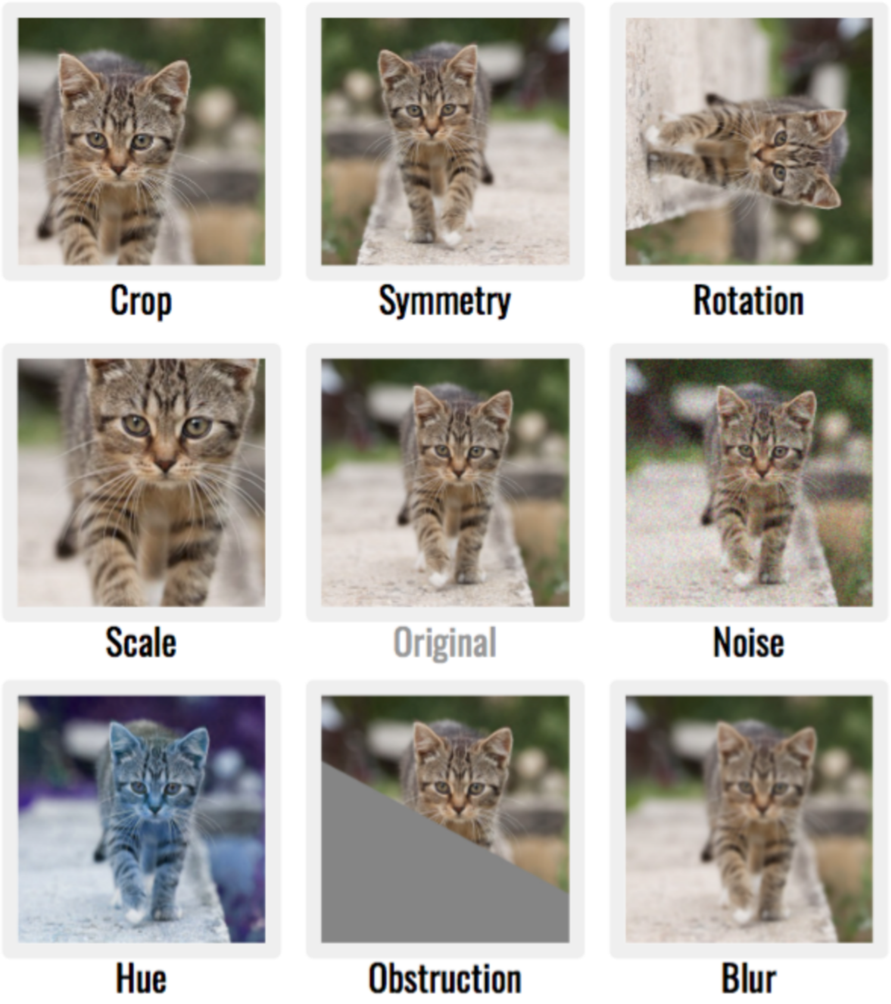
Risk Minimization $f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}} R(f)$

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

Empirical Risk Minimization $\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} R_\delta(f)$

논문에서 의미하는 augmentation이란?

일반적으로 보이는 다음 이미지들은 아님



Different augmentations

Src : https://medium.com/@wolframalphav1.0/easy-way-to-improve-image-classifier-performance-part-1-mixup-augmentation-with-codes-33288db92de5

Mixup이 제시한 기여점

Contribution Motivated by these issues, we introduce a simple and data-agnostic data augmentation routine, termed *mixup* (Section 2). In a nutshell, *mixup* constructs virtual training examples

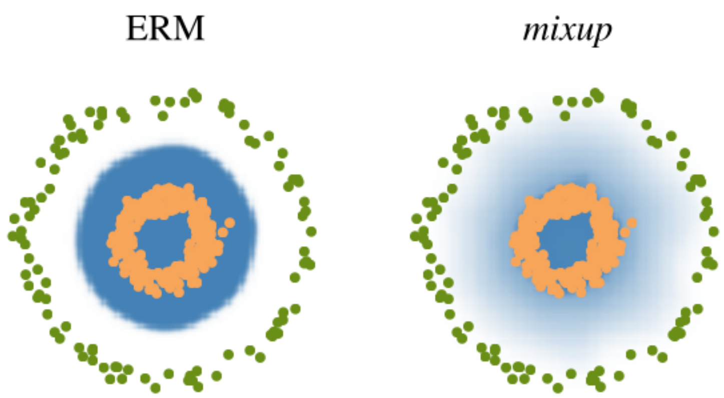
$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \qquad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j, \qquad \text{where } y_i, y_j \text{ are one-hot label encodings}$$

(x_i, y_i) and (x_j, y_j) are two examples drawn at random from our training data, and $\lambda \in [0, 1]$. Therefore, *mixup* extends the training distribution by incorporating the prior knowledge that linear interpolations of feature vectors should lead to linear interpolations of the associated targets. *mixup* can be implemented in a few lines of code, and introduces minimal computation overhead.

Simple Code Sample & 결과 시각화

```python
# 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.

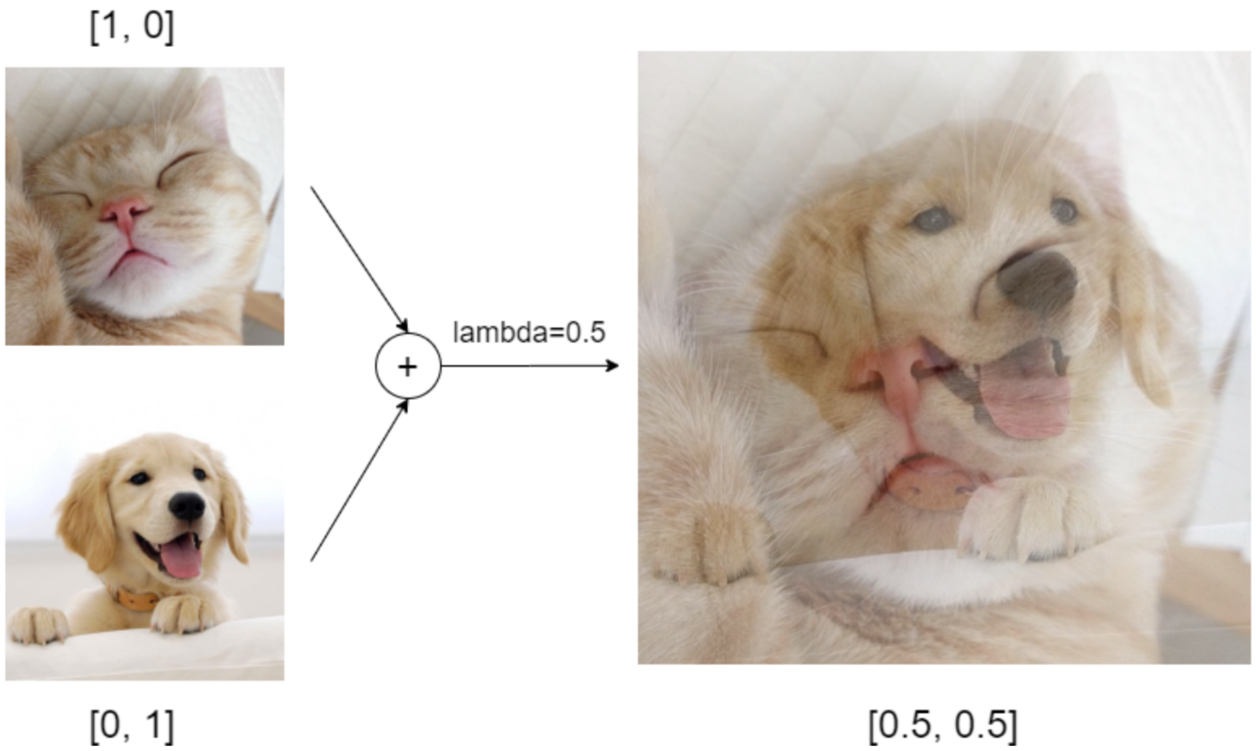


(b) Effect of *mixup* ($\alpha = 1$) on a toy problem. Green: Class 0. Orange: Class 1. Blue shading indicates $p(y = 1|x)$.

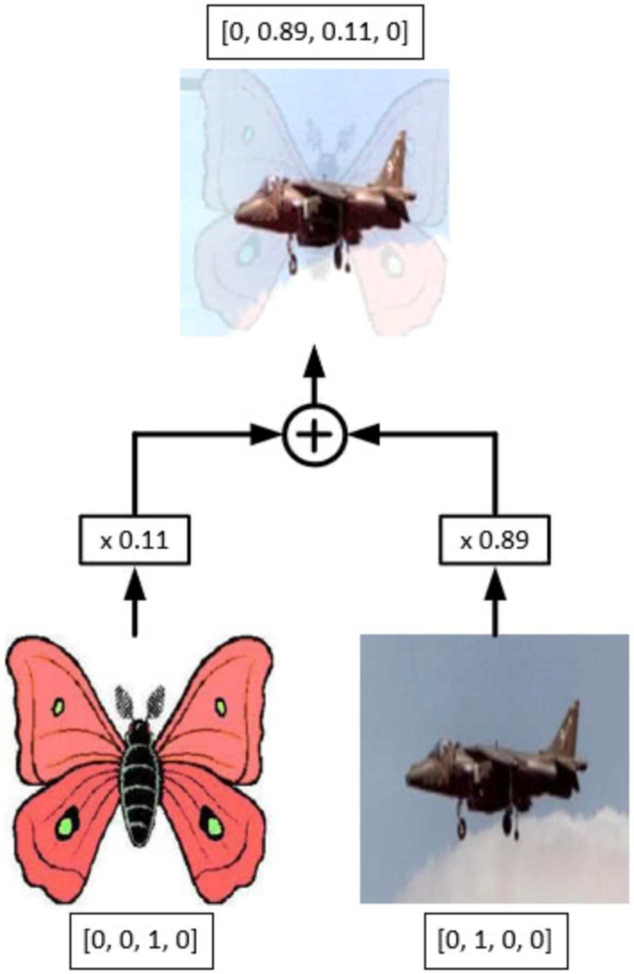
Figure 1: Illustration of *mixup*, which converges to ERM as $\alpha \to 0$.

몇가지 Mixup sample

Mixup in Action



Mix-up works by blending 2 images with alpha % from image_1 and (1-alpha) % from image_2

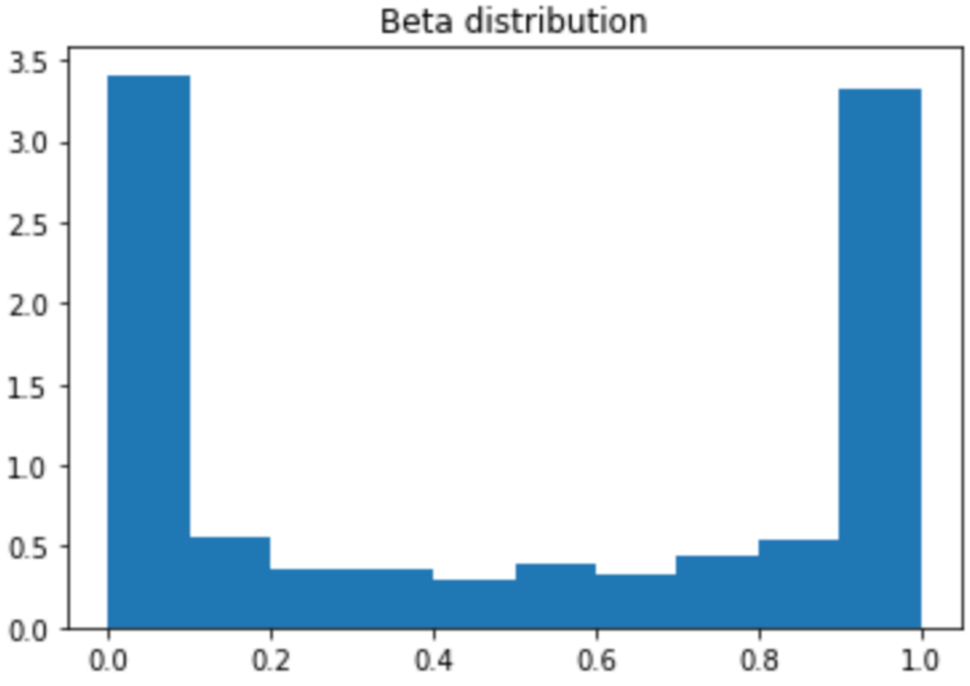


Src : https://medium.com/@wolframalphav1.0/easy-way-to-improve-image-classifier-performance-part-1-mixup-augmentation-with-codes-33288db92de5

λ Distribution 샘플

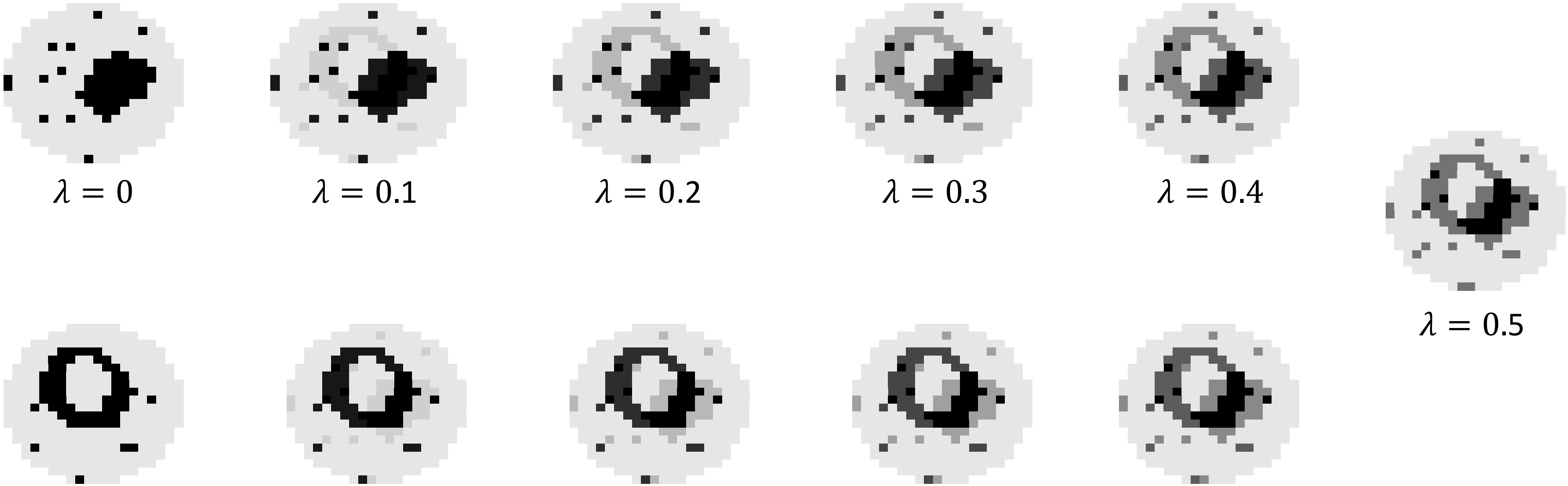
$\alpha \in [0.1; 0.4]$ leads to improved performance, smaller α creates less mixup effect, whereas, for large α, mixup leads to underfitting.

As you can see in the following graph, given a small $\alpha = 0.2$, beta distribution samples more values closer to either 0 and 1, making the mixup result closer to either one of the two examples.



λ 의 변화에 따른 Mixup 효과

$$\lambda x_1 + (1 - \lambda) x_2$$



$\lambda = 0$ $\lambda = 0.1$ $\lambda = 0.2$ $\lambda = 0.3$ $\lambda = 0.4$

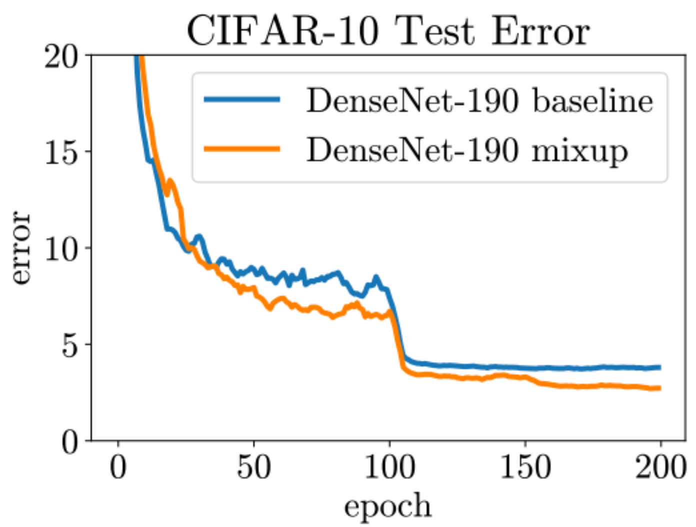
$\lambda = 0.5$

$\lambda = 1$ $\lambda = 0.9$ $\lambda = 0.8$ $\lambda = 0.7$ $\lambda = 0.6$

Image Classification 적용 사례

Dataset	Model	ERM	*mixup*
CIFAR-10	PreAct ResNet-18	5.6	**4.2**
	WideResNet-28-10	3.8	**2.7**
	DenseNet-BC-190	3.7	**2.7**
CIFAR-100	PreAct ResNet-18	25.6	**21.1**
	WideResNet-28-10	19.4	**17.5**
	DenseNet-BC-190	19.0	**16.8**

(a) Test errors for the CIFAR experiments.



(b) Test error evolution for the best ERM and *mixup* models.

Figure 3: Test errors for ERM and *mixup* on the CIFAR experiments.

Voice Recognition 적용 사례

Model	Method	Validation set	Test set
LeNet	ERM	**9.8**	**10.3**
	mixup ($\alpha = 0.1$)	10.1	10.8
	mixup ($\alpha = 0.2$)	10.2	11.3
VGG-11	ERM	5.0	4.6
	mixup ($\alpha = 0.1$)	4.0	3.8
	mixup ($\alpha = 0.2$)	**3.9**	**3.4**

Figure 4: Classification errors of ERM and *mixup* on the Google commands dataset.

라벨 오류 비율에 따른 Mixup의 효과

Label corruption	Method	Test error		Training error	
		Best	Last	Real	Corrupted
20%	ERM	12.7	16.6	0.05	0.28
	ERM + dropout ($p = 0.7$)	8.8	10.4	5.26	83.55
	mixup ($\alpha = 8$)	**5.9**	6.4	2.27	86.32
	mixup + dropout ($\alpha = 4, p = 0.1$)	6.2	**6.2**	1.92	85.02
50%	ERM	18.8	44.6	0.26	0.64
	ERM + dropout ($p = 0.8$)	14.1	15.5	12.71	86.98
	mixup ($\alpha = 32$)	11.3	12.7	5.84	85.71
	mixup + dropout ($\alpha = 8, p = 0.3$)	**10.9**	**10.9**	7.56	87.90
80%	ERM	36.5	73.9	0.62	0.83
	ERM + dropout ($p = 0.8$)	30.9	35.1	29.84	86.37
	mixup ($\alpha = 32$)	25.3	30.9	18.92	85.44
	mixup + dropout ($\alpha = 8, p = 0.3$)	**24.0**	**24.8**	19.70	87.67

Table 2: Results on the corrupted label experiments for the best models.

Application of Mixup

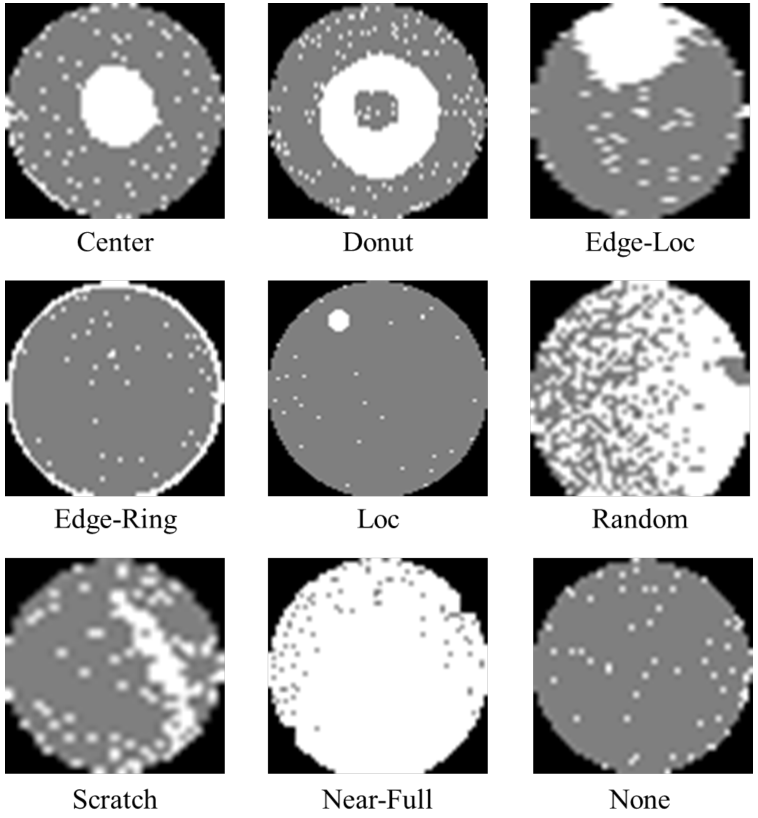
- Mixup을 어떻게 활용할 수 있을까요?

- 논문에서의 Mixup의 역할
    - Mixup을 사용하여 기존 Single label classification의 성능을 향상

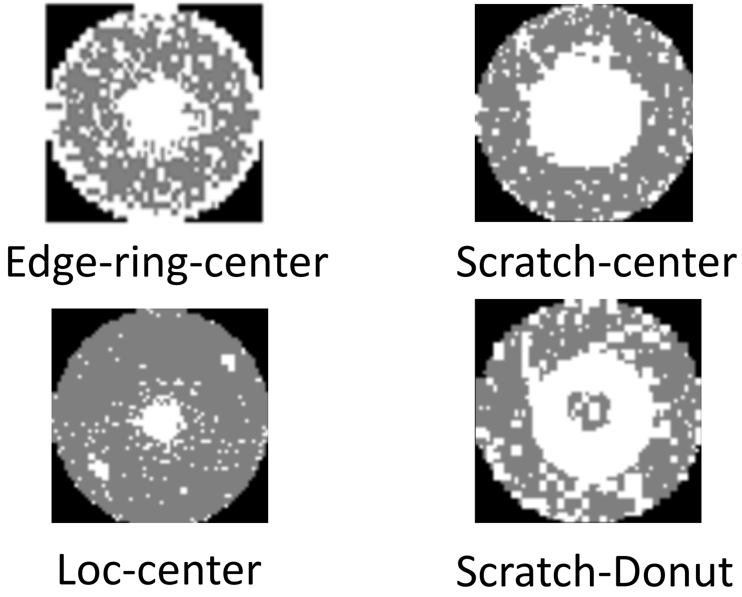
- Mixup의 중간과정과 본질을 생각하면
    - Multi-label 문제에 적합!

WM-811k

Single-type defect pattern (included)



Mixed-type defect pattern (not included)



진행 중 연구 소개

- **믹스업 활용시 복합 결함 데이터 분류 정확도 향상 ('20.11 대한산업공학회)**
  - 단순 학습 시 복합 패턴 분류 성능(baseline)은 72%
  - 믹스업 학습 시 복합 패턴 분류 성능 92% 수준으로 향상
  - Baseline 대비 정확도 20% 향상됨.

Accuracy		Alexnet	VGGnet-16	Resnet-18	Resnet-34	Resnet-50
단순 패턴 분류	단순 학습 (baseline)	0.977 (0.001)	0.981 (0.001)	0.978 (0.000)	0.977 (0.000)	0.977 (0.001)
	믹스업 학습 (제안방법)	0.970 (0.001)	0.974 (0.001)	0.971 (0.001)	0.970 (0.001)	0.972 (0.001)
복합 패턴 분류	단순 학습 (baseline)	0.715 (0.023)	0.678 (0.035)	0.738 (0.008)	0.750 (0.012)	0.749 (0.004)
	믹스업 학습 (제안 방법)	**0.872** (0.015)	**0.964** (0.020)	**0.928** (0.019)	**0.895** (0.013)	**0.902** (0.011)

단순 패턴 분류 성능은 거의 유지

복합 패턴 분류 성능은 대폭 향상

- **개선된 Mixup 방법 연구 및 다양한 테스트 진행 중**

감사합니다.