

Classification 성능 향상을 위한 다양한 Mixed Base Data Augmentation의 비교 및 분석

장원이

서울대학교 컴퓨터공학부

Abstract

양질의 데이터 확보의 어려움 때문에 Deep Learning을 위해 다양한 Data Augmentation 기법들이 연구되어 활용되고 있다. 본 프로젝트에서는 Image와 관련된 학습을 할 때, Image 자체를 변형 시켜 데이터를 증강하는 방법들에 대해 다룬다. Cutout, Mixup, Cutmix 등과 같은 기법들에 대해 알아보며, 이들을 활용하여 CIFAR-100 Dataset에서의 Classification을 학습 시켜 비교 분석한다. 각 연구논문에서 제공되는 소스 코드들을 활용하여 pytorch로 구현하였으며, 다음 주소에서 확인할 수 있다.

<https://github.com/jammythedreamer/thesis.git>

1 INTRODUCTION

Deep Learning에 있어서 방대한 데이터는 필수적이다. 하지만, 학습에 적합한 데이터를 구하는 것은 쉽지 않으며, 데이터를 구했다 하더라도 Labeling을 하는데도 많은 작업이 요구된다. 그렇기 때문에 한정된 데이터에서 더욱더 높은 정확도의 네트워크 학습을 하기 위해 Data Augmentation(데이터 증강) 기법들이 사용된다. Image에 대해서는 flip, crop, gray scale, rotate, shift, rescale, shear, stretch 등을 통해 하나의 Image로 여러 Image를 만들어 낼 수 있다. 이 밖에도 mixed-based 데이터 증강 기법들도 있는데, Mixup[3], Cutmix[4], Augmix[5] 등이 그렇다. Augmix의 경우는 하나의 Image를 사용하는 기법이기에 본 프로젝트에서는 다루지 않을 것이다. 그 외에도 Cutmix의 이전 기법인 Cutout[2]을 구현하여 비교군으로 이용할 것이다. 추가로 기존의 연구로 증명된 방법이 아닌 Divmix라는 방법과, Mixup과 Cutmix를 합친 Cutmixup를 제안하고 구현하여 실험을 진행하고 비교 분석할 것이다.

2 BACKGROUND

본 프로젝트의 실험은 기존에 사용되는 많은 방법을 바탕으로 진행되었다. 정확한 비교분석을 위해서는 모든 경우에 대해 진행해야 하지만, 현실적으로는 불가능하기에 일반적으로 자주 사용되고 성능이 좋다고 알려진 것들을 선정하여 사용하였다. 다음은 선정된 방법들과 그에 대한 간단한 설명, 그리고 선정 이유에 대한 내용이다.

2.1 CNN Architecture

ResNet[1]: Residual Network의 준말로써, 신호 표현을 직접하는 대신, 잔류(residual) 표현 함수를 학습함으로써, 최대 152개까지의 Layer로 구성된 매우 깊은 네트워크를 가질 수 있다. 이는 Residual Block을 기반으로 구현되며, 이전 Layer에서 다음 Layer로 입력을 fit하기 위해 입력의 수정 없이 shortcut connection을 도입하여 더 깊은 네트워크를 가질 수 있게 한다. Image Classification에서 좋은 성능을 보이고, 지금까지도 사용되기에 선정하였다. 여기서는 Layer가 50개인 ResNet-50을 사용한다.

2.2 Loss function

Cross Entropy: Image Classification에서 예측값이 [0,1]에 있을 때 성능을 측정할 때 사용한다. class가 M개이고, 정답 여부를 나타내는 y, 예측값을 p로 나타낼 때 식은 다음과 같다.

$$-\sum_{c=1}^M y_{o,c} \log(p_{o,c})$$

Multi class의 문제를 다루기 때문에 자주 사용되고 성능도 좋은 Cross Entropy Loss를 선정하였다.

2.3 Optimizer

SGD: Stochastic Gradient Descent의 약자로, 전체 데이터(Batch)에 대해 Loss function을 계산하는 대신, 일부 데이터(Mini-Batch)를 사용하여 Loss function을 계산한다. 전체 데이터에 대해 계산하는 Batch Gradient Descent에 비해 다소 부정확할 수 있으나, 계산속도가 훨씬 빠르고 같은 시간에 더 많은 step을 갈 수 있어 정확도를 높인다. ResNet-50을 사용하기에 적합하다고 생각하여 선정하였다.

2.4 Data Augmentation

본 프로젝트의 주제인 Data Augmentation의 여러 기법에 대한 간단한 설명과 이해를 돕기 위한 사진들이다. 아래의 사진들은 CIFAR-10의 이미지 중 임의의 이미지들에 대해 여러 기법들은 적용한 것이나, 실제로 학습을 진행할 때에는 batch size도 크고, 부가적인 작업도 있기에 조금 차이가 있을 수 있다.

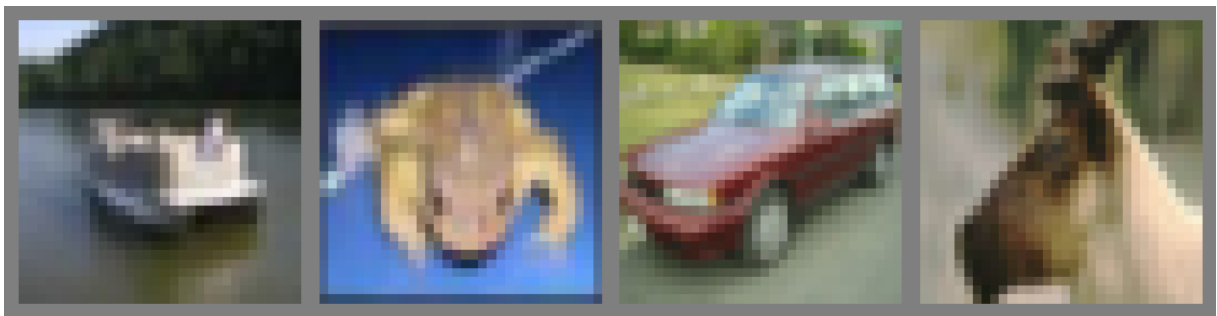


Figure 1: 원본 이미지(ship, frog, car, deer)

Cutout[2]: image에 일부분을 오려내는 방법이다. 정확히는 image의 특정 영역의 pixel 값을 모두 0으로 바꾸는 것이다. 기존의 연구를 통해 해당 기법의 성능은 지워지는 영역의 모양보다는 크기가 중요하다는 결과가 있었다. 따라서 본 실험에서는 32 x 32 pixel의 이미지에서 cutout length를 16으로 하여, 최대 16x16의 정사각형 영역을 들어내는 작업을 실행할 것이다. cutmix prob을 설정값으로 넣어 cutout을 하는 확률을 설정한다.

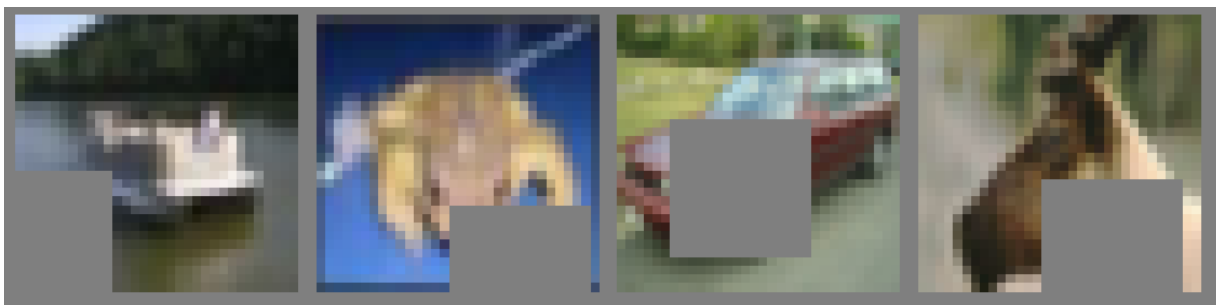


Figure 2: cutout 이미지(ship, frog, car, deer)

Mixup[3]: mixup은 mixed-based data augmentation 기법으로 2가지의 이미지를 섞는다. loss는 mixup을 통해 만들어진 이미지의 비율에 따라 label을 나눈 것처럼 계산한다. 아래의 예시에서는 이해를 돕기 위해 순차적인 mixup을 했는데, 본래의 알고리즘에서는 batch 안에서 랜덤한 이미지를 선택해 mixup한다. mixup alpha를 설정값으로 넣어 beta함수를 통해 mixup비율을 통해 랜덤성을 가진다.



Figure 3: mixup 이미지(ship-frog, frog-car, car-deer, deer-ship)

Cutmix[4]: cutmix는 cutout에서 일부 영역을 0으로 바꾸는 대신, 이 영역에 다른 이미지의 일부를 가져와 새로운 이미지를 만드는 것이다. 본 프로젝트에서는 cutmix length를 16으로 하였고, cutout prob을 설정값으로 넣어 cutmix의 확률을 설정한다. loss는 label을 영역의 넓이 비율만큼 나눈 것처럼 계산한다. mix하는 이미지는 batch에서 가져오며, random crop으로 가져오는 것이 아닌 cutout한 영역과 같은 영역을 가져온다.

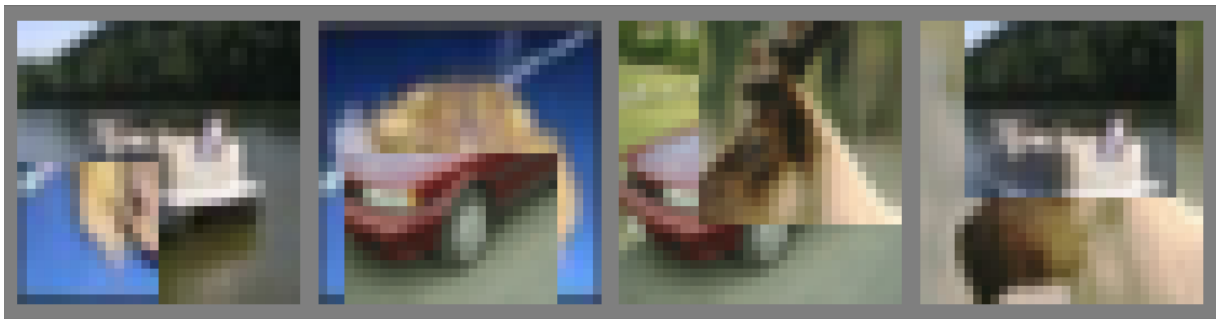


Figure 4: cutmix 이미지(ship+frog, frog+car, car+deer, deer+ship)

Cutmixup: mixup과 cutmix를 함께 사용하는 기법으로 이다. 본래의 cutmix에서 사용될 영역을 덮어쓰기가 아닌 mixup방식으로 적용한다. loss는 cutmix에서 사용된 영역의 넓이 비율에 따른 label 배분과 mixup에서 사용된 비중에 따른 label 계산을 합쳐서 계산한다.

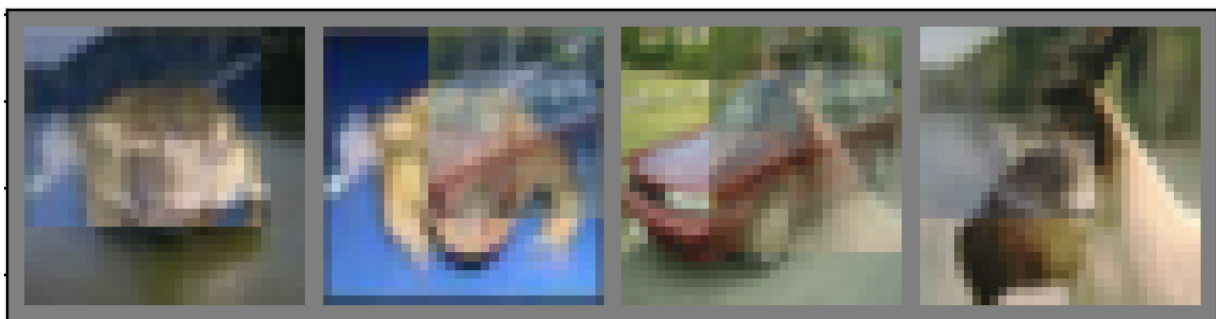


Figure 5: cutmixup 이미지 (ship+frog, frog+car, car+deer, deer+ship)

Divmix: 위의 기법들과는 달리 기존의 사용되거나 검증된 방법은 아니다. 새로운 기법을 찾기 위한 시도 중 하나로, 이미지를 네 부분으로 나누어 하여 섞는 방법이다. 사용되는 이미지는 batch내에서 random하게 3개를 골라 사분면에 할당한다. loss계산은 label을 1/4씩 계산한다. divmix prob을 이용해 확률을 설정해준다.

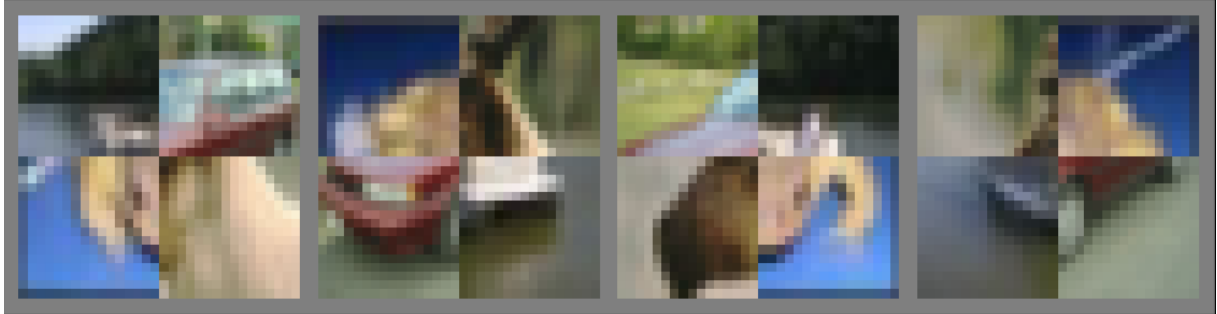


Figure 6: divmix 이미지 (모두 ship+frog+car+deer)

위의 방법의 특성을 비교해서 정리하면 다음과 같다.

	Cutout	Mixup	Cutmix	Cutmixup	Divmix
Usage of full image region	✗	✓	✓	✓	✓
Regional dropout	✓	✗	✓	✓	✓
Mixed image & label	✗	✓	✓	✓	✓

3 EXPERIMENTS

실험은 Google Colab을 통해 진행하였다. 학습네트워크로는 ResNet-50, mini-batch는 256, training epoch은 300, lr(learning rate)는 0.1, alpha는 300으로 설정하였으며, Dataset은 CIFAR-100을 사용하였다. 그 결과는 *Table 1* 같다.

Standard를 기준으로 설정값에 따라 Cutout, Mixup, Cutmix, Cutmixup, Divmix 모두 유의미한 성능향상을 보였다. Cutout의 경우 prob가 1.0인 경우가 더욱 큰 성능향상을 보였다. Mixup(alpha 0.4)와 Cutmix(prob 1.0)에서 Standard보다 3% 정도의 성능향상을 보였다. 현재의 학습환경이 CIFAR-100 dataset에 최적화가 되어있지 않아, 전체적인 정확도가 낮음을 고려하더라도 유의미한 성능향상이라 할 수 있다. Top 1 Error는 비슷하지만, Top 5 Error에서 Cutmix의 정확도가 Mixup보다 1%가량 높은 것도 주목할 만하다.

본 프로젝트에서 제안한 Divmix와 Cutmixup모두 Standard를 기준으로 유의미한 성능향상을 보였다. 특히, Cutmixup(alpha 1.0, prob 1.0)은 진행했던 방법 중 가장 높고, 독보적인 성능향상을 보였다. Cutmixup의 경우 prob가 1.0인 경우가 0.5인 경우보다 더 큰 성능 향상을 보이는 경향을 확인할 수 있었다. Divmix는 성능향상을 하긴 했지만, 유사한 방법인 Cutmix에는 미치지 못하였다.

	Top 1 Error (%)	Top 5 Error (%)
Standard	28.65	8.08
Cutout	26.02	6.78
Mixup	25.69	7.54
Cutmix	25.48	6.53
Cutmixup (proposed)	25.11	6.39
Divmix (proposed)	26.89	6.76

Table 1 : 각 Data Augmentation 기법을 사용하였을 때의 Top 1 Error와 Top 5 Error

4 DISCUSSION

위 실험을 통해 우리는 첫 번째로 Divmix가 Cutmix를 뛰어넘지 못하였다는 것과 두 번째로 Mixup과 Cutmix를 합친 Cutmixup이 더욱 큰 성능향상을 이루어 냈다는 사실을 확인할 수 있었다. 이에 대해 ablation study와 이론적 분석을 진행할 것이다.

4.1 Divmix ablation study & Analysis

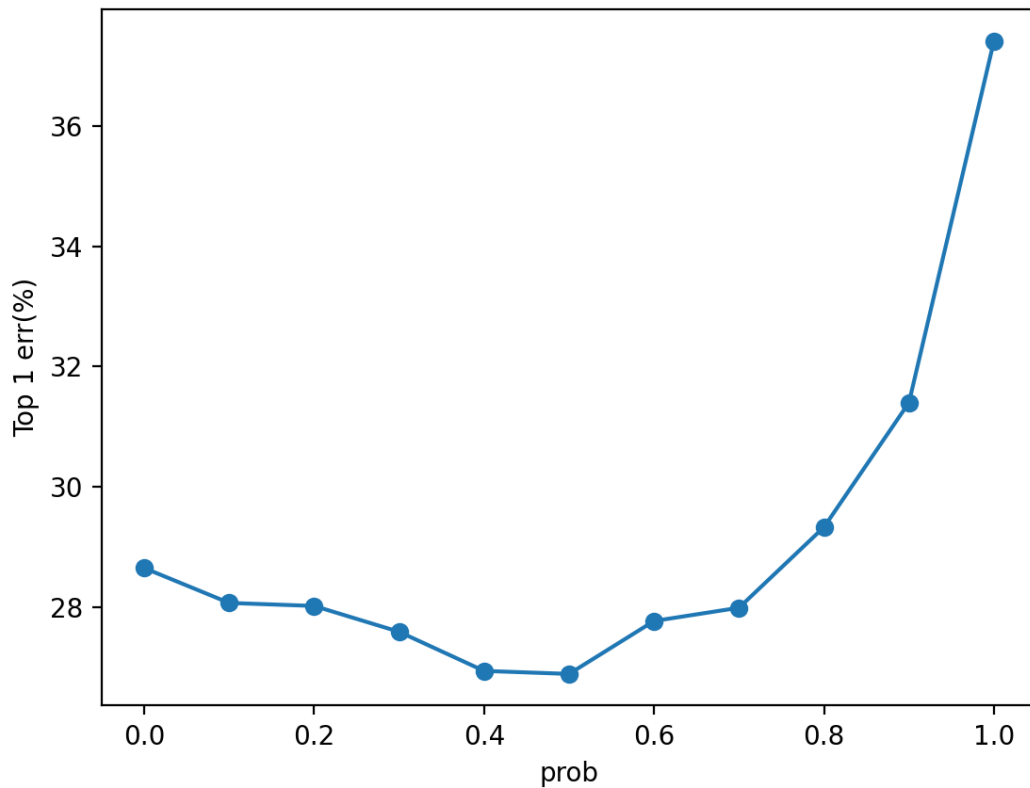


Figure 7: Divmix prob에 따른 Top 1 error 그래프

divmix prob을 변화시키면서 가장 성능이 높은 지점을 찾아보았다. 0에서부터 확률이 높아질수록 성능이 높아지다가 0.5에서 최고성능을 보였다. 이후에는 확률이 올라갈수록 오히려 성능이 떨어지는 것을 확인할 수 있었다.

	Top 1 Error (%)	Top 5 Error (%)
Standard	28.65	8.08
Cutout (prob 1.0, mask 0)	26.02	6.78
mask 0.001	26.26	7.01
mask 0.005	27.28	7.12
mask 0.01	28.11	7.45
mask 0.05	28.48	8.3
mask 0.1	28.6	7.66
mask 0.2	28.46	7.75

Table 2 : Cutout에서 mask값에 따른 Top 1 Error와 Top 5 Error

먼저, Divmix가 실패한 이유를 분석해보자면 4장의 image를 합친다는 특징은 있지만, 근본적으로 Cutmix의 아류와 같은 느낌이다. Cutmix의 경우 Cut되는 영역이 랜덤이기에 Data Augment의 관점에서 유의미한 입력데이터가 되지만, Divmix의 경우 영역이 고정되어 있다. 그렇기에 Data Augment를 통한 큰 효과를 내지 못하는 것이다. 이점은 대부분의 대상이 가운데에 위치하고 있는 CIFAR-100 Dataset에서 더욱 치명적인 단점으로 작용한다. 이런 이유로 본 프로젝트에서 제안한 Divmix는 실패한 방법이라 할 수 있다.

4.2 Cutmixup ablation study & Analysis

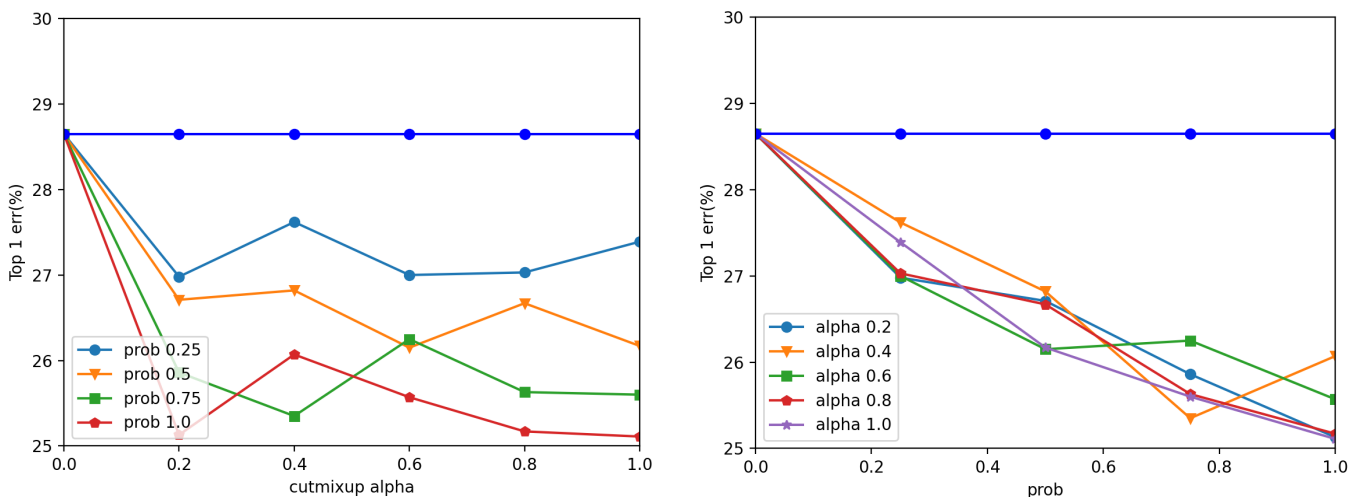


Figure 8: Cutmixup의 prob와 alpha에 따른 Top 1 error 그래프

Cutmixup의 ablation study결과 alpha값의 영향보다는 prob의 영향이 컸다. cutmixup의 확률인 높아질수록 성능이 향상되었고, 항상 cutmixup을 하였을 때, 가장 높은 성능을 보였다. 이는 mixup과정에서 beta함수를 사용하여 cutmixup을 하기에, 이 부분에서 이미 prob과 마찬가지로 역할을 하는 것으로 보인다. 따라서, 가장 Cutmixup기법의 영향을 높인 상황에서 최고 성능을 보인다고 추측한다.

	Top 1 Error (%)	Top 5 Error (%)
Standard	28.65	8.08
Cutout (prob 1.0, mask 0)	26.02	6.78
mask 0.001	26.26	7.01
mask 0.005	27.28	7.12
mask 0.01	28.11	7.45
mask 0.05	28.48	8.3
mask 0.1	28.6	7.66
mask 0.2	28.46	7.75

Table 2 : Cutout에서 mask값에 따른 Top 1 Error와 Top 5 Error

Cutmixup에서 성능향상의 원인을 분석한다. Cutmix와 Cutmixup의 차이점 중 하나가 Cutmixup에서는 원래의 이미지의 전체적인 부분이 남아 있는 것이다. 이 점을 증명하기 위해 Cutout에서 mask를 0으로 하는 것이 아닌 0.001~0.2 사이의 값으로 두고 실험을 진행하였다.

Table 2에서 mask의 값이 커짐에 따라 정확도가 낮아지는 경향이 있음을 알 수 있었다. mask는 0일 때가 가장 효과적이라는 결론과 함께 Cutmixup의 성능향상의 원인에 대한 추측을 증명할 수 없었다.

설명할 수 있는 다른 관점으로는 Cutmix와 Mixup의 방법을 융합시키면서 더 큰 Data Augmentation 효과를 낼 수 있었다는 점이다. 우리가 이런 mixed-base Data Augmentation 기법을 사용하는 것이 입력 데이터를 늘리기 위함이었다. 두 방법이 합쳐지며 유의미한 Data를 더 많이 만들어낼 수 있게 되어 보다 높은 성능 향상을 이루어낸 것으로 추측된다.

5 CONCLUSION

Image Classification에서 사용되는 Data Augmentation 기법인 Cutout, Mixup, Cutmix에 대해 알아보고 구현하고 실험하여 결과를 확인해보았다. 또한, Divmix와 Cutmixup을 제안하고, 구현하여 실험을 진행하였다. 그 결과 Standard보다 향상된 성능을 보여주었고, Cutmixup은 더욱더 높은 성능 향상을 보여주었다. Divmix의 실패원인을 Dataset의 특성과 부족한 Data Augmentation으로 분석하였다. Cutmixup의 성능 향상 요인을 찾기 위해 실험을 하였으나 성공하지는 못하였고, 더욱 강한 Data Augmentation이 주요했다고 판단된다.

본 프로젝트를 통해 Data Augmentation에서 성능향상에 영향을 끼치는 것은 적절한 random 변수를 넣어 다양하고 많은 입력데이터를 만들어 내는 것과 Dataset 특성에 대한 이해라는 결론을 내릴 수 있었다.

REFERENCES

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016
- [2] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.
- [3] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2017.
- [4] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. ICCV, 2019.
- [5] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer and Balaji Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. arXiv preprint arXiv:1912.02781, 2020.