[[1]](#footnote-1)

Enhanced Classifier via Pyramid Network with Cut-mix and Label Smoothing

Kun Kim, Jiwan Kim, Seongwon Choi

*초록*— 이번 competition의 목적은 CIFAR-10 test dataset의 정확도 최대화이다. 목적 달성을 위해 ResNet, DenseNet, EfficientNet, Preact-Resnet, PyramidNet Model비교 분석하였고 PyramidNet Model을 Base Line Model로 선정하였다. 모델 선정 이후 Batch normalization, Cut-mix, Label Smoothing, Hyper Parameter의 조절이 성능을 높이는데 주요한 역할을 했고, 결과적으로 PyramidNet model에 Label Smoothing 및 Cut-mix를 사용한 model이 test dataset에서 92.315%의 정확도를 보였다.

# 소개

인공지능에서 이미지 인식은 기계가 마치 사람처럼 사진이나 동영상으로부터 사물을 인식하거나 장면을 이해하는 것으로 정의할 수 있다. 이미지 인식은 이미지 내 특정 사물을 분류(Classification), 여러 사물을 동시에 검출(Detection), 사물들을 픽셀 단위로 식별하여 분할(Segmentation)하는 태스크(Task) 등이 있다. 특히 본 Competition의 종목인 이미지 인식 같은 경우 ILSVRC(ImageNet Large Scale Visual Recognition Challenge) 등에서 여러 딥러닝 모델을 통해 경쟁하는 등의 모습을 보이고 있다. 2015년 ILSVRC에서 사람의 인식률 (94.90%)을 추월 (96.43%)하고, 2020년에는 사람을 한참 뛰어넘는 수준 (98.7%)으로 진화했다. 이번 Competition의 목적은 Cifar10의 9만 개의 Train set과 9만 개의 Validation set을 이용하여 2M 내의 Parameter를 사용한 최적의 모델을 찾아내 Test set accuracy를 최대화하는 것이다. dataset은 32x32픽셀의 컬러 이미지가 포함되어 있으며, 각 이미지는 10개의 클래스로 Labeling 되어있다. Competition에서 성능을 향상시키기 위해 Regularization, data augmentation을 적용하였다. 여러 기법들을 비교 분석하기 위한 실험들을 실행하였으며 RandomHorizontalFlip, RandomCrop, RandomAffine, ColorJitter, Cut-mix, Shake-Drop, Label Smoothing등 다양한 기법들을 사용하였다. 최종적으로는 Cut-mix, Label Smoothing을 사용하여 Test Case에 대한 Acc를 최대화하였다.

# 관련 연구

200만개 이하의 파라미터를 만족하는 딥러닝 모델로 최적의 성능을 내기 위해 경량 버전이 존재하거나 파라미터 개수를 효율적으로 줄일 수 있는 모델(ResNet, DenseNet, PyramidNet)과, 제한된 dataset을 효과적으로 학습에 이용하기 위한 여러 가지 방법의 데이터 증강 기법을 조사하였다.

## 딥러닝 모델

1. ResNet

ResNet은 Residual Learning개념을 도입하여 일반 Network의 깊이에 따른 Gradient Vanishing 문제를 해결하였다. Residual network의 구조는 입력 값에 대한 weight layer 적용 이후 바로 출력으로 이어지는 일반 네트워크와 달리 입력 값을 출력에 skip connection으로 연결한다.

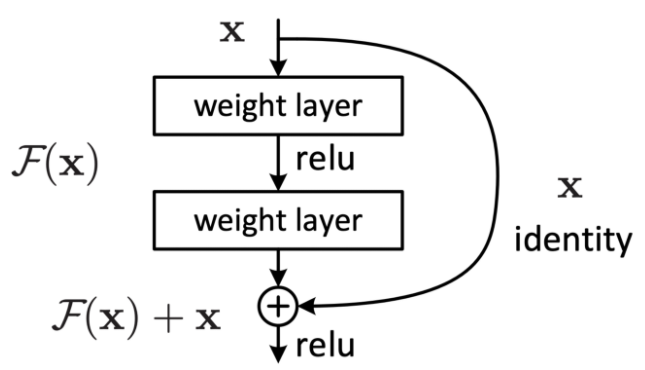


그림 1. Residual Building Block [1]

이와 같은 구조는 출력 값이 이전 layer의 입력 값(identity)을 포함하고 있기 때문에 연산과정에서 발생하는 Gradient Vanishing 문제를 방지한다. 결과적으로 성능 저하 없이 네트워크를 이전보다 더 깊게 쌓을 수 있게 되었다.

2. DenseNet

이전 layer들의 출력에 대한 축적이 다음 layer에 연결되는 방식을 사용했다. 입력값(identity)을 출력에 단순히 더하는 ResNet과 달리, 이전 layer들을 다음 layer와 연결해주는 개념이다. [2]

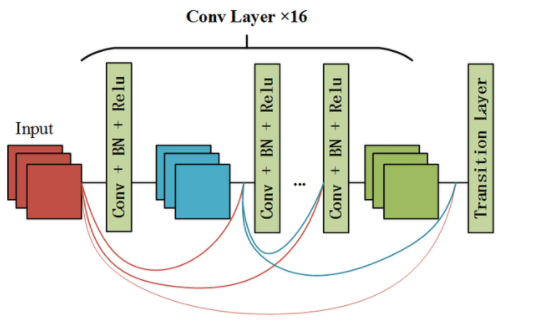


그림 2. Dense block [3]

이처럼 densely connected된 네트워크는 앞 쪽에 위치한 정보를 여러 층을 지나도 유지하게 되어 Gradient Vanishing 문제를 해결할 수 있다. 이 네트워크는 이전 layer의 feature map이 지속적으로 축적되어 무거워 보일 수 있지만 이와 같은 학습의 효율성 때문에 결과적으로 파라미터 수를 크게 줄일 수 있다.

3. EfficientNet

해당 모델은 network block을 창안하기 보다는 성능 향상을 위한 일반적인 방법인 scaling up을 효율적으로 다루는 것을 중점으로 한다. 제한된 크기 안에서 grid search방법으로 최적 depth, width, resolution을 찾아낸 후에, 모델을 더 큰 구조로 확장 시킬 수 있다. [4]

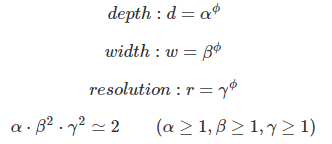


그림 3. Efficient-Net scaling method

4. PyramidNet

Residual network 기반의 모델의 경우 depth가 깊어질 때 width가 절반으로 감소하고 channel을 두배로 늘리는 것이 일반적인데 이 때의 급격한 width의 변화를 감소시키는 것을 중점으로 한 모델이다. Alpha값에 따라 layer마다 점진적으로 증가시킬 width의 정도를 결정하게 된다. 대표적으로 Additive Mode와Multiplicative Mode를 사용하며 N residual units에 대해 다음과 같은 output 차원을 이루게 된다. [5]

⌊Dk−1 + α/N⌋, (if 2 ≤ k ≤ N + 1) ··· Additive

⌊Dk−1 · ⌋, (if 2 ≤ k ≤ N + 1) ··· Multiplicative

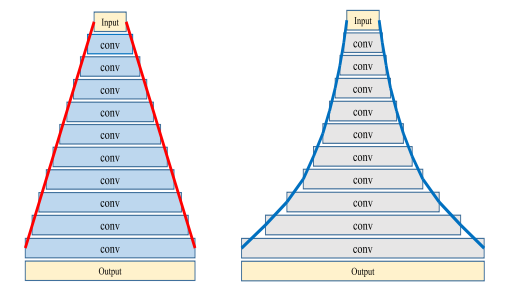


그림4. Additive PyramidNet and Multiplicative PyramidNet

## 데이터 증강 기법

Image Classification에 있어서 데이터 셋이 한정되어 학습 성능이 저하되는 경우, 데이터 셋에 변형을 주어 절대적인 데이터 셋 크기를 늘려주는 방법을 사용한다.

1. Affine Translation

Cropping, Scaling, Shearing, Rotation 등 Data Augmentation의 일반적인 기법들을 원본이 가진 직선, 평면을 유지한 채로 사용하는 방법이다. 원본을 왜곡하지 않으면서 데이터 증강을 할 수 있는 방법이다. [6]

2. Color Jitter

이미지에 기하학적 변환을 주는 것이 아니라 이미지의 색과 관련된 값을 조정함으로써 Data Augmentation을 하는 방법이다. 이미지의 채도 조절, 픽셀 값 조정 등을 사용한다. [7]

3. Cut-Mix

Training 데이터 셋 안에서 두 이미지를 결합하여 새로운 이미지를 생성하는 방법이다. 하나의 이미지 안에서 일정한 부분을 cropping하고 해당 region을 다른 이미지의 region으로 대체한다. 새로운 이미지의 label은 섞인 두 이미지의 pixel 비율로 정하게 된다. [8]

## 정규화 기법

1. Shake Drop

Shake-Shake 정규화의 연장선에 있는 방법으로 forward/back propagation시 convolutional layer의 출력에 noise를 주는 기존 방법에 음수 값을 추가적으로 더해준다. 이미지의 feature를 직접 조작하지 않기 때문에 정규화 기법으로 소개되긴 하지만 전반적인 학습에 있어서 Data Augmentation과 같은 효과를 낼 수 있다. [9]

2. Label Smoothing

Dataset의 labeling이 잘못 저장되어 있을 경우 loss값의 오차가 커지는 것과 동시에 overfitting을 방지하는 방법으로, label target값을 0또는1로 한정하지 않고 상수 에 대해 다음 연산 과정을 거치게 된다.

= (1−α) + α/K

결과적으로 label target이 0~1사이의 값을 갖게 되어

모델이 정확하게 예측하지 못한 부분에 대한 손실이 감소된다. [10]

# 제안 방법

최종적으로 파라미터 개수를 2M로 제한하면서 여러 가지 Base model을 실험하였다. 그 결과 PyramidNet이 가장 높은 성능을 보였고 해당 Model을 중심으로 실험을 수행하였다. 또한 PyramidNet의 성능에 가장 큰 영향을 주는 Depth 및 Alpha를 변화하며 실험을 수행하였고 최종적으로 depth, alpha의 값은240, 40이었다. 그리고 이 Model의 파라미터 개수는 약 1.97M를 가졌다.

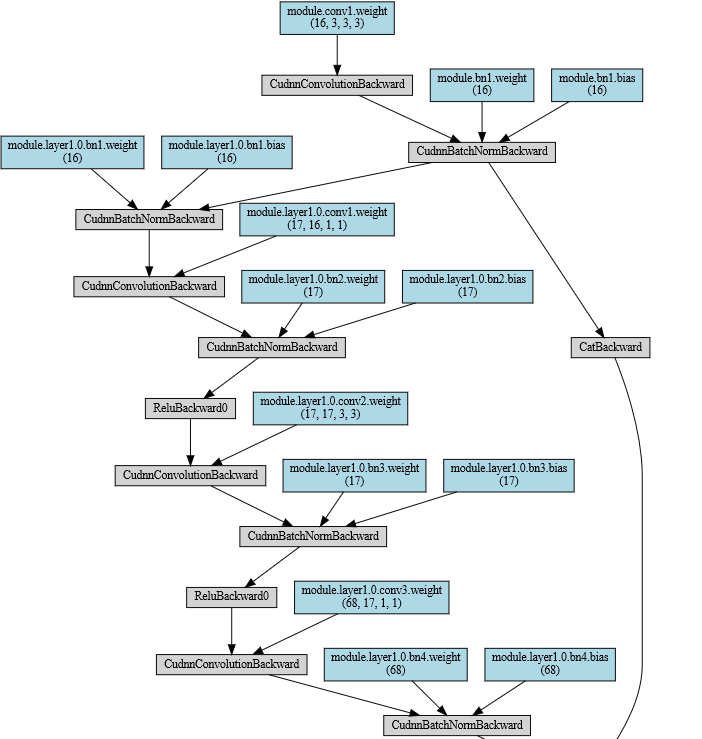


그림5. Model Structure (Input Layer)

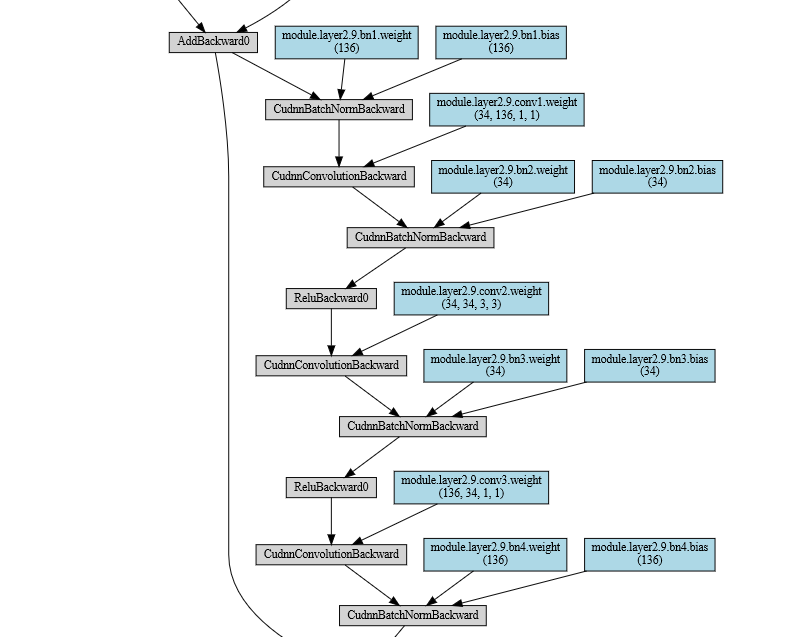


그림6. Model Structure (Middle Layer)

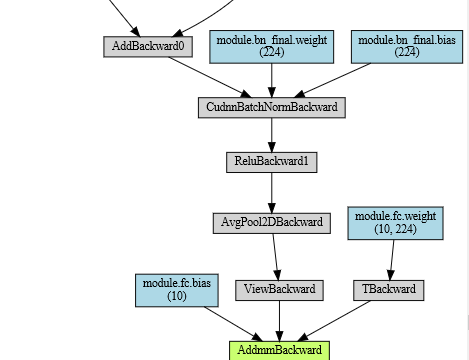


그림7. Model Structure (Output Layer)

그림8. Original Image and Cut-mix Image

기존의 Model을 Training하면 epoch 150 이상에서 Overfitting 현상이 발생하였다. 따라서 Augmentation을 통해 일반화 능력을 향상시키고자 Cut-mix 기법을 사용하였다. 해당 기법은 랜덤으로 사진의 일부를 잘라내고 이를 섞는 것이다. 기존 이미지만을 사용할 경우 일부 Pixel에 의존하여 예측하는 경향성이 생길 수 있고 이는 Overfitting이 될 확률이 크다. 따라서 일부 Pixel이 아닌 다양한 Pixel에 대해서 Label을 예측하도록 학습을 진행하여 Overfitting을 방지하고자 하였다.

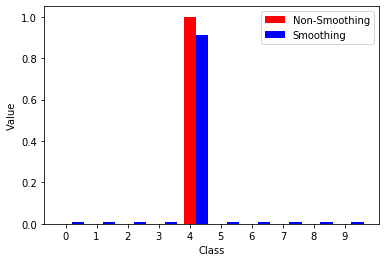


그림9. Smoothing & Non-Smoothing Chart

마지막으로 label-smoothing regularization기법을 사용하였다. 해당 기법을 통해 Mislabeling에 대한 영향을 줄일 수 있었으며 Cut-mix와 같은 기법에서 여러 Label에 대한 Loss를 고려할 수 있었다. 본 실험에서의 label-smoothing 파라미터는 0.1로 지정하였다. 그 결과 위의 차트처럼 Fourth Class가 Target일 경우, [0,0,0,1,0,0, ...] 가 아닌 [0.01,0.01, 0.01,0.91,…]으로 값을 부여 하였다.

# 실험

## 실험조건

실험을 수행한 모델은 PyramidNet을 사용하였다. PyramidNet에는 이전 Layer의 Width에 비해 얼마나 늘릴 것인지 할 수 있다. 또한 Width 증가 방식에는 Linear하게 Width를 증가시키는 Additive Mode와 Exponential하게 증가시키는 Multiplicative Mode가 있다. 본 실험에서는 Additive Mode를 적용하여 input과 가까운 초기 Layer의 Feature Map이 보존되도록 하였다. PyramidNet에 가장 큰 영향을 미치는 Hyper Parameter는 Depth와 Width를 결정하는 Alpha이다. 본 실험에서는 Depth = 158, Alpha = 56을 적용하였다. 이외의 실험 파라미터 같은 경우 Batch Size는 256으로 진행하였고 Scheduler는 Multisteplr을 사용하여 [60,120,80]에 0.1을 순차적으로 곱해주었다. 모멘텀은 그대로 0.9를 사용하였으며 가중치 감쇠 지수는 를 사용하였다.

초기에는 Train set만 사용하였으나 Valid set을 추가적으로 학습에 이용하자 Acc의 큰 상승이 있었고 Valid set을 포함한 18만장을 학습으로 사용하였다.

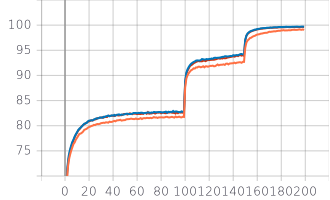
실험 환경은 Google Colab Pro를 사용하였으며 T4 혹은 P100 GPU를 할당 받았다.

## 최신 모델간 비교 실험

표1. Performance Graph Per Model

위의 표는 Cifar10 Train Set을 사용하여 각 모델을 3회 Train한 후 평균 성능을 비교한 결과표이다. ResNet, DenseNet, Eff-NetB0, Preact-Resnet, PyramidNet 등 총 5개의 Model을 비교한 후 Test Data set에 대한Acc를 비교하였다. 그 결과 PyramidNet, EfficientNetB0, Preact-Resnet, DenseNet, ResNet 순서로 우수한 성능을 보였다. 본 결과를 토대로 이번 실험에 있어 PyramidNet을 사용하였다.

## 하이퍼 파라미터에 대한 실험



* 158 Depth 56 Alpha
* 240 Depth 40 Alpha
* 110 Depth 70 Alpha

그림10. Performance Graph per HyperParameter about Train set

PyramidNet 같은 경우 네트워크의 깊이를 결정하는 Depth와 너비를 결정하는 Alpha가 주요한 Hyper Parameter

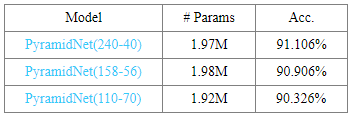
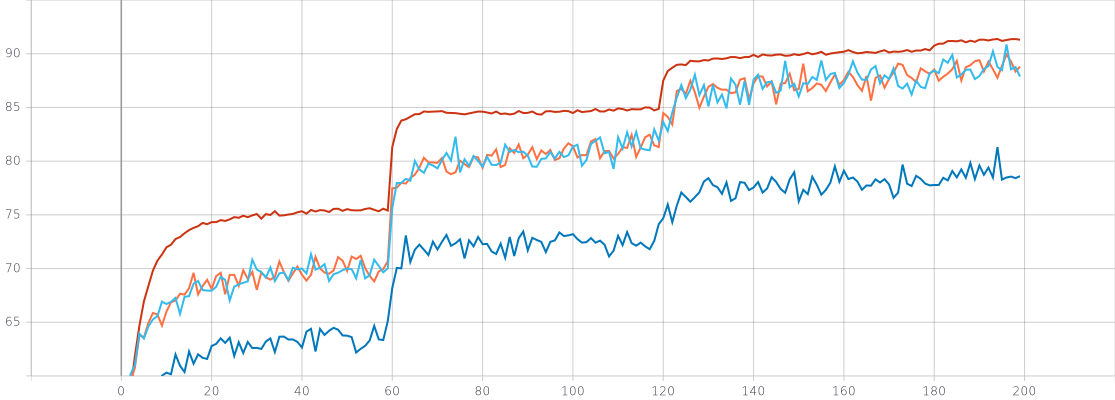
로 작용한다. 이에 어떠한 값을 가질 때 최적의 성능을 갖는지 평가하기 위해 Depth와 Width를 다양하게 구성하여 평가를 수행하였다. Depth가 Alpha 값에 비해 큰 경우, Alpha 값이 Depth에 비해 큰 경우, Depth 및 Alpha가 비슷한 경우 총 3가지로 실험을 구성하였다. 그 결과 각 Case 모두 Overfitting 현상이 일어났다. 먼저 158 Depth의 경우 최종 Acc가 99.7%로 가장 높았고 240 Depth는 99.64%, 마지막으로 110 Depth는 99.04%를 기록하였다.

표2. Comparison Table of regularization and augmentation About Test set

Test set을 대상으로 Acc를 측정한 결과 이전과 다소 다른 결과가 산출되었다. 먼저 Train set에 대해서 99%의 Acc를 보였음에도 불구하고 Test set에 대한 Acc는 그보다 현저하게 낮았다. 이를 통해 Overfitting을 추론할 수 있었다. 또한 158 Layer가 Train set에 대해 가장 높은 Acc를 가졌던 결과에 반해 Test set에 대해서는 240 Layer의 Case가 가장 높은 결과를 산출하였다.

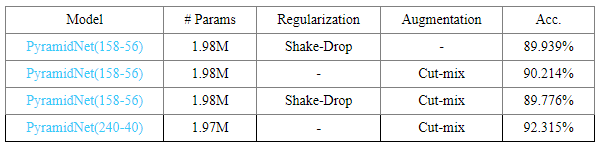
## 데이터 증강 기법에 대한 비교 실험



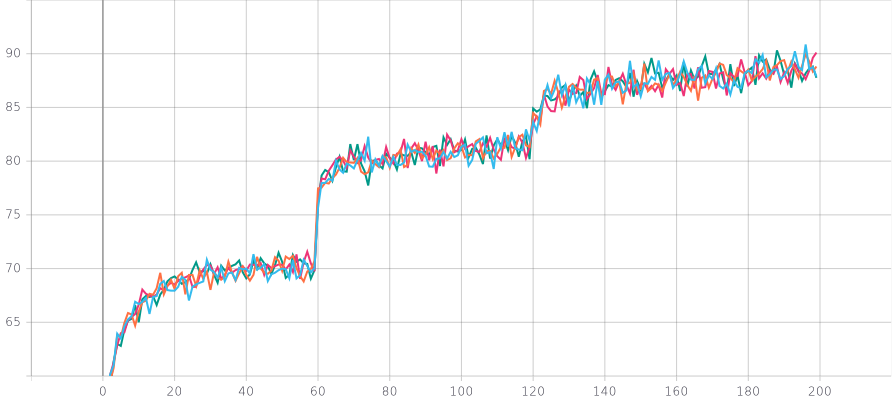
* Case1 – Both
* Case2 – Shake-Drop
* Case3 – Cut-mix (158)
* Case4 – Cut-mix (240)

그림11. Comparison graph of regularization and augmentation about Train set

위의 그래프는Data Augmentation 및 Regularization을 적용했을 때, Train Set에 대한 결과표이다. 실험은 Shake-drop, Cut-mix, Shake-drop & Cut-mix 총 4개의 Case를 두고 수행하였다. 먼저 Shake-drop만 적용한 Case2의 경우, 91.31%로 가장 높은 Acc를 보였다. 다음으로 158 Layer에 Cut-mix만 적용한 Case3는 Acc는 약 89.94%를 보였으며, 240 Layer에 Cut-mix를 적용한 Case4의 Acc는 약 90.86%를 기록하였다. 마지막으로 모두 적용한 Case1의 경우 81.31%로 가장 낮았다.

  
표3. Comparison Table of regularization and augmentation About Test set

하지만 Test Set에 대한 Acc는 위의 결과와 다름을 볼 수 있었다. 위의 표는 Test Set에 대해 Case1,2,3,4를 정리한 결과이다. Layer 240에 Cut-mix만 적용했을 때, 92. 315%로 가장 높은 Acc를 보였다. 그 다음 Layer 158에 Cut-mix를 적용했을 경우에 92.214%를 보였으며, Layer 158에 Shake-drop만 적용했을 경우에는 90.214% 정도의 Acc를 보였다. 마지막으로 모두 적용한 경우에는 89.939%로 가장 낮은 Acc를 산출하였다.



* Case1 – Smoothing & Cut-mix (158)
* Case2 – Cut-mix (158)
* Case3 – Smooting & Cut-mix (240)
* Case4 – Cut-mix (240)

그림12. Comparison graph of Cut-mix and Label Smoothing About Train set

마지막으로Cut-mix와 Label Smoothing을 적용하여 비교 실험을 수행하였다. Layer 240에 Cut-mix만을 넣은 Case4가 90.86%로 가장 높았다. 이후 Smoothing을 함께 적용한 Case3가 90.31%로 높았으며, Layer 158에 모두 적용한 Case1이 90.11%를 기록하였다. 마지막으로 Layer 158에 Cut-mix만 적용한 Case2는 89.94%를 기록하였다.

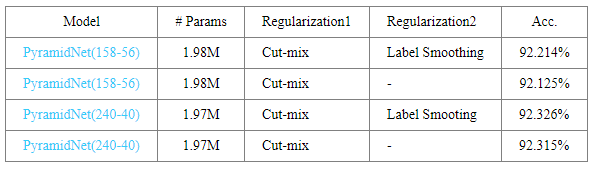


표4. Comparison Table of Cut-mix and Label Smoothing About Test set

Test set에 대해서 실행을 수행했을 때 이전의 Train set에 대한 결과와 조금 달랐다. Cut-mix를 적용하고 Label Smoothing을 한 경우가 Cut-mix만 적용한 Case보다 높은 Acc를 보였다. 최종적으로Layer 240에 Cut-mix와 Smoothing을 적용한 Case3가 가장 높은 Acc인 92.326%의 결과를 보였다.

# 결론

ResNet, DenseNet, PyramidNet, EfficientNet 등의 모델을 사용하며 이번 Competition에 어떤 모델이 가장 적합한지 실험하였고 Base Model로 PyramidNet을 선정하였다. 또한 Generalization ability 향상을 위해 Cut-mix, Shake-Drop, Label Smoothing등을 사용하였다. 초기에는 Model의 성능에 있어 Depth와 Alpha모두 중요한 요소라고 생각하였고 적절한 균형 지점을 찾고자 하였다. 하지만 실험을 진행한 결과 Alpha보다 Depth를 늘렸을 때 더 좋은 결과를 산출하였다. 또한 2M 이하의 파라미터를 가졌음에도 Overfitting 현상이 빈번한 것을 알 수 있었고 앞서 언급한 여러 기법들을 통해 성능을 높일 수 있었다.

모델 경량화 및 일반화 능력 향상에 대한 요구는 끊임없이 발생하고 있다. 더불어CCTV, Tablet PC 등의 소형 임베디드 기기에 딥러닝을 적용하려는 시도 또한 많아지고 있다. 본 연구와 같이 제한된 파라미터 아래 여러 기법 등을 시도해 봄으로써 임베디드 기기에 적용 가능한 모델을 설계한다면 치안, 복지 등 많은 분야에 적용이 가능할 것으로 보인다.

# 참고문헌

[1] Kaiming He et al. “Deep Residual Learning for Image Recognition” IEEE 2016.

[2] Gao Huang et al. “Densely Connected Convolutional Networks” IEEE 2017

[3] Pengming Feng et al. “Embranchment CNN based Local Climate Zone Classification using SAR and Multispectral Remote Sensing Data” State Key Laboratory of Space-Ground Integrated Information Technology, CAST, Beijing, 100095, China 2019

EMBRANCHMENT CNN BASED LOCAL CLIMATE ZONE CLASSIFICATION USING SAR

AND MULTISPECTRAL REMOTE SENSING DATA

[4] Mingxing Tan, Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks” ICML 2019.

[5] Dongyoon Han, Jiwhan Kim, Junmo Kim. “Deep Pyramidal Residual Networks” CVPR 2017

[6] E. Pang, D. Hatzinakos. “An efficient implementation of affine transformation using one-dimensional FFTs” IEEE 1997

[7] Chanachai Puttaruksa, Pinyo Taeprasartsit “Color Data Augmentation through Learning Color-Mapping Parameters between Cameras” IEEE 2018

[8] Sangdoo Yun et al. “CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features” ICCV 2019

[9] Yoshihiro Yamada, Masakazu Iwamura, Koichi Kise. “ShakeDrop regularization” ICLR 2018

[10] Rafael Müller, Simon Kornblith, Geoffrey Hinton.   
“When Does Label Smoothing Help?” NeurIPS 2019

**First A. Author** was born in Daejeon, Republic of Korea, 1997. He graduated from Kyung Hee University in Yeong-tong with a degree in computer Engineering in 2021.

At 2015, He won an award on Citizen Idea Contest from the Ministry of Public Administration and Security.

From 2015 to 2016, He did a mentor for Major Subject in Dasom, a Crew in Kyung Hee univ’s Software College. From 2016 to 2017, He did a mentor for College freshman in Humanitas College. From 2020.03 to 2020.08, He participated in an internship with Seoul Digital Foundation. During Internship, He did Big Data-Preprocessing, Algorithm Programming, IT Infrastructure Building. From 2020.10 to 2020.12, He Participated in Google Boot Camp for Machine Learning. He studied about Machine Learning’s various method and took a course, Cousera for machine learning. He got accepted to NHN Entertainment. Since 2021, He will join NHN Entertainment Infra Team.

He is interested in various things in Computer Engineering.

He aims to be a Backend Engineering that can many things for customers. his final dream is to make application that have a machine Learning. And Start-up by this Application.

**Second B. Author** was born in Suwon, Republic of Korea, 1998. He graduated from Kyung Hee University in Yeong-tong with a degree in Computer Engineering in 2023.

He was accepted to join the Korean army in 2021. He was very interested in games. He was ranked in the 300th rank around November 2020 in a team fight tactics. He was also in the top 0.01 percent of the league of legend.

He was more interested in machine learning than in other fields of computer science. In 2019, he worked on a project to improve image resolution through machine learning.

He aims to do a job involving games and machine learning. His dream is to make a team fight tactics machine with machine learning to rank team fight tactics first in the world. Also, selling it to become a millionaire and become a young boss.

**Third C. Author** was born in Suwon, Republic of Korea, 1993. He graduated from Kyung Hee University in Yeong-tong with a degree in Computer Engineering in 2021.

He has worked as a cook at Canada when he was a sophomore, and after returning to Kyung Hee University in 2019, he became interested in machine learning. He recently participated in a project about “Estimation of COVID-19 Transmission Risk" using object detection. His current goal is to work in the field of machine learning.

1. [↑](#footnote-ref-1)