



[P1] 이미지 분류 Wrap-up Report

[점수 및 순위](#)

[Final Submit의 Options](#)

[Inference and Ensemble](#)

[Model](#)

[Data](#)

[Training](#)

[성능 향상을 위해 시도한 실험](#)

[실험 List](#)

[실험의 근거와 의의](#)

[P1을 통해 얻은 교훈](#)

[개선해야 할 점](#)

[느낀 점](#)

점수 및 순위

- Private LB / 0.7638 / 11위
- Public LB / 0.7790 / 8위

Final Submit의 Options

Inference and Ensemble

- 384 x 320 , 300 x 256 Size로 학습한 총 10개의 Model을 Soft Voting으로 Ensemble

Model

- EfficientNet-B4 를 Backbone으로 사용
- 18개의 Class를 예측하는 모델로 구성
- Dropout의 확률을 70%로 높여 일반화 성능 향상

Data

- Rotate(10), Horizontal Flip 등 소량의 Augmentation만 적용
- Center Crop을 통해 배경과 같은 Noise를 제거
- Model의 Inference Probability 등을 활용해 Noise Data의 Label을 Cleansing

- 한 사람의 정보가 Leaking되지 않도록 사람 기준으로 Validation Set 구성
- 성별과 나이를 기준으로 Stratified K Fold 수행

Training

- LR Scheduler를 사용하지 않고, LR은 1e-4로 사용
- AdamP Optimizer 사용
- 전체 Class 분포에 따른 Weighted Cross Entropy Loss 사용
- GPU Memory에 따라 가장 큰 Batch Size 사용 (32, 16)

성능 향상을 위해 시도한 실험

다양한 Idea와 설정으로 실험을 수행했다.

Name	Last Modified
custom_data	a day ago
0000_Testing Place.ipynb	a day ago
0329_Code.ipynb	10 days ago
0329_Train Data Distribution.ipynb	8 days ago
0330_Code.ipynb	9 days ago
0331_Code_v1.MakeBaseline.ipynb	9 days ago
0331_Code_v2.Baseline.ipynb	8 days ago
0401_AgeCheck.ipynb	8 days ago
0401_Code_v1.CenterCrop.ipynb	8 days ago
0401_Code_v2.GrayScale.ipynb	8 days ago
0401_Code_v3.AgeFilter.ipynb	8 days ago
0401_Code_v4.FlipRotation.ipynb	8 days ago
0402_Code_v1.CosineAneling.ipynb	7 days ago
0402_Code_v2.CosineSGD.ipynb	7 days ago
0402_Code_v3.AdamP.ipynb	7 days ago
0402_Code_v4.SGDP.ipynb	7 days ago
0402_Code_v5.FocalLoss.ipynb	7 days ago
0402_Code_v6.WeightCELoss.ipynb	6 days ago
0402_Code_v7.StepLR.ipynb	6 days ago
0403_Code_v1.Madgrad.ipynb	6 days ago
0403_Code_v2.MixPrecision.ipynb	6 days ago
0403_Code_v3.LR.ipynb	6 days ago
0403_Code_v4.CosineWarm.ipynb	6 days ago
0403_Code_v5.WeightDecay.ipynb	6 days ago
0403_Code_v6.WeightFocalLoss.ipynb	6 days ago
0403_Code_v7.F1Loss.ipynb	5 days ago
0404_CleanLab.ipynb	5 days ago
0404_Code_v1.Baseline.ipynb	5 days ago
0404_Code_v2.FlipRot.ipynb	5 days ago
0404_Code_v3.CheckNoiseData.ipynb	4 days ago
0404_Code_v4.CenterCrop.ipynb	4 days ago
0404_Code_v5.SameLabelMixup.ipynb	5 days ago

Version별 Code 1

Name	Last Modified
0404_CleanLab.ipynb	5 days ago
0404_Code_v1.Baseline.ipynb	5 days ago
0404_Code_v2.FlipRot.ipynb	5 days ago
0404_Code_v3.CheckNoiseData.ipynb	4 days ago
0404_Code_v4.CenterCrop.ipynb	4 days ago
0404_Code_v5.SameLabelMixup.ipynb	5 days ago
0404_Code_v6.SameLabelMixup.ipynb	5 days ago
0405_Clustering.ipynb	4 days ago
0405_Code_v1.Baseline.ipynb	4 days ago
0405_Code_v2.Bn.ipynb	3 days ago
0405_Code_v3.ResNet50.ipynb	4 days ago
0405_Code_v4.ResNeXt50.ipynb	4 days ago
0405_Code_v5.MultiLoss.ipynb	3 days ago
0405_Code_v6.GenderNoiseData.ipynb	3 days ago
0406_Code_v1.Baseline.ipynb	3 days ago
0406_Code_v2.BaselineB4.ipynb	3 days ago
0406_Code_v3.CleansData.ipynb	3 days ago
0406_Code_v4.Augs.ipynb	3 days ago
0406_Code_v5.ThreeCLF.ipynb	2 days ago
0406_Code_v6.AugsB4.ipynb	2 days ago
0407_Code_v1.EightDimCLF.ipynb	2 days ago
0407_Code_v2.Seed43BaselineB4.ipynb	a day ago
0407_Code_v3.18C_Baseline.ipynb	2 days ago
0407_Code_v4.HistoEQ.ipynb	2 days ago
0407_Code_v5.Random8C.ipynb	2 days ago
0407_Code_v6.ColorJitter.ipynb	2 days ago
0407_Code_v7.TrainAll.ipynb	2 days ago
0408_Code_v1.Dropout.ipynb	2 days ago
0408_Code_v2.DropoutB4.ipynb	a day ago
0408_Code_v3.CC-384-320.ipynb	a day ago
0408_Code_v4.DropoutTesting.ipynb	a day ago
0408_Code_v4.EnsembleCode.ipynb	a day ago

Version별 Code 1

Name	Last Modified
00. NoAug, b0	8 days ago
01. CenterCrop, b0	8 days ago
02. GrayScale, Crop, b0	8 days ago
03. 59AgeFilter, Crop, b0	8 days ago
04. 58AgeFilter, Crop, b0	8 days ago
05. 57AgeFilter, Crop, b0	8 days ago
06. Flip, Rotation, 58, Crop, b0	8 days ago
07. AdamP, 58, Crop, b0	7 days ago
08. Focal, AdamP, 58, Crop, b0	7 days ago
09. WeightCE, AdamP, 58, Crop, b0	6 days ago
10. StepLR, Focal, AdamP, 58, Crop, b0	6 days ago
11. Focal, Madgrad, 58, Crop, b0	6 days ago
12. CosineWarm, WeightCE, AdamP, 58, Crop, b0	6 days ago
13. WeightDecay, WeightCE, AdamP, 58, Crop, b0	5 days ago
14. WeightFocal, AdamP, 58, Crop, b0	5 days ago
15. F1Loss, AdamP, 58, Crop, b0	5 days ago
16. FlipRot, WCE, AdamP, 58, Crop, b0	5 days ago
17. CenterCrop, 0404Baseline(g=1)	4 days ago
18. FlipRot, CenterCrop, 0404Baseline	4 days ago
19. B4, 0405Baseline	4 days ago
20. MultiLoss, 0405Baseline	3 days ago
21. MultiLoss, B4, 0405Baseline	3 days ago
22. CleansData, MultiLoss, B0, 0405Baseline	3 days ago
23. Augs, CleansData, MultiLoss, B0, Baseline	3 days ago
24. Augs, CleansData, MultiLoss, B4, Baseline	2 days ago
25. Seed43, B4, Baseline	a day ago
26. Dropout, B4	a day ago
27. Dropout, B4, CC	a day ago
28. Dropout, B4, NCC	a day ago
DM_DS_Nomal	a day ago
Dropout_Big	a day ago
Dropout_Big_Midium	a day ago

Version별 산출물

실험 List

- Center Crop Size (384×320, 320×256, 224×224 등)
- Age Filter (59~, 58~, 57~, 56~ 등)
- Optimizer (Adam, SGD, Momentum, SGDP, Madgrad, SAM 등)
- Loss Function (CE, Focal, F1, Multi-Loss 등)
- Model Architecture (Multi-Head Model, Eight Dimension Output Model)
- LR Scheduler (Cosine Annealing, Step LR 등)
- Low Precision Training
- Weight Decay Rate
- Augmentation (Flip, Rotation, Random Brightness Contrast, HSV, Gray Scale 등)
- Mix-up (Same Label 등)
- Backbone Model (ResNet34, ResNet50, ResNeXt50 등)
- Data Cleansing (CleanLab, Probability Threads Holds 등)

- Histogram Equalization
- Drop Out Rate (0.3, 0.5, 0.7 등)

실험의 근거와 의의

Center Crop

얼굴 부근을 제외한 정보는 Task 상 Noise라고 판단하여, 실험을 진행했다.

대회 초반에 224×224 Size를 통해 성능 향상을 얻었고, 작은 Size를 통해 각종 실험을 빠르게 수행할 수 있었다. (수많은 실험을 통해서 더 많은 성능 향상을 얻을 수 있었다.)

대회 마지막에는 Size를 변경해가면서 옷차림과 같은 정보도 포함하는 Model을 구성했고, 유의미한 Size의 Model을 Ensemble하여 최종 결과를 생성했다.

Age Filter

60세 이상의 Data가 현저히 적은 Class Imbalance 문제를 해결하기 위해 50대 후반의 Data를 60세로 Labeling하고자 했다.

먼저, 각 Age의 Data를 시각화 하여 확인했다. 개인적으로는 50세 후반과 60세를 구분하기 어렵다고 판단했다.

이에 59세부터 57세까지 Labeling을 변경해가며 제출해보았고, 58세에서 가장 높은 성능 향상을 확인했다. 이 방법도 최종 결과까지 사용되었다.

Optimizer

Data에 적합한 Optimizer를 찾기 위해 성능을 실험했다.

최초에는 Adam을 Base로 활용했으나, 이에 대한 검증이 필요하다고 생각했다.

LR을 고정한 채로 Optimizer만 변경하여 성능을 검증했고, 편차가 심한 Optimizer들에 대해서는 LR과 Batch Size 등을 조정해가며 성능을 실험했다.

결론적으로 AdamP가 가장 좋은 성능을 보였고, 이를 Default로 설정해 사용했다.

Loss Function

최초의 Baseline은 Cross Entropy Loss를 Default로 사용했고, Task에 적합한 Function을 찾고자 실험을 진행했다.

먼저, Class Imbalanced 문제에 대해 집중하여 Loss Function 실험을 설계했다.

Focal Loss에 대해서는 Gamma 값을 0~5까지 변경해가며 수행했고, 이 당시 Gamma=5가 가장 높은 성능을 보였다.

F1 Loss는 그 자체로 높은 성능이 나오지 않았고, 이후 Multi-Loss 실험시 추가적으로 사용되었다.

Multi-Loss는 각 Loss의 성능을 고려하여 비율을 조정해가며 실험을 수행했다.

Weight Cross Entropy는 18개의 Class의 분포를 역수로 하여 가중치로 사용했다. 이 때, 각 값의 가중치를 줄 수 있도록 Custom 하였고, 2배의 값을 사용한 것이 가장 높은 성능을 보였다.

Model Architecture

마스터님이나 멘토님의 말씀처럼, 18개의 Class를 예측하는 것보다는 각 대분류별 예측이 더 합리적이라고 판단했다.

먼저 Backbone으로부터 3개의 출력을 갖는 Multi-Head Model을 구성했고, 이에 따라 Pipeline을 개선하여 성능을 실험했다.

위 실험에서 좋은 성능을 얻지 못해, 이번에는 8차원의 Output을 출력하는 Model을 구성했다.

각 차원은 마스크착용 상태(3), 성별(2), 나이(3)를 의미했고, 마치 Multi-label의 형태의 Task로 구현했다.

위 Architecture에 적합한 Hyper-Parameter를 Search하다가, 시간적인 한계로 기존에 사용하던 18개 Class를 한번에 예측하는 Model을 사용했다.

LR Scheduler

LR을 더 낮게 줄 경우, Minimum을 더 잘 찾을 수 있으리라 판단했다.

먼저, Step LR Scheduler를 통해 Step Size와 Gamma를 조정해가며 실험을 수행했다. 더 높은 성능이 나오지 않아, Cosine Annealing LR Scheduler와 Warm Restart를 실험했다.

Epoch의 수가 높지 않아, T_0(주기)를 설정하기 어려웠고, 각종 실험 끝에 사용하지 않기로 결정했다.

Low Precision Training

Low Precision Training을 구현할 Code를 찾던 중, 토론 게시판에 이와 관련된 Code가 게시되어 실험을 진행했다.

당시 실험 기준으로 약 20~30%정도 학습 속도가 향상 되었으나, 다소 성능이 떨어지는 문제가 있었다.

실험 당시 모든 기준을 Public Score로 하고 있었기 때문에, 이에 대한 혼동이 생기지 않도록 사용하지 않기로 결정했다.

Weight Decay

학습의 수렴속도가 빠른 것이 확인 되어서, Model이 과적합 되지 않고, 일반적인 성능을 내기를 기대하는 마음에 Weight Decay 값을 변경해가며 실험했다.

결론적으로 성능 향상에 도움이 되지 않아, 사용하지 않기로 결정했다.

Augmentation

Data의 일반화와 증강을 위해 실험을 진행했다.

최초에는 변화가 Augmentation이나, 강도를 높게하여 Data를 증강했었으나 성능에 악영향이 생겼다. Test나 Validation Data와의 Gap이 생겼기 때문이라고 생각한다.

이후 Image에 큰 변화가 생기지 않는 Augmentation을 수행했고, 일부 Augmentation에서 성능 향상을 확인할 수 있었다.

Image를 크게 변형시키지 않는 Augmentation이 좋은 성능을 발휘하는 만큼, TTA와 Data Up-sampling을 사용하지 않기로 결정했다.

Mix-up

Data 증강의 방법으로 Mix-up에 대한 실험을 진행했다.

Idea는 60대 이상의 Data를 Mix-up의 Source로 자주 등장시켜서, Data에 대한 학습을 더 많이 수행토록 하려고 했다.

하지만, 이는 Data Augmentation의 강도가 낮아서 쉽게 과적합이 되는 문제가 발생했다.

일반화 성능을 얻기 위해서 Same Label에 대해서 Mix-up도 수행했으나, 많은 Error가 발생했다.

특히, Data Loader의 Worker가 중단되는 Error가 지속적으로 발생했고, 이 부분을 해결하지 못한채 사용을 중단했다.

Backbone Model

성능 대비 Parameter의 수가 적은 EfficientNet-b0를 Default로 위 실험들을 진행했다.

최종 성능을 이끌어 내거나 Ensemble에 사용할 Model을 찾기 위해 다양한 실험을 진행했다.

위 실험을 통해 구축된 환경이 EfficientNet에 특화되어 있어서인지, ResNet 계열의 Model은 성능이 현저히 낮았다. 다른 Setting에도 변화를 주었으나, 결국 최종 결과에 사용하지는 못했다.

EfficientNet은 b3, b4, b5, b7에 대해서 실험을 진행했고, b4에서 가장 좋은 성능을 보였기 때문에, 최종 Model로 선정되었다.

Data Cleansing

일부 Labeling이 잘못된 Data로 인해 성능이 저하되고 있을 것이라고 판단해, 이를 수행했다.

Noise Data를 색출하기 위한 방법으로 Anomaly Detection과 같은 방법을 찾아보았으나, 구현하기에는 큰 어려움이 따랐다.

군집화를 통한 이상치 탐지나, 평균으로부터의 유사도 측정 등의 Idea는 있었으나 배보다 배꼽이 커지는 경우가 되는 것 같았다.

결국, 토론 게시판에 올라온 CleanLab을 공부하여 사용했다. 이외에도 단순한 Model의 Threads Holds를 정하고, Probability를 통해 Noise Data를 색출해 Labeling을 변경해주었다.

실제로 Noise Data가 많지 않았고, 성능 향상에 큰 도움이 되지는 못했다고 판단한다.

Histogram Equalization

사람이 찍은 사진이기에, 전체적으로 어둡거나 밝은 Image가 있으리라고 판단했고 이를 정규화하는 것이 좋을 것이라 생각했다.

이를 적용해서 시각화해보았으나, 오히려 사람이 보기에는 어려운 Image가 생성되었다.

Model 역시 좋은 성능을 내지 못했고, 사용하지 않기로 결정했다.

Drop-out Rate

Drop-out의 비율이 높을수록 강건한 모델이 생길 것이라고 판단했다.

EfficientNet의 Dropout Layer를 변경해가며 실험을 진행했고, 0.7에서 유의미한 성능 향상을 얻을 수 있었다. 이는 최종 결과에까지 사용되었다.

P1을 통해 얻은 교훈

개선해야 할 점

- 실험에 대한 근거와 이유를 조금 더 탄탄하게 다지고, 유의미한 실험의 비율을 높여야 하겠다.
- 논문 등을 읽고 아이디어를 검증 할 수 있는 실력을 갖춰야 하겠다.
- 실험 Code를 더 체계적으로 관리해야 하겠다.
- Neptune이나 Tensor Board 등의 Tool을 익혀서 Option과 성능을 체계적으로 관리해야 하겠다.
- 아이디어를 구현할 수 있는 Coding Skill을 갖춰야 하겠다.
- Python IDLE 형태로 Project를 수행하는 연습을 해야 하겠다.
- 1등 하신 분처럼 한 Block만 바뀌도 실험을 수행할 수 있도록 _Code를 구성해야 하겠다.
- Inference와 Ensemble의 시간을 아끼기 위해, Test Data에 대한 Probability를 Pickle 등으로 저장해야 하겠다.

느낀 점

이번 대회를 진행하면서, 이 분야의 '실력'이 무엇을 뜻하는 것인지 알게 된 것 같다.

일정 수준까지는 노력과 다양한 실험을 통해서 성능을 개선할 수 있었으나, 그 이상으로 가기 위해선 '실력'이 필요한 것 같았다. 이 분야의 실

력은 다음과 같은 것들이라고 느꼈다.

- 남들이 모두 적용하는 단순한 Idea를 넘어서, Task에 맞는 Idea를 구상하거나 논문 등을 통해 찾아볼 수 있는 능력
- 이러한 Idea를 Code로 구현하고, 적용하는 능력

이외에도 많은 것들이 있겠지만, 이번에 내가 뼈저리게 느낀 점은 위와 같았다.

더 좋은 성적을 거두지 못해 아쉬운 마음도 있지만, 이는 나의 실력의 문제였다고 생각한다.

정말 내공과 실력이 있는 사람이 되기 위해서 훨씬 더 노력해야겠다는 생각이 가득해졌다.