

# 부스트캠프 랩업 리포트 가이드

## <기술적인 도전>

본인의 점수 및 순위

LB 점수 **accuracy 78.7778% / f1 score 0.7321 / 103등**

- 검증(Validation) 전략

1. 우선 좋은 성능을 보이는 모델과 **argumentation** 방법들을 찾기 위해 제공된 데이터셋을 80%는 **training dataset**으로, 20%는 **test dataset**으로 나누어 사용 하였습니다.
2. 모델과 **argumentation** 방법에 대해 몇가지 후보군을 정한 다음 전체 **dataset**에 대해 5개의 **fold**를 만들어 학습에 사용 하였습니다.
3. **k**개의 **fold**를 만들어 학습에 사용하지 않고 전부 **training dataset**으로 사용하였고, 학습의 종료 시점은 **loss**의 적절한 수렴 지점을 정하여 결정 하였습니다.

- 사용한 모델 아키텍처 및 하이퍼 파라미터

1. 아키텍처: resnext50\_32x4d
  - a. LB 점수 : accuracy 78.7778% / f1 score 0.7321
  - b. Training Time Augmentation
    - i. `Resize((512, 384), Image.BILINEAR),`  
`CenterCrop(304),`  
`RandomChoice([RandomHorizontalFlip(p=0.5),`  
`RandomVerticalFlip(p=0.5),`  
`RandomRotation(30)]),`  
`RandomChoice([ColorJitter(brightness=(1,1.1)),`  
`ColorJitter(contrast=(0.2, 3)),`  
`ColorJitter(saturation=(0.1, 3)),`  
`ColorJitter(hue=(-0.5, 0.5)),`  
`ColorJitter(0.1, 0.1, 0.1, 0.1),`  
`RandomGrayscale(p=0.1)]),`  
`ToTensor(),`  
`Normalize(mean=[0.548, 0.504, 0.479],`  
`std=[0.237, 0.247, 0.246]),`  
`RandomErasing(p=0.5,`  
`scale=(0.02, 0.33),`  
`ratio=(0.3, 3.3),`  
`value=0,`  
`inplace=False)`  
`img_size = 304 x 304`
  - c. Test Time Augmentation

- i. `Resize((512, 384), Image.BILINEAR),`  
`CenterCrop(304),`  
`ToTensor(),`  
`Normalize(mean=[0.485, 0.456, 0.406],`  
`std=[0.229, 0.224, 0.225])`
  - d. Loss
    - i.  $(1 - f1 \text{ factor}) * \text{cross entropy} + f1 \text{ factor} * f1 \text{ loss}$  ( $f1 \text{ factor} = 0.85$ )
  - e. optimizer
    - i. MADGRAD
  - f. 추가로 적용한 방법
    - i. f1 loss와 Imbalanced Dataset Sampler를 사용하여 accuracy만 오르고 f1 score가 같이 오르지 않는 문제를 해결 하였습니다.
    - ii. 예측을 잘 수행하지 못하는 data를 직접 확인해보고 Mask에 대한 label이 잘못되어 있는 data에 대해서는 label을 수정해주고, 마스크 착용 유무와 성별이 애매한 데이터는 outlier로 판단하여 제외시킨 후 학습하였습니다.
    - iii. 마스크 착용 유무, 성별, 나이에 대한 classification을 따로 학습한 후 3개 모델이 각각 추론한 결과를 토대로 최종적으로 종합 하였습니다.
    - iv. 전반적으로 batch size가 클수록 더 좋은 모델 성능을 보이는 경향이 있어 accumulation을 이용하여 batch size를 256이 되도록하여 학습 하였습니다.
    - v. MAD Grad를 optimizer로 사용하여 수렴 속도가 굉장히 빠르면서 학습이 안정적으로 진행되었고, 주어진 dataset 전체를 학습에 사용하였기 때문에 epoch를 1로, learning rate를 0.0001로 설정 하였습니다.

## ● 시도했으나 잘 되지 않았던 것들

1. GaussianBlur, GaussianNoise, ImageNetPolicy 등의 다양한 argumentation 방법을 사용해보았지만 성능이 좋지 않았습니다.
2. data imbalance 문제가 가장 심한 age group에 대한 classification의 성능을 높이기 위해 FocalLoss, LabelSmoothingLoss 등 loss에 대한 다양한 방법과 조합을 시도하여 보았지만 성능 향상이 없었습니다. 그 중 LabelSmoothingLoss를 약간 변형하여 인접한 age group에 대해 먼 age group 보다 높은 확률을 부여하는 방식으로 gaussian distribution 모양의 label을 생성하였고, 이와 동시에 어린 나이는 넓은 범위의 group으로, 나이가 들수록 좁은 범위의 group이 되도록 class를 나누어 학습을 하는 방법도 시도해보았지만 성능 향상은 없었습니다.
3. 그리고 pytorch에서 제공하는 pretrained 모델을 거의 전부 시도해보았지만 resnext50\_32x4d 모델보다 높은 성능을 보이는 모델을 찾지 못하였습니다.
4. 5개의 fold를 나누어 각각 학습을 하였고, 그 다음 결과를 soft voting 방식으로 추론하였지만 성능 향상은 없었다.
5. 피어 세션을 통해 age group을 나누는 기준중 하나를 60세로 하지 않고 58세나 59세로 하면 성능이 좀 더 좋아진다는 조언을 듣고 시도해보았지만 성능 향상이 있지는 않았다.

## <학습과정에서의 교훈>

주어진 문제에 대한 다양한 해결 방법이 있고, 그 중 최적의 해결 방법을 찾는 것이 순위를 가른다는 생각이 들었습니다. 하지만 혼자서 최적의 해결 방법을 찾는 것은 많은 실험을 해야하며 그에 따라 시간이 오래 걸리게 되기 때문에 같은 목적을 가진 많은 사람들의 의견을 들어보고 서로의 실험과 결과를 공유하는 것이 더 높은 순위로 올라갈 수 있는 방법이란 생각을 가지게 되었습니다.

저는 실제로 저의 상식내에서 문제를 해결하기 위해 단순히 **efficientnet b0 model**과 **adam optimizer**로만 학습을 하였고, 근거없이 이 방법이 가장 좋을 것이라고 생각하였습니다. 하지만 피어세션에서 만난 동료들을 통해 **resnet**도 좋은 성능을 보인다는 점을 알게 되어 다양한 모델에 대해 실험해 볼 필요가 있다는 생각하게 되었습니다. 또한 최근에 나온 **madgrad**라는 **optimizer**가 **adam**보다 안정적으로 빠르게 학습을 한다는 정보도 얻게 되어 적용할 수 있었습니다.

**U stage**에서는 만날 수 없었던 많은 동료들을 만나게 되고, 문제를 바라보는 다양한 관점을 들을 수 있어 이번 피어세션이 저에겐 정말 유익하다고 느껴졌고, 시간이 지날수록 문제를 더 잘 해결하고 싶은 욕심에 더욱 적극적으로 참여하게 되었습니다. 그리고 **stage 1** 이후의 **stage 2,3,4**를 같이 할 팀원들을 얻을 수 있게 된 점이 정말로 좋았던 점입니다.

## <마주한 한계와 도전숙제>

### 아쉬웠던 점들

1. 문제를 해결하기 위해 제 나름대로 계획을 세운 다음 피드백을 통해 점차 개선된 모델을 만들며 진행했어야 했는데 모든 경우에 대한 실험을 해보고 싶은 욕심 때문에 정작 문제에 대한 성능 향상은 시간내에 이루지 못한 점이 아쉬웠습니다.

### 한계/교훈을 바탕으로 다음 스테이지에서 새롭게 시도해볼 것

1. 다음 **stage**부터는 초반에 피어세션과 팀원들과의 소통을 좀 더 적극적으로 활용하여 많은 아이디어를 얻고 다양한 시도를 해본 다음 후반에는 좁혀진 선택지 안에서 세세한 부분을 고치며 정답에 가까워질 수 있도록 해보려고 합니다.