# 농업 환경 변화에 따른 작물 병해 진단 AI 경진대회

TEAM 닉네임이제일어려워

# Contents



LG AI Research

# | 팀원 및 역할 소개

## 조승제 (팀장)

- 베이스라인 모델 설계 및 구현 (Image Captioning Model)
- 베이스라인 코드 구축 및 관리
- Multiprocessing을 사용한 Inference Time 개선

## 김종현 (팀원)

- 환경 변수 데이터(CSV) 전처리 (minmax, standard scaling)
- 이미지 Augmentation 실험
- 전반적인 모델 훈련 및 추론

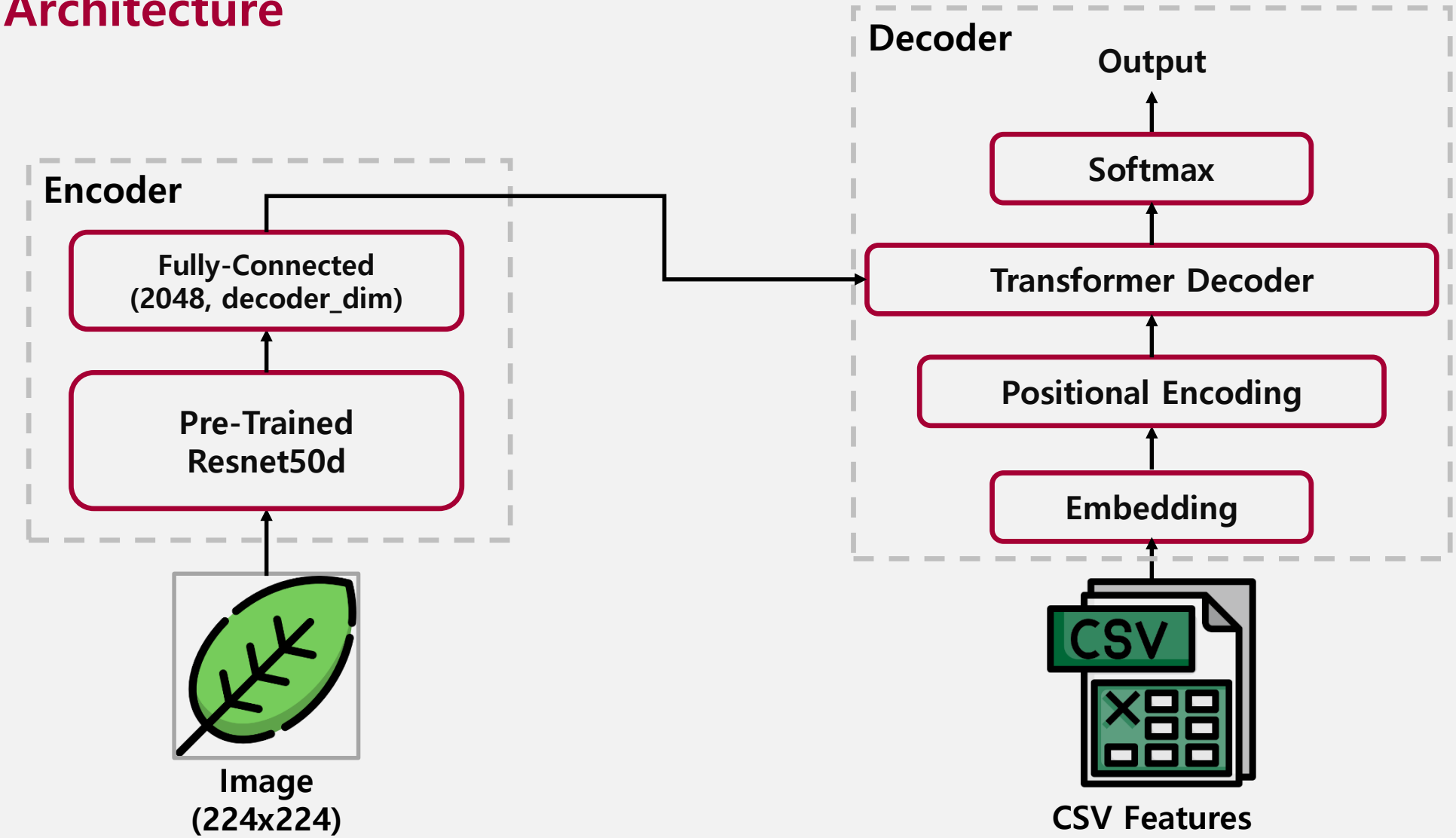
# 문제 접근 방식 (문제 분석)

- Train / Test Set의 비율이 약 1:10이고, Train Set의 크기는 5,767개로 적은 편에 속함
- 적은 Train Set으로, 전체 Test Set에 대해 Robust하고 Generalization된 모델 개발 목표
- 환경 변수 데이터(CSV)에서 90% 이상의 결측치를 가지는 column을 삭제하여 전처리

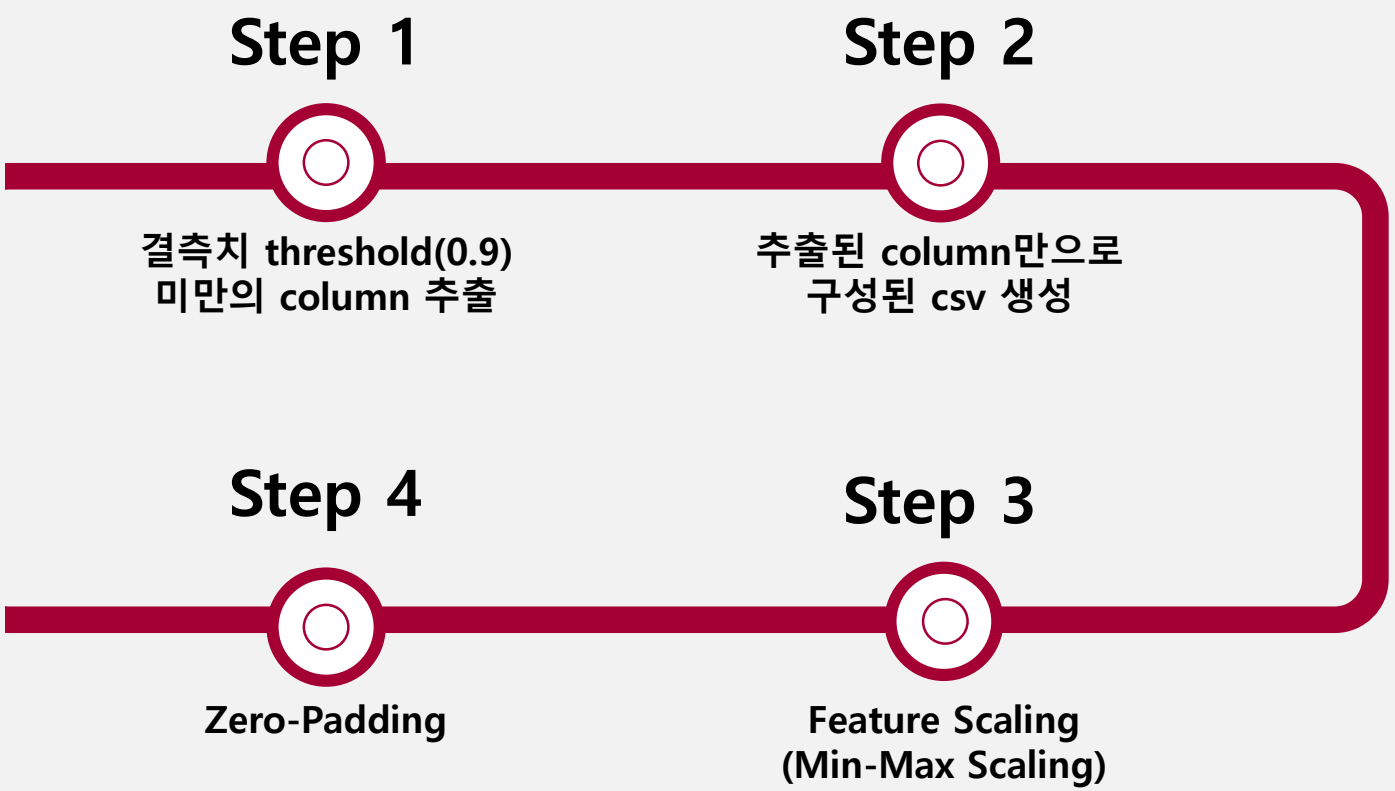
# 모델 설계 및 구현

- 기존의 Baseline code와 동일한 Encoder-Decoder 구조
- Baseline Decoder (LSTM)에서는 Encoder Output인 image 정보를 학습하지 않음
- Image Data + Time-Series Data 상관관계를 함께 학습시키기 위해 Vanilla Transformer Decoder 사용
- 내부 실험 시 LSTM보다 Transformer Decoder의 F1 Score가 더 높아 Transformer Decoder 최종 사용

# Model Architecture

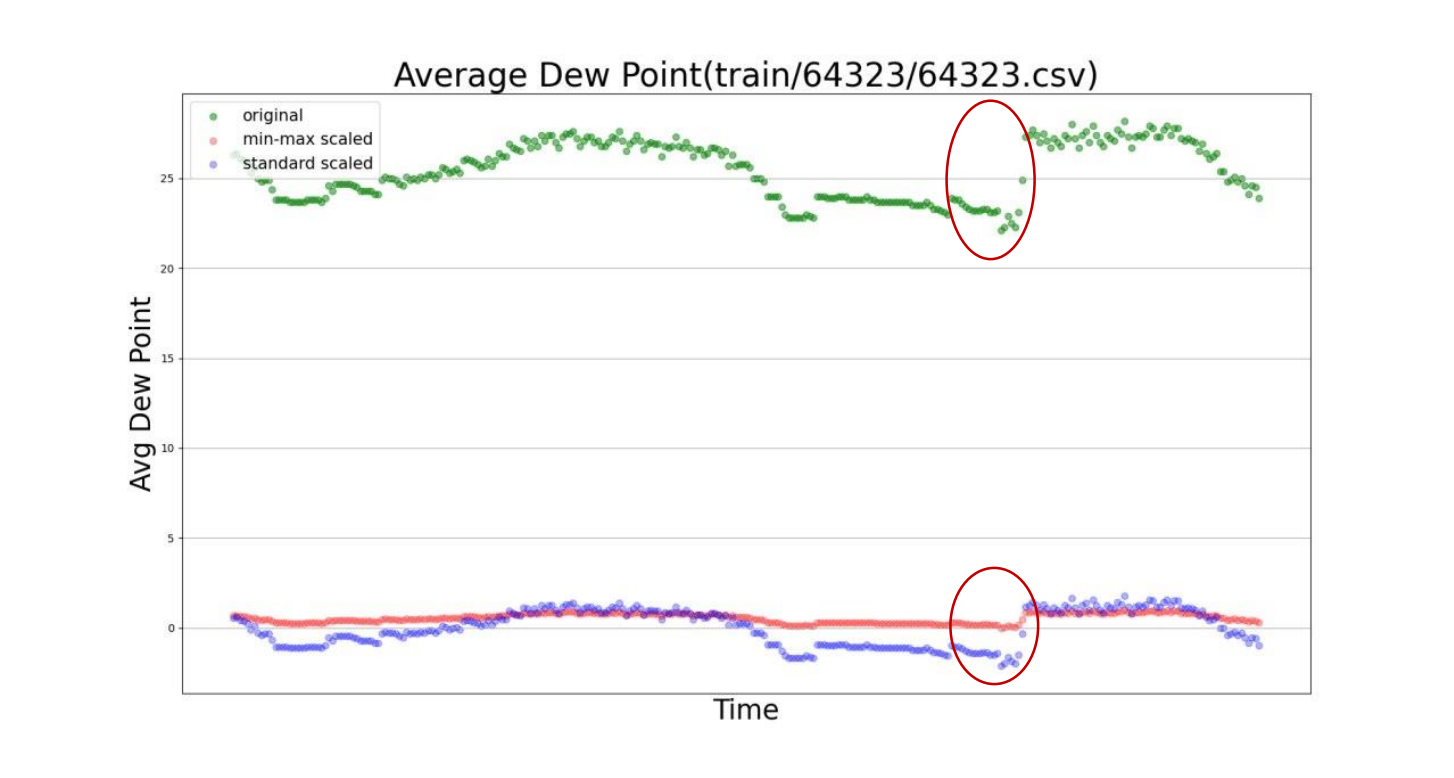


# | 환경 변수 데이터 전처리


Step 1
결측치 threshold(0.9)
미만의 column 추출
Step 2
추출된 column만으로
구성된 csv 생성
Step 3
Feature Scaling
(Min-Max Scaling)
Step 4
Zero-Padding

# Feature Scaling

- CSV 데이터의 각 Feature(온도, 이슬점 및 습도)에 대한 측정단위가 다르므로, Scaling 수행
- Normalization, Standardization 두 방식에 대해 실험




| Normalization |
|---|
| 스케일링 시 최대, 최소값이 사용 |
| 피처의 크기가 다를 때 사용 |
| [0, 1] (또는 [-1, 1]) 사이의 값으로 스케일링 |
| 분포에 대해 모를 때 유용 |
| MinMaxScaler, Normalizer |
| **Standardization** |
| 스케일링 시 평균과 표준편차가 사용 |
| 평균이 0, 표준편차가 1인 것을 확인하고 싶을 때 (그렇게 만들고 싶을 때) 사용 |
| 특정 범위로 제한되지 않음 |
| 피처가 정규분포(가우시안 분포)를 따를 경우 유용 |
| StandardScaler, RobustScaler |

# Feature Scaling

- Feature 분포 파악이 어려우므로 Normalization이 더 효과적일 것이라고 가정
- Submission 시 동일조건에서 MinMax Scaling Method의 Score가 더 높았음

| | | | | | |
|---|---|---|---|---|---|
| 645709 | mixup_kfold_result.csv<br>loss_based, minmax, label_smoothing 0.1 edit | **MinMax Scaling** | 2022-02-03 00:00:53 | 0.9504343732<br>- | ☑ |
| 646435 | mixup_standard_kfold_result.csv<br>loss_based, standard, mixup, resnet50, label smoothing 0.1 edit | **Standard Scaling** | 2022-02-04 00:18:43 | 0.9483455209<br>- | ☐ |

# Image Augmentation

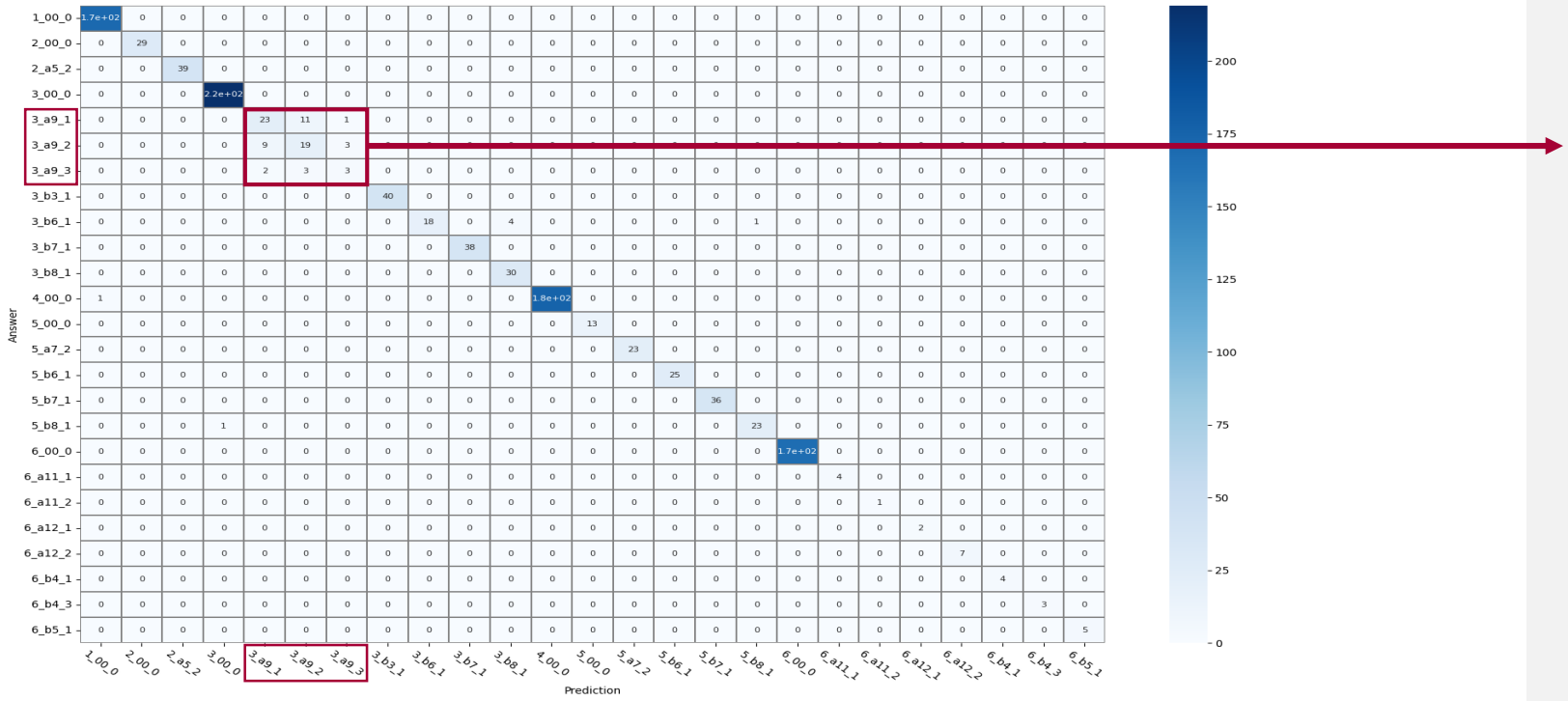
- 내부 실험 결과, 다양한 Augmentation 적용 시 병해 부위 훼손 가능성 발견
- 따라서 원본 이미지를 유지하는 Rotate, Flip 류의 Augmentation만 적용

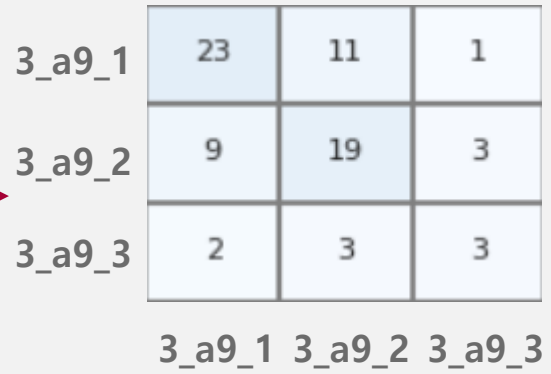
- ✓ Rotate(30, p=.5)
- ✓ RandomRotate90(p=.5)
- ✓ Resize(224, 224)
- ✓ HorizontalFlip(p=.5)
- ✓ VerticalFlip(p=.5)
- ✓ Normalize (ImageNet)


LG AI Research

# Image Augmentation

- Augmentation만 적용시 Validation Set에 대한 Score 하락 원인 파악
- 파프리카 흰가루병 (3_a9_X) Label의 예측 정확도가 낮음

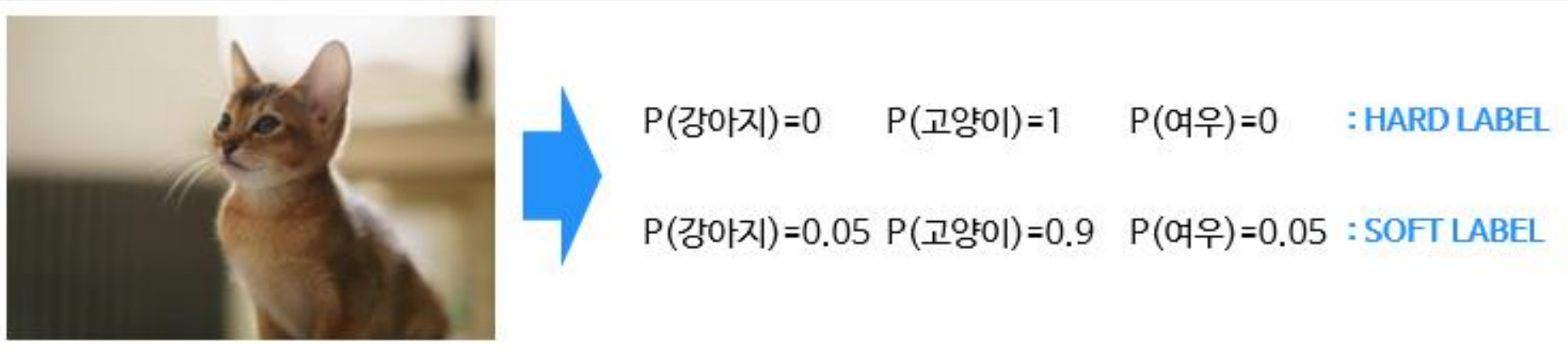



LG AI Research

# Label Smoothing

- 기존의 One-Hot Vector로 분류 시 예측 값이 over-confidence 해짐
- 실제 Label이 가장 높은 확률 값을 가지되, 정답을 제외한 다른 클래스를 균일 분포 (Uniform Distribution)로 만들어 over-confidence 방지 (Hard Label → Soft Label)
- Submission Score 기준 약 0.5% 향상 (+0.0041)

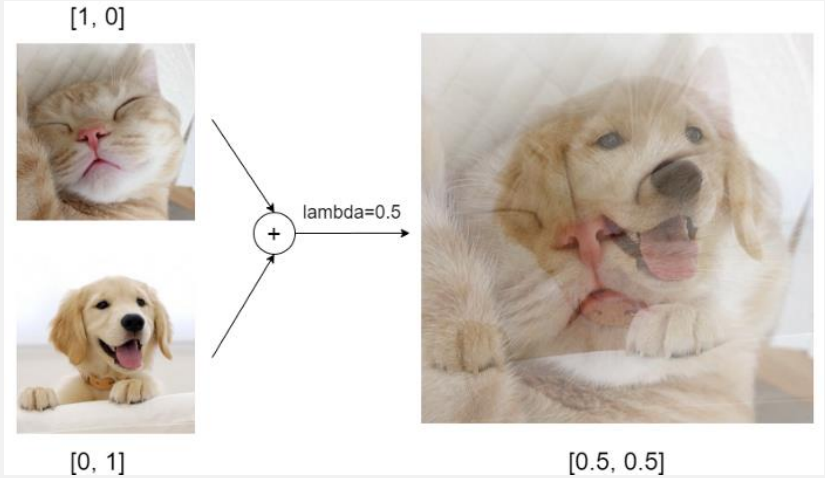


# Mixup

- 학습을 진행할 때 무작위로 두 개의 샘플 $(x_i, y_i)$, $(x_j, y_j)$를 뽑아 $(\hat{x}, \hat{y})$를 만들어 학습에 사용
- $\lambda \in [0, 1]$는 Beta$(\alpha, \alpha)$에서 추출 → $\alpha$값이 1일 때, 균일 분포 (Uniform Distribution)
- 동일조건에서 Mixup 사용시, Submission Score 향상 (+0.0035)

$$\begin{aligned} \hat{x} &= \lambda x_i + (1-\lambda) x_j \\ \hat{y} &= \lambda y_i + (1-\lambda) y_j \end{aligned}$$

| | | | | | |
|---|---|---|---|---|---|
| 645709 | mixup_kfold_result.csv<br>loss_based, minmax, label_smoothing 0.1 edit | With Mixup | 2022-02-03 00:00:53 | 0.9504343732<br>- | ☑ |
| 645422 | kfold_loss_based_result.csv<br>label smoothing 0.1, soft voting edit | Without Mixup | 2022-02-02 13:42:51 | 0.9469021365<br>- | ☐ |




LG AI Research

# Test Dataset Inference

- RTX 3090
- Batch Size 128 기준
- Best Model (Mixup + 5-fold)
- TTA를 적용해도 성능 향상이 없음
- Multiprocessing : 5

| | Best Model | Best Model + TTA | Best Model + Multiprocessing | Best Model + Multiprocessing + TTA |
|---|---|---|---|---|
| Time | 25m 40s | 41m 40s | 7m 50s | 17m 50s |
| Score (Public) | 0.95043 | 0.95043 | 0.95043 | 0.95043 |

# 결론

- 파프리카 흰가루병 분류 난이도가 높았고, 학습 데이터셋의 크기가 작아 Generalization한 모델을 만들기 어려웠음
  - ✓ K-Fold Validation, Soft Voting, Mixup 사용

- 실험 과정에서 K-Fold Validation, Test Time Augmentation으로 Inference 시간이 기하급수적으로 늘어남
  - ✓ Multiprocessing을 통해 Inference 시간 단축

- Encoder 모델로 EfficientNetB7, ResNet200 등 큰 모델을 사용해도 성능 차이가 크지 않았음
  - ✓ 비교적 크기가 작은 ResNet50을 사용하여, 성능과 추론 시간에서 이점을 가짐

# 한계점

- 파프리카 흰가루병 분류를 위해 시간을 많이 사용해서 보다 다양한 실험을 하지 못함
  - ✓ 파프리카 흰가루병만의 분류를 위한 모델 구성을 시도하는 등 생각보다 많은 시간을 사용
  - ✓ 다양한 Scheduler, Hyper Parameter (weight decay 등) 실험을 하지 못함
  - ✓ Class imbalance 문제를 해결하기 위한 Focal Loss를 사용하지 못했음
  - ✓ 환경 변수 데이터(CSV)의 결측치를 0으로 처리하여, 의미 없는 값을 학습시킴
  - ✓ Mean/Median Imputation와 같은 방법을 사용하지 못함

# Q & A

LG AI Research

# 감사합니다.

LG AI Research