



Fast campus

upstage

Upstage AI Lab

Computer Vision Seminar | 2025. 07.
11(금)

목차

01. 팀 소개
02. 경진대회 수행 절차 및 방법
03. 분석 인사이트 및 결과
04. 회고

01

팀 소개

팀장/팀원 소개
협업 방식

* [Five Guys] 팀 각오 입력



팀장
이승민
데이터 엔지니어/
model 중심 실험



팀원
이경도
AI/데이터분석
이미지 전처리/모델 실험



팀원
김재덕
NLP/의공학과
팀원



팀원
이상원
LLM/건축공학과
팀원



팀원
최웅비
자동화/컴퓨터공학과
EDA, 모델 테스트

경진대회 협업 방식

: Computer Vision [대회] Classification

협업 방식 요약

1. 정기 회의

- 주 2~3회 10시 or 14시에 zoom에서 1시간의 회의를 진행.

2. 진행 상황 공유

- 회의에서 개인 작업 진행도, 해결할 과제 및 발굴한 기법 공유

3. 전문성 강화

- 각자 관심 분야(증강, 모델, 하이버파리미터 등)를 심화 실험하고, 효과적인 기법을 발굴해 파이프라인 반영

4. 성과 고도화

- 정기 공유를 통해 서로의 성과를 결합 고도화 및 개선

02

경진대회 수행 절차 및 방법

목표 수립
수행 내용 / 수행 결과

경진대회 목표 수립

: Computer Vision [대회] Classification

주제

Document Type Classification | 문서 타입 분류

문서는 금융, 보험, 물류, 의료 등 도메인을 가리지 않고 많이 취급됩니다. 이 대회는 다양한 종류의 문서 이미지의 클래스를 예측합니다.

목표

목표

- * 전처리부터 학습, 추론까지의 CV 파이프라인을 각자 구현 및 실습
- * 팀 전체 고득점 달성

개요

소개 및 배경 설명

- * 이번 Computer Vision 대회는 문서 이미지의 타입을 분류하는 모델을 개발하고, 데이터 증강 및 하이퍼파라미터 튜닝 등 다양한 기법을 적용해, 성능을 향상시키는 목표로 진행.

기간

2025. 06. 30 ~ 2025. 07. 10

경진대회 수행 내용

: Computer Vision [대회] Classification

1
* 개발 환경 구축

vscode, cursor
wandb, hydra,
torchLightning

2
* 데이터 분석

mislabeling
class imbalance
도메인 갭 (train<->test)

3
* Feature 엔지니어링

VoronoiTessellation
GaussianBlur
Rotate
HorizontalFlip
Normalize

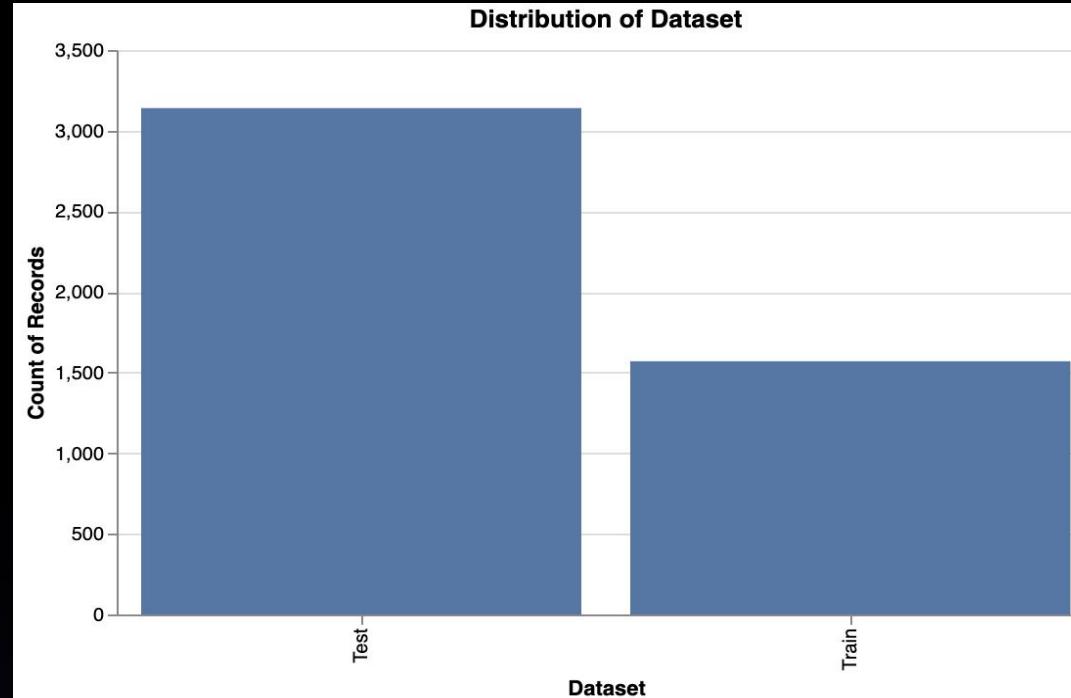
4
* 모델 선택 학습 및 평가

model : resnet50, efficientnet,
convnext
평가 : f1 score, acc, class f1

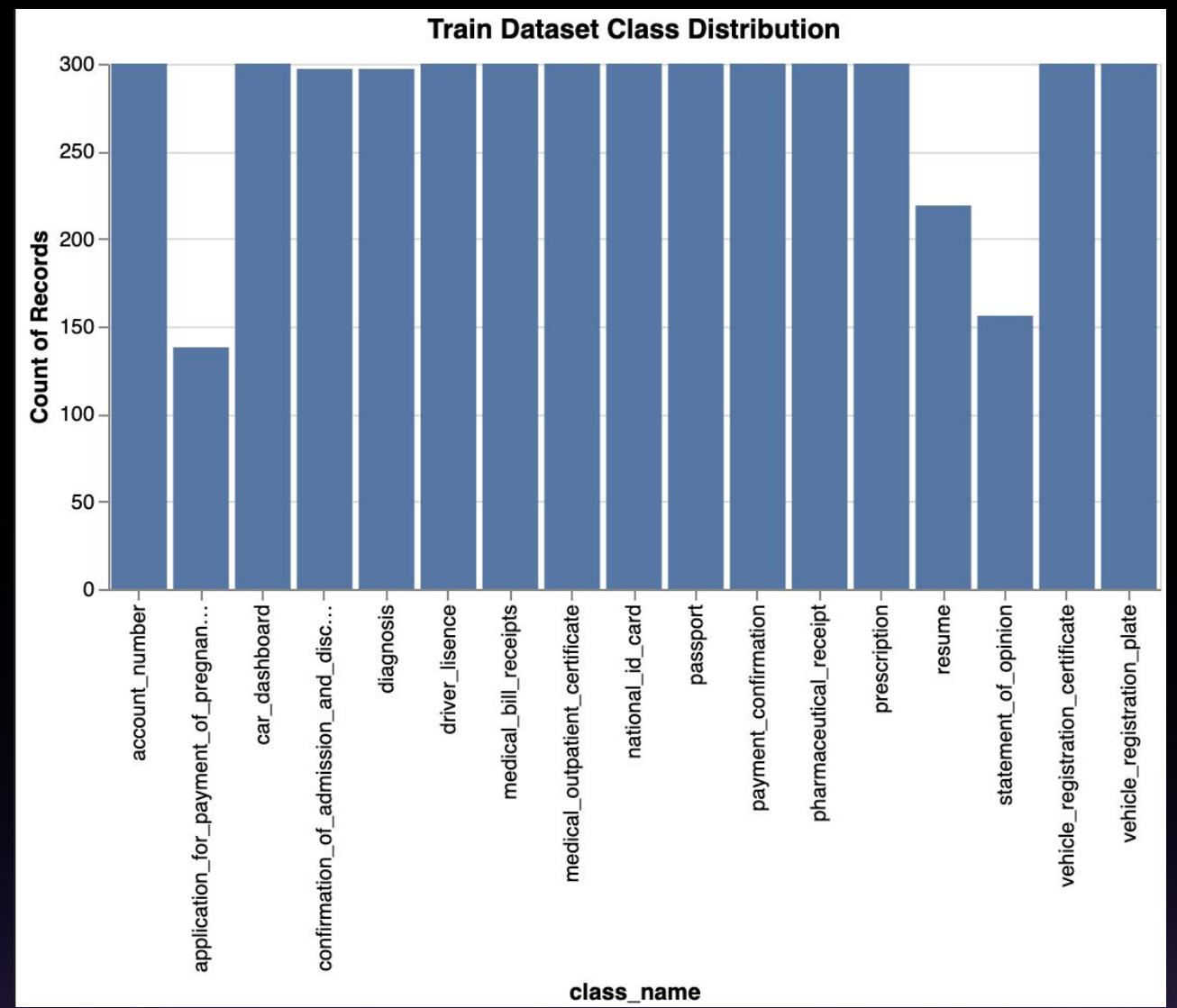
경진대회 수행 결과

: Computer Vision [대회] Classification

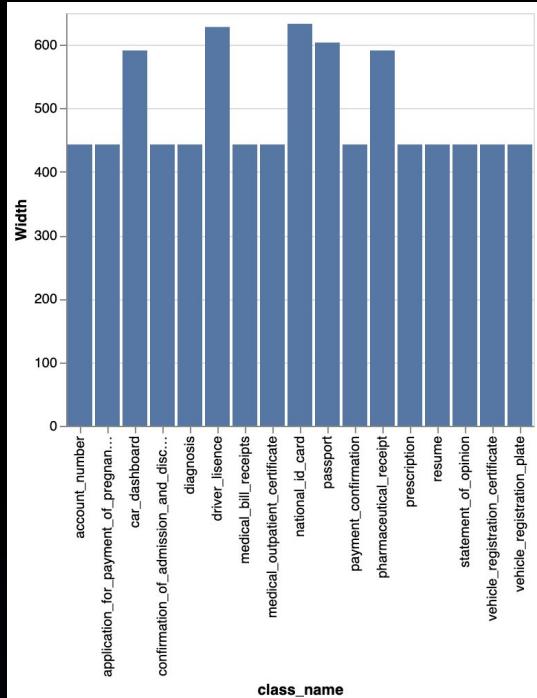
* 결측값 확인 및 데이터 전처리, 모델 실험 등에 관해 팀에서 시도한 점의 결과 이미지와 요약 정리



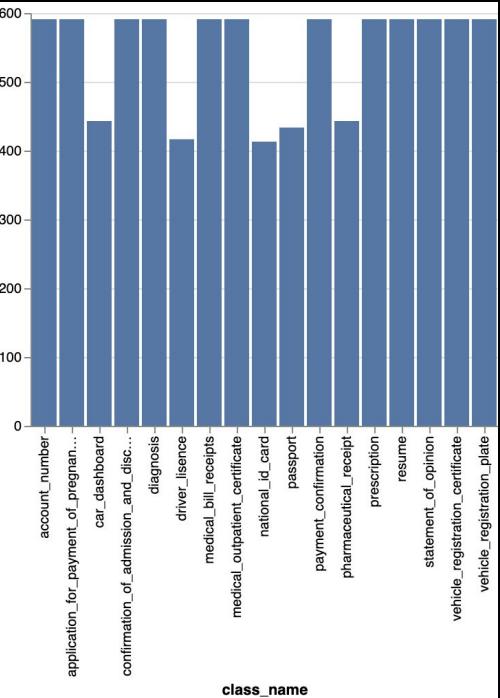
- train data가 test data에 비해 적음.



- 각 클래스별 데이터
分布



- 클래스별 이미지의 높이, 길이
비교



경진대회 수행 결과

: Computer Vision [대회] Classification

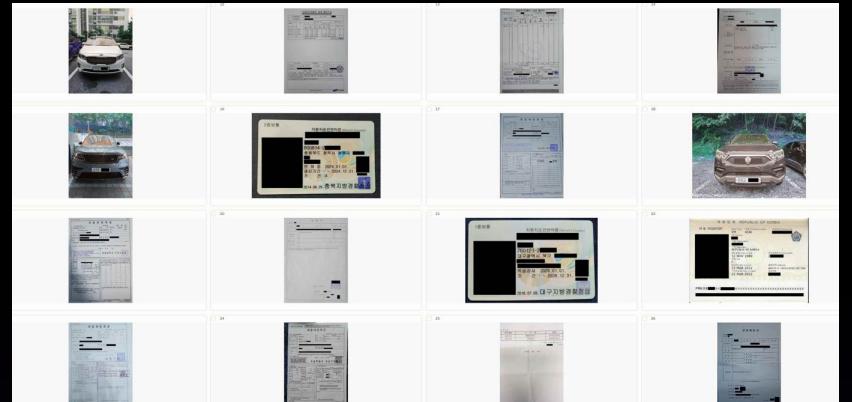
* 핵심 문제 분석: 도메인 갭(Domain Gap)

가장 중심적인 문제는 학습 데이터셋과 테스트 데이터셋 간의 심각한 분포 불일치, 즉 "도메인 갭(Domain Gap)"입니다. 모델이 한 종류의 데이터로 학습된 후, 전혀 다른 종류의 데이터로 테스트되고 있습니다.

주요 근거

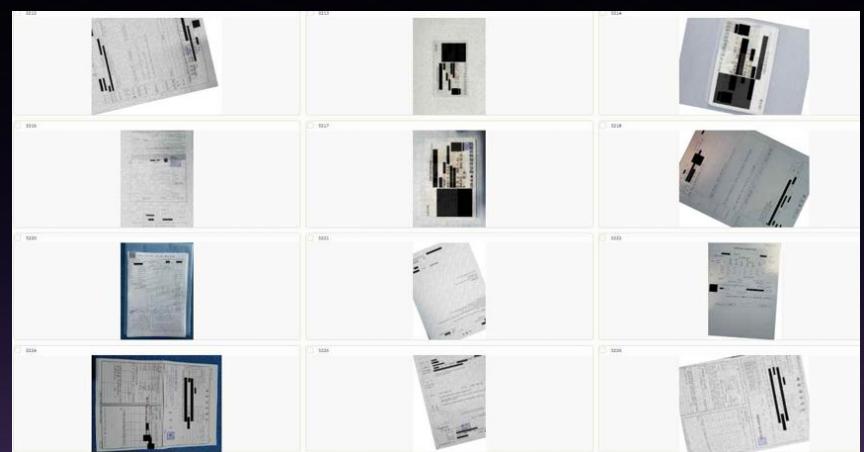
A) 정량적 불일치:

분석 결과, 테스트셋(12.6°)과 학습셋(1.9°)의 평균 회전 각도에서 554%의 상대적 차이가 발견되었습니다. 또한 테스트셋은 과다 노출 이미지의 비율이 학습셋(20%)에 비해 2배 이상(46%) 높습니다.



B) 정성적 불일치:

테스트 이미지는 까다로운 실제 환경에서의 휴대폰 촬영 특성을 보입니다.
심각한 원근 왜곡 (비스듬한 각도에서 촬영된 사진).
그림자가 있는 어둡고 균일하지 않은 조명. 배경의 간섭 및 일부만 보이는 문서.
상당한 수준의 회전.



결론:

현재의 평가는 신뢰할 수 없습니다. 모델이 테스트 환경에 대해 준비되지 않았으므로, 결과는 모델의 실제 능력을 반영하지 못합니다. 이 도메인 갭을 줄이는 것이 최우선 과제입니다

경진대회 수행 결과

: Computer Vision [대회] Classification

* 우선순위 기반 데이터 증강 전략(Augmentation Priority)

학습 / 테스트 데이터 간의 큰 차이가 성능 저하의 가장 큰 원인입니다.

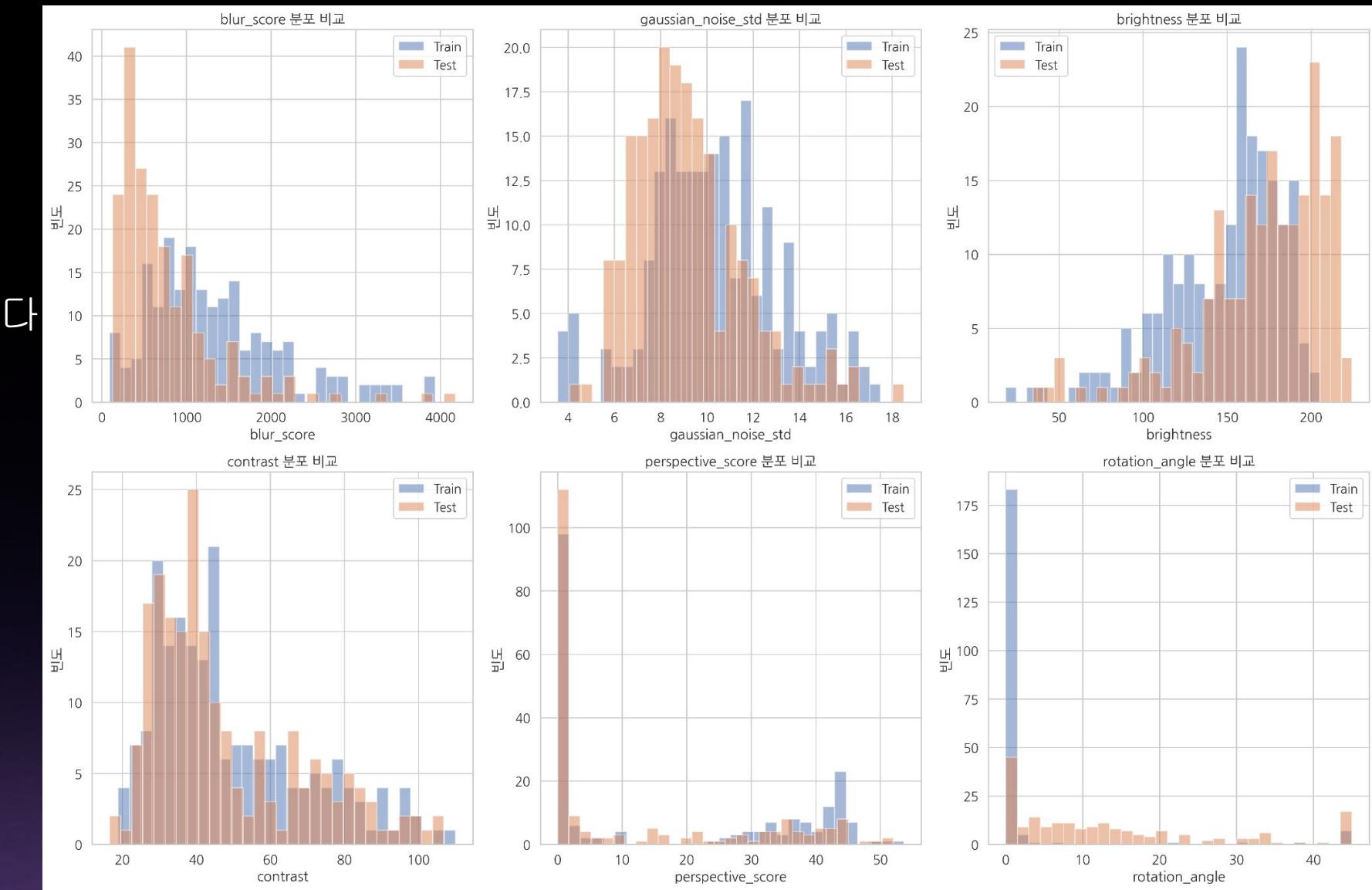
이 문제를 해결하려면, 테스트셋에서 발견된 실제 환경 조건을

시뮬레이션하도록 학습 데이터를 증강해야 합니다.

#1 우선순위: 회전과 상하 반전 증강

이유: 실제 성능 저하의 가장 큰 원인입니다.

방법: 회전 (Rotation): 모델은 회전에 불변(invariant)하도록 학습 되도록 했습니다



#2 우선순위: 3번 label 특징인 조명 및 색상 증강:

이유: 테스트셋에 과다 노출된 이미지가 훨씬 많습니다.

방법: 무작위 밝기 및 대비 조정을 사용하도록 했습니다.

#3 우선순위 3: 노이즈 및 블러 증강:

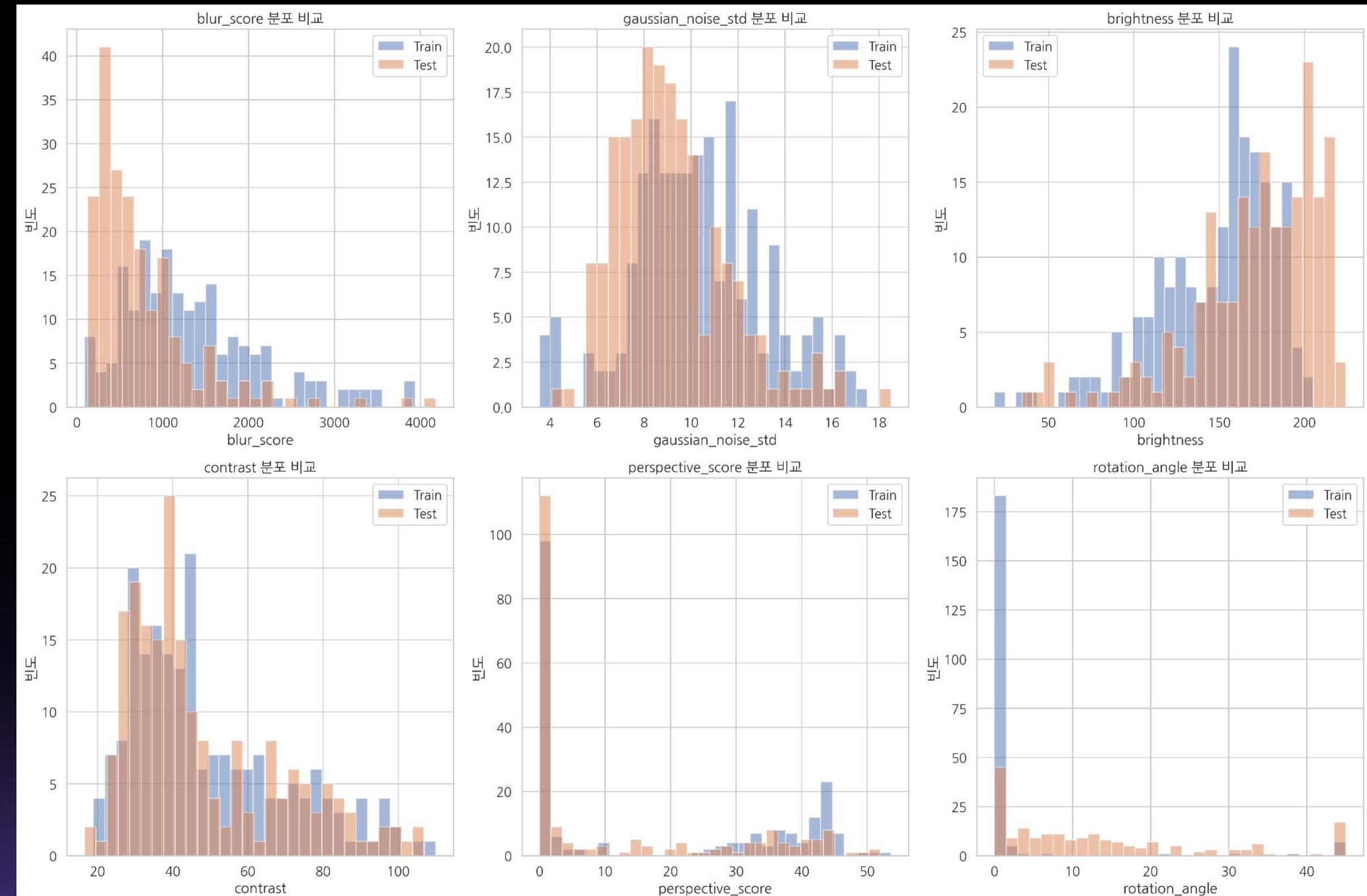
이유: 데이터셋 간에 노이즈 유형(가우시안 vs. 임펄스)과 블러 수준이 다릅니다.

방법: 노이즈와 블러 수준이 유사하게 학습하도록 했습니다. (*실제 적용 안했음)

경진대회 수행 결과

: Computer Vision [대회] Classification

* Train Vs. Test 데이터의 정량적 차이



경진대회 수행 결과

: Computer Vision [대회] Classification

* batch mixup : f1 score +0.3

```
def training_step(self, batch, batch_idx):
    images, targets = batch
    if hasattr(self, 'mixup_fn') and self.mixup_fn is not None:
        images, targets = self.mixup_fn(images, targets)
```

경진대회 수행 결과

: Computer Vision [대회] Classification

- * 각 클래스의 데이터를 10배로 증강(회전, 색반전, GaussianBlur, 좌우반전 적용): f1스코어 + 0.24

```
transform = A.Compose([
    A.InvertImg(p=0.5),
    A.HorizontalFlip(p=1.0),
    A.GaussianBlur(p=0.3),
    A.Rotate(limit=30, border_mode=0, value=(255,255,255), p=1.0),
    A.Resize(224, 224),
])
```

```
for label in sorted(df['target'].unique()):
    class_df = df[df['target'] == label]
    img_names = class_df['ID'].tolist()
    n_origin = len(img_names)
    n_to_add = min(FIT_COUNT, n_origin * 10) - n_origin
    if n_to_add <= 0:
        continue
```

- * 모든 클래스의 데이터를 1500까지 증가: f1스코어 + 0.04

```
for label in sorted(train_df['target'].unique()): #
    class_df = train_df[train_df['target'] == label] #
    # img_names는 'train/0a1b8d3c52d64052.jpg'와 같은 형태
    img_ids_with_prefix = class_df['ID'].tolist()
    n_origin = len(img_ids_with_prefix) #
    fit_count = FIT_COUNT_MAP.get(label, 1500) #
    n_to_add = fit_count - n_origin #
    if n_to_add <= 0: #
        continue
```

BEST

: Computer Vision [대회] Classification

초기 설정: 학습 중 Albumentations와 Augraphy 학습 도중에 적용 → 저장 용량 ↓, 과적합 방지

실험 결과: Augraphy를 사전 적용하고, 학습 중에는 Albumentations만 적용 시 F1-score 개선 ($0.9363 \rightarrow 0.9577$)
→ 결론: Augraphy는 사전 처리, Albumentations는 학습 중 적용이 성능에 더 효과적

Augraphy (학습 중 사용 X, 사전 처리로만 사용)

- 배경 생성:
 - PatternGenerator와 Delaunay 무조건 적용 ($p=1.0$)
- 텍스처 추가:
 - NoiseTexturize, BrightnessTexturize, PaperFactory

Albumentations

- Normalize, HorizontalFlip, VerticalFlip, Rotate

BEST

: Computer Vision [대회] Classification

손실 함수:

- Focal Loss: 분류 어려운 샘플에 집중
- Class-Balanced Loss: 소수 클래스에 더 큰 가중치 부여
- 결합해서 CB-Focal Loss

소수 클래스/분류하기 어려운 클래스:

- Hard Negative Mining: 학습 전 어려운 negative 샘플 선별
- Online Hard Example Mining (OHEM): 학습 중 손실 큰 샘플만 사용
- 사용자는 둘 중 하나 선택 가능, 동시 사용 비추천 → F1-score 다소 하락 ($0.9289 \rightarrow 0.9265$)

경진대회 수행 결과

BEST : Computer Vision [대회] Classification

수행 결과

최적화 및 스케줄링

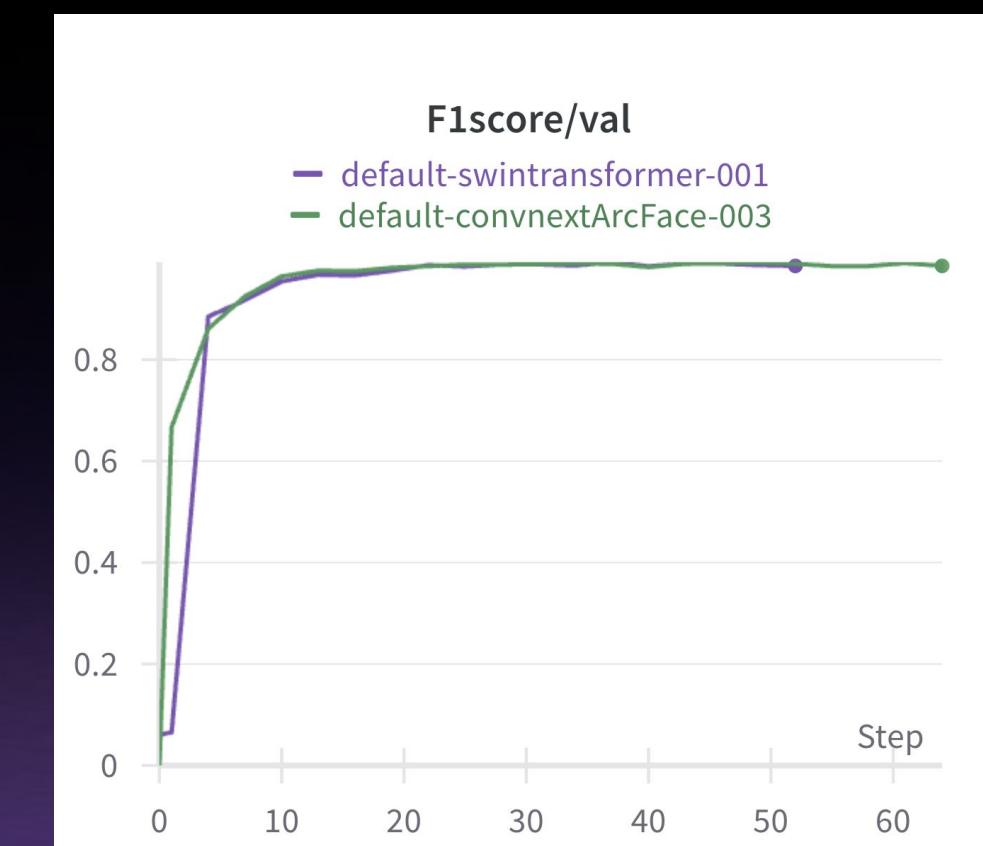
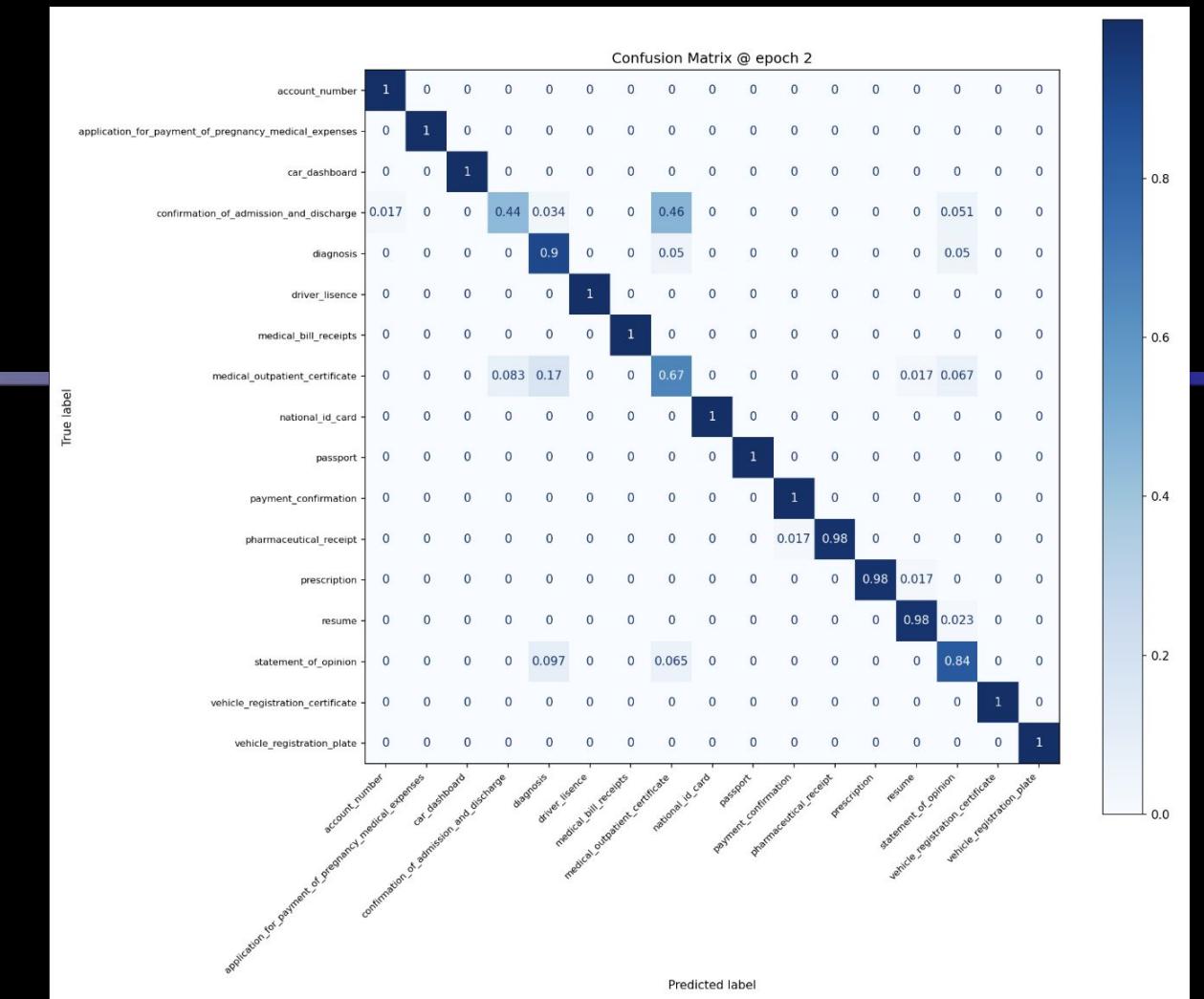
- AdamW 옵티마이저 + CosineAnnealing 스케줄러
- Validation F1@ 10 epoch 동안 향상 없으면 조기 종료

예측 시

- TTA(Test-Time Augmentation) 적용 → F1-score 감소 ($0.9598 \rightarrow 0.9553$)

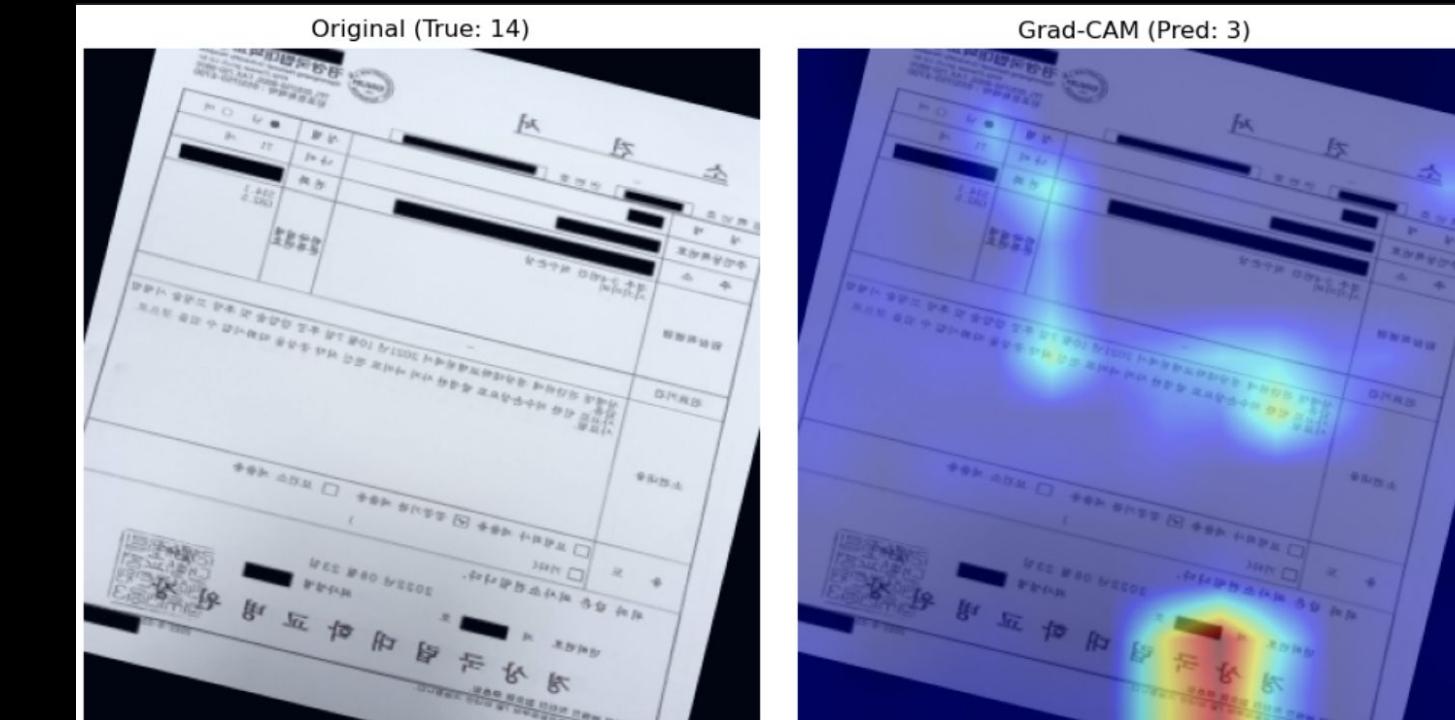
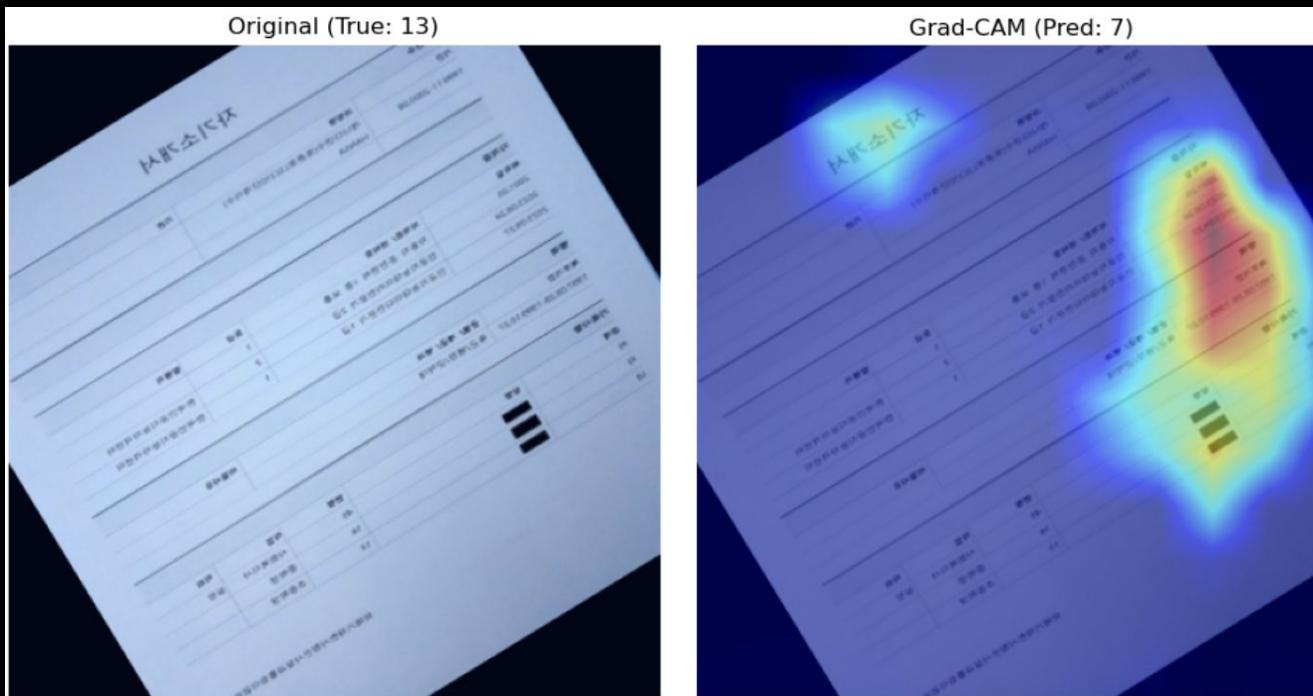
학습 중 분석

- 각 epoch 종료 시 Confusion Matrix 이미지 생성



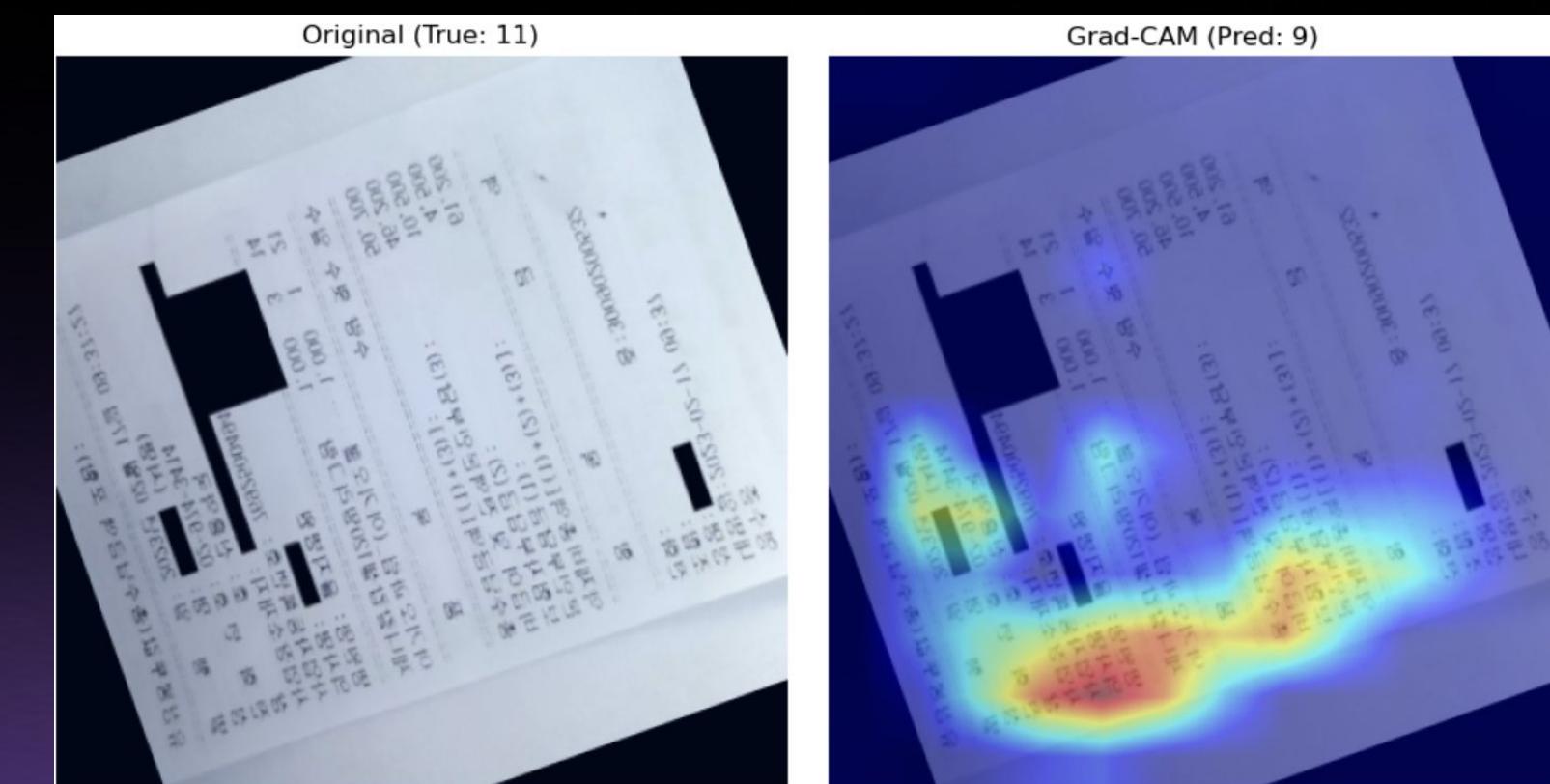
BEST

: Computer Vision [대회] Classification



학습 후 분석

- Grad-CAM으로 validation에서 오분류 이미지 시각화



경진대회 인사이트 공유

: Computer Vision [대회] Classification

02. 인사이트 도출

학습/테스트 데이터 불일치(Domain Gap)

우리의 학습 데이터는 대부분 스캔된 것처럼 깨끗하지만, 실제 테스트 데이터는 휴대폰으로 찍은 것처럼 왜곡과 노이즈가 심합니다. 이 데이터 환경의 차이가 성능 저하의 가장 큰 원인입니다. 실제 환경과 유사하게 학습 데이터를 만들 수 있으면 성능을 높일 수 있습니다.

정확도 개선의 한계

(Resnet50) 테스트 정확도가 79%에 머무는 이유는, 특정 각도 이상으로 기울어진 문서 이미지에 대한 오답률이 높기 때문입니다. 따라서 회전(rotation) 데이터 증강을 강화하면 성능을 크게 개선할 수 있습니다.

신뢰도(Confidence) 분석

모델이 정답을 맞히더라도, 특히, 상하 반전된 이미지에 대해서는 예측 신뢰도가 50% 이하로 매우 낮게 나타납니다. 상하좌우 반전에 대한 데이터 증강을 집중적으로 수행하여 모델의 확신도를 높여야 합니다.

번외

: Computer Vision [대회] Classification

- * 잘못 라벨링된 데이터 정정

계획

CLIP 임베딩 추출(이미지를 의미 기반으로 벡터 공간에 표현하는 방법) → FAISS(Facebook AI

Similarity Search: 대규모 벡터 데이터를 빠르게 검색 or 군집화)로 클러스터링

-> 혼합도 높은 클러스터 구하기

-> 혼합도 높은 클러스터 순서대로 잘못 라벨링된 데이터 fiftyone으로 수동 검토

번외

: Computer Vision [대회] Classification

- * 잘못 라벨링된 데이터 정정

CLIP 임베딩 추출 → FAISS로 클러스터링

```

from transformers import CLIPProcessor, CLIPModel #clipmodel: 모델 자체, clipprocessor: clipmodel 전처리기
import torch
from torchvision.transforms import Compose, Resize, ToTensor, Normalize
#
# CLIP 준비
clip_model = CLIPModel.from_pretrained("openai/clip-vit-base-patch32").eval().cuda()
clip_processor = CLIPProcessor.from_pretrained("openai/clip-vit-base-patch32")

def extract_clip_features(image_paths, batch_size=32):
    features = []
    for i in tqdm(range(0, len(image_paths), batch_size)):
        batch = [Image.open(p).convert("RGB") for p in image_paths[i:i+batch_size]]
        inputs = clip_processor(images=batch, return_tensors="pt", padding=True).to("cuda")
        with torch.no_grad():
            outputs = clip_model.get_image_features(**inputs)
            outputs = outputs / outputs.norm(p=2, dim=-1, keepdim=True) # normalize
        features.append(outputs.cpu())
    return torch.cat(features)

clip_features = extract_clip_features(df["file_path"].tolist())

```

```

    ↘ import faiss
    import numpy as np
    from sklearn.preprocessing import normalize

    # clip_features는 torch.Tensor → numpy 변환 후 float32로 변환
    features_np = clip_features.numpy().astype("float32")

    # optional: 정규화 (cosine 기반 거리 계산에 유리함)
    features_np = normalize(features_np, axis=1)

    # 클러스터 수 (예: 30)
    n_clusters = 30

    # FAISS k-means 초기화
    kmeans = faiss.Kmeans(d=features_np.shape[1], k=n_clusters, niter=300, verbose=True, seed=42)
    kmeans.train(features_np)

    # 클러스터 할당
    _, cluster_ids = kmeans.index.search(features_np, 1)
    df["cluster"] = cluster_ids.flatten()

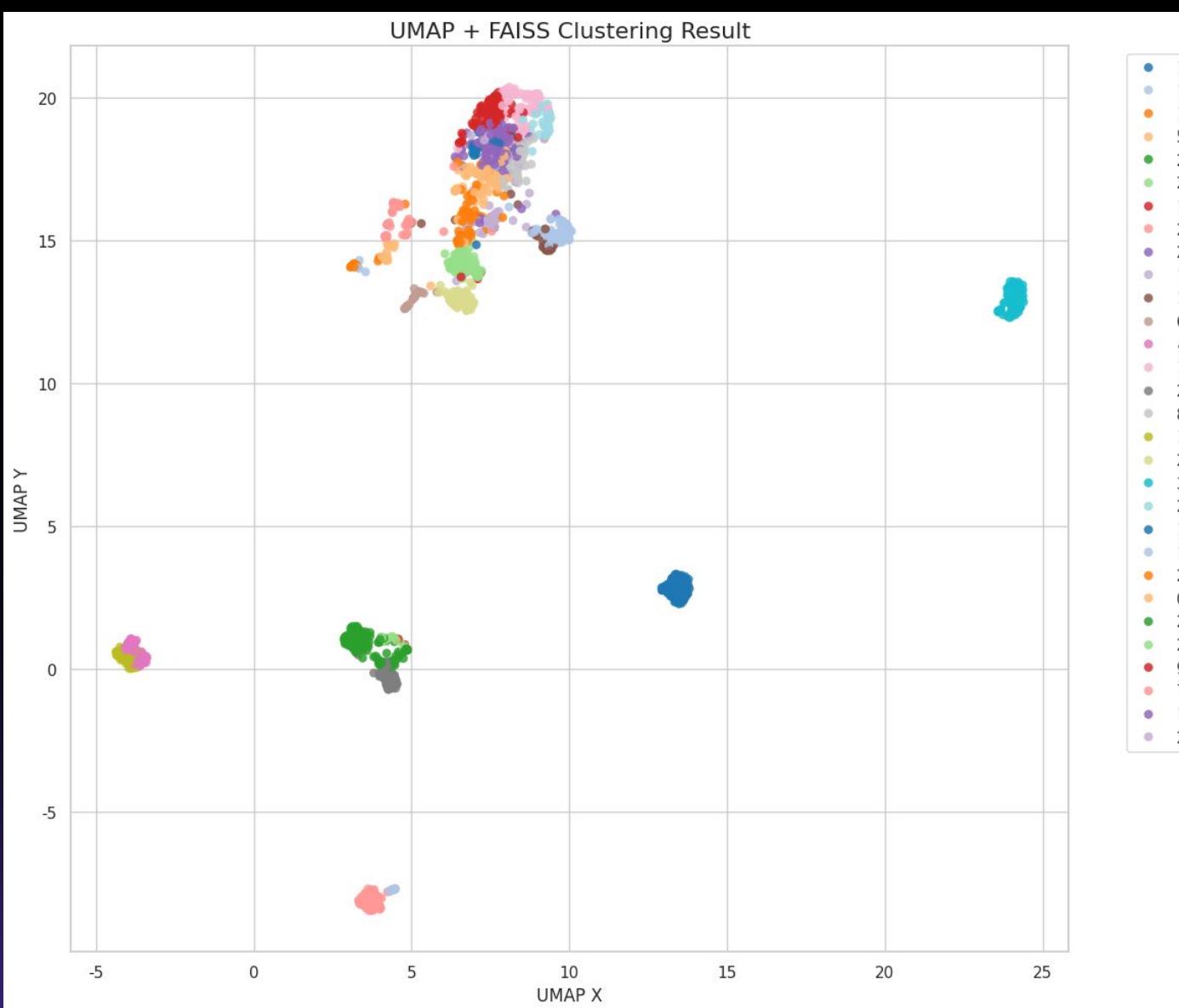
```

번외

: Computer Vision [대회] Classification

- * 잘못 라벨링된 데이터 정정

CLIP 임베딩 추출 → FAISS로 클러스터링(시각화 결과)



분석 결과

뚜렷한 경계의 클러스터가 존재하나

상단부에 혼합된 점들이 분포한 클러스터가 다수 존재
(분류를 혼동하거나 하나의 클래스가 다수의
클러스터에 분포했을 가능성)

fifty one으로 직접 검토 필요

경진대회 수행 결과

번외

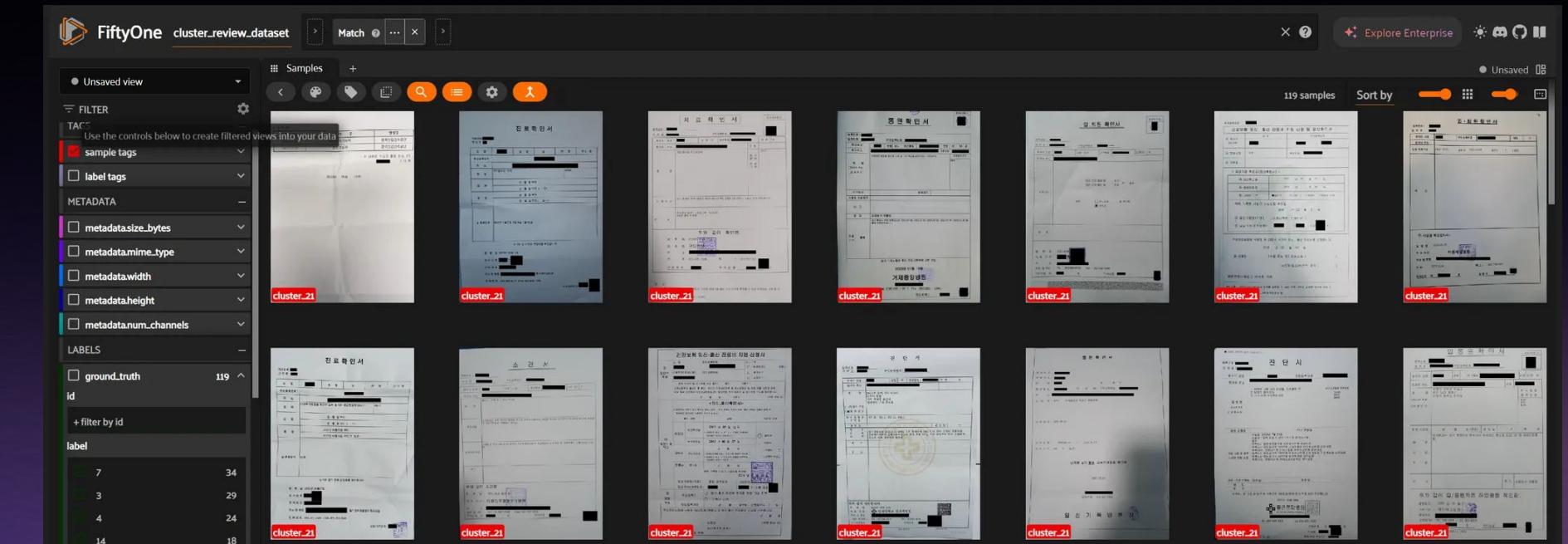
: Computer Vision [대회] Classification

수행 결과

- * 잘못 라벨링된 데이터 정정

혼합도 높은 클러스터 순서대로 잘못 라벨링된 데이터 -> fiftyone으로 해당 클러스터 열기
-> label 별로 선택해서 직접 이미지 내 잘못된 라벨링 데이터 검토 및 변경

cluster	mixture_score	classes	혼합도 구간	우선순위	이유
0	5	0.716216	[1, 2, 4, 6, 7, 10, 12, 13]	고혼합	1 다수 클래스 혼합으로 라벨 오류 가능성 높음
1	21	0.714286	[1, 2, 3, 6, 10, 11, 14]	고혼합	2 다수 클래스 혼합으로 라벨 오류 가능성 높음
2	8	0.709091	[2, 5, 6, 10, 11, 12]	고혼합	3 다수 클래스 혼합으로 라벨 오류 가능성 높음
3	15	0.615385	[1, 3, 6, 10, 11, 13]	고혼합	4 다수 클래스 혼합으로 라벨 오류 가능성 높음
4	12	0.600000	[1, 5, 6, 10, 12, 14]	고혼합	5 다수 클래스 혼합으로 라벨 오류 가능성 높음
5	13	0.457143	[1, 2, 3, 4, 5, 7, 11]	고혼합	6 다수 클래스 혼합으로 라벨 오류 가능성 높음
6	27	0.358974	[5, 8]	고혼합	7 다수 클래스 혼합으로 라벨 오류 가능성 높음
7	24	0.357143	[5, 8]	고혼합	8 다수 클래스 혼합으로 라벨 오류 가능성 높음
8	9	0.333333	[5, 8]	중혼합	9 클래스 소수 혼합, 시각적 유사 클래스 포함
9	1	0.303030	[2, 4, 6]	중혼합	10 클래스 소수 혼합, 시각적 유사 클래스 포함
10	19	0.250000	[5, 10]	중혼합	11 클래스 소수 혼합, 시각적 유사 클래스 포함
11	16	0.240741	[2, 4, 5, 6, 10, 11, 12, 13]	중혼합	12 클래스 소수 혼합, 시각적 유사 클래스 포함
12	26	0.119048	[5, 8]	저혼합	13 단일 클래스 중심으로 혼합도 낮음
13	20	0.114286	[5, 6, 10, 11, 13]	저혼합	14 단일 클래스 중심으로 혼합도 낮음
14	7	0.074074	[1, 2, 3, 4, 5]	저혼합	15 단일 클래스 중심으로 혼합도 낮음



번외

: Computer Vision [대회] Classification

- * 잘못 라벨링된 데이터 정정

결과: 총 9개의 잘못 라벨링된 데이터 발견 및 정정

ID	target_before	before_label	target_after	after_label
22	02ebb92c43006832.jpg	11	약국 영수증	10 진료비 납입확인서
38	0583254a73b48ece.jpg	11	약국 영수증	10 진료비 납입확인서
192	1ec14a14bbe633db.jpg	14	소견서	7 진료확인서
340	38d1796b6ad99ddd.jpg	11	약국 영수증	10 진료비 납입확인서
428	45f0d2dfc7e47c03.jpg	3	입퇴원확인서	7 진료확인서
723	7100c5c67aecadc5.jpg	3	입퇴원확인서	7 진료확인서
862	8646f2c3280a4f49.jpg	7	진료확인서	3 입퇴원확인서
1095	aec62dc6d7af97cd.jpg	3	입퇴원확인서	14 소견서
1237	c5182ab809478f12.jpg	4	진단서	14 소견서

번외

: Computer Vision [대회] Classification

* 프로젝트 진행 과정

◆ [학습 중심 단계: CLIP 중심 파이프라인]		
단계	구성 요소	세부 설명
0	데이터 로딩	정제된 <code>train.csv</code> 불러오기 (ID, target 기반)
1	클래스 불균형 보정	소수 클래스 oversampling, SMOTE 등 적용
2	Augmentation 적용	rotation, blur, noise, distortion 등 오염 시뮬레이션
3	CLIP Processor 전처리	<code>CLIPProcessor(images=..., return_tensors="pt")</code>
4	Dataset & DataLoader 구성	커스텀 Dataset + <code>pixel_values</code> , <code>label</code> 반환
5	모델 구성	<code>CLIPModel.vision_model</code> + Classifier Head

6	손실 함수 설정	SCE, Label Smoothing 등 noise-robust loss
7	학습 루프	forward → loss → backward
8	EarlyStopping / Curriculum	epoch 증가에 따라 augmentation 강도 조절
9	Test 데이터 추론	동일한 CLIPProcessor 적용 → 예측 수행
10	결과 저장	<code>ID</code> , <code>target</code> 형식 CSV 저장

번외

: Computer Vision [대회] Classification

- * 프로젝트 진행 과정

결과: 실제 결과는 실패한 모델

confusion matrix에 예측 결과가 0인 클래스 다수 존재

precision/recall/f1-score의 저성능으로 유의미한 결과 도출 불가

03

분석 인사이트 및 결과

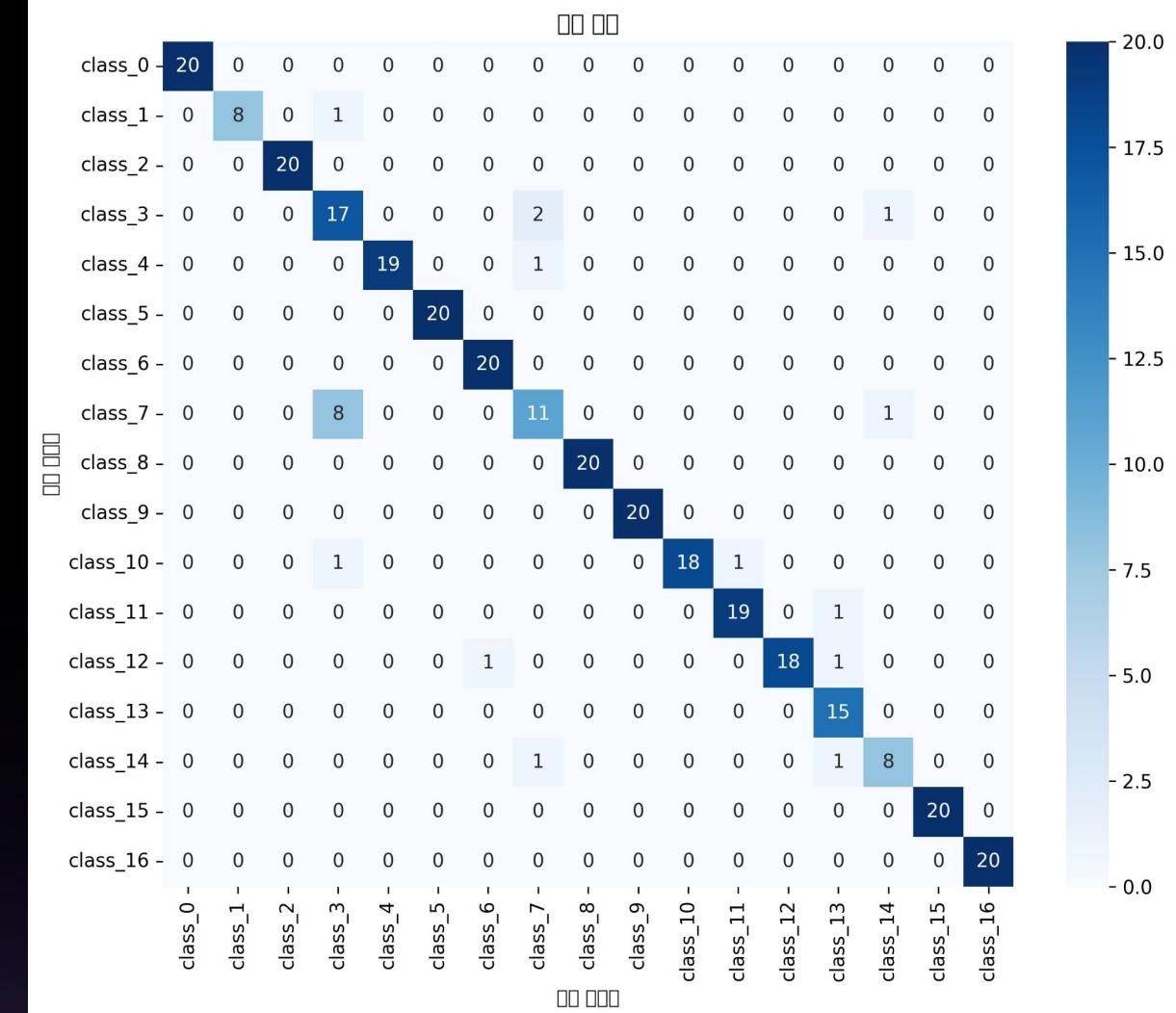
문제 및 인사이트 도출
해결 방법 및 결과

경진대회 인사이트 공유

: Computer Vision [대회] Classification

01. 문제 발생 배경 및 원인 분석

3,7 번 이미지들이 비슷하게 생김 + 잘못된 라벨링 존재



경진대회 인사이트 공유

: Computer Vision [대회] Classification

03. 해결방법

1. 오버샘플링
2. loss 가중치
3. 7번 라벨 임의 17번 라벨 새로
분류

04. 결과

1 오버 샘플링이 가장 유효

경진대회 수행 결과

: Computer Vision [대회] Classification

* 클래스 불균형 & 어려운 클래스 해결 : 오버샘플링 f1 score + 0.1

```
oversample_labels = [3,4,7,14]
mask = train_data['target'].isin(oversample_labels)
df_dup1 = pd.concat([train_data[mask]] * 7, ignore_index=True)
df_dup2 = pd.concat([train_data[~mask]] * 6, ignore_index=True)

df_aug = pd.concat([df_dup1, df_dup2], ignore_index=True)
train_data = df_aug.sample(frac=1, random_state=config.get('seed', 42)).reset_index(drop=True)
```

경진대회 수행 결과

: Computer Vision [대회] Classification

* 클래스 불균형 & 어려운 클래스 해결 : Focal Loss f1score -0.01

```
_target_: src.losses.focal.FocalLoss
alpha: 1.0
gamma: 2.0
class_weights: [1.0, 1.0, 1.0, 2.5, 2.5, 1.0, 1.0, 2.5, 1.0,
1.0, 1.0, 1.0, 1.0, 1.0, 2.5, 1.0, 1.0]

ce_loss = F.cross_entropy(inputs, targets, reduction='none',
weight=class_weights)
pt = torch.exp(-ce_loss)
focal_loss = alpha * (1 - pt) ** gamma * ce_loss

if reduction == 'mean':
    return focal_loss.mean()
elif reduction == 'sum':
    return focal_loss.sum()
```

경진대회 수행 결과

: Computer Vision [대회] Classification

* 클래스 불균형 & 어려운 클래스 해결 : 7 label 임의 분류 새로운 label 생성 f1 score (시간 부족)

통원 확인서

병록번호	[REDACTED]	주민등록번호	[REDACTED]		
환자성명	[REDACTED]	생년월일	[REDACTED]	관련/설명	13/F
환자주소	상세불명의 혁주측만증, 상세불명의 부위 전화: [REDACTED]				
병 명					
내원일	20230629				
통원기간					

상기란자는 : 2023-06-29 | 부터 : 2023-06-29 | 까지 통원하였음을 확인합니다. (1 일간)

발 행 일 : 2023년 06월 29일
병원명 : 인서프라임병원
병원 주 소 : [REDACTED]
면허 번 호 : 제 [REDACTED] 호
의사 성 명 : [REDACTED]

진료 확인서

차트번호	[REDACTED]	연번호	[REDACTED]		
성 명	[REDACTED]	성 별	남	연령	만 35 세
주민등록번호					
주 소					
병 명	신경후리병증을 동반한 요추 및 기타 주간관찰(G55.1*) M511				
입 원	년 월 일부터				
	년 월 일까지 (간)				
통 원	2023년 06월 15일부터				
	2023년 06월 15일까지 (1 일간)				
실 통원일자	15일				

상기와 같이 진료 받았음을 확인합니다.

발 행 일 2023년 06월 27일
의사 성 명 [REDACTED]
면허 번 호 [REDACTED]
주소 및 영장 [REDACTED] 활기찬마취통증의학과의원
전화 번 호 055-313-2500 (FAX) 055-313-2501
의료기관의장 [REDACTED]

04

회고

우리 팀의 목표 달성도
느낀점 및 향후 계획

경진대회 회고

: Computer Vision [대회] Classification

Point 1

우리 팀의 처음 목표에서 어디까지 도달했는가

- * 전팀원 전부 고득점 -> 2명 고득점 달성

Point 2

우리 팀이 잘했던 점

- * 각자 작업한 연구에 대한 빠른 공유 및 코드 리뷰

Point 3

협업하면서 아쉬웠던 점

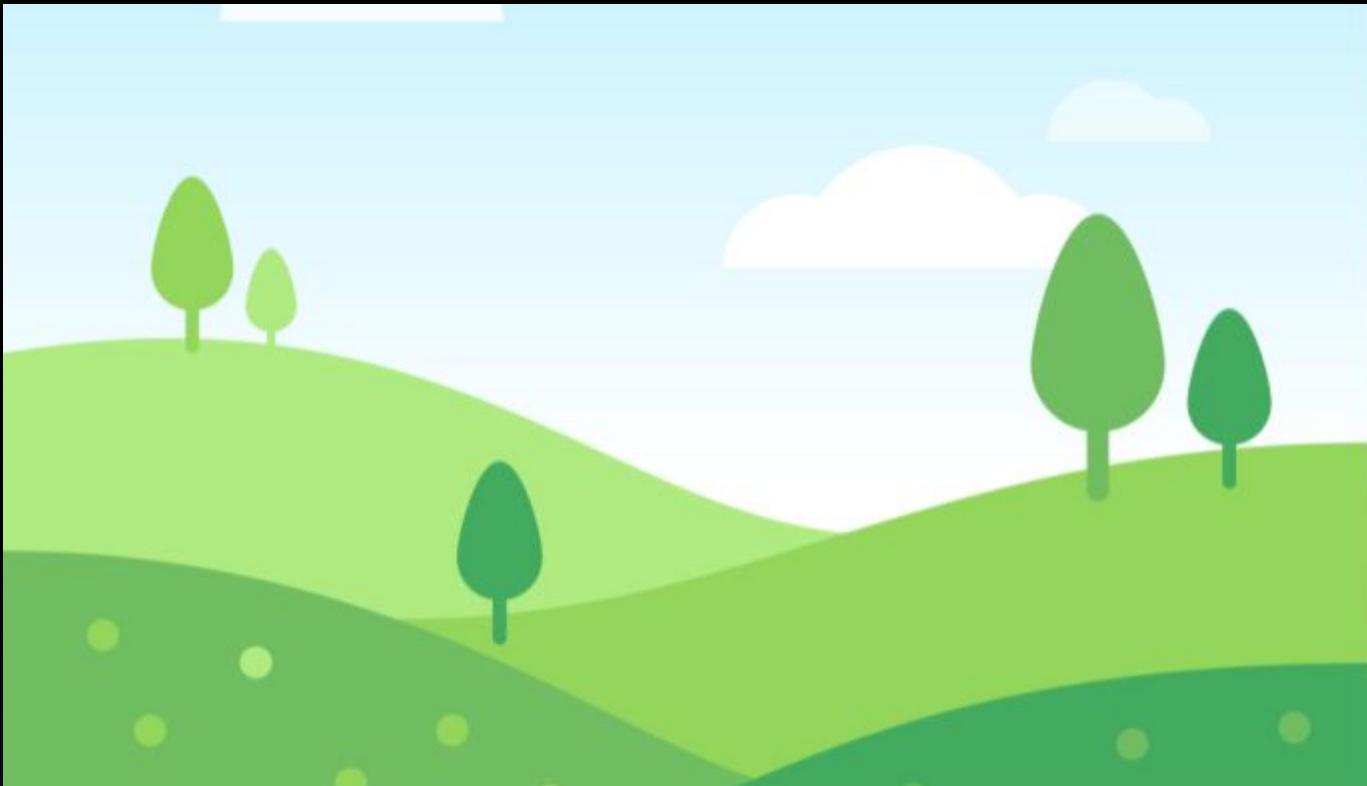
- * 공유한 결과를 개인 프로젝트에 실현 및 적용하기 여려웠음.

첫 번째 경진대회 진행 소감 및 앞으로의 계획에 대해 정리해 작성해주세요.

경진대회 진행 소감

: Computer Vision [대회] Classification

그룹 사진을 첨부해주세요.



빠른 성과를 위해 회전·반전·가우시안 노이즈·믹스업 등 오프라인 증강을 활용해 다양한 모델과 이미지 크기를 중심으로 실험을 진행했으나, 반복적인 증강 처리로 처리 속도에 한계가 있었고 데이터의 기본적 특성을 충분히 검토하지 못한 점이 아쉬웠다. 앞으로는 단순 증강과 모델 튜닝에 앞서 데이터 분포, 클래스별 샘플 특성, 노이즈·결측치 패턴 등을 심층적으로 분석하여, 그 결과에 기반한 맞춤형 전처리·증강 전략을 수립함으로써 모델의 성능과 학습 효율을 동시에 개선하겠다.

* 이승민
인공지능 모델 개발 할 때 프로젝트 설정 하는 능력이 매우 중요하며 재대로 하지 않을 경우 체계적으로 연구를 진행하기가 어려운 것 같습니다.

* 최웅비
파이토치와 비정형 데이터 분석이 처음이라 어려움이 많았지만, 팀원이 미리 작성해둔 코드와 다른 조원들의 GitHub 코드를 참고하며 개선해 나갈 수 있었고, 그 결과 좋은 점수를 받을 수 있었던 것 같습니다.

* 이경도
처음 모델 설계 및 방향 설정 과정이 부족해서 생각보다 유의미한 결과를 얻지 못했던 것 같다. 막상 코드 구현보다는 실제 모델 설계 (모델 선택, 앙상블 등)와 전처리 과정(데이터 증강)이 굉장히 중요한데 이 부분을 간과함. 또한 코드 및 하이퍼파라미터 버전 관리를 적극적으로 사용하지 않아 프로젝트 진행에 어려움을 겪음. 이 부분도 보완해야 할 필요성을 느낀.

* 김재덕
강의를 통해 먼저 학습을 할 때는 여러기법들을 적용해봐야지라는 생각을 했지만, 막상 대회를 진행하니 적용해보니 못한 것이 많았습니다.
또 데이터 증강을 제외하고는 다른 부분에는 소홀히 하였기에, 그 부분 추후 추가적인 작업을 할 예정입니다.

Life-Changing Education

감사합니다.
