

코스포 x 데이콘 자동차 충돌 분석 AI경진대회

안익균, 좌대현

Final Result : Leaderboard

코스포 x 데이콘 자동차 충돌 분석 AI경진대회 채용

채용 | 알고리즘 | 동영상 | 비전 | 분류 | Macro F1 Score

₩ 상금 : 채용

🕒 2023.02.06 ~ 2023.03.13 09:59

+ Google Calendar

👤 965명 📅 마감



참여중

1	miinngdok		0.79169	71	일 년 전
2	yuseok		0.77369	27	일 년 전
3	베이스라인		0.7704	99	일 년 전
4	설빙 더 아이스		0.75015	84	일 년 전
5	tjr		0.74428	1	일 년 전
6	Redix6		0.74246	46	일 년 전
7	jsh0551		0.72929	64	일 년 전
8	잇준		0.72535	29	2일 전
9	망기도한		0.72054	25	일 년 전
10	재애애영		0.71505	16	일 년 전
11	중요한건찍이지않는마음		0.70453	42	일 년 전

About Task : Label information

블랙박스 영상을 보고 충돌 여부, 날씨, 시간을 분류하는 video classification

crash	ego-involve	weather	timing	label
No	-	-	-	0
Yes	Yes	Normal	Day	1
Yes	Yes	Normal	Night	2
Yes	Yes	Snowy	Day	3
Yes	Yes	Snowy	Night	4
Yes	Yes	Rainy	Day	5
Yes	Yes	Rainy	Night	6
Yes	No	Normal	Day	7
Yes	No	Normal	Night	8
Yes	No	Snowy	Day	9
Yes	No	Snowy	Night	10
Yes	No	Rainy	Day	11
Yes	No	Rainy	Night	12

- **Crash** : 차량 충돌 여부 (No/Yes)
- **Ego-involve** : 본인 차량과 충돌 여부 (No/Yes)
- **Weather** : 날씨 상황 (Normal/Snowy/Rainy)
- **Timing** : 낮과 밤 (Day/Night)

- 차량 블랙박스 영상 데이터 5초
- 데이터는 초당 10프레임으로 총 50프레임
- 불균형한 레이블 - 특정 레이블에 대한 저조한 학습
 - Label 6 : 사고 + 본인 차량 충돌O + 비 + 밤
 - Label 10 : 사고 + 본인 차량 충돌 X + 눈 + 밤

```
train_df['label'].value_counts()
```

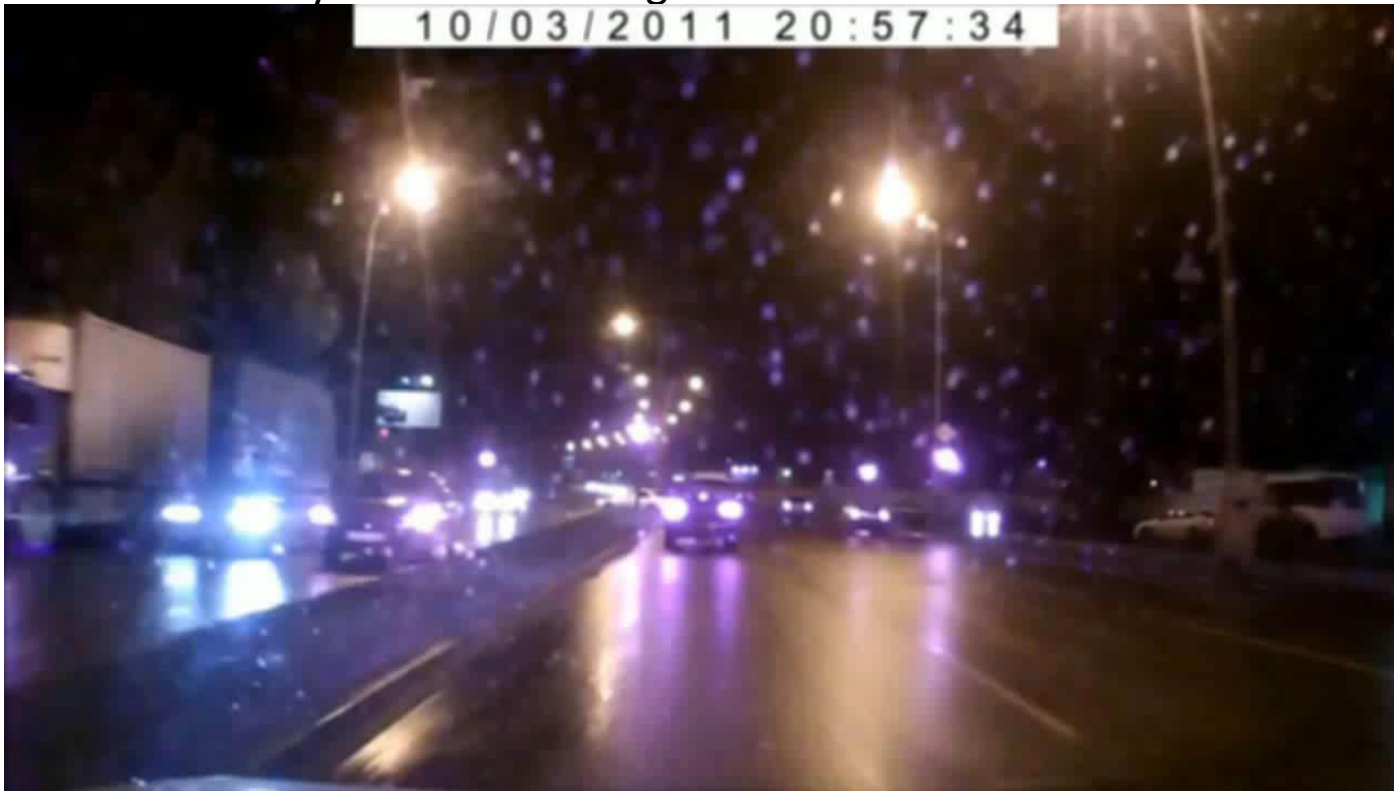
```
label
0      1783
1       318
7       317
3        78
2         51
9         34
11        33
8         30
5         28
4         13
12         6
10         4
6          3
Name: count, dtype: int64
```

Data : Incorrect label

- 잘못 지정된 레이블 - 일반화 능력 저하, 경계 값 분류 오류

→ train 데이터에 대해 직접 확인 후 레이블 다시 지정

ex> Label 5 <Day> --> Label 6 <Night>



- **Macro F1 score**

- 각 클래스마다 F1 score 계산 후 평균
- 모든 클래스 F1에 동일한 가중치 비도가 낮은 데이터 예측 중요

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Models : Used models

Video Model

3D CNN

Base-line

ResNet 3D

Final

Swin3D

Image Model

ResNet

Base-line

EfficientNet

VIT

ConvNeXt

Swin-Transformer

Final

Base Model : 3D CNN

```
class BaseModel(nn.Module):
    def __init__(self, num_classes=13):
        super(BaseModel, self).__init__()
        self.feature_extract = nn.Sequential(
            nn.Conv3d(3, 8, (1, 3, 3)),
            nn.ReLU(),
            nn.BatchNorm3d(8),
            nn.MaxPool3d(2),
            nn.Conv3d(8, 32, (1, 2, 2)),
            nn.ReLU(),
            nn.BatchNorm3d(32),
            nn.MaxPool3d(2),
            nn.Conv3d(32, 64, (1, 2, 2)),
            nn.ReLU(),
            nn.BatchNorm3d(64),
            nn.MaxPool3d(2),
            nn.Conv3d(64, 128, (1, 2, 2)),
            nn.ReLU(),
            nn.BatchNorm3d(128),
            nn.MaxPool3d((3, 7, 7)),
        )
        self.classifier = nn.Linear(1024, num_classes)
```

- Conv3d : 4개
- Linear 1개
- 13개 Label 대해 학습 후 예측
- public 점수: 0.19
- private 점수: 0.21

Model Train : Method

Method 1

crash

ego
Involve

weather

timing

Method 2

crash
+
ego

weather

timing

Method 3

crash
+
ego

weather

Image

timing

Model Train : For 4 Class

Method 1

Test F1: 0.484 / 0.590

crash

☐ Val F1 : 0.99326 ResNet 3D-18

ego
Involve

☐ Val F1 : 0.90359 ResNet 3D-18

☐ Val F1 : 0.89623 Swin3D

weather

☐ Val F1 : 0.88831 ResNet 3D-18

☐ Val F1 : 0.74585 Swin3D

timing

☐ Val F1 : 0.95678 ResNet 3D-18

☐ Val F1 : 0.90359 Swin3D

Model Train : For 3 Class

Method 2

Test F1: 0.559 / 0.628

crash
+
ego

☐ Val F1 : 0.96029 ResNet 3D-18

weather

☐ Val F1 : 0.88831 ResNet 3D-18

timing

☐ Val F1 : 0.95678 ResNet 3D-18

Model Train : Image Classification to weather

Method 3

Test F1: 0.725/0.607

crash
+
ego

☐ Val F1 : 0.96029 ResNet 3D-18

weather

Image

- ☐ Val F1 : 0.87193 ResNet50
- ☐ Val F1 : 0.87521 EfficientNet
- ☐ Val F1 : 0.89163 ConvNeXt
- ☒ Val F1 : 0.92509 Swin-Transformer
- ☐ Val F1 : 0.88282 ViT

timing

☐ Val F1 : 0.95678 ResNet 3D-18

이미지 분류 Why?

- Weather는 맥락의 중요성 낮음
- 많은 Train data 활용 가능
- Frame 0, 25, 49 사용

Frame 0



Frame 25



Frame 49



Data augmentation : transforms

Original



Horizontal Flip



Rotate 25 degrees



Color Jitter



Random Resized Crop



- **ColorJitter**

- brightness : 밝기
- contrast : 대비
- saturation : 채도
- hue : 톤
조절

Loss function with class weight

- Rainy, Snow 에 대해 가중치 부여

```
# 가중치 계산
class_counts = [2400, 800, 660]
class_weights = [1.0 / count for count in class_counts]
class_weights = torch.tensor(class_weights, dtype=torch.float).to(device)

def train(model, optimizer, train_loader, val_loader, scheduler, device):
    model.to(device)

#가중치 추가
criterion = nn.CrossEntropyLoss(weight=class_weights).to(device)
```

3 Frame Inference

- Frame 0, 25, 49 Inference and Hard-voting
- Robust performance



Weather : Ablation study

ResNet 50	Data augmentation	Weight	Val F1	Test F1
1st	X	x	0.82641	-
2nd	Horizontal Flip, Rotation, ColorJitter, Crop, Random Erase	x	0.87193	0.467 / 0.575
4th	2nd + Vertical Flip - ColorJitter	x	0.82904	-
6th	2nd - Random Erase	o	0.82786	0.518 / 0.734
7th	2nd	o	0.86038	0.618 / 0.723

- augmentation 적용 시 성능이 높아짐
- Vertical Flip은 성능 하락 , ColorJitter는 성능 상승 예상
- Random Erase -> Robust Model

New datasets

- 현 데이터와 유사한 날씨 Dataset 부재

Semi-supervised learning

- label 0은 crash에 대한 정보만 존재

crash	ego-involve	weather	timing	label
No	-	-	-	0

- 적용 시점 기준 Best Weather Model 사용
- Only use threshold over 0.8

Data addition Results

- Snow에 대해서 잘 예측
- Rainy에 대해서는 밤인 경우가 많았음

Discussion

- 눈이 쌓인 구별가능한 특성 때문에 Snow 잘 예측
- 사용한 모델 또한 Rainy에 대해 좋은 특징 추출 실패
- 젖은 땅을 통해 모델이 비를 예측 → 밤에 어두운 땅을 비로 혼동

- **CFG**

- **crash+ego**

- `Img_size: 128, batch_size: 4, lr: 1e-5`
 - `augmentation : Horizontal Flip, Rotate, Brightness, Gauss Noise`

- **weather**

- `Img_size: 224, batch_size: 32, lr: 1e-5`
 - `augmentation : Horizontal Flip, Rotate, ColorJitter, Crop, RandomErase`
 - `Label weight`
 - `Semi-supervised Learning`

- **timing**

- `Img_size: 128, batch_size: 16, lr: 1e-5`
 - `augmentation : Horizontal Flip, Rotate, Brightness, Gauss Noise`
 - `k-fold`

- Public 1st Model

- Model : ResNet 3D-18 + Swin-base
- Public : 0.725
- Private : 0.607

- Private 1st Model

- Model : ResNet 3D-18 + ResNet50
- Public : 0.618
- Private : 0.723

- Robust Model

- Model : ResNet 3D-18 + Swin-base
- Public : 0.687
- Private : 0.688

Score stagnation

- 각 레이블에 대한 결과를 합치는 과정에서 발생 예상

Prediction	Crash+ego	weather	timing	Prediction
1	O	O	O	correct
2	X	X	X	incorrect
3	O	O	O	correct

Prediction	Crash+ego	weather	timing	Prediction
1	O	O	X	incorrect
2	O	X	O	incorrect
3	X	O	O	incorrect

Various Models Ensemble

- Used Model : ResNet3D, ViT, ResNet50, Swin-base, ConvNeXt
 - Public : 0.604
 - Private : 0.663
- Why? -> Low original performances

Discussion : Limitations

- 원본 데이터의 낮은 품질 + 어려운 특징 (Rainy)
- 추가 데이터셋 부재
- 생성형 모델 전용 실패

Fake Images



- 각 Label에 대한 test score 알 수 없음
- semi-supervised learning 불완전한 적용

Thank you!