과적차량 및 정상차량 분류를 위한 YOLOv5 기반 프로젝트 기술 보고서

목차

서론

본 보고서에서는 YOLOv5를 활용하여 과적차량과 정상차량을 분류하는 프로젝트에 대해 기술합니다.

교통 당국이 과적차량에 대한 적절한 제재를 취할 수 있도록 도움을 주는 이 프로젝트는 과적차량이 도로 및 교량 구조물에 초래할 수 있는 피해를 예방하는데 큰 도움이 됩니다.

YOLOv5 소개 및 주요 특징

YOLOv5(You Only Look Once version 5)는 객체 탐지를 위한 신속한 딥러닝 기반 알고리즘입니다. YOLOv5는 다음과 같은 주요 특징을 가지고 있습니다.

- 빠른 추론 속도: YOLOv5는 실시간 객체 탐지에 적합한 빠른 추론 속도를 제공합니다.
- 높은 정확도: YOLOv5는 기존 YOLO 버전들보다 높은 정확도를 가지고 있어, 객체 탐지 분야에서 널리 사용되고 있습니다.
- 사용자 친화적 구조: YOLOv5는 PyTorch 기반으로 개발되어 있어, 사용자가 쉽게 수정하고 확장할 수 있습니다.

주어진 데이터셋 구조

본 프로젝트에서 사용하는 데이터셋은 다양한 환경에서 촬영된 과적차량 및 정상차량 이미지와 이미지에 대한 라벨(json)로 구성됩니다. 주어진 데이터셋의 구조는 다음과 같습니다.

주어진 데이터셋 구조

```
New_Sample
├── 원천 데이터
│   └── TS1.대형차
│       ├── 불법 차량
│       │   └── *.jpg
│       └── 정상 차량
│           └── *.jpg
└── 라벨링 데이터
    └── TS1.대형차
        ├── 불법 차량
        │   └── *.json
        └── 정상 차량
            └── *.json
```

Ultralytics에서 요구하는 데이터셋 포멧 구조

```
New_Sample
├── image
│   ├── train
│   │   └── *.jpg
│   └── val
│       └── *.jpg
└── labels
    ├── train
    │   └── *.txt
    └── val
        └── *.txt
```

주어진 데이터셋 label Annotation


```
"FILE":[{
        "FILE_NAME":"A01_B01_C04_D01_0703_E06_F07_1_1.jpg",
        "COLLECTIONMETHOD":"CCTV",
        "DAY/NIGHT":"주",
        "LANE":"다선",
        "ROADNUMBER":"아트센터대로",
        "PLACE":"아트센터교2(송도동73)",
        "IDCODE":"F07",
        "DATE":"2021.07.03",
        "WEATHER":"맑은날",
        "RESOLUTION":"1920*1080",
        "MAKE":"한화테크윈",
        "MODELNAME":"XNO-6020R",
        "FILESIZE":"333392",
        "BOUNDINGCOUNT":"1",
        "ITEMS":[
                {
                "DRAWING":"Box",
                "SEGMENT":"대형차",
                "BOX":"722.02,479.77,456.74,307.75",
                "POLYGON":"",
                "PACKAGE":"불법차량",
                "CLASS":"적재불량",
                "COVER":"덮개개방",
                "COURSE":"후면우측",
                "CURVE":"정상주행"
                }
```


과적차량 Annotation json format

모델 구성 및 훈련

본 프로젝트에서 사용하는 YOLOv5 모델은 기본 구조를 유지하면서 과적차량과 정상차량 분류를 수행할 수 있도록 수정되었습니다. 모델 훈련 과정은 다음과 같습니다.

- data.yaml설정: 올바른 데이터인식과 학습을 위해 yaml파일을 생성합니다
- 라벨데이터 json을 txt로 변환: 학습을 위해 주어진 json 라벨데이터를 txt로 변환합니다.
- 학습 파라미터 설정: 학습률, 배치 크기, 에포크 수 등의 학습 파라미터를 설정합니다.
- 모델 훈련: 훈련 데이터를 사용하여 모델을 훈련시키며, 검증 데이터를 사용하여 모델의 일반화 성능을 평가합니다.
    - 본 프로젝트에선 360개의 이미지 데이터와 yolov5s, 50에포크로 학습되었습니다.

모델 구성 및 훈련

```
# X1, Y1, X2, Y2 (라벨링구분이 Box인 경우 필수)
box = item['BOX'].split(',')
x1, y1, x2, y2 = float(box[0]), float(box[1]), float(box[2]), float(box[3])

# 바운딩 박스 좌표값을 이용하여 YOLO 형식의 라벨 정보 생성 x
dw = 1. / img_width
dh = 1. / img_height

x = x1 + x2 / 2.0
y = y1 + y2 / 2.0
w = x2
h = y2
```

```
# %cat /content/dataset/data.yaml
%cat /content/drive/MyDrive/DataSample/overloaded/data.yaml

names:
- 불법차량
- 정상차량
nc: 2
train: /content/drive/MyDrive/DataSample/overloaded/train.txt
val: /content/drive/MyDrive/DataSample/overloaded/val.txt
```

	from	n	params	module	arguments
0	-1	1	3520	models.common.Conv	[3, 32, 6, 2, 2]
1	-1	1	18560	models.common.Conv	[32, 64, 3, 2]
2	-1	1	18816	models.common.C3	[64, 64, 1]
3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	2	115712	models.common.C3	[128, 128, 2]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	3	625152	models.common.C3	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	1182720	models.common.C3	[512, 512, 1]
9	-1	1	656896	models.common.SPPF	[512, 512, 5]
10	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	361984	models.common.C3	[512, 256, 1, False]
14	-1	1	33024	models.common.Conv	[256, 128, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	90880	models.common.C3	[256, 128, 1, False]
18	-1	1	147712	models.common.Conv	[128, 128, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20	-1	1	296448	models.common.C3	[256, 256, 1, False]
21	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	1182720	models.common.C3	[512, 512, 1, False]
24	[17, 20, 23]	1	18879	models.yolo.Detect	[2, [[10, 13, 16, 30, 33, 23],

YOLOv5s summary: 214 layers, 7025023 parameters, 7025023 gradients, 16.0 GFLOPs

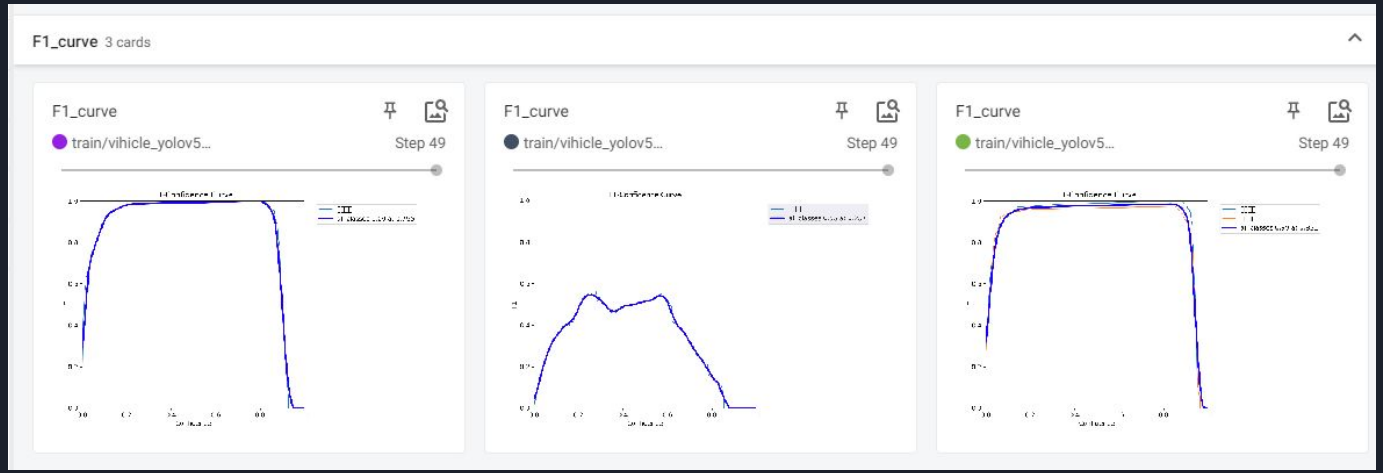
```
%cd /content/yolov5/

!python train.py --img 640 --batch 16 --epochs 50 --data /content/drive/MyDrive/DataSample/overloaded/data.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt --name vihicle_yolov5s_results
```

성능 평가 및 결과

성능 평가 및 결과 본 프로젝트에서는 다음과 같은 평가 지표를 사용하여 모델의 성능을 평가합니다.

- 정밀도(Precision): 과적차량으로 예측한 결과 중 실제로 과적차량인 비율
- 재현율(Recall): 실제 과적차량 중 모델이 과적차량으로 예측한 비율
- F1 점수(F1 Score): 정밀도와 재현율의 조화 평균으로 계산되는 성능 지표



성능 평가 및 결과

- 약 40분간의 학습을 걸쳐 모델을 완성하였으며 대형차량에 한해서 정확히 인식할 수 있다.
- 깃을 통하여 test_program.ipynb를 배포하여 간단한 모델 테스트를 진행할 수 있다.
- 우수한 결과를 보이진 않으나 어느정도 불법차량과 정상차량을 분간한다.

결론 및 향후 계획

- 코끼리와 같이 사이즈가 과적차량과 비슷할 경우 잘 못 인식하는 경우가 있다.
- 몇몇 소형차량의 경우 불법차량으로 인식한다.
- 이러한 모델의 부정확성은 학습된 모델의 데이터량이 적기때문이라고 판단된다.
- 더 나은 모델을 얻기위해, 소형차, 중형차에 대한 과적차량과 정상차량데이터 학습시킬 계획이다.