YOLOv5를 활용한 운전자 부주의 행동 검출 개체 인식 프로젝트

TAVE 8기 딥러닝 프로젝트 1팀
박지애 이민재 이승은 이은호

1. 프로젝트 진행 배경 및 목적

- Computer Vision 분야에 관심이 있는 팀원들

  : Image Classification 〉〉〉 Object Detection

- AI 놀이터: 실전 AI 도전 !!!

  “ 운전자 부주의 행동 검출 객체 인식 ”

2. 프로젝트 진행 과정

1) 데이터 수집

〉〉〉 https://ai-korea.kr/playground/selectActualPlaygroundTask.do

Train: 운전자 얼굴 이미지(.jpg) 파일 & bounding box 좌표(.json) 파일

각 16,410개

class 종류: face, eye_opened, eye_closed, mouth_opened, mouth_closed, phone, cigar

Test: 운전자 얼굴 이미지(.jpg) 파일 3,598개

2. 프로젝트 진행 과정

2) 데이터 전처리

운전자 얼굴 이미지(.jpg) 파일에 따른 bounding box 좌표 (.json) 파일을

YOLOv5 모델에 학습시킬 수 있도록 .txt 파일로 바꿔주기

: 기존의 COCO style을 YOLO style의 좌표로 변환 (16,410 개의 파일로)

& bounding box 좌표 앞에 label 번호 넣어주기 (0, 1, 2, 3, 4, 5, 6)

Note.

class 종류: face, eye_opened, eye_closed, mouth_opened, mouth_closed, phone, cigar

```
# Yolo bounding box1 저장

names = ['eye_opened', 'eye_closed', 'mouth_opened', 'mouth_closed', 'face', 'phone', 'cigar']
with open(train_path+'/labels.json') as f:
    json_data = json.load(f)
    json_anno = json_data["annotations"]
    json_anno = json_anno[:8000]
    for json_img in tqdm(json_anno):
        img_id = str(json_img['file_name'])
        txt_name = str(img_id.split('.')[0])+'.txt'
        txt_dir = os.path.join(yolo_boundingbox_directory1, txt_name)
        img_dir = train_path + '/images/' + img_id
        f_txt = open(txt_dir, 'w')
        img = cv2.imread(img_dir, cv2.IMREAD_COLOR)
        w = img.shape[1]
        h = img.shape[0]
        classes = []
        for img_obj in json_img['objects']:
            class_id = str(names.index(img_obj['class'])) + " "
            f_txt.write(class_id)
            img_pos = img_obj['position']
            a,b,c,d = img_pos
            img_pos = convert((w,h),(a,c,b,d))
            for i in range(len(img_pos)):
                input = str(img_pos[i])
                f_txt.write(input)
                f_txt.write(' ')
            f_txt.write('₩n')
```

2. 프로젝트 진행 과정

3) 모델 훈련 및 최적화

협업 방식 정하기:

Google Drive 딥러닝 실전 스터디 공유 폴더

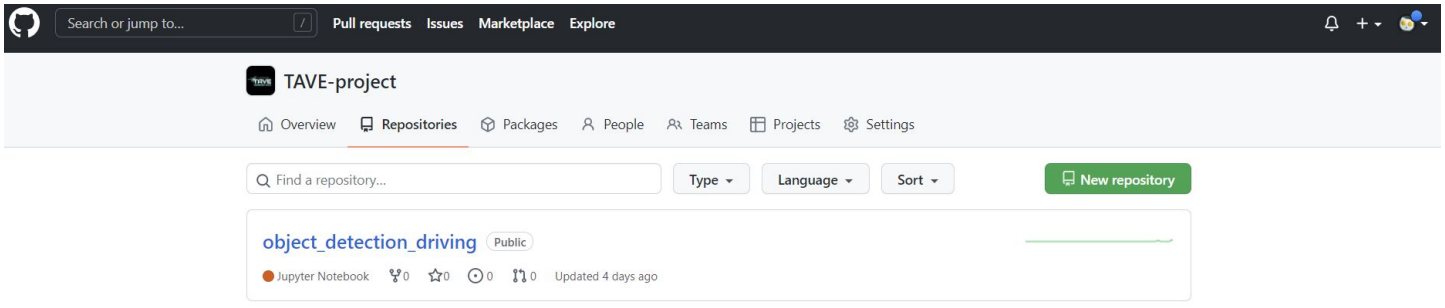
2. 프로젝트 진행 과정

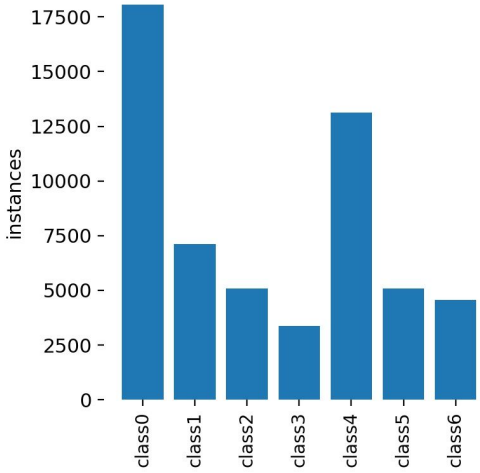
3) 모델 훈련 및 최적화

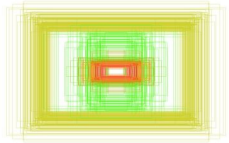
협업 방식 정하기:

github organization 구성, hyperparameter tuning 배분

(batch size: 16, 32, 64 / epoch: 10 - 25: 런타임, gpu_mem 등 상황 고려)




instances
17500
15000
12500
10000
7500
5000
2500
0
class0
class1
class2
class3
class4
class5
class6



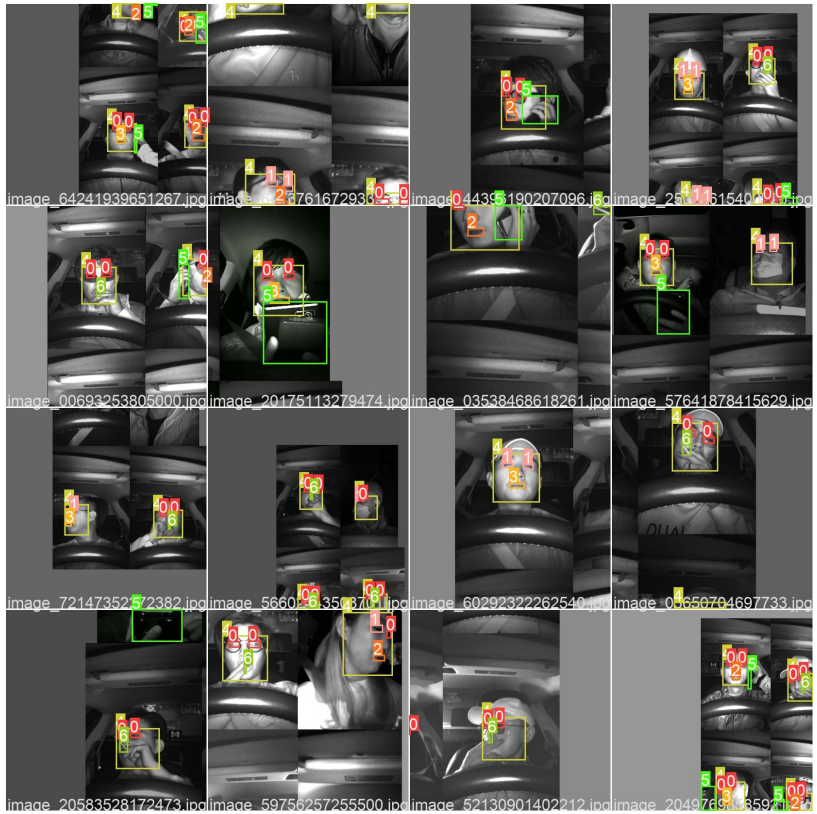
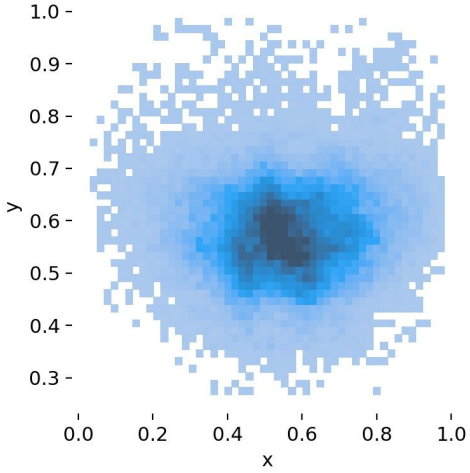
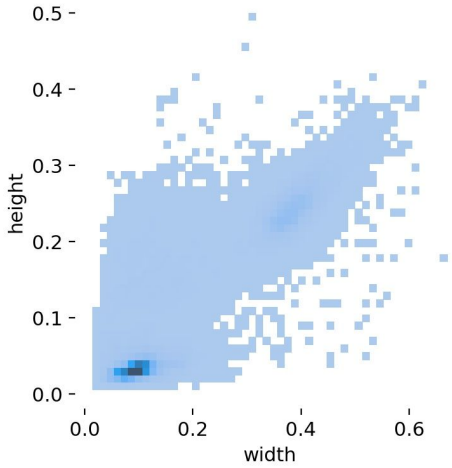


image_64241939651267.jpg
image_0443951902070 96.jpg
image_00693253805000.jpg
image_20175113279474.jpg
image_03538468618261.jpg
image_57641878415629.jpg
image_20583528172473.jpg
image_59756257255500.jpg
image_52130901402212.jpg


y
x
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.0
0.2
0.4
0.6
0.8
1.0


height
width
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.2
0.4
0.6

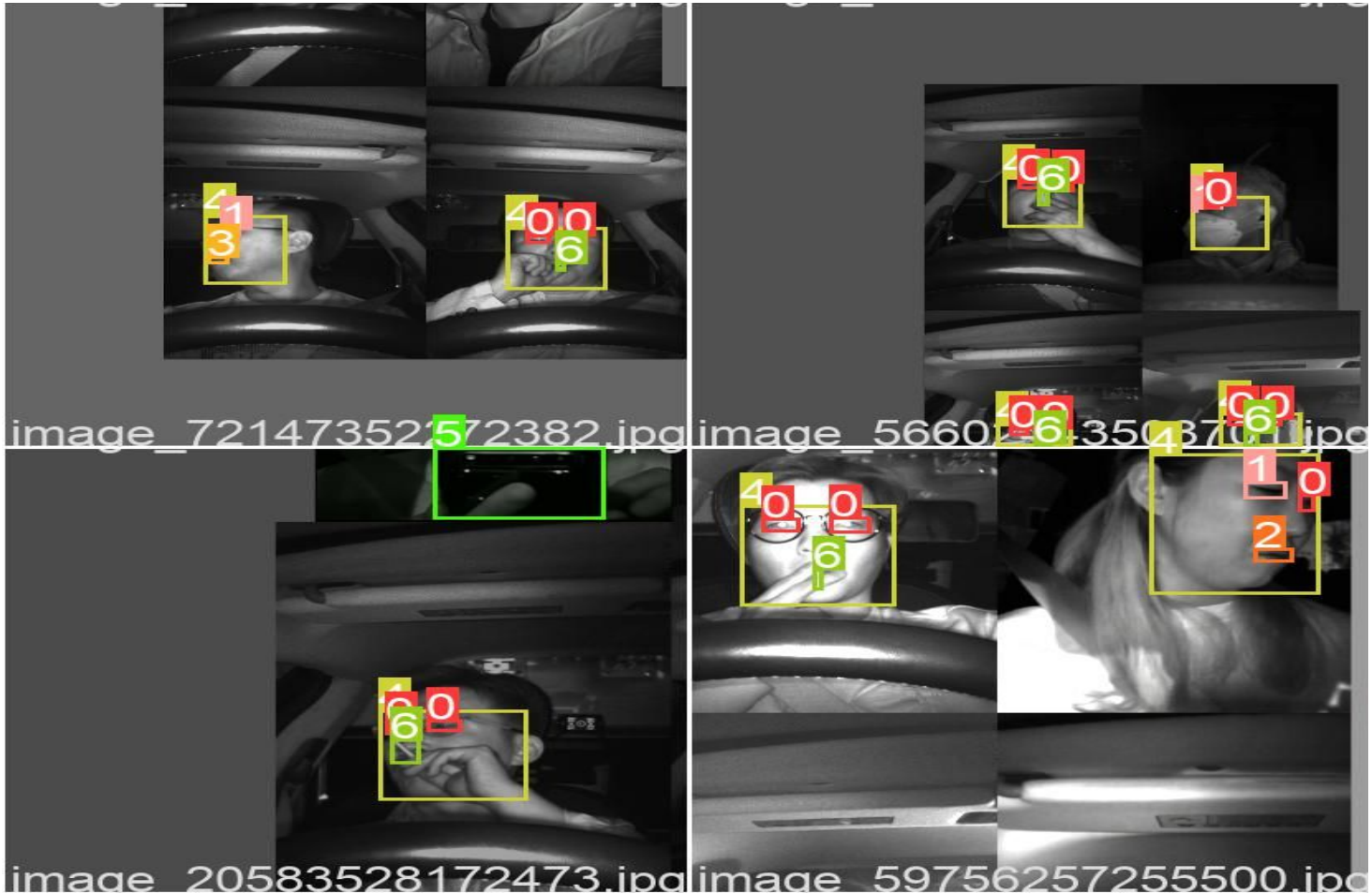


image_721473522572382.jpg
image_5660
image_20583528172473.jpg
image_59756257255500.jpg

2. 프로젝트 진행 과정

3) 모델 훈련 및 최적화

Google Drive

export >>> images

>>> labels (각 16,410개)

& data.yaml 파일을 생성해서 class 종류와 개수(nc)를 표현해주기

& yolov5 >>> models >>> yolov5s.yaml 파일의 nc개수 바꾸기

2. 프로젝트 진행 과정

4) 모델 테스트

Test 이미지 파일 (3,598개)로 테스트 진행

yolov5 > detect > labels

(result_labels.zip, test 이미지의 객체 bounding box를 담은 .txt 파일 3,598개)

```
# 테스트

!python /content/drive/MyDrive/careless_driving/yolov5/detect.py --weights /content/drive/MyDrive/careless_driving/yolov5/runs/train/yolov5s_results6/weights/best.pt --img 416 --conf 0.5 --source "/content/drive/MyDrive/careless_driving/test/images/*.jpg"
```

2. 프로젝트 진행 과정

5) 이론 학습

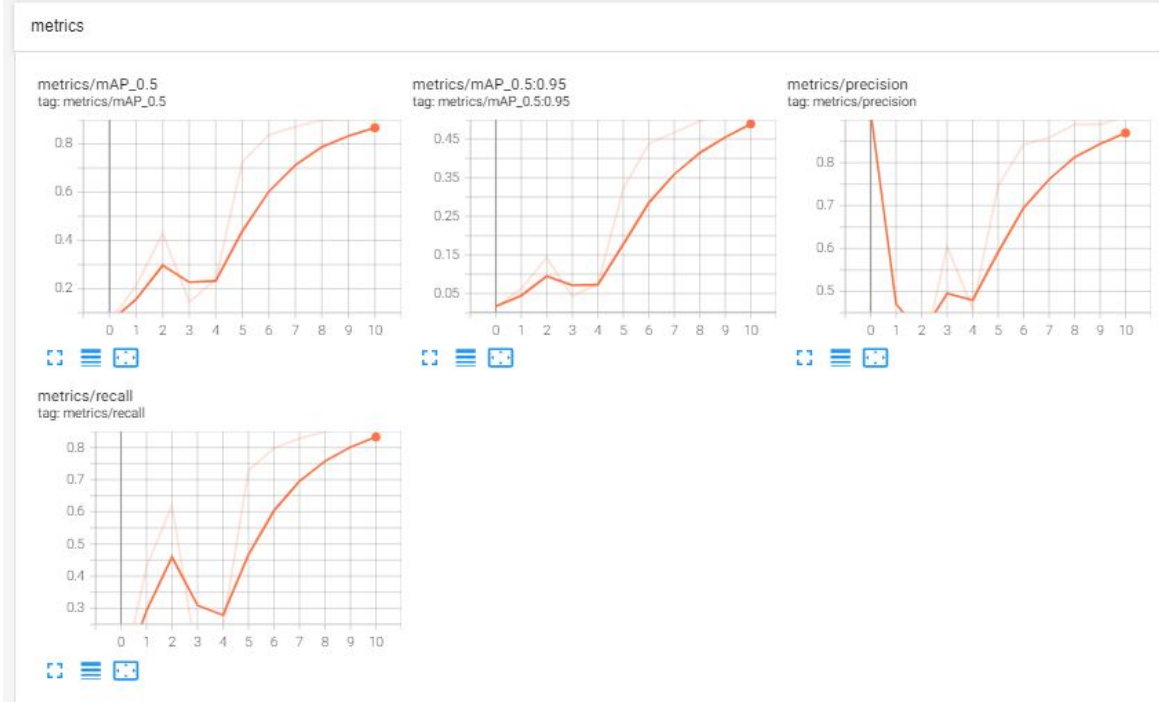
Object Detection 관련 개념 및 모델 발전 순서,

R - CNN 및 YOLO 모델 이론, evaluation 이론 (mAP, PR curve) 등

>>> https://docs.google.com/document/d/1mFQsYLkeFlmUumnwmmaeMXrSqJxLqZ3D/edit

3. 프로젝트 결과

batch_size = 32, epochs = 11



3. 프로젝트 결과

batch_size = 64, epochs = 23

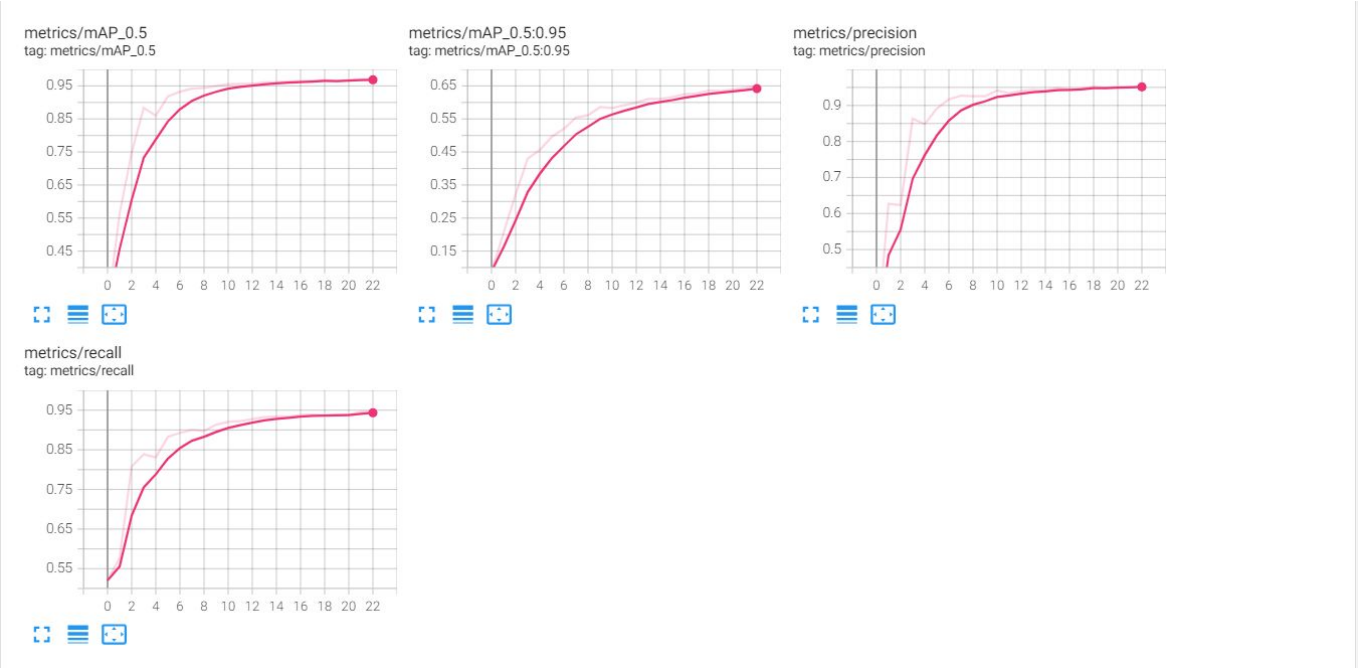
epoch	train/box_loss	train/obj_loss	train/cls_loss	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95	val/box_loss	val/obj_loss	val/cls_loss	x/lr0	x/lr1	x/lr2
0	0.083135	0.032768	0.030698	0.24712	0.52029	0.2789	0.08833	0.072181	0.023133	0.018403	0.00205	0.00205	0.08155
1	0.056991	0.02579	0.015143	0.627	0.57677	0.5634	0.20536	0.048301	0.020664	0.010649	0.0040954	0.0040954	0.062995
2	0.047339	0.023805	0.010374	0.62346	0.80758	0.74797	0.32197	0.047251	0.01918	0.0081277	0.0060828	0.0060828	0.044383
3	0.041255	0.022672	0.0083973	0.86351	0.83856	0.88316	0.43001	0.03708	0.018421	0.0068561	0.0079699	0.0079699	0.02567
4	0.037352	0.021725	0.0074326	0.84811	0.83087	0.86032	0.45606	0.033874	0.018159	0.0065385	0.0094434	0.0094434	0.0094434
5	0.034539	0.021327	0.0067584	0.89184	0.88279	0.91806	0.49616	0.030043	0.016881	0.0056187	0.0094434	0.0094434	0.0094434
6	0.032458	0.020611	0.0061861	0.917	0.89271	0.9326	0.52012	0.030503	0.016659	0.0052803	0.0091406	0.0091406	0.0091406
7	0.031	0.020353	0.0057831	0.92711	0.89981	0.94182	0.55446	0.028153	0.016289	0.00488	0.0087804	0.0087804	0.0087804
8	0.030334	0.020071	0.0055718	0.92535	0.89738	0.94439	0.56081	0.027669	0.016154	0.0048684	0.0083684	0.0083684	0.0083684
9	0.029193	0.019659	0.0053319	0.92549	0.91325	0.94923	0.58584	0.025911	0.015751	0.0044799	0.0079112	0.0079112	0.0079112
10	0.028725	0.019425	0.0051625	0.94149	0.92067	0.9551	0.58347	0.026437	0.01558	0.0043253	0.007416	0.007416	0.007416
11	0.028019	0.019125	0.0050483	0.93381	0.92237	0.95613	0.5916	0.025173	0.015281	0.0042255	0.0068906	0.0068906	0.0068906
12	0.027501	0.018818	0.0048461	0.94057	0.92726	0.95645	0.5991	0.025102	0.015237	0.0041388	0.0063432	0.0063432	0.0063432
13	0.026942	0.018899	0.0047209	0.94328	0.93251	0.96051	0.61065	0.02462	0.015069	0.0039864	0.0057826	0.0057826	0.0057826
14	0.026563	0.018468	0.004584	0.94102	0.93444	0.96185	0.6102	0.024387	0.014974	0.0039433	0.0052174	0.0052174	0.0052174
15	0.026064	0.018444	0.0043892	0.94868	0.93362	0.964	0.61623	0.024198	0.014861	0.0038934	0.0046568	0.0046568	0.0046568
16	0.02571	0.018158	0.0043702	0.9432	0.93799	0.96428	0.62499	0.02382	0.01479	0.0037402	0.0041094	0.0041094	0.0041094
17	0.025297	0.017865	0.0042705	0.94689	0.93869	0.9656	0.62721	0.023501	0.01462	0.0036356	0.003584	0.003584	0.003584
18	0.025075	0.017779	0.0041396	0.95315	0.93687	0.96782	0.63548	0.023267	0.01462	0.003548	0.0030888	0.0030888	0.0030888
19	0.024776	0.017511	0.0040907	0.9486	0.9386	0.96452	0.63481	0.023198	0.014519	0.0035426	0.0026316	0.0026316	0.0026316
20	0.024377	0.017468	0.0039792	0.95174	0.93805	0.96768	0.63873	0.022974	0.014476	0.0034908	0.0022196	0.0022196	0.0022196
21	0.024108	0.017404	0.0038956	0.95143	0.94602	0.96961	0.64408	0.022834	0.014329	0.0033827	0.0018594	0.0018594	0.0018594
22	0.023831	0.017413	0.0038219	0.95326	0.9471	0.97034	0.64867	0.022646	0.0143	0.0033148	0.0015566	0.0015566	0.0015566

3. 프로젝트 결과

batch_size = 32, epochs = 11

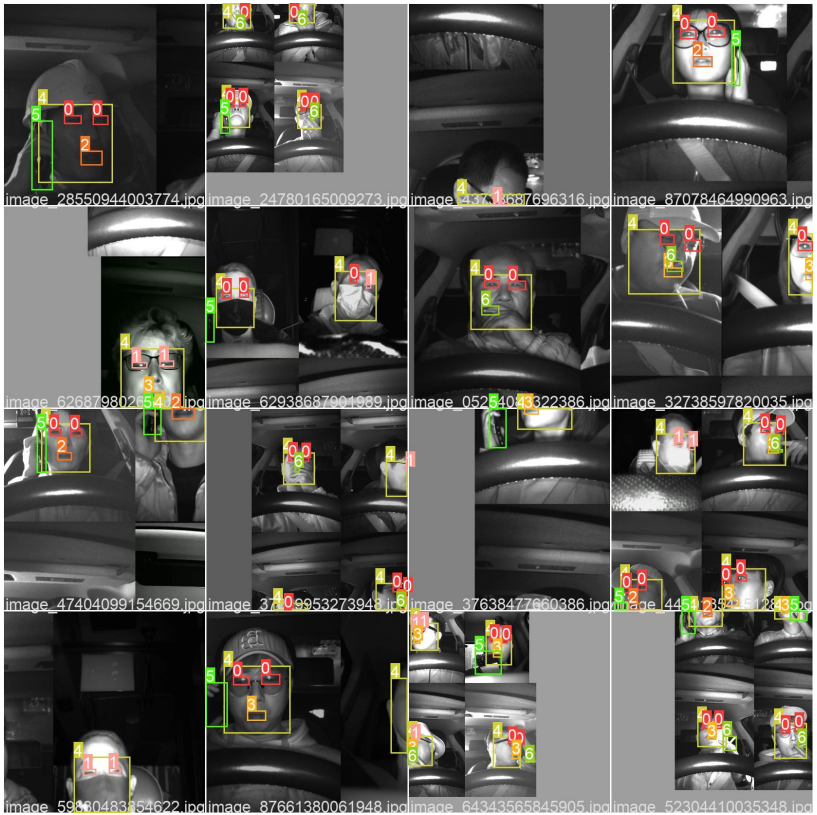
3. 프로젝트 결과

batch_size = 64, epochs = 23



3. 프로젝트 결과

batch_size = 64, epochs = 23



3. 프로젝트 결과

submission 파일 생성 (.json file >>> “objects” > “class” 와 [, , ,] bounding box, “confidence_score” columns 생성)

Public Private

순위	팀명	점수	최종제출일
1	TAVE8	1.000000000000000	2022-01-08 14:35:17
2	테이브	0.596897734400000	2022-01-10 12:23:31
3	별따닥이	0.524185824700000	2021-12-08 21:48:17
4	희곰	0	2021-11-26 13:44:44

{} results.json ×

C: > Users > kongs > Desktop > vscodeproj > {} results.json


```
{"annotations": [{"image_id": 0, "file_name": "image_46375024492458.jpg", "public": false, "objects": [{"class": "eye_opened", "position": [179, 650, 217, 686], "confidence_score": "0.60105"}, {"class": "mouth_closed", "position": [195, 772, 263, 818], "confidence_score": "0.82115"}, {"class": "eye_opened", "position": [241, 617, 297, 654], "confidence_score": "0.83334"}, {"class": "phone", "position": [246, 624, 528, 855], "confidence_score": "0.8707"}, {"class": "face", "position": [170, 577, 477, 885], "confidence_score": "0.88079"}]}, {"image_id": 1, "file_name": "image_89987118503447.jpg", "public": true, "objects": [{"class": "cigar", "position": [534, 885, 597, 929], "confidence_score": "0.78244"}, {"class": "eye_opened", "position": [596, 726, 662, 769], "confidence_score": "0.81285"}, {"class": "eye_opened", "position": [476, 697, 554, 743], "confidence_score": "0.82485"}, {"class": "face", "position": [323, 652, 673, 992], "confidence_score": "0.93019"}]}, {"image_id": 2, "file_name": "image_42157893326079.jpg", "public": false, "objects": [{"class": "eye_opened", "position": [606, 602, 657, 641], "confidence_score": "0.71440"}, {"class": "eye_opened", "position": [494, 585, 567, 629], "confidence_score": "0.84092"}, {"class": "mouth_opened", "position": [548, 746, 629, 786], "confidence_score": "0.84765"}, {"class": "phone", "position": [633, 644, 783, 823], "confidence_score": "0.91966"}, {"class": "face", "position": [359, 540, 664, 862], "confidence_score": "0.95603"}]}, {"image_id": 3, "file_name": "image_28678340697300.jpg", "public": true, "objects": [{"class": "phone", "position": [583, 624, 673, 846], "confidence_score": "0.77831"}, {"class": "mouth_closed", "position": [345, 720, 436, 752], "confidence_score": "0.84488"}, {"class": "eye_opened", "position": [420, 593, 498, 637], "confidence_score": "0.85054"}, {"class": "eye_opened", "position": [293, 587, 371, 631], "confidence_score": "0.85149"}, {"class": "face", "position": [246, 547, 549, 834], "confidence_score": "0.95401"}]}, {"image_id": 4, "file_name": "image_75699664474558.jpg", "public": false, "objects": [{"class": "eye_opened", "position": [268, 584, 324, 622], "confidence_score": "0.76843"}, {"class": "eye_opened", "position": [364, 576, 430, 614], "confidence_score": "0.82198"}, {"class": "mouth_opened", "position": [301, 710, 381, 770], "confidence_score": "0.8693"}, {"class": "phone", "position": [184, 646, 288, 802], "confidence_score": "0.89827"}, {"class": "face", "position": [260, 530, 564, 849], "confidence_score": "0.95871"}]}, {"image_id": 5, "file_name": "image_60389125934197.jpg", "public": false, "objects": [{"class": "cigar", "position": [320, 685, 366, 713], "confidence_score": "0.74749"}, {"class": "eye_opened", "position": [270, 538, 336, 577], "confidence_score": "0.80132"}, {"class": "eye_opened", "position": [388, 542, 464, 582], "confidence_score": "0.83099"}, {"class": "face", "position": [247, 490, 552, 807], "confidence_score": "0.93407"}]}, {"image_id": 6, "file_name": "image_28506827056630.jpg", "public": false, "objects": [{"class": "eye_opened", "position": [250, 643, 306, 679], "confidence_score": "0.73246"}, {"class": "eye_opened", "position": [349, 621, 424, 664], "confidence_score": "0.85241"}, {"class": "cigar", "position": [311, 723, 338, 797], "confidence_score": "0.85649"}, {"class": "face", "position": [237, 574, 568, 914], "confidence_score": "0.92939"}]}, {"image_id": 7, "file_name": "image_81961864778467.jpg", "public": false, "objects": [{"class": "eye_opened", "position": [595, 589, 658, 625], "confidence_score": "0.80493"}, {"class": "eye_opened", "position": [463, 599, 543, 641], "confidence_score": "0.83396"}, {"class": "mouth_opened", "position": [528, 743, 632, 796], "confidence_score": "0.88250"}, {"class": "phone", "position": [663, 598, 799, 773], "confidence_score": "0.92265"}, {"class": "face", "position": [340, 546, 677, 877], "confidence_score": "0.96610"}]}, {"image_id": 8, "file_name": "image_64362011059155.jpg", "public": true, "objects": [{"class": "mouth_closed", "position": [311, 656, 407, 706], "confidence_score": "0.50850"}, {"class": "mouth_opened", "position": [312, 659, 406, 708], "confidence_score": "0.58004"}, {"class": "phone", "position": [537, 531, 651, 844], "confidence_score": "0.74855"}, {"class": "eye_opened", "position": [268, 527, 327, 567], "confidence_score": "0.85134"}, {"class": "eye_opened", "position": [377, 500, 456, 546], "confidence_score": "0.85857"}, {"class": "face", "position": [258, 452, 585, 789], "confidence_score": "0.96878"}]}, {"image_id": 9, "file_name": "image_35222986823119.jpg", "public": false, "objects": [{"class": "eye_opened", "position": [440, 553, 506, 590], "confidence_score": "0.65521"}, {"class": "eye_opened", "position": [316, 559, 393, 599], "confidence_score": "0.78396"}, {"class": "mouth_opened", "position": [390, 689, 467, 738], "confidence_score": "0.82206"}, {"class": "phone", "position": [609, 631, 712, 967], "confidence_score": "0.84482"}, {"class": "face", "position": [233, 508, 521, 804], "confidence_score": "0.95359"}]}, {"image_id": 10, "file_name": "image_24941705549506.jpg", "public": true, "objects": [{"class": "eye_opened", "position": [589, 582, 645, 621], "confidence_score": "0.77456"}, {"class": "eye_opened", "position": [472, 558, 550, 602], "confidence_score": "0.83725"}, {"class": "mouth_opened", "position": [498, 717, 576, 763], "confidence_score": "0.86468"}, {"class": "phone", "position": [587, 619, 720, 802], "confidence_score": "0.90775"}, {"class": "face", "position": [340, 519, 659, 821], "confidence_score": "0.95714"}]}, {"image_id": 11, "file_name": "image_43024285958128.jpg", "public": false, "objects": [{"class": "eye_opened", "position": [482, 637, 534, 674], "confidence_score": "0.76649"}, {"class": "mouth_opened", "position": [441, 761, 504, 809], "confidence_score": "0.80830"}, {"class": "eye_opened", "position": [382, 637, 445, 675], "confidence_score": "0.84533"}, {"class": "phone", "position": [501, 714, 601, 883], "confidence_score": "0.8942"}
```

4. 결론 및 제언

1) 프로젝트 결과 요약

- Object Detection
- YOLOv5 model customizing
- hyperparameter tuning
- 이론 학습

4. 결론 및 제언

2) 프로젝트 한계 및 보완 방향

- data imbalance 최소화
- YOLOv5 small, medium, large, xlarge 모델을 모두 테스트
- Google Colaboratory GPU 사용량…
- Vision Transformer (ViT) 등 다양한 모델 테스트
- epoch, batch_size 더 다양하게 튜닝 (Colab 런타임의 한계…)

Reference

- [운전자 부주의 행동 검출 객체 인식 모델] https://github.com/PJunhyuk/2021AICompetition-03
- [YOLOv5 customizing] https://www.youtube.com/watch?v=T0DO1C8uYP8&t=479s, https://developeryoung.tistory.com/15?category=879762
- [YOLO style bounding box] https://eehoeskrap.tistory.com/357