

Object Detection

Wrap up Report

2021/09/27 ~ 2021/10/14

컴퓨터구조 (7 조)



목차

목차

I.	프로젝트 개요.....	3
1.	프로젝트 주제 및 개요	3
2.	활용 장비 및 재료 (개발 환경 등).....	3
3.	기대 효과.....	3
II.	프로젝트 팀 구성 및 역할.....	3
III.	프로젝트 수행 절차 및 방법	3
IV.	프로젝트 수행 결과.....	4
1.	탐색적 분석 및 전처리	4
	분석	4
2.	모델 개요 및 실험.....	4
3.	모델 선정 및 분석 & 모델 평가 및 개선	5
4.	시연 결과	5
V.	자체 평가 의견.....	6
1.	잘한 점들.....	6
2.	그 외 시도했으나 잘 되지 않았던 것들	6
3.	아쉬웠던 점들	6

I. 프로젝트 개요

1. 프로젝트 주제 및 개요

재활용 쓰레기 Detection

이미지에서 쓰레기를 Detection하는 모델을 만들어 분리 배출 문제를 해결한다.

2. 활용 장비 및 재료 (개발 환경 등)

GPU : AI Stages - NVIDIA V100

Python IDE : Jupyter Notebook, VSCode / Visualization tool : Wandb

3. 기대 효과

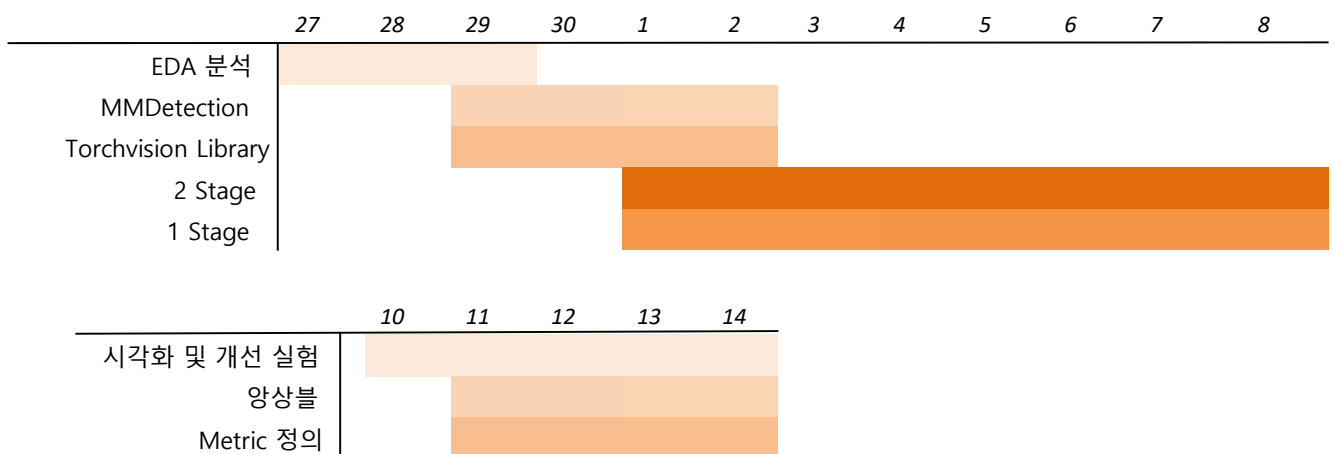
카메라로 비춰진 사람 얼굴 이미지만으로 마스크 정착용 여부를 자동으로 가려낼 수 있는 시스템이 공공장소 입구에 갖춰져 있다면 적은 인적자원으로도 충분히 검사가 가능할 것이다.

II. 프로젝트 팀 구성 및 역할

- **고재욱**(팀원) : 데이터 EDA, 시각화, 앙상블 코드 작성, Faster RCNN을 통해 Anchor Box 크기 수정 실험
- **김성민**(팀원) : PM, 시각화, 앙상블 코드 수정, detectron2, torchvision lib 이용한 코드 실험, universe 101, PVT-B3, DCN 돌리고 성능 확인, swin 기반 기능 적용 및 실험
- **박지민**(팀원) : RetinaNet, Yolo, EfficientDet 실험
- **박진형**(팀원) : DetectoRS 실험
- **심세령**(팀원) : Validation dataset 나누기, 새로운 metric 정의, Readme 등의 문서 정리, yolo-X, PVT-B5 돌리고 성능 확인, RetinaNet 기준 실험
- **윤하정**(팀원) : vfnets 실험, yolo 실험, loss 실험, 모델 ensemble

III. 프로젝트 수행 절차 및 방법

- Gantt chart



IV. 프로젝트 수행 결과

1. 탐색적 분석 및 전처리

분석

학습 데이터 : 이미지 4883장 (Train 80% / Valid 20%)

Input : (1024, 1024, 3) 이미지

Output : 각 이미지 Bounding Boxes

2. 모델 개요 및 실험

Ensemble	Input size: (1024, 1024)
	Skip box threshold: 0.01 / IoU threshold: 0.5
Swin_Transformer	Backbone : cascade rcnn image-resize : (1024, 1024), (2048, 2048) Neck: FPN, PAFPN anchor box : scales=[8], ratios=[0.5, 1.0, 2.0], ->0.33, 3.0 도 추가해서 실험 strides=[4, 8, 16, 32, 64]) ->2 도 추가해서 실험 rcnn : classification loss - focal, cross entropy rpn proposal 의 N - 1000, 2000 Optimizer : AdamW, Adam, SGD learning rate : 0.0001, 0.0001, 0.05 learning schedule : cosine annealing, stepLR ->(추후에 cosine annealing 이 좋아서 그것만 적용)
PVT_B3,B5	Backbone : RetinaNet Optimizer : AdamW
DetectoRS	Backbone : <u>cascade_rcnn_r50_fpn</u> (Resnet or ResNext) Optimizer : AdamW
EfficientDet	backbone : tf_efficientnet_b4 neck : BIFPN Optimizer : type = AdamW, lr = 0.0001 bbox head : RetinaHead

3. 모델 선정 및 분석 & 모델 평가 및 개선

모델	선정 이유 및 개선 방법
Swin_Transformer, PVT	swin 의 성능이 papers with code 기준 sota 였음 swin 과 PVT 모두 backbone 단에서 transformer 을 적용한 초창기 모델임 최종적인 구조가 one, two stage detector 과 동일해서 수정이 용이했음
DetectoRS	feature pyramid 를 Recursive 하게 진행하는 Recursive Feature Pyramid 를 사용함으로써 low level 과 high level 의 feature 를 최대한 모두 가져갈 수 있도록 만든 모델이기 때문에 좋은 성과를 낼 수 있을 것이라 생각하여 선택
	기존 모델에서 <u>cascade rcnn r50 fpn</u> 과 resnet 으로 교체하고 Optimizer 를 AdamW 를 적용 함
EfficientDet	one-stage 모델 중에 성능과 속도가 좋다고 알려진 모델 BIFPN 을 사용하여 neck 부분에서 bias 를 주는 것이 좋은 성과를 낼 것이라고 예상함
	수정 없이 d4 를 구현 했을 때 성능이 0.3 정도로 매우 낮았음 Optimizer 수정 => AdamW Dataset split => validation 적용

4. 시연 결과

모델	모델 성능	결과물 사진 mAP : 0.53, mAP: 0.602
Swin_Transformer	0.559 - val : 0.580	
DetectoRS	0.538 - val : 0.571	
EfficientDet	0.447	
PVT-B3, B5	0.517 - val : 3- 0.563, 5-0.515 (B3 가 valid 기준 높은 성능 -> 원인 : lib 차이일 것으로 판단)	

V. 자체 평가 의견

1. 잘한 점들

- 베이스라인에 있는 것 외에도 다양한 모델을 조사 및 구현해보았다.
- 평가 지표에 대한 고민, 우리만의 metric을 만들어서 단일 모델이 많은 문제를 해결할 수 있게 하려는 시도
- 포기하지 않고 끝까지 다양한 시도를 하여 배운 것이 많았다.
- 좋은 팀원을 만나 협업을 배울 수 있다.

2. 그 외 시도했으나 잘 되지 않았던 것들

- One-stage 모델의 낮은 성능
- YOLOv4 구현 실패
- False positive 줄이기

3. 아쉬웠던 점들

- EDA 관련 기준의 부족, valid의 gt, prediction을 비교하는 지표, 시각화를 늦게 진행함.
- 라이브러리의 다양성이 부족했음
- 초반에 모델만을 돌리는 것에 집중
- VIT기반 모델들이 fine-tune시 사용한 data가 많기도 하며 일정 성능 이상을 뽑기 위해선 data가 많이 필요하다는 특징에 따라 빠른 속도로 최고 valid에 수렴하였으며, 모델 수정을 거쳐도 극적이 성능 개선이 있진 않았음.
- (실험엔 다른 모델이 좀더 좋았을 수 있을 것 같음)