



김성민

T2038 (Male)

팀원들과 가치를 만드는 개발자

생년월일

1996.05.17

보유 기술

- PyTorch, Numpy, Pandas, opencv
- C, Java, Python, linux
- git, notion, slack

연락처 및 사이트

연락처: 010-4879-7216

mail: 7tkfkd@naver.com

github: <https://github.com/ksm0517>

blog: <https://blog.naver.com/7tkfkd>

서울시 금천구 시흥대로84가길 26

병역사항

군필

국적

한국

프로젝트 (제가 기여한 부분)

Ocr in Editor(이미지 속 글자 번역을 하는 편집 툴)

3글자 이내 이미지 기준 수행시간 2초 이내, F1-score 0.62

- PM, front end 구축, 모델링

모델 경량화 및 최적화(Classification)

2000장 inference - 58초, F1-score 0.68

- 팀원들에게 baseline을 해설 및 로깅 환경 구축

Segmentation (이미지 속 쓰레기 검출)

LB rank 14 -> 9, private mIoU 0.720

- Engineering Tool, swin 모델을 활용한 실험

Object Detection(이미지 속 쓰레기 검출)

F1 score 0.4 -> 0.5

- Engineering Tool, swin 모델, metric 설정, 모델 hyper parameter tune

Classification(마스크, 성별, 나이 분류)

F1 score 0.771(2위)

- customized 라벨 스무딩, 수도 라벨링, 프로젝트 일정 관리

학력

명지대학교

물리학, 2015.03~2019.08

- 학점 : 3.49/4.5

- 수강 과목 : 전산물리, 광학, 반도체, 수리물리

- 동아리 : 응원단(응원단장)

외부교육이수사항

부스트캠프

2021.08~이수중

인공지능 기초, CV

classification, object detection, segmentation

data annotation(ocr) project

KGITBANK

2017.09~2017.12

Java, JSP, 네트워크 기초

취미

- 일렉기타

- 방송 편집

- CS 유튜브 보기(ex - 운영체제 강좌, docker 강좌)

관심사항

제조업 및 영상처리 분야 딥러닝을 통한 문제 해결



이름 T2038 김성민
 연락처 010-4879-7216
 사이트 <https://github.com/ksm0517>
 메일 7tkfd@naver.com

Portfolio

my goal(To Do) : 팀원들과 가치를 만드는 개발자

- 지금 하는 일이 결과에 미치는 영향에 대한 통찰력, 자료들에 대한 지식 및 전달력
- 팀원들이 실험에 집중할 수 있는 환경을 만들어 주어야 함(ex- lib, 사용법)

오이(OCR in Editor)

RESULT



before
applying



after
applying

기능 및 성능

편집 중 영상에 있는 자막을 번역하고 싶을 때,
 버튼을 누르면 번역된 이미지가 나오는 기능
 - 3글자 이내 이미지 기준 수행시간 2초 이내, F1-score 0.62

제가 한 역할

1. PM
 - 자료 조사 및 전달
(Mish activation, pyqt5, base64, CTC loss, CRNN 등)
 - NABC 사업 계획 모델에 맞추어 팀원들과 프로젝트를 구성
 - frontend는 기능을 구현해야 하는 것에 맞추어 일정 작성
 - modeling은 baseline 구축까지 일정을 작성 후
모델 성장을 위해 해야 할 일, 하면 좋은 일, 긴가민가 한 일
3가지로 나누어 상의 한 뒤, 역할 배분
 - 코드 이해를 못한 팀원을 따로 만나서 전체 구조 해설
 - 발표문 작성 및 발표
2. frontend
 - docks, tab, action UI 구현으로 기틀 형성
 - 글자 배경, 글자색으로 쓸 색을 추출하는 기능 구현
3. model
 - CRNN을 Fine-tuning
Mish를 적용해 gradient 흐름을 좋게 함,
마지막 블록의 num-class를 맞춰줌
 - wandb 활용 모니터링 환경 구축
 - shared convolution(backbone) 구축
 - test code 작성

추후 개선 사항

- 단어 단위가 아닌 문장 단위를 받아 왔을 때의 처리
- transformer등 최신 모델을 활용한 성능 개선
- 다양한 언어 지원
- 잘못된 자막을 받아왔을 경우
영상 편집자가 수정할 수 있는 기능

Segmentation

RESULT



기능 및 성능

사진에 있는 쓰레기를 종류 별로 segmentation
 - LB rank 14 -> 9, private mIoU 0.720

제가 한 역할 (문제 -> 해결)

1. 박스 위에 붙어 있는 테이프를 일반쓰레기로 잘 검출하지 못함
 - CRF를 적용
(성능이 하락함 -> 과거 적용된 기술인 만큼 swin과 맞지 않았던 듯함)
2. Class imbalance
 - Object Augmentation
(데이터가 부족한 라벨의 데이터를
다른 사진에 합성해서 사용)
3. confusion metric을 통해 본 결과 플라스틱, 일반쓰레기, 종이끼리
서로 혼동이 있음
 - rgb shift
4. 물체의 다양한 사이즈, 각도에 대응
 - scale TTA, rotate
5. valid상 적용한 모델을 마다 잘맞추는 라벨이 따로 있었음
 - valid를 모니터링 하고 특정 라벨만 각 모델별 우선순위를
부여해서 ensemble

(Engineering Tool)

- mmdetection, mmsegmentation lib 환경 구축
- Visualize tool
- psuedo labeling
- ensemble
- mmsegmentation에 alumentation 적용 가능하게 구현
- object augmentation debugging
- Swin base를 기반으로 실험을 진행



경량화

RESULT



Glass

Input

Output

기능 및 성능

Input : 쓰레기 사진 -> Output : classified label
- 2000장 inference - 58초, F1-score 0.68

제가 한 역할

1. 팀원들에게 Baseline 해설
 - Optuna를 활용해 model을 searching
 - W&B를 활용한 logging
2. AdamW, NAdam, RAdam 등 optimizer을 적용 및 해설
3. Lr scheduler(Cos annealing, Cycle) 적용 및 해설

Detection

RESULT



Before Ensembled

Ensembled

기능

Input : 쓰레기 사진 -> Output : detected picture
- F1 score 0.4->0.5

제가 한 역할(문제 -> 해결)

1. 작은 물체 검출력이 좋지 못함
 - 모델을 고정한 상태로 PAFPN, 학습 resolution 높이기, anchor box의 다양화
2. 동일한 물체에 박스가 여러개 생김
 - rpn_proposal줄이기
3. mAP에선 무작정 박스를 높이는 것이 수치가 높음
 - 팀원들과 좋은 metric을 고민해봄.
(우선은 f1을 설정하고 이를 개선하기 위해 노력)
4. Class Imbalance
 - Focal loss를 적용
5. convolution 기반 가벼운 모델의 실험
 - Deconvolution Net(DCN)실험

Classification

RESULT



Input

Output

기능 및 성능

Input : 사람 사진 -> Output : 성별, 마스크 착용 여부, 나이대
- F1 score 0.771(2위)

제가 한 역할(문제 -> 해결)

1. 인공지능 프로젝트에서 무엇을 해야할지 모르는 팀
 - 대회에서 제시하는 F1 score을 높일 방법을 '해야할일, 하면 좋은일'로 나누어 팀원들과 brain storming
 - 일정을 작성하고 상의하여 일정 확립 및 진행
 - 1주차에 비해 0.13가량의 F1 score을 개선
2. 나이 분류 모델의 성능이 낮음
 - 정해진 나이 구역 이외에도 정답 라벨을 약간 주는 label smoothing을 적용 (0.07가량의 성능 향상)
3. 테스트 데이터를 활용
 - Pseudo labeling을 구현하고 적용