

1 주차 활동 보고서

발표자: 최승환

일자: 2023-07-11

목차

- | 01 1 주차 활동 소개
- | 02 1 주차 활동 결과
- | 03 2 주차 목표 설정
- | 04 현재 진행 상황

1 주차 활동 소개

0703 (1일차)

- 주차별 추진 일정 및 내용 숙지
- 노트북 인계 및 개발환경 구축
- 딥러닝 기초: MNIST 데이터셋을 이용한 손글씨 숫자 인식 수행

0706 (4일차)

- 검출 성능에 대한 고민
- 더 나은 데이터 선정 방법

0704 (2일차)

- 해외 데이터 300장으로 차량번호판 인식

0705 (3일차)

- 해외 데이터 1000장으로 차량번호판 인식

0707 (5일차)

- 대구디지털혁신진흥원(DIP)에서 외부 교육 참가

1 주차 활동 과정 - 1일차

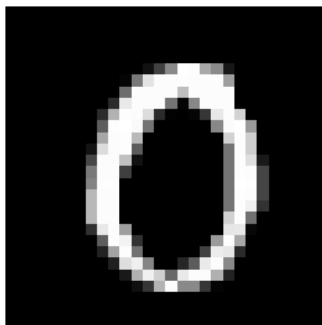
1. Visual Studio Code 설치
2. 아나콘다(파이썬) 설치 및 vscode 연동
3. 파이썬 라이브러리 설치
4. [Colab](#)에서 간단한 손글씨 숫자 인식 프로그램 구현

개발 환경

- ✓ Windows 10
- ✓ Python=3.8
- ✓ Opencv-python>=4.8.x
- ✓ Numpy>=1.21.6
- ✓ Pillow>=9.3.0
- ✓ Torch>=2.0.1

Accuracy: 0.7199999690055847

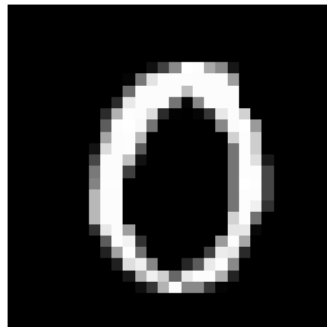
result = 0



10 에포크

Accuracy: 0.7999999523162842

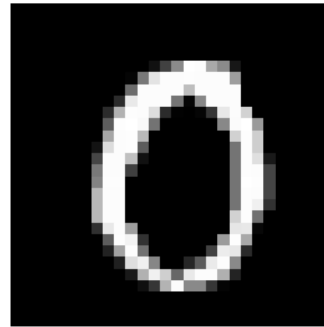
result = 0



20 에포크

Accuracy: 0.824999988079071

result = 0



40 에포크

1 주차 활동 과정 - 2일차

1. 차량번호판 객체인식
2. Kaggle에서 해외 데이터 300장 수집
3. Roboflow에서 데이터 라벨링 및 데이터 증강 후 훈련 데이터 600장 생성
4. [Colab](#)에서 YOLOv7으로 학습

파라미터

- ✓ Batch-size: 16
- ✓ Epochs: 100
- ✓ model: yolov7x.pt
- ✓ Image-size: 640x640

1. 데이터 수집



2. 데이터 라벨링



3. 딥러닝

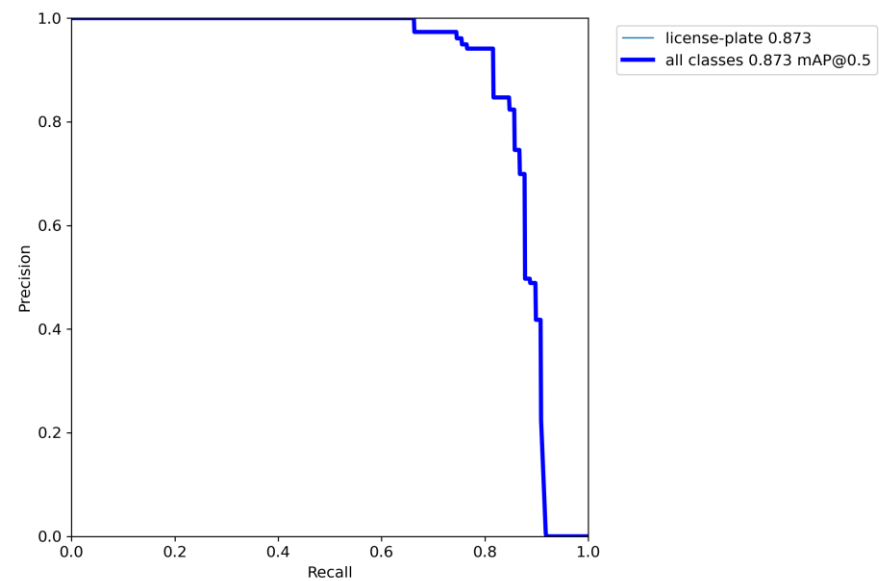


1 주차 활동 과정 - 2일차

4. 테스트



5. 평가



1 주차 활동 과정 - 2일차

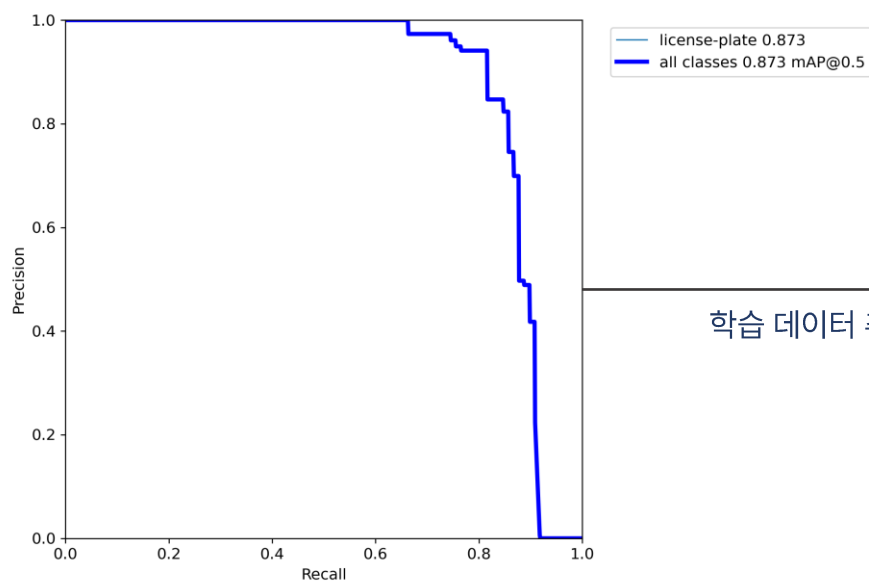
낮은 정답률 → 더 많은 데이터 수집 필요

1 주차 활동 과정 - 3일차

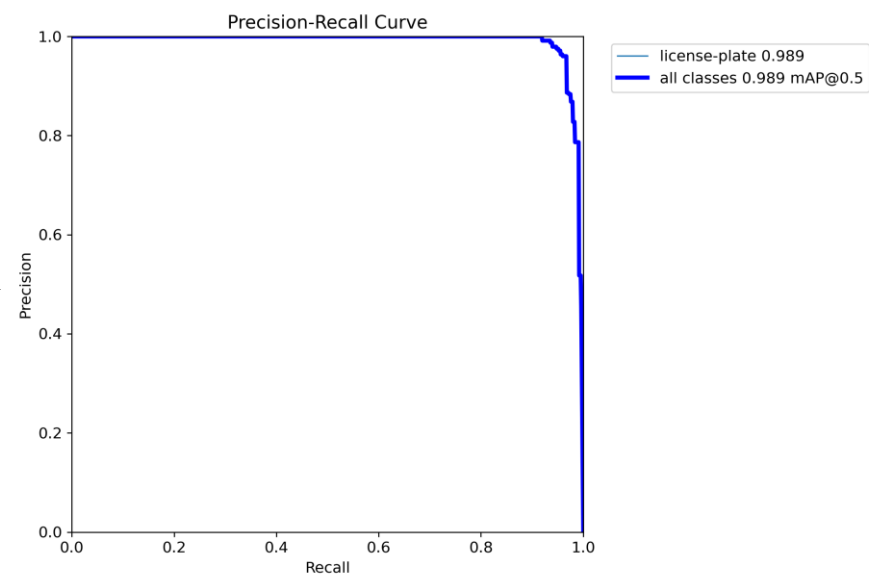
1. 차량번호판 객체인식
2. Kaggle에서 해외 데이터 1000장 수집
3. Labellmg 툴을 사용하여 데이터 라벨링
4. [Colab](#)에서 YOLOv8로 업그레이드하여 학습

파라미터

- ✓ Batch-size: 16
- ✓ Epochs: 100
- ✓ model: yolov7m.pt
- ✓ Image-size: 640x640



학습 데이터 추가



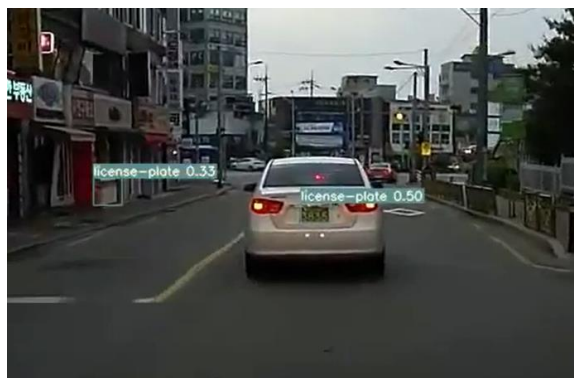
1 주차 활동 과정 - 3일차

- 인식률 자체는 좋아졌지만 거짓 양성의 비율이 증가하였다.
- 차량번호판의 데이터 개수와 질이 여전히 좋지 못했다.
- 데이터가 일관적이지 못하다.
- 실시간 영상에 대한 인식률이 저조하다. (특히, 국내 번호판)

엄격한 데이터 선정과 국내 차량번호판 데이터가 필요

1 주차 활동 과정 - 4일차

목표: 거짓 양성 비율을 줄이기 위한 데이터 수집



잘못된 데이터 선정

- 지금까지 일반적이지 않은 번호판들을 라벨링 했었다.

1 주차 활동 과정 - 4일차

데이터 선정 방식

- 차량번호가 잘 보이는가?
- 번호판 색과 모양이 대체로 일관적인가?
- 약 500개 수집



1 주차 활동 과정 - 6일차

데이터 라벨링 → 데이터 학습 → 평가

- 오히려 떨어진 검출율과 정확도
- 과적합의 문제 발생 가능성
- 너무 적은 데이터량

2 주차 목표 설정

새 프로젝트 진행을 위한 영상처리 기술 공부

- Opencv 라이브러리를 활용한 영상처리
- 이미지 색상 표현/변환/분리
- 이미지 연산
- 에지 검출
- 노이즈 필터링
- 회전, 어파인 변환, 원근 변환
- 로컬 호스트 브라우저에서 위 기능들 구현

현재 진행 상황

1. Opencv 라이브러리 활용
2. Flask 웹 프레임워크를 사용하여 웹캠 및 비디오 출력
3. 딥러닝 연산을 위한 스트리밍 방법
 - [Jinja](#) 템플릿을 사용하여 HTML 동적 프로그래밍

개발 환경

- ✓ Windows 10
- ✓ Python=3.8
- ✓ Opencv-python>=4.8.x
- ✓ Flask>=2.3.2

