

에코빈 진행 보고서

작성 및 검토 확인란

구분	성명	년 월 일	서명
작성자	이지훈	2023.05.16.	

개정 이력

개정일자	버전	개정내용	작성자	확인자
2023.05.09	1.0	05/09 진도보고서	이지훈	-
2023.05.16	2.0	05/16 진도보고서	이지훈	

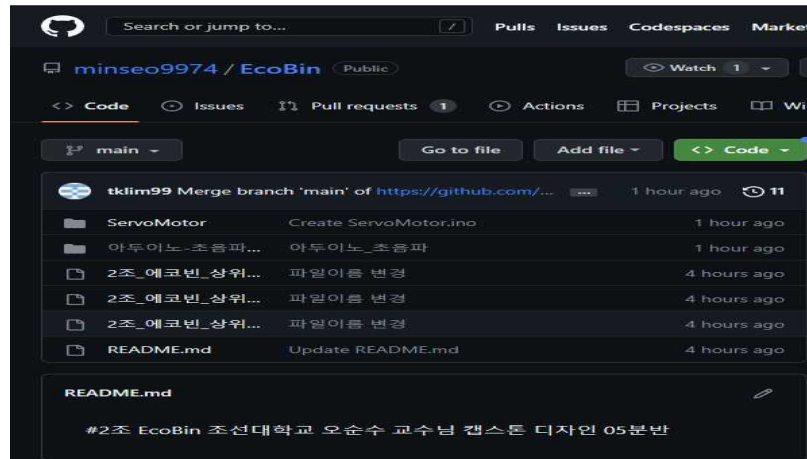
목 차

제 1 장. 진도보고서	1
제 1 절 5월 9일 진도보고서	3
제 2 절 5월 16일 진도보고서	6

제 1 장. 진도보고서

제 1 절 5월 9일 진도보고서

- 버전관리와 협업을 위해 github 프로젝트를 생성함.
- 이후 <https://github.com/minseo9974/EcoBin>를 통해 진행 상황을 올리고 피드백 활동 진행함.



[그림 1-1] github 프로젝트 생성

[표 1] 에코빈 물품 조사

물품	가격 (원)	결제방식	수량 (개)
ESP32 IOT 사물인터넷 WiFi + 블루투스 듀얼모드 아두이노 보드 모듈	5,980	선불 결제 (네이버페이)	1
아두이노 ESP32 DevKitC V4 38p 블루투스 개발 보드	10,980(배송비 포함)		1
아두이노 SG-90 SG90서보모터	5,280		4
아두이노 스텝핑 모터 키트	57,400(배송비 포함)		1
ESP32 WIFI + Bluetooth 일체형 개발보드	11,000	후불 결제 (에듀이노)	1
우노 R3 DIP 호환보드	23,000		2
초음파센서 HC-SR04	4,400		4
라즈베리파이 카메라모듈 V2	43,500		1
라즈베리파이 NOIR 적외선 카메라모듈	35,900		1

- 현재 틀 제작을 위한 재료 제외, 기능 물품을 구매 완료 후 승인 대기 중 임.
- 선불 결제 : 79,640원, 후불 결제 : 117,800원, 총 197,440원을 사용함.

상태	작성일	결제방법	신청금액	작업
교수승인중	2023-05-02	현금영수증(사업자용)	79,640원	보기 취소
교수승인중	2023-04-28	전자(세금)계산서(청구/영수)	117,800원	보기 취소

[그림 1-2] 제작비 지급 신청 현황

- 에듀이노는 코딩 교육을 위한 교구, 전자부품을 전문적으로 취급하는 코딩 교구 전문 쇼핑몰임.
- 후불 결제 절차가 자세히 안내되어 있어 쉽게 이용할 수 있으며, 다른 학교에서도 많이 사용하는 쇼핑몰임.



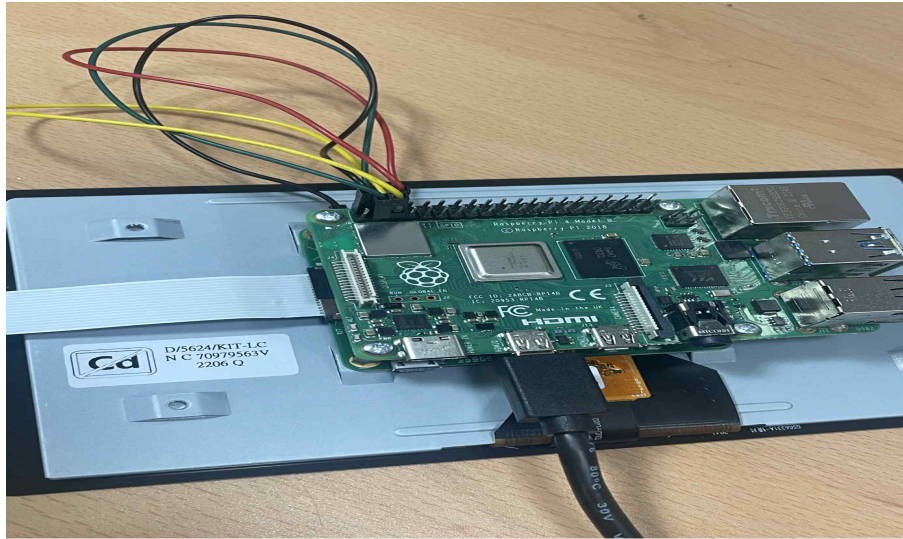
[그림1-3] 에듀이노 후불결제

- 영상인식을 위한 라즈베리파이 OS를 <https://www.raspberrypi.com/software/>의 Raspberry Pi Imager v1.7.4를 이용하여 64비트 OS를 설치함.



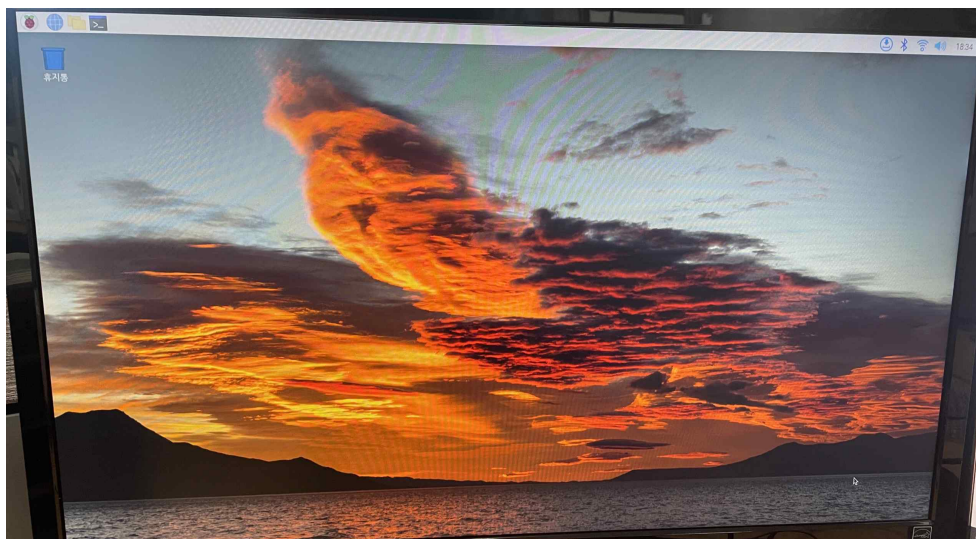
[그림 1-4] Raspberry Pi Imager

- 라즈베리파이에 OS가 담긴 SD카드, 전원 연결하여 초기 라즈베리 환경을 설정함.



[그림 1-5] 라즈베리파이 연결

- 모니터와 라즈베리파이를 HDMI선으로 연결하여 모니터에서 라즈베리파이를 사용함.



[그림 1-6] 라즈베리파이와 연결된 모니터 화면

- 라즈베리파이를 사용 준비 완료함.

제 2 절 5월 16일 진도보고서

■ 지원금 신청 반려 재신청

· 반려사유 : 53,900 영수증이 없음. 또한 나머지 증빙이 잘못됨, 물품구매사진은 재료를 한곳으로 모은 후 1장으로 찍고, 영수증도 하나의 PDF로 묶어서 연번1번 간에만 첨부. "연번1번"외 연번2번부터는 어떠한 증빙도 철하지 말것.

[그림2-1] 반려사유

- 링크사업단에서 지원금 신청을 그림[2-1]와 같은 사유로 반려함.
- 서류를 재수정하는 과정 중 본인의 실수로 신청취소가 진행되었음.
- 현재 재 신청 완료 후 교수님 승인 대기 상태임.

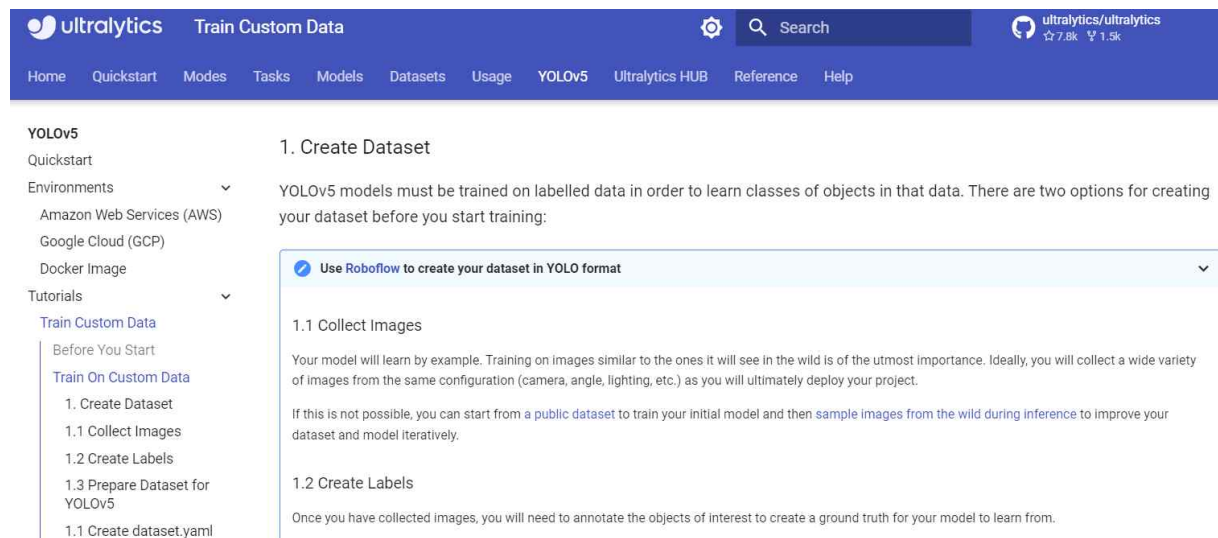
■ 데이터 세트 수집 및 구축

- 데이터세트 수집은 2가지 방법으로 진행함

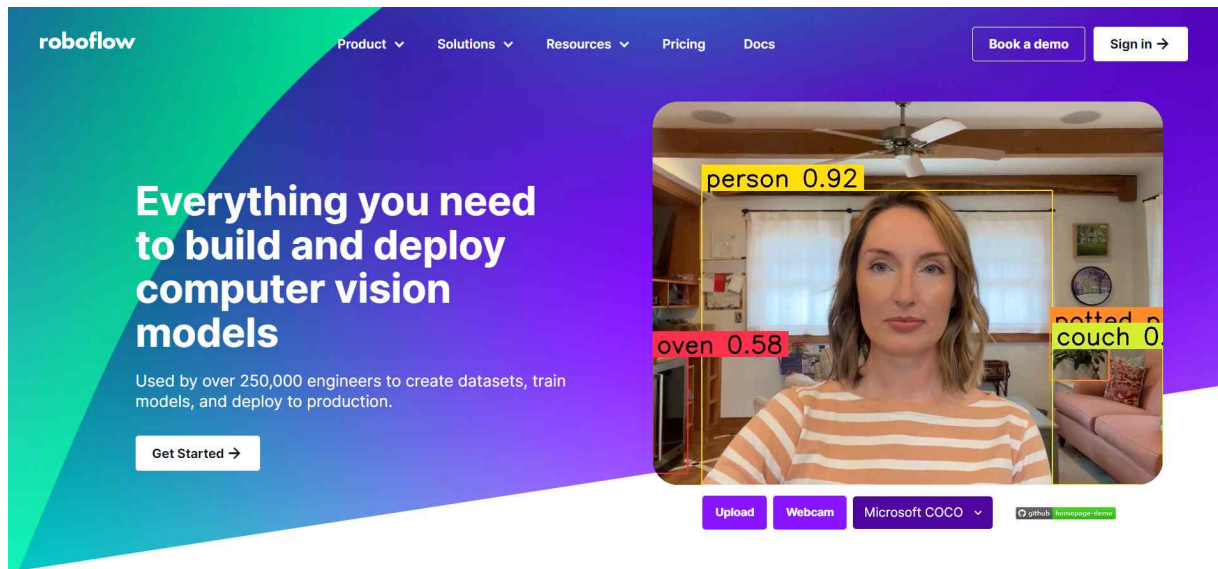
1. YoloV5의 github에서 추천하는 roboflow 홈페이지에서 Yolo형식의 데이터세트 다운로드
2. Ai허브에서 신청한 '생활 폐기물 이미지' 중 플라스틱과 캔 이미지 다운로드, labeling 프로그램을 통한 이미지 라벨링 작업

Roboflow (<https://roboflow.com/>)

- roboflow는 YOLO Github(<https://github.com/ultralytics/yolov5/>)에서 Custom Data를 만들기 위해 추천하는 사이트임.
- 다양한 컴퓨터 비전 모델을 만들기 위해 원시 이미지를 사용자 지정 형식에 따라 제공함.



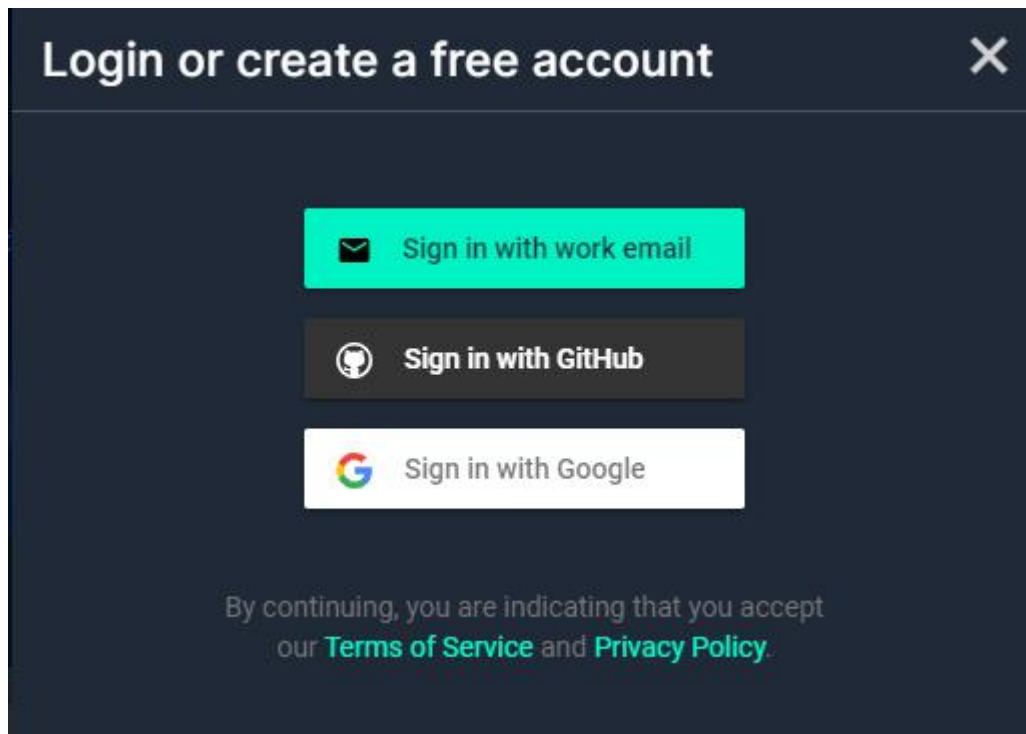
[그림2-2] Yolo에서 추천하는 Roboflow



[그림 2-3] Roboflow 홈페이지

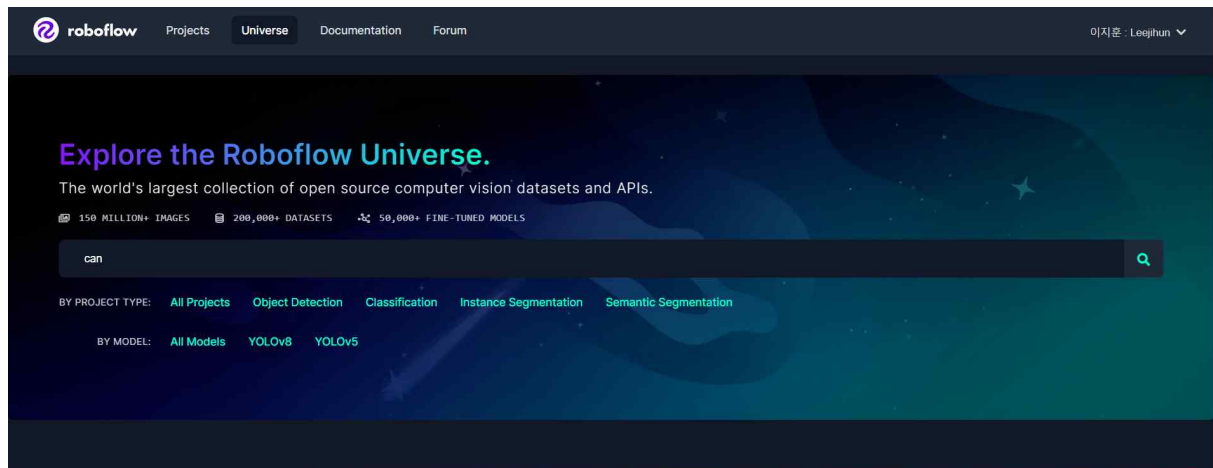
▪ roboflow 데이터 수집 과정

– roboflow 회원 가입



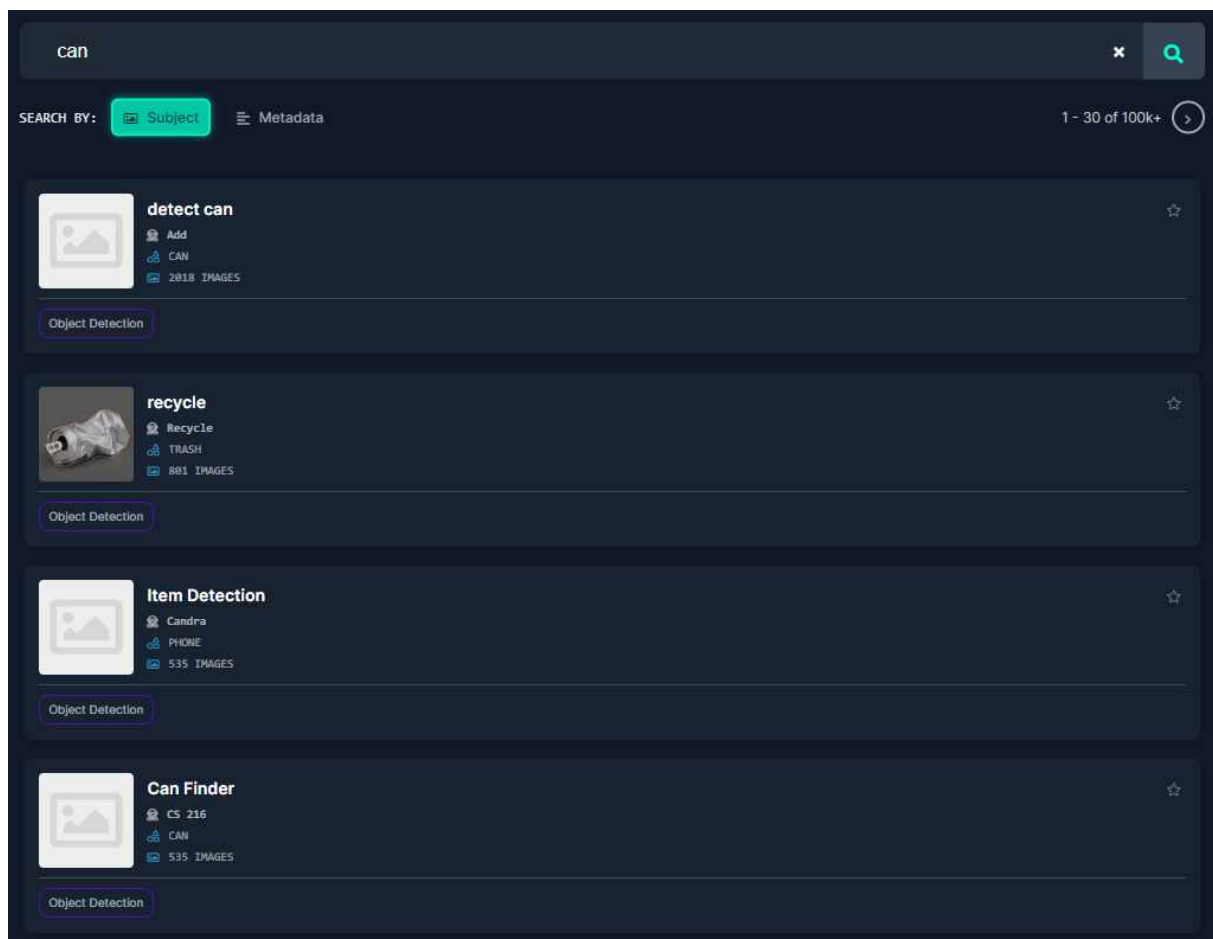
[그림 2-4] roboflow 회원가입

- Roboflow의 Universe 접속함
- 검색창에 찾고자 하는 데이터 검색함 - can검색



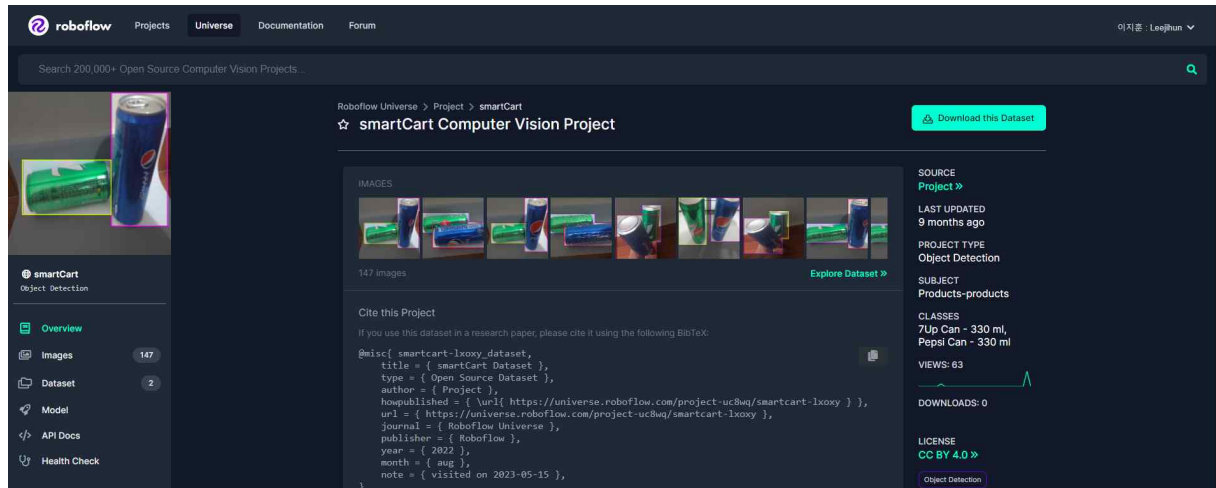
[그림 2-5] Roboflow Universe 검색

- 다른 사용자가 공유한 데이터셋 및 이미지가 나열됨



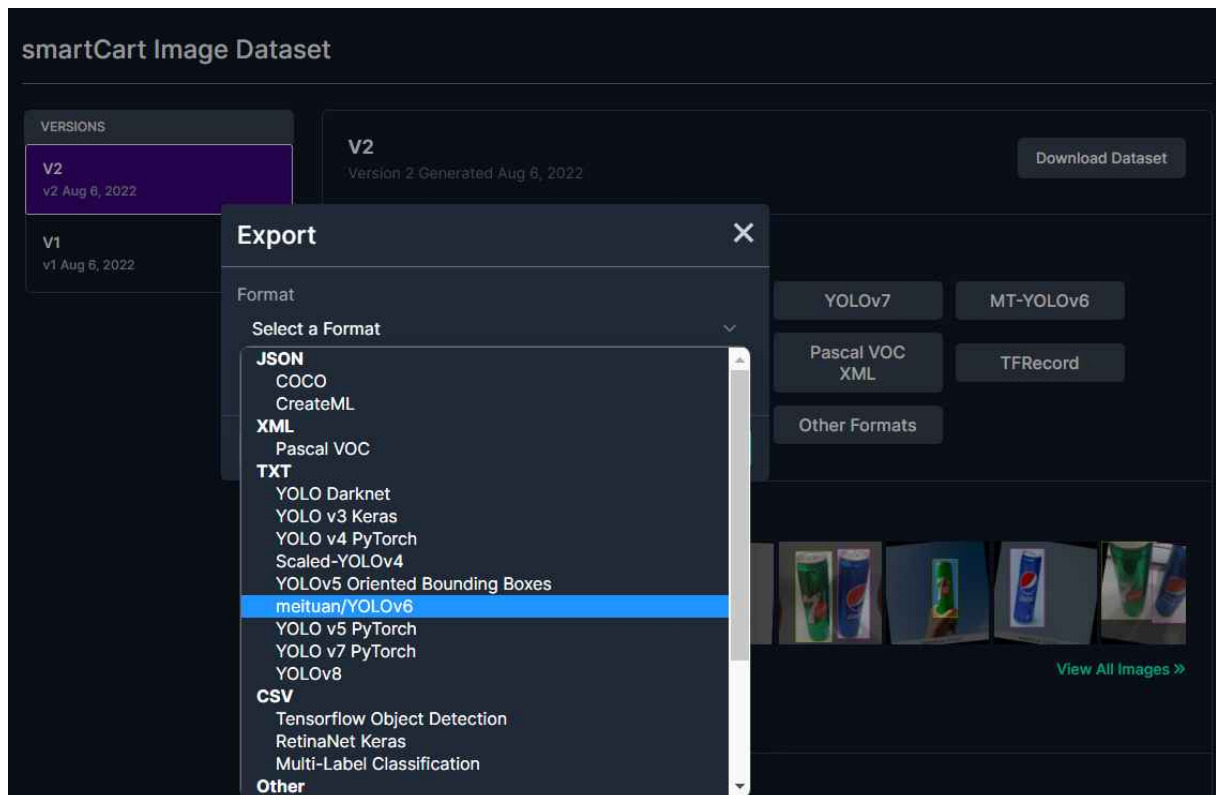
[그림 2-6] can 검색 결과

- 원하고자 하는 데이터셋, 이미지를 선택하여 다운로드 진행



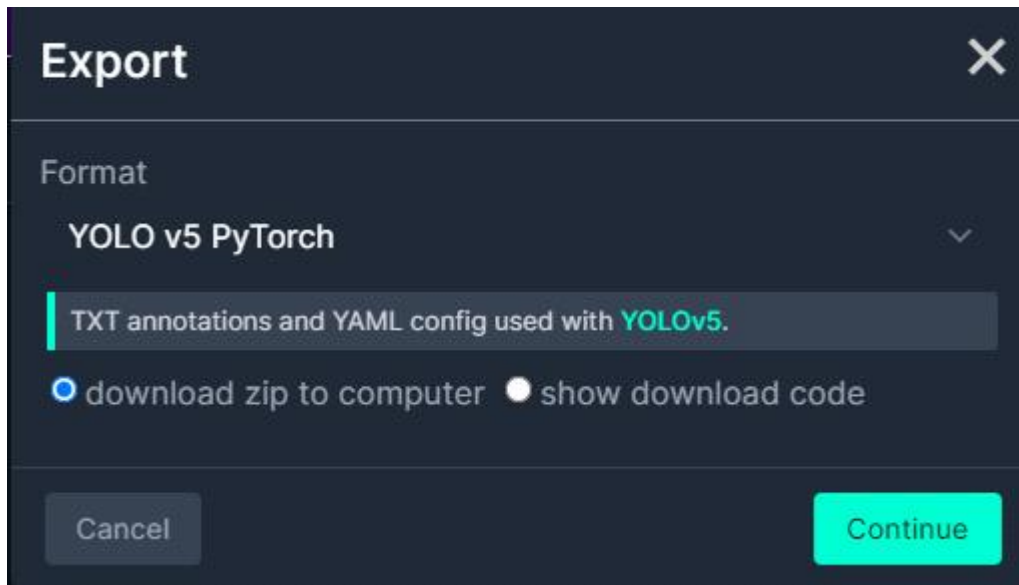
[그림 2-7] 선택한 can 데이터셋

- Download this Dataset을 클릭하면 사용자가 사용하려는 모델의 데이터형식을 설정할 수 있음.



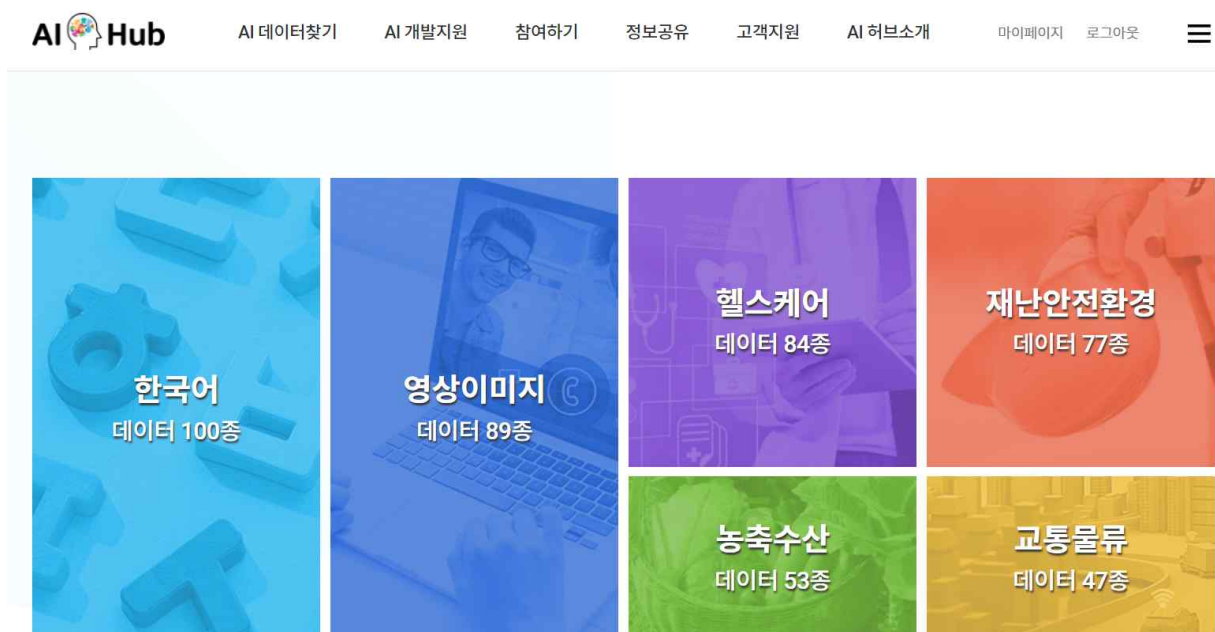
[그림 2-8] 데이터 형식 설정

- download zip to computer : 컴퓨터에 zip형식으로 다운로드
- show download code : code를 실행해 다운로드



[그림 2-9] 다운로드 방식

- 컴퓨터에 zip형식으로 데이터셋을 다운 받아 수집 진행함.
- Ai hub(<https://aihub.or.kr/>)
- Ai hub는 Ai기술 및 제품·서비스 개발에 필요한 Ai 인프라(Ai데이터, Ai SW API, 컴퓨팅자원)을 지원하는 Ai 통합 플랫폼
- 한국어, 영상이미지, 헬스케어, 재난안전환경 등 Ai학습용 다양한 데이터를 제공함.



[그림 2-10] Ai hub

■ AI허브 데이터 수집 과정

- 사용하고 싶은 데이터를 제공하는 절차에 맞춰 신청함.

데이터명	생활폐기물 데이터 활용-한류
유형	이미지
신청목적 *	<div>선택</div> <div>▼</div>

☐ **본인은 AI 허브 데이터를 이용함에 있어 다음과 같은 사항을 준수할 것에 동의합니다. (필수)**

1. 본 시데이터 등을 이용할 때에는 반드시 한국지능정보사회진흥원의 사업결과임을 밝혀야 하며, 본 시데이터 등을 이용한 2차적 저작물에도 동일하게 밝혀야 합니다.
2. 국외에 소재하는 법인, 단체 또는 개인이 시데이터 등을 이용하기 위해서는 수행기관 등 및 한국지능정보사회진흥원과 별도로 합의가 필요합니다.
3. 본 시데이터 등의 국외 반출을 위해서는 수행기관 등 및 한국지능정보사회진흥원과 별도로 합의가 필요합니다.
4. 본 시데이터는 인공지능 학습모델의 학습용으로만 사용할 수 있습니다. 한국지능정보사회진흥원은 시데이터 등의 이용의 목적이나 방법, 내용 등이 위법하거나 부적합하다고 판단될 경우


※ 데이터 다운로드는 PC에서만 가능합니다.

다운로드

[그림 2-11] 데이터 신청

- 데이터 신청 완료하여 다운로드를 받을 수 있음

데이터 신청 내역

**데이터 다운로드
신청 절차**


1.접수완료 : 데이터 신청 건이 접수된 상태입니다.

3.자동승인 : 데이터 다운로드가 자동승인 된 상태입니다.

※ 안심존 데이터의 경우 안심존 사용신청 상태가 제공됩니다

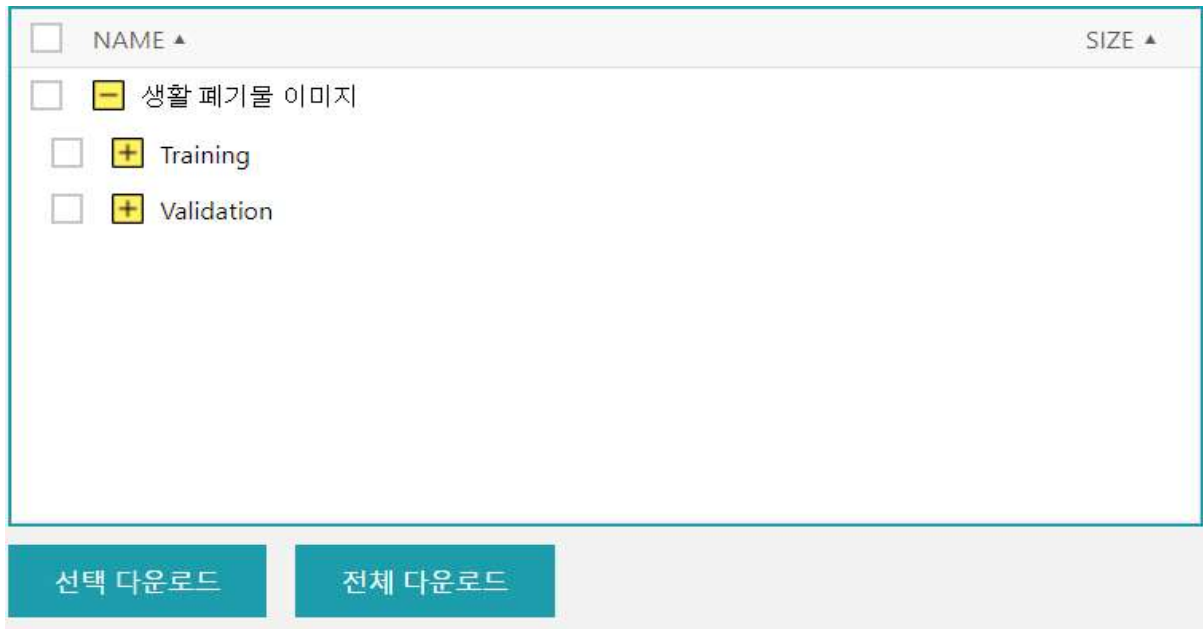
2.승인완료 : 데이터 다운로드 승인이 완료된 상태입니다.

4.반려 : 데이터 다운로드 신청이 반려된 상태입니다.

번호	분야	데이터명	구분	파일유형	상태	다운로드
1	재난안전환경	생활 폐기물 이미지		비디오, 이미지	자동승인	다운로드

[그림 2-12] 데이터 승인

- 데이터는 학습용과 검증용 데이터로 구분되며 학습용 데이터를 수집함.



[그림 2-13] 다운로드 내용

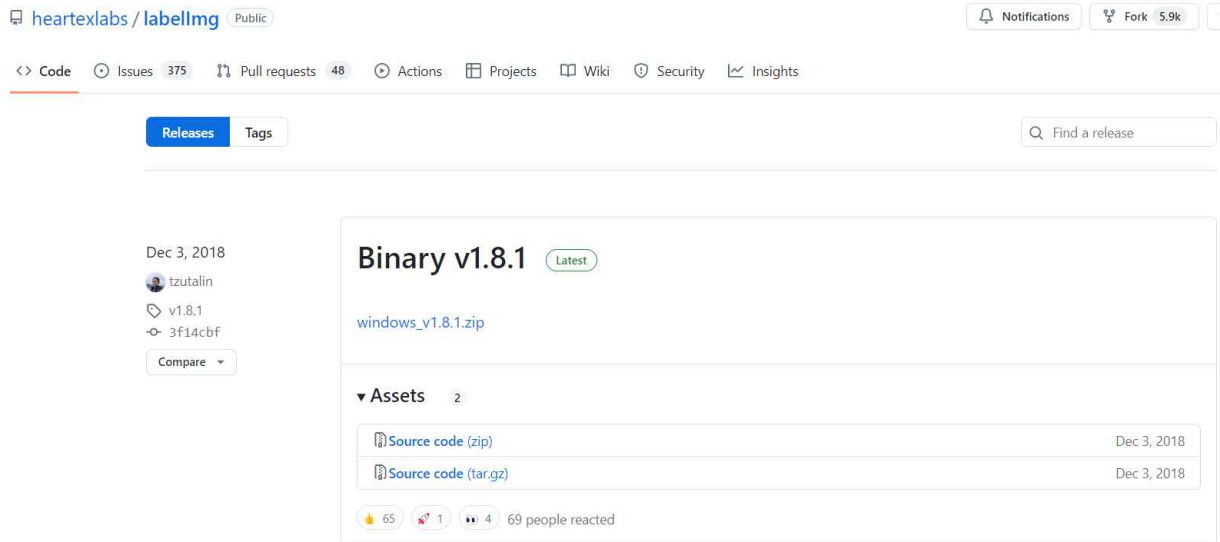
- 다운로드를 진행함.



[그림 2-14] 데이터 다운로드

■ Ai hub 이미지 라벨링 작업

- 원시 이미지를 Yolo에 적용하기위해 이미지 라벨링 작업이 필요함.
- <https://github.com/heartexlabs/labellmg/releases>에서 제공하는 labellmg 프로그램을 이용함.



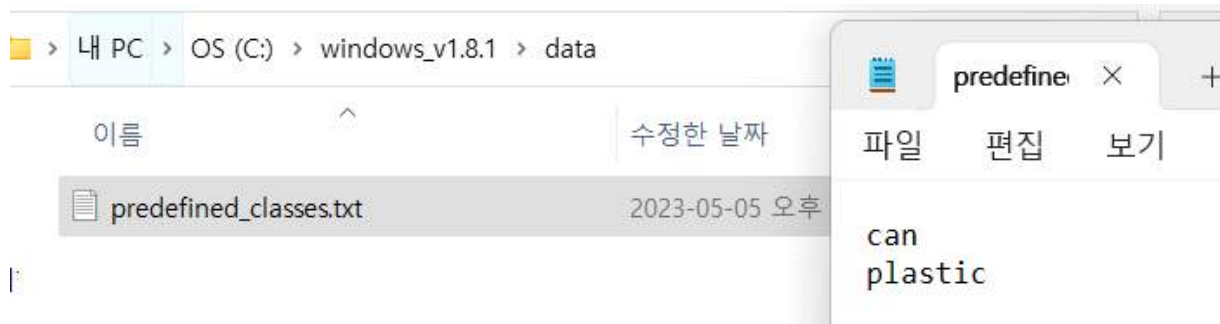
[그림 2-15] labellmg 프로그램 git hub

- Binary v1.8.1 zip 파일을 다운로드함.



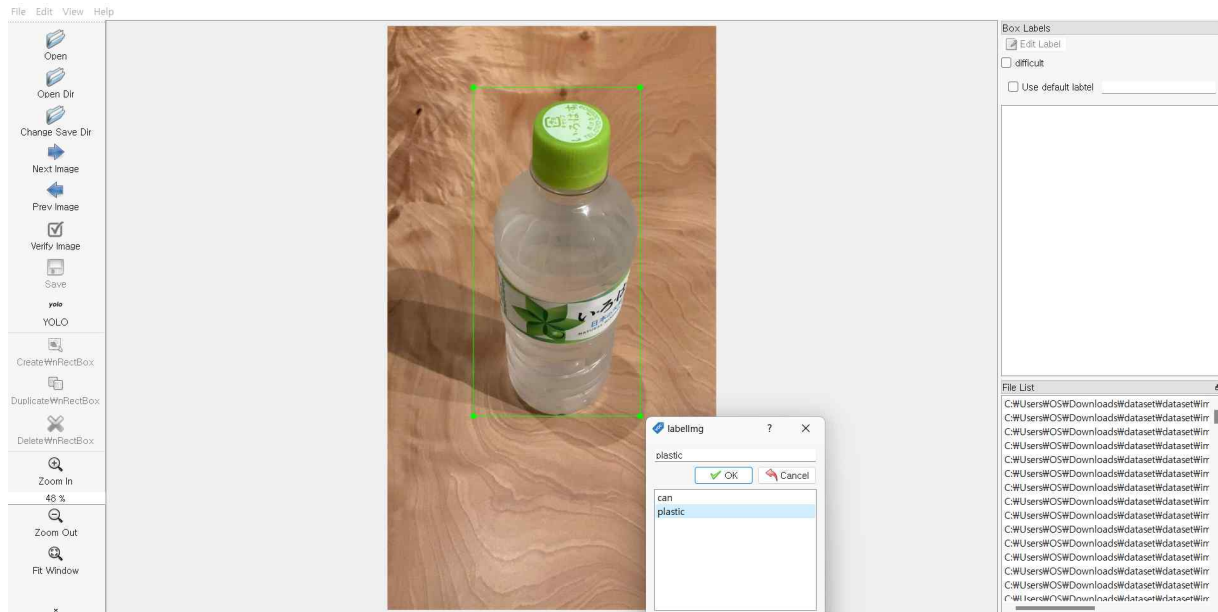
[그림 2-16] labellmg 파일

- 라벨링 작업의 클래스를 구별 하기위해 data -> predefined_classes.txt 파일의 내용을 수정함.
- 캔, 플라스틱 두 개의 클래스로 라벨링 작업을 실시함.



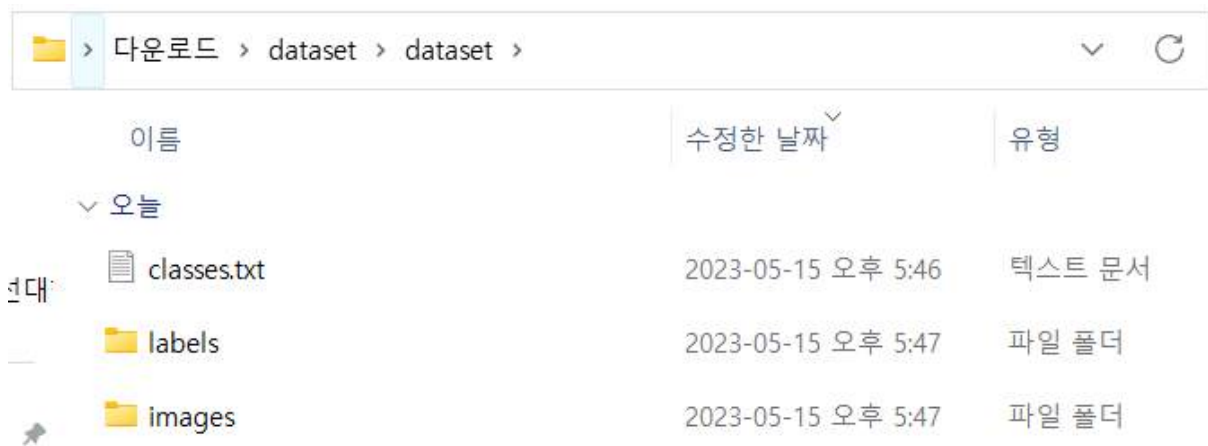
[그림 2-17] labellmg 클래스 수정

- labeling 프로그램을 실행 후, 'YOLO'데이터 파일 형식으로 변경 뒤 라벨링 작업을 실시함
- plastic과 can의 클래스를 선택 후 폴더에 저장하여 image파일과 label파일을 분리함.

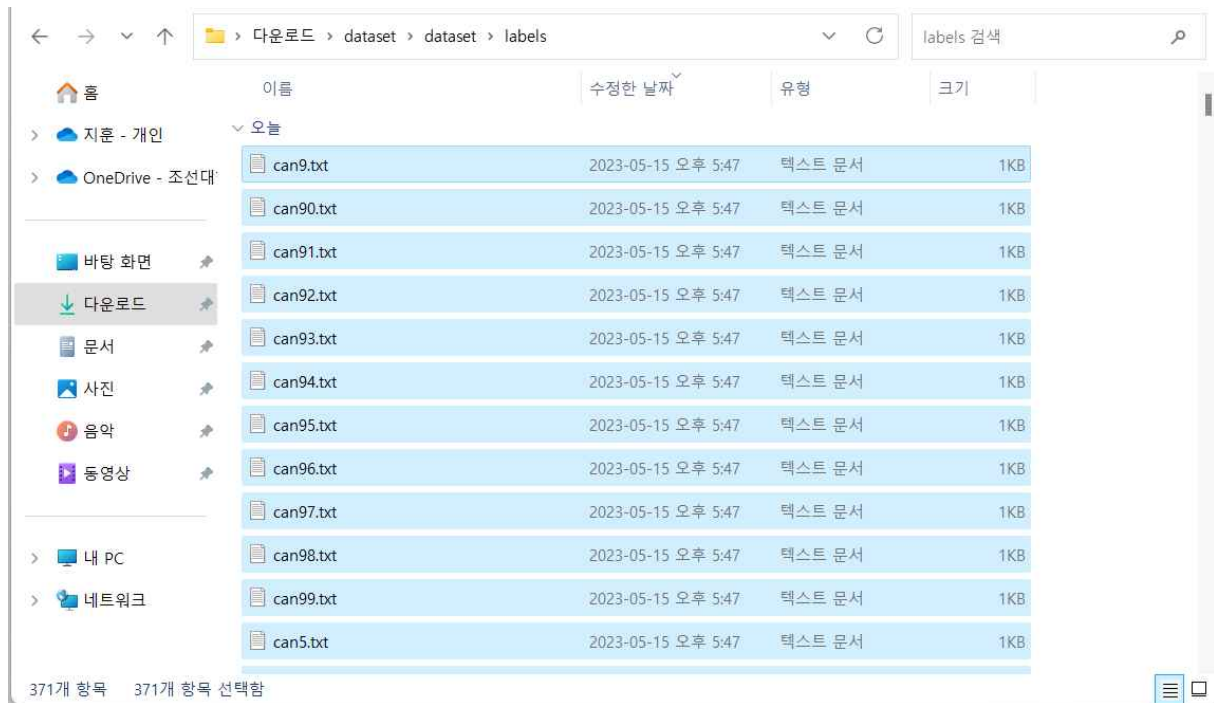


[그림 2-18] 라벨링 작업

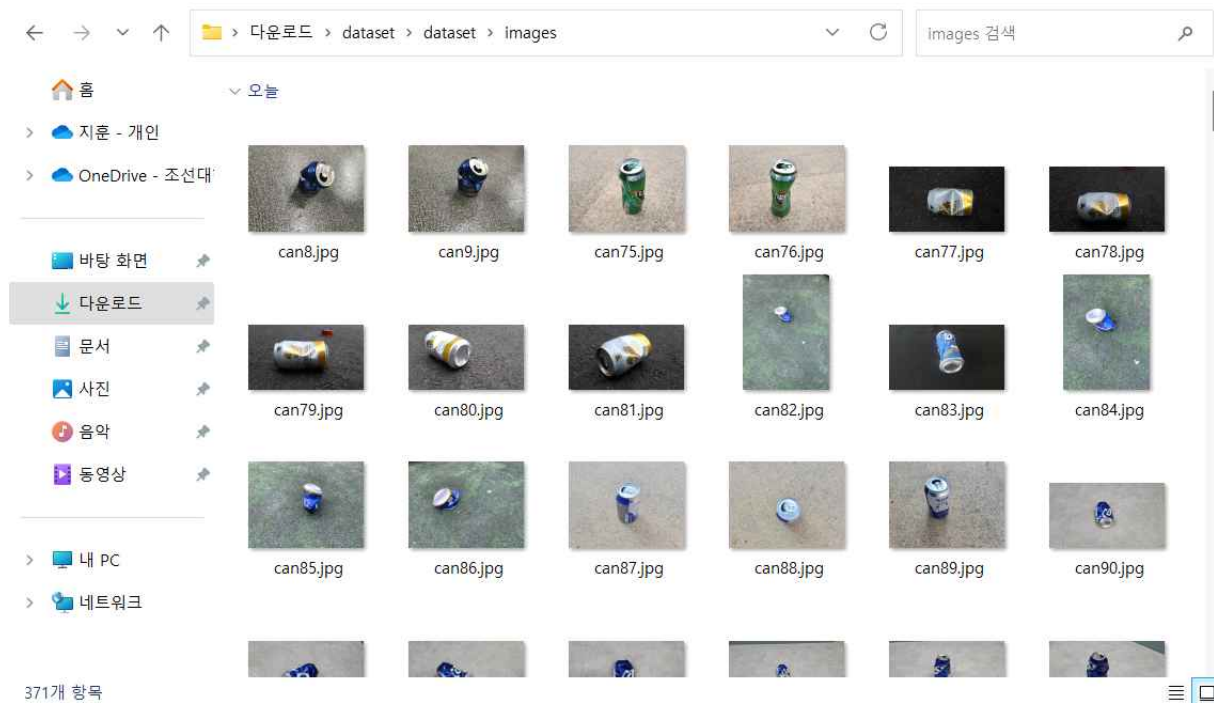
- classes.txt 파일은 사용된 클래스를 설명하는 텍스트 문서임
- labels는 labeling파일을 통해 도출된 label의 txt문서를 저장함
- image는 라벨링하기 위한 원시 이미지 데이터임



[그림 2-19] 데이터셋 파일

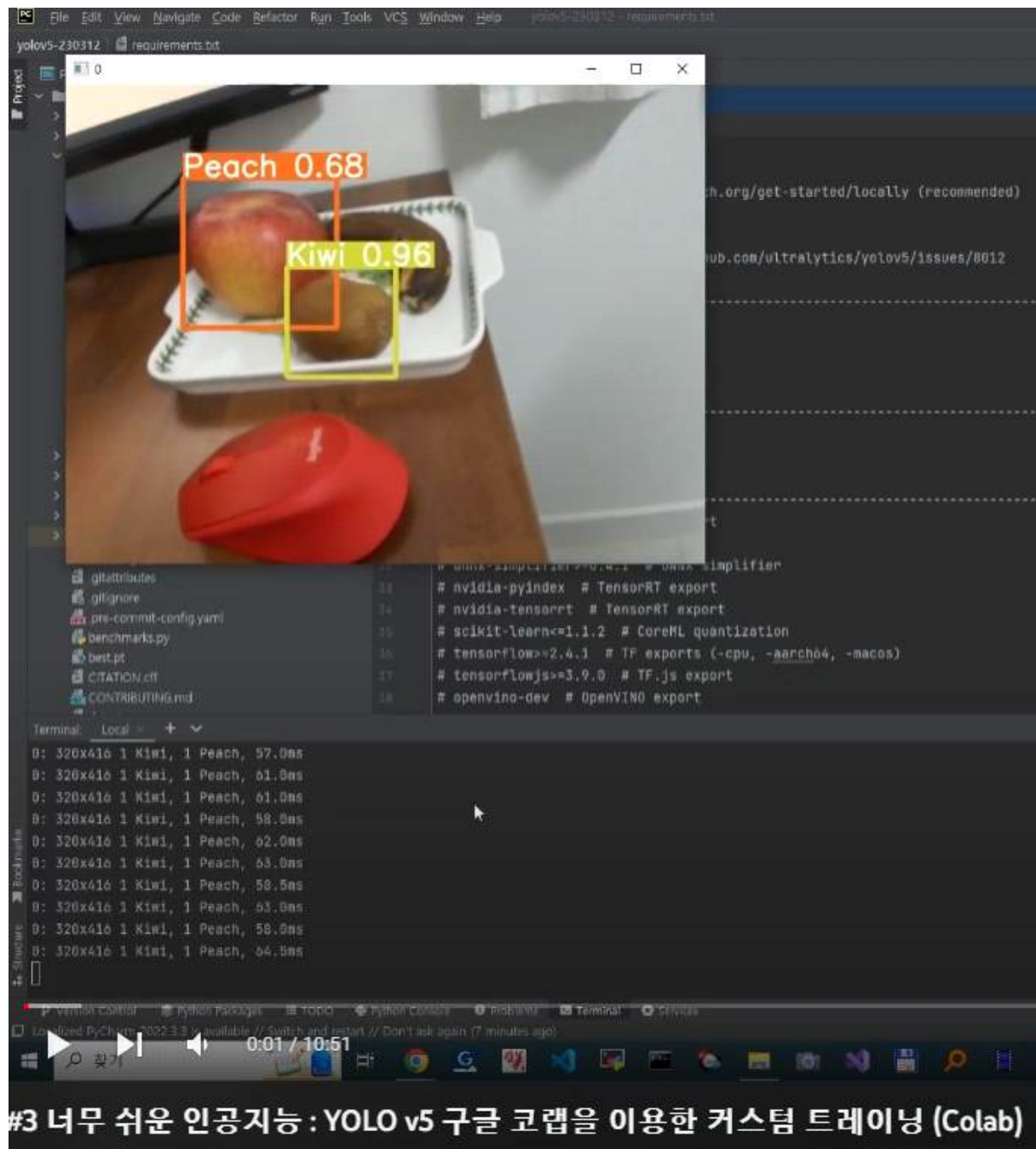


[그림 2-20] 라벨링 txt데이터



[그림 2-21] 캔과 플라스틱 이미지 데이터

- 현재 캔 200개, 플라스틱 171개를 수집하여 완료하였음.
- 각 팀원들이 700개씩 수집하여 총 2100개(캔 1050개, 플라스틱 1050개)의 데이터를 YOLO 모델에 입력하여 1차적으로 분류를 진행할 예정임.
- Yolo v5 시연
 - 데이터셋 구축 후 <https://www.youtube.com/watch?v=U-wkRQ8U3GE>에서 YOLO v5의 영상을 보며 코랩으로 Yolo v5 모델 사용법을 익힘.



[그림 2-22] 구글 코랩을 이용한 커스텀 트레이닝

- roboflow의 synthetic-fruit 데이터셋을 이용하여 과일을 분류하는 yolo모델을 시연

```

[1] pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="bGyz7mKSHey9W2HlmV0")
project = rf.workspace("brad-dwyer").project("synthetic-fruit")
dataset = project.version(10).download("yolov5")

[2] git clone https://github.com/ultralytics/yolov5.git

[10] %cd yolov5

!pip install -r requirements.txt

!python train.py --img 416 --batch 10 --epochs 10 --data ./content/yolov5/data.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt --name fruit_yolov5s_results

train: weights=yolov5s.pt, cfg=./models/yolov5s.yaml, data=./content/yolov5/data.yaml, hyp=./data/hyps/hyp.scratch-low.yaml, epochs=10, batch_size=10, imgsz=416, rec
github: up to date with https://github.com/ultralytics/yolov5
Comet: run 'pip install comet_ml' to automatically track, visualize and remotely train YOLOv5 in Comet
TensorBoard: Start with 'tensorboard --logdir runs/train', view at https://localhost:6006/
Downloading https://ultralytics.com/assets/Arial.ttf to /root/.config/ultralytics/Arial.ttf ...
100% 755k/755k [00:00<00:00, 27.4MB/s]
Downloading https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5s.pt to yolov5s.pt ...
100% 14.1M/14.1M [00:00<00:00, 63.9MB/s]

Overriding model.yaml nc=80 with nc=63

      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]
  
```

[그림 2-23] yolo모델 시연

- 학습 진행 상황을 확인할 수 있음

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
0/9	1.14G	0.06566	0.0385	0.1004	40	416: 100% 1000/1000 [03:40<00:00, 4.54it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 50/50 [00:12<00:00, 4.09it/s]
	all	1000	2803	0.00884	0.755	0.0155 0.00444
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
1/9	1.14G	0.04707	0.03492	0.09724	45	416: 100% 1000/1000 [03:26<00:00, 4.85it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 50/50 [00:10<00:00, 4.79it/s]
	all	1000	2803	0.0068	0.55	0.0104 0.0024
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
2/9	1.14G	0.0442	0.03326	0.09645	62	416: 100% 1000/1000 [03:21<00:00, 4.96it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 50/50 [00:09<00:00, 5.08it/s]
	all	1000	2803	0.0112	0.853	0.0267 0.0087
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
3/9	1.14G	0.0411	0.0321	0.09493	43	416: 100% 1000/1000 [03:23<00:00, 4.92it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 50/50 [00:10<00:00, 4.69it/s]
	all	1000	2803	0.0115	0.891	0.0482 0.0163
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
4/9	1.14G	0.04014	0.03154	0.09146	25	416: 100% 1000/1000 [03:22<00:00, 4.95it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 50/50 [00:09<00:00, 5.04it/s]
	all	1000	2803	0.481	0.204	0.105 0.0425
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
5/9	1.14G	0.03965	0.03112	0.08618	58	416: 100% 1000/1000 [03:20<00:00, 4.99it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 50/50 [00:10<00:00, 4.70it/s]
	all	1000	2803	0.368	0.353	0.159 0.0643
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
6/9	1.14G	0.03879	0.03055	0.08197	37	416: 100% 1000/1000 [03:25<00:00, 4.86it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 50/50 [00:10<00:00, 4.89it/s]
	all	1000	2803	0.336	0.42	0.212 0.0892
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
7/9	1.14G	0.03791	0.03007	0.07886	51	416: 100% 1000/1000 [03:26<00:00, 4.85it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 50/50 [00:10<00:00, 4.92it/s]
	all	1000	2803	0.375	0.44	0.282 0.108

[그림 2-24] 학습 진행과정

- 분류 결과 사과 이미지를 Redcurrant로 0.21확률을 인식하는 것을 확인함
- 데이터베이스를 살펴본 결과 과일 이미지 뿐만 아니라 과일이 아닌 이미지가 많아 질 좋은 데이터로 학습되지 않음을 확인함
- 질 좋은 데이터를 수집하여 YOLO v5시연 경험을 통해 앞으로 캔과 플라스틱을 분류를 수행할 예정임.

