

▶ AI\_TRAIN\_v11.exe 알집 파일을 D:\WAI SVT Training\_mk 경로에서 풀어줘야함

- '1)train\_FRCNN\_res50.bat', 'model\_lib\_v2.py', 'faster\_rcnn\_resnet50\_v1\_800x1333\_batch1.config', 'model\_main\_tf2\_FRCNN\_res50.py', 'model\_main\_tf2.py'

모두 Train\_Tool > 최종본 안에 있는 파일들로 교체해야함

- 2) **CMD** (속성 들어가서) 관리자권한실행 체크박스에 **체크**

- 3) 환경변수 설정

- ex: TensorBoard에 54.8K로 나오면 train\_result의 ckpt도 마지막 기록까지 다 나눔

▶ 훈련툴(GUI) 보수\_06.09~06.12

SVT model prep

Check Training Files

2025.06.12.version

Choose Directory

.bmp to .jpg

Check files

Create Label Map

Change Class Number

Enter class name to be changed

Enter new class name

Change Class Name

기존 라벨명 입력

변경할 라벨명 입력

Change Label Name

Choose Options

Augmentation Info

Horizontal Flip

Vertical Flip

Rotation

CLAHE

Auto Label XML 조정

Enter row number

Enter column number

Row by Column

Color to Gray

BGR to RGB

RGB to BGR

Save

Proceed Training

100.00%

☒ 훈련 종료 후 18시 이후면 PC 종료

Run Training File

Check Training Result

Extract model

Enter checkpoint

Step 54800 → Loss: 0.0010

## ▶ 환경 및 초기화

- 필수 라이브러리 : PyQt6, OpenCV, PIL, TensorFlow, numpy 등 다양한 이미지/딥러닝/GUI관련 패키지
- 관리자 권한 체크 : 관리자 권한으로 설정이 되어있지 않을 시, 자동으로 권한 상승
- Qt 스타일 DLL 자동 제거 및 검사 : 호환성 문제 있는 DLL 삭제 및 경고 출력
- 환경 경로 체크 : 주요 폴더 존재 여부 확인

## ▶ 훈련 스레드(TrainingThread)

- 학습 배치파일 실행 : train\_FRCNN\_res50.bat
- 실시간 Loss/Step 추출 : 콘솔 로그에서 Loss, Step 실시간 파싱
- 최저 Loss 및 Step 팝업창 : 훈련 결과 중 가장 낮은 Loss와 해당 Step 팝업창 띄움
- 훈련 종료 감지 및 후처리 : 종료 메시지 감지 후 PC 종료 또는 GpuNormal 설정 복구
- 딥러닝 연산속도 활성화 : XLA On/Off 에 따른 컴파일러 활성화/비활성화

## ▶ 메인 GUI 클래스(ClassAugChanger)

- PyQt6 기반 GUI : 버튼, 입력창, ProgressBar
- 이미지 증강 기능 :
  - CLAHE, 수평/수직 플립(뒤집기), 수평/수직 회전 이동 등 다양한 증강
  - XML 박스 자동 변형 및 확장
- 클래스/라벨 관리 :
  - 클래스, 라벨링명 변경, 라벨맵 생성, config파일 내 클래스 수 자동 갱신
- 훈련 실행 및 결과 확인 :
  - 훈련 실행, 실시간 Loss 표시, 훈련 완료 시 최적 Loss/Step 팝업창
  - 체크포인트 입력 및 모델 추출

## ▶ 기타

- 이미지 포맷 변환 : bmp -> jpg, BGR to RGB, 컬러 -> 그레이 변환
- 진행상황 **ProgressBar** : 모든 주요작업에 진행률 표시
  - H Flip, V Flip, H Shift, V Shift, 회전, CLAHE, XML 객체 추가, Auto Label XML 조정, 행렬 변환, 훈련 시작
- 모든 입력칸 내용 고정

## ▶ .zip 을 풀었을 시

**[경고] qmodernwindowsstyle (10).dll 메타데이터 이상 감지 (실행 불안정 가능성)**  
라는 오류메시지가 발생하면

- 1. 해당 DLL 삭제
  - (PyQt6/Qt6/plugins/styles 폴더에서qmodernwindowsstyle\*.dll 파일 수동 삭제)
  - 2. GUI가 깨지거나 실행이 안 되면, **PyQt6 삭제 후 재설치**
- 경고만 뜨고 프로그램 정상 작동하면 무시해도 됨

## ▶ 한글화 GUI (06.17)

SVT model prep

Check License

라이선스 확인

MAC address

시리얼번호: 52350396

Check Training Files

2025.09.15

폴더 선택

.bmp to .jpg

파일 확인

라벨 맵 생성

클래스 번호 변경

기존 클래스명:

변경할 클래스명:

클래스명 변경

XLA: ON

Choose Image Augmentation Options

Augmentation Info

수평 뒤집기(H Flip)

수평 이동(H Shift)

회전(Rotation)

수직 뒤집기(V Flip)

수직 이동(V Shift)

CLAHE

XML 객체 추가(단일 이미지에 복수 객체 추가)

Auto Label XML 조정(복수 이미지에 복수 파일 추가)

행 번호:

열 번호:

행렬 변환(Row by Column)

Color to Gray

BGR to RGB

RGB to BGR

Proceed Training

0%

☐

훈련 종료 후 설정된 시각 이후 PC 종료

00:00

훈련 시작

훈련 파일 결과 확인

모델 추출

체크포인트 입력:

☐ x 10

☒ x 20

☐ x 30

XLA: ☒ On ☐ Off

실시간 Loss: -

Smart Vision Tech Inc.

SVT

## ▶ 라이선스 확인

- 라이선스 확인 버튼을 누르고 'Licensed' 가 떠야만 훈련툴 사용 가능
- 'No license' 가 뜨면 시리얼넘버 경로를 확인하라는 팝업창이 생성되며, 훈련툴을 사용할 수 없음
- 시리얼번호는 각 PC마다 고유 번호를 가지고 있음

▶ 폴더 선택

- 이미지 수정 및 훈련을 하기 전 폴더 선택

▶ .bmp to .jpg

- .bmp 이미지를 .jpg 이미지로 변환

▶ 파일 확인

- 이미지와 XML 파일의 개수가 같은지 확인, 그렇지 않으면 짝이 없는 파일을 삭제
- XML 파일에 빈 클래스가 포함되어 있는지 확인, 그렇다면 빈 파일 삭제

▶ 라벨 맵 생성

- 라벨 맵을 만들고 클래스 이름 나열

▶ 클래스 번호 변경

- .config train 파일에서 이전 클래스 번호로 변경

▶ 클래스명 변경

- 각 입력란에 클래스명 입력 후 클래스명 변경 클릭 시 원하는 클래스명으로 변경

▶ XLA: ON/OFF

- model\_main\_tf2\_FRCNN\_res50.py 의 환경변수 ENABLE\_XLA 값에 따라 TensorFlow 의 XLA(JIT) 컴파일러를 활성화 or 비활성화
- *XLA(On)* = *활성화*: 딥러닝 연산속도 빨라질 수 있음

▶ 뒤집기(Flip)

- 수평 뒤집기(H Flip)는 수평으로 뒤집힌 증강 이미지 생성
- 수직 뒤집기(V Flip)는 수직으로 뒤집힌 증강 이미지 생성
- XML 주석도 생성

▶ 이동(Shift)

- 수평 이동(H Shift)은 수평으로 이동된 증강 이미지 생성
- 수직 이동(V Shift)은 수직으로 이동된 증강 이미지 생성
- XML 주석도 생성

▶ 회전(Rotation)

- 일반적으로 객체의 위치가 변경되지 않는 이미지에 사용
- XML 주석도 생성

▶ CLAHE

- 모든 이미지에 CHALE 전처리 적용
- 원본과 동일하므로 XML 파일은 생성되지 않음

▶ XML 객체 추가(단일 이미지에 복수 객체 추가)

- 하나의 이미지 .xml 파일에 object 가 (1pixel = 0.03mm) 씩  
up, down, left, right, expand, shrink 6개 자동 추가 저장

▶ Auto Label XML 조정(복수 이미지에 복수 파일 추가)

- 하나의 이미지와 .xml파일 (1pixel = 0.03mm) 씩  
up, down, left, right, expand, shrink (JPG, XML) \* 6 자동 추가 저장

▶ 행렬 변환(Row by Column)

- 이미지를 행과 열 단위로 나누고, 한 단계씩 이동
- XML 주석도 생성
- 이미지 크기는 행/열 단위로 나눌 수 있어야 됨

▶ Color to Gray

- 흑백 이미지 폴더 생성

#### ▶ BGR to RGB

- BGR 에서 변환된 RGB 파일 폴더 생성

#### ▶ RGB to BGR

- RGB 에서 변환된 BGR 파일 폴더 생성

#### ▶ 훈련 시작

- 훈련 파일을 실행하고 Chrome 브라우저에서 텐서보드 실행
- 버튼 클릭 시 비활성화됨, 실수로 훈련 버튼 또 누를 일 없음
- 훈련이 끝나면 자동으로 ProgBar 100% 되고, '훈련 완료' 팝업창 뜸

#### ▶ 훈련 파일 결과 확인

- D:\WAI\_SVT\_Training\_mkWtrain\_result 폴더 있으면 '훈련 완료' 팝업창
- D:\WAI\_SVT\_Training\_mkWtrain\_result 폴더 없으면 '훈련 완료X' 팝업창

#### ▶ 모델 추출

- Ckpt 입력 시 숫자 저장 및 Run Training File 버튼 비활성화
- 숫자입력칸 밑에 Checkpoint is updated: ckpt-\*\*\* 뜸

#### ▶ 시간을 설정하고 PC 종료 **체크박스**를 누른 뒤 훈련을 진행하면 PC 강제 종료

- 단, "훈련 종료 후 설정된 시각 이후 PC 종료" 체크박스를 선택하지 않거나,
- 체크박스를 선택하더라도 설정 시각 이전에 훈련이 종료됐다면 PC는 **종료되지 않음**
- 또 강제종료가 되지 **않은** PC는 **GpuNormal.bat 자동 실행**



### ▶ Python RadioButton

- EpochNo 로 annos의 jpg \* 20 값 고정
- RadioButton x 10, x20, x30 누르면 그 값에 맞게 곱해져서 자동 저장
- 최소 10,000step, 최대 300,000step 이며, 그 사이 훈련 step 값은 Epoch.No.txt 값 고정