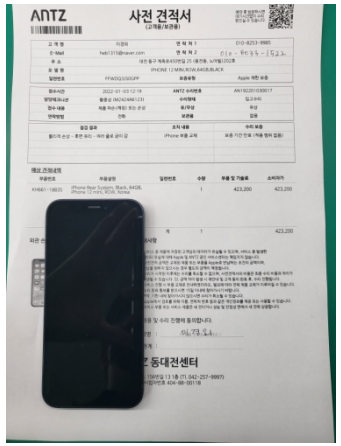
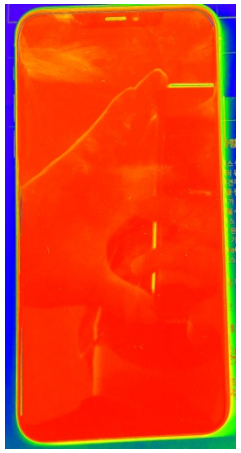
모델 작업

이미지 가공



원본 데이터

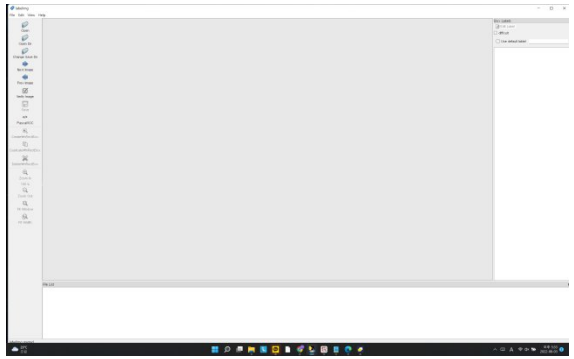
1차 이미지 가공



2차 이미지 가공

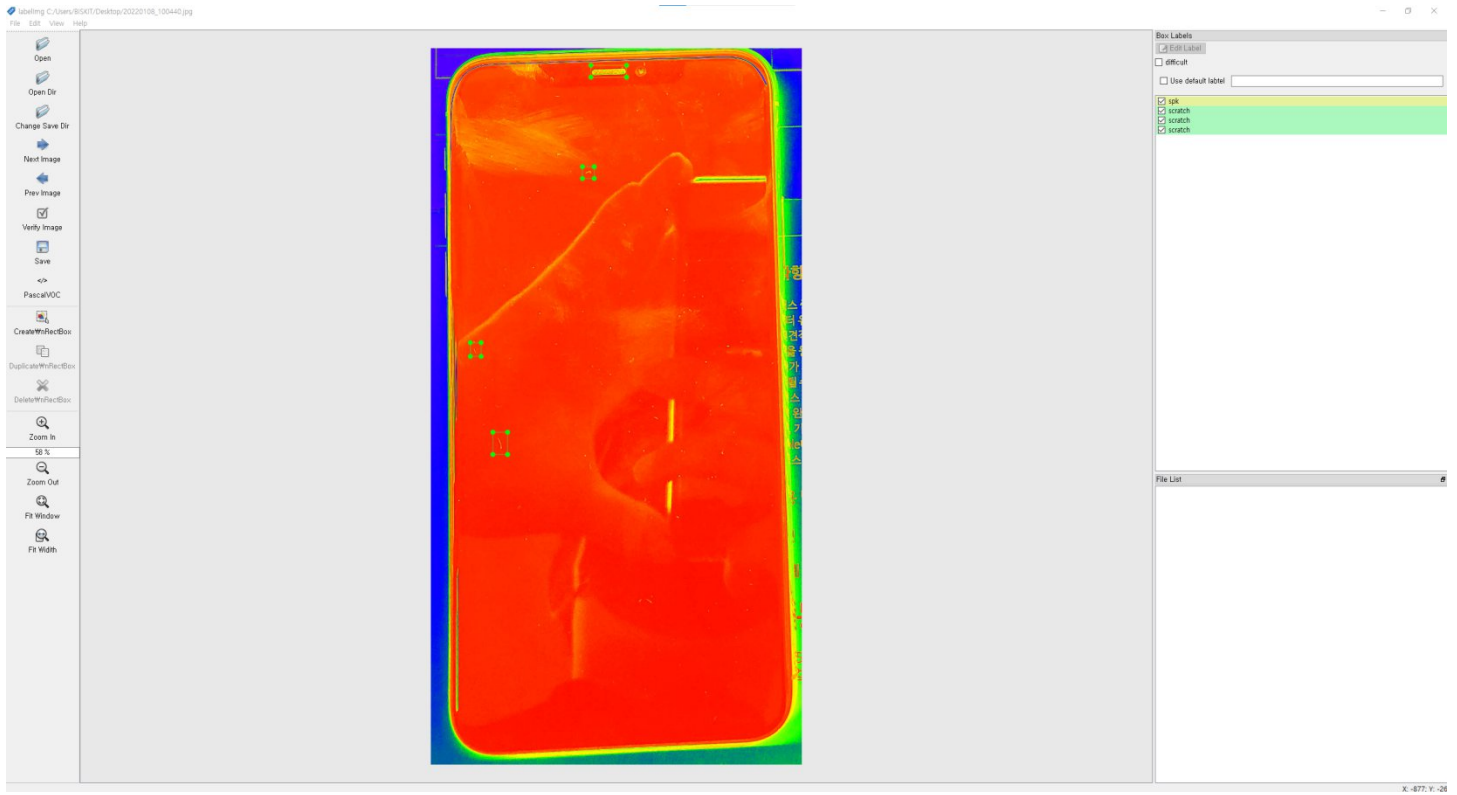
Label 작업

Classes 정보
- Crack : 깨짐
- Scratch : 긁힘
- Spk: 스피커



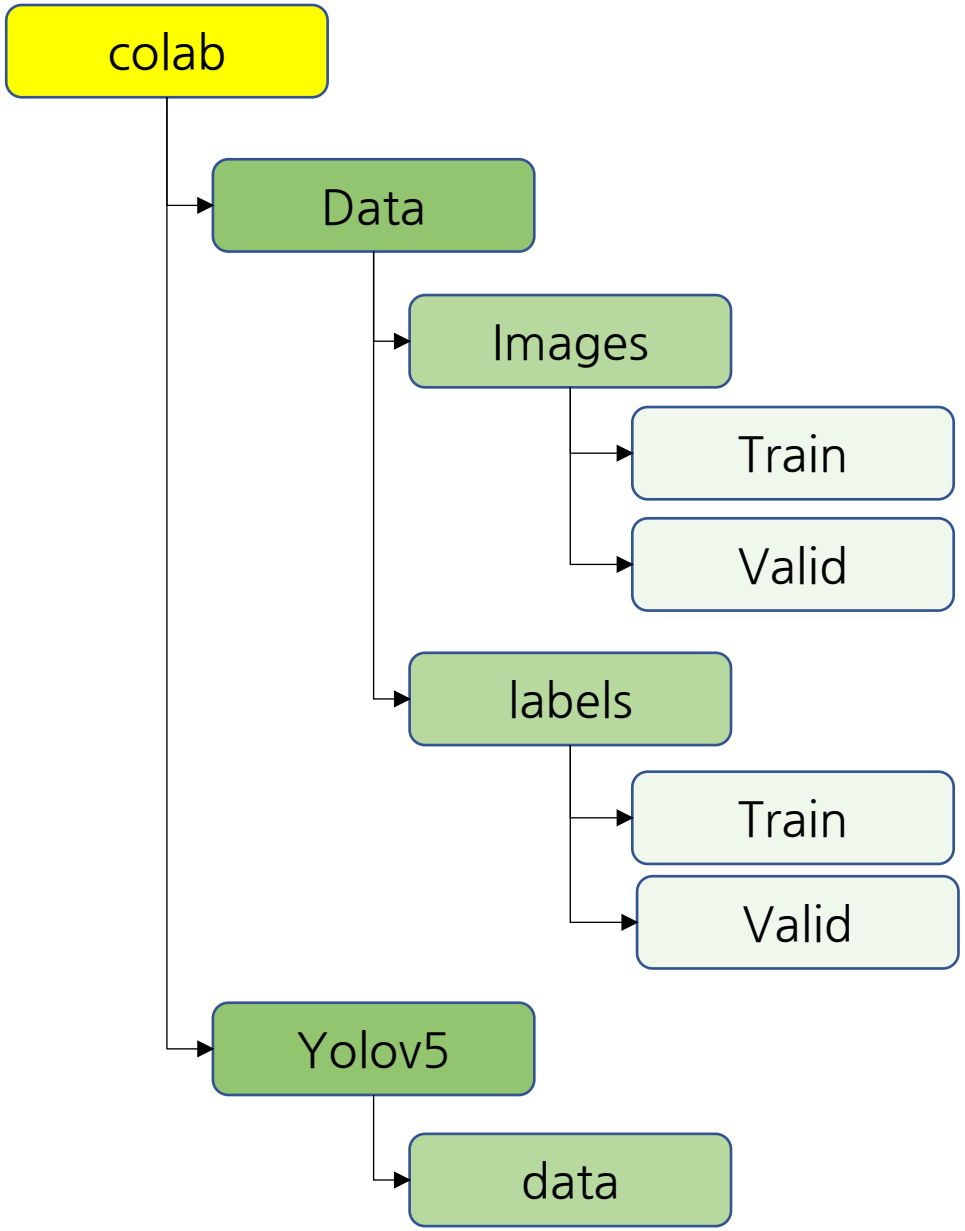
labelimg 프로그램을 이용하여 box label작업

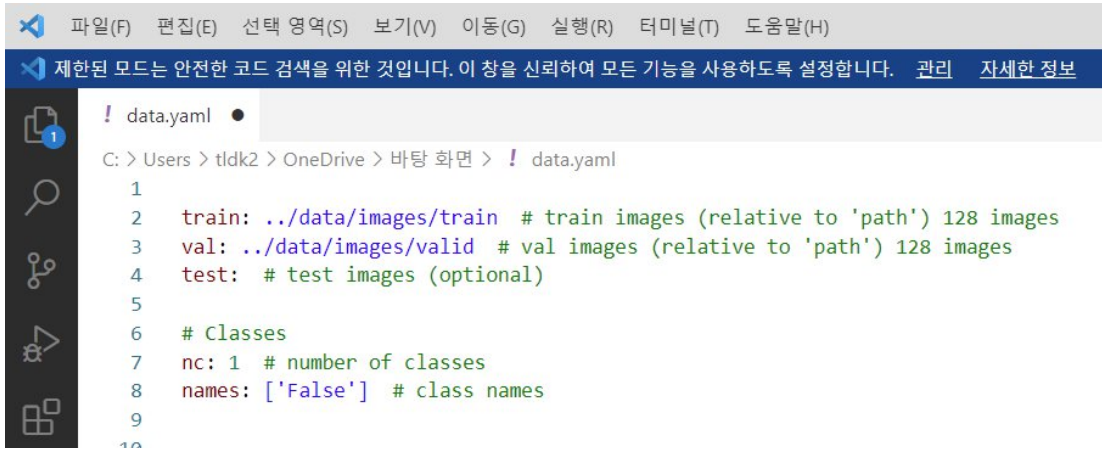
Label 작업 후 이미지/txt 파일로 저장



Dataset

학습 정보
전체 이미지 : 3109장
Train : 2817장
Val : 292장





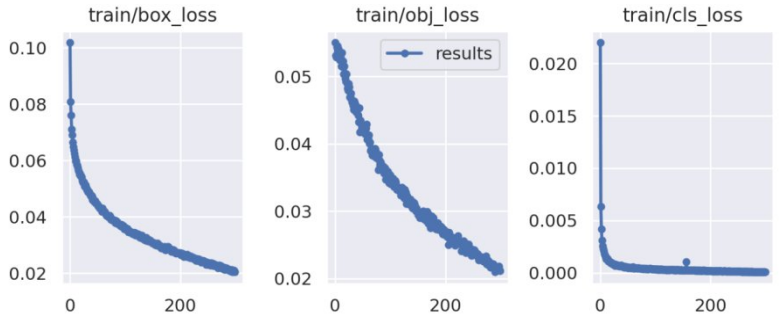
Data.yaml 제작
- train 파일 위치 설정
- val 파일 위치 설정
- Nc classes 개수 설정
- Classes name 설정

(Yolov5 data 파일에 소스 저장)

Model train

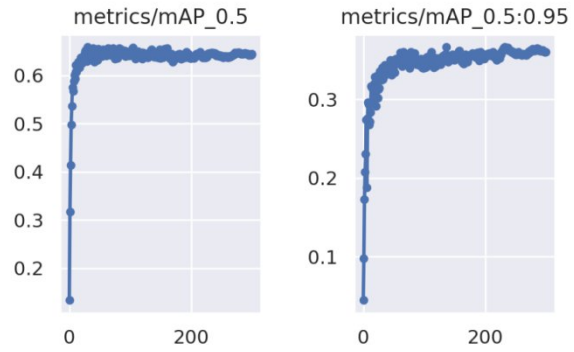
모델 학습 시 하이퍼 파라미터를 수정
Batch, Epochs값 등을 조절하면서 학습을 진행하였습니다. Batch, epochs값 등을 조절하는 이유는 과적합 방지를 하기 위해서 입니다.
(과적합 이란 새로운 데이터에 일반화되지 않는 상황을 의미합니다.)
사용할 모델은 다양한 이미지가 들어오기 때문에 과적합 방지를 위해 Batch, Epochs값 등을 수정하면서 학습을 진행하였습니다.

※ 학습 데이터에 따라 batch, epochs 등 지정하는 값이 달라지며 라벨 방식에 따라 학습되는 값도 변화가 있을 수 있습니다.

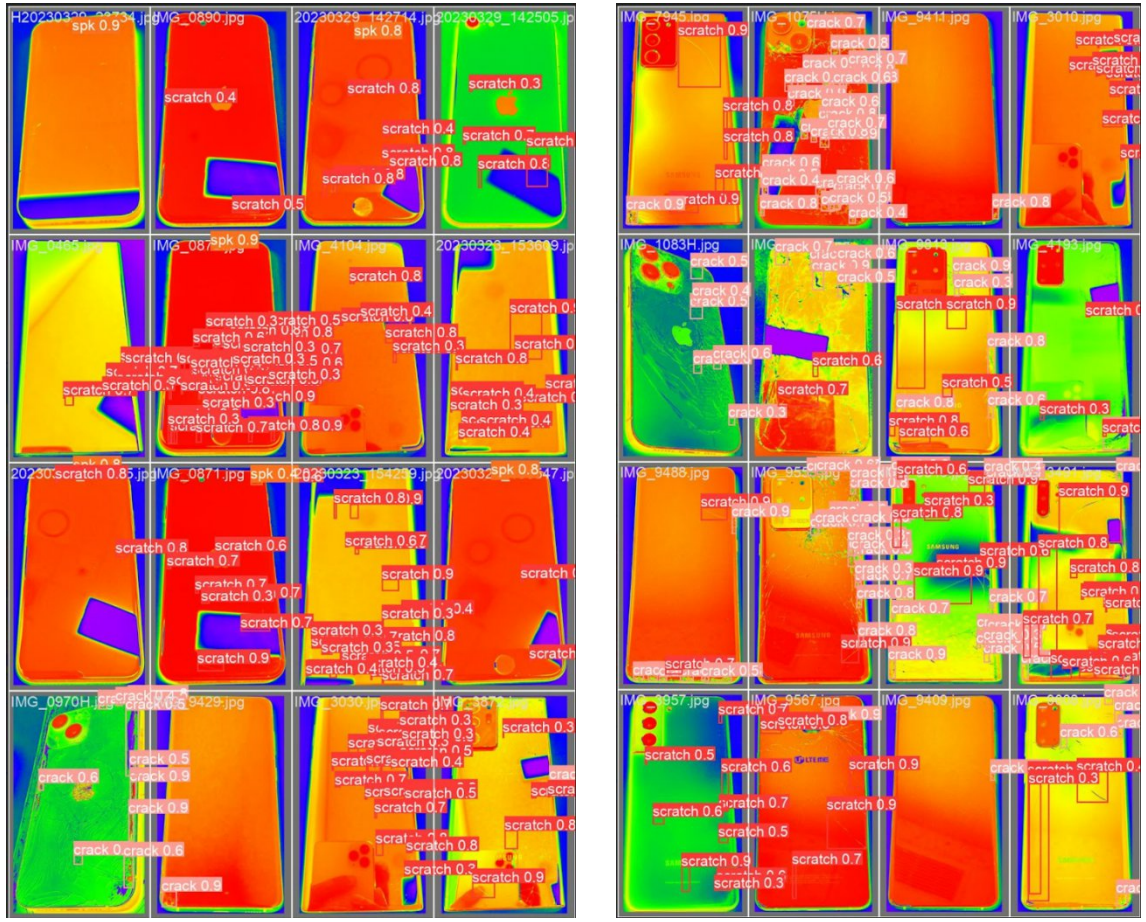


Batch, epochs값 등 수정했을 때 모델 검증 결과 자료

```
YOLOv5x summary: 322 layers, 86186872 parameters, 0 gradients, 203.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 10/10 [00:06<00:00,  1.49it/s]
                   all        293       2162        0.8      0.722      0.738      0.508
               scratch        293       1948       0.57      0.448      0.447      0.193
                 crack        293        159      0.848       0.78      0.816      0.658
                   spk        293         55      0.981      0.938      0.951      0.674
```



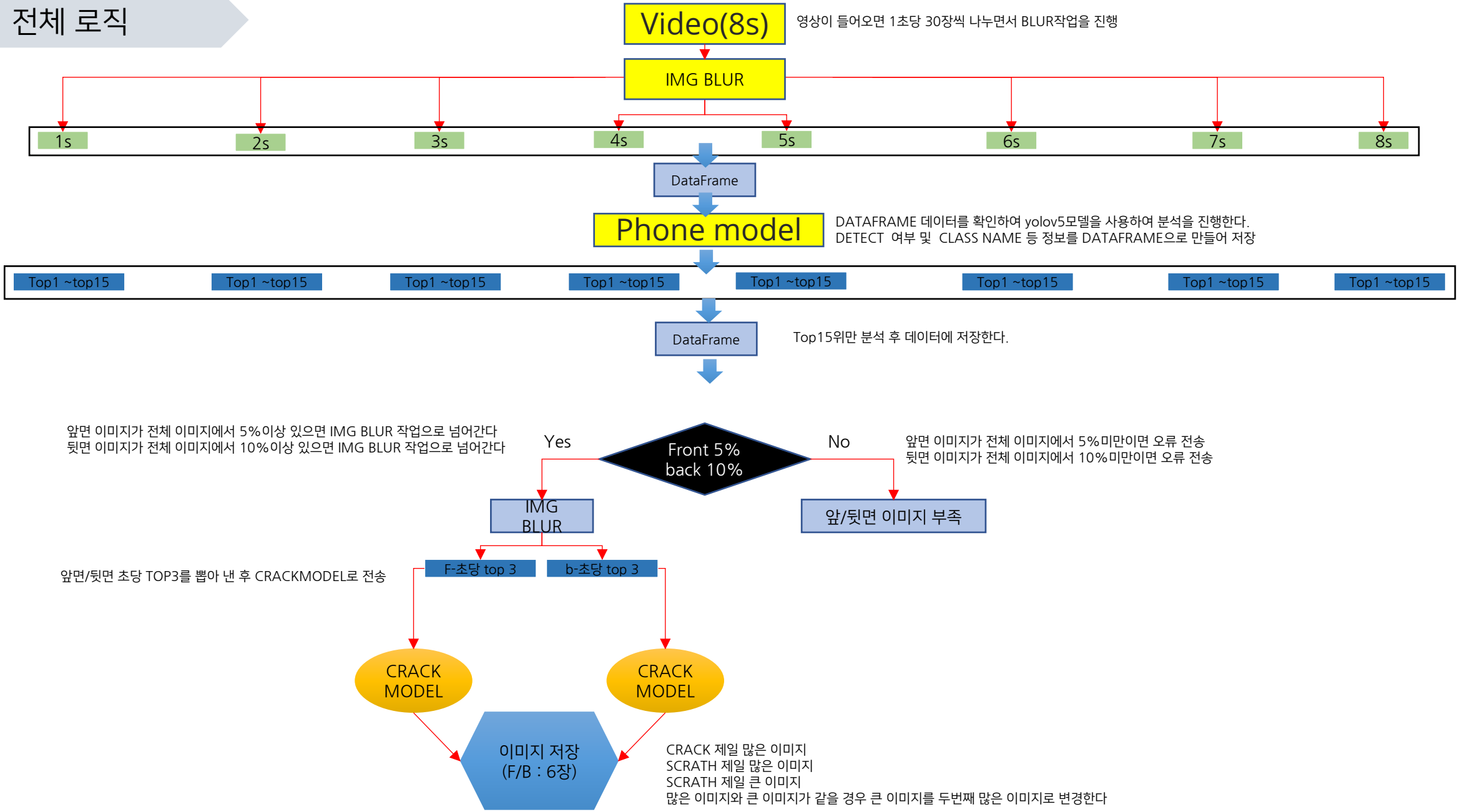
모델 검증 데이터



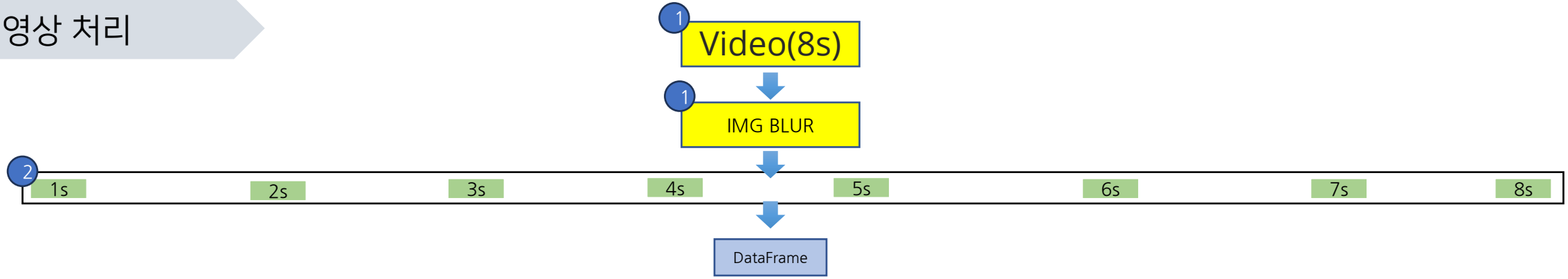
검증데이터 결과 값

외관 분석 코드 리뷰

전체 로직


Video(8s)
영상이 들어오면 1초당 30장씩 나누면서 BLUR작업을 진행
IMG BLUR
1s
2s
3s
4s
5s
6s
7s
8s
DataFrame
Phone model
DATAFRAME 데이터를 확인하여 yolov5모델을 사용하여 분석을 진행한다.
DETECT 여부 및 CLASS NAME 등 정보를 DATAFRAME으로 만들어 저장
Top1 ~top15
Top1 ~top15
Top1 ~top15
Top1 ~top15
Top1 ~top15
Top1 ~top15
Top1 ~top15
Top1 ~top15
DataFrame
Top15위만 분석 후 데이터에 저장한다.
앞면 이미지가 전체 이미지에서 5%이상 있으면 IMG BLUR 작업으로 넘어간다
뒷면 이미지가 전체 이미지에서 10%이상 있으면 IMG BLUR 작업으로 넘어간다
Yes
Front 5%
back 10%
No
앞면 이미지가 전체 이미지에서 5%미만이면 오류 전송
뒷면 이미지가 전체 이미지에서 10%미만이면 오류 전송
IMG
BLUR
앞/뒷면 이미지 부족
앞면/뒷면 초당 TOP3를 뽑아 낸 후 CRACKMODEL로 전송
F-초당 top 3
b-초당 top 3
CRACK
MODEL
CRACK
MODEL
이미지 저장
(F/B : 6장)
CRACK 제일 많은 이미지
SCRATH 제일 많은 이미지
SCRATH 제일 큰 이미지
많은 이미지와 큰 이미지가 같을 경우 큰 이미지를 두번째 많은 이미지로 변경한다

영상 처리



1

영상이 들어오고 영상을 이미지로 변경하여 blur처리하는 구간
영상이 들어왔을 때 opencv를 사용하여 이미지로 변경하여 이미지 수량 및 blur체크한다.

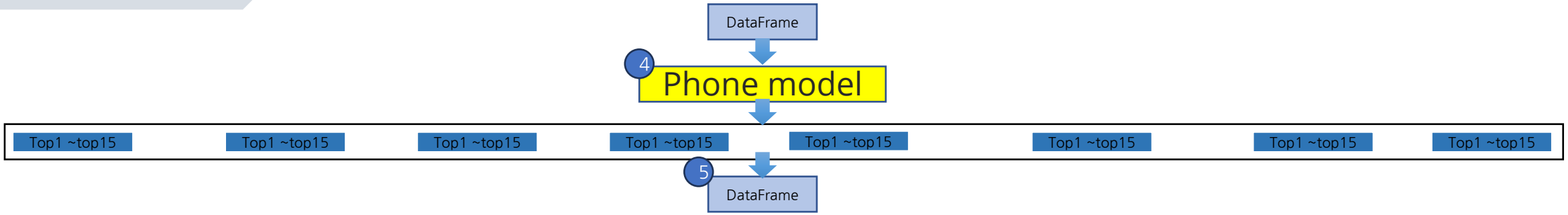
Blur 설명 : 분산(픽셀 값의) 값은 이미지의 선명도나 경계 부분을 나타내는데 사용될 수 있습니다.
분산이 클수록 이미지에서 높은 주파수 성분이 많다는 의미이므로 이미지의 선명도가 높거나 경계가 확실하게 표시되는 특징을 가지고 있을 가능성이 높다.
모델에 이미지를 넣기 전 화질이 좋은 이미지를 체크하기 위해 위와 같은 기술을 사용하여 blur체크를 하는 방식이다.

2

Fps계산하는 코드 및 이미지 초 구별하는 구간
- 이미지가 생성되는 시간을 체크하여 저장한다.

ORIGINAL_IMG	ORIGINAL_TIME	ORIGINAL_BLURS	TRUE_TIME	ORIGINAL_BLUR_RANK	UP1	UPL	UP2	UPR	UP3	BK1	BKL	BK2	BKR	BK3	START	END	BLACK	FLIP
1	0	1.037508621	29.44279094	28	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12	1.69E+12
2	0	1.001781382	29.44279094	30														
3	0	1.021306521	29.44279094	29														
4	0	1.058230092	29.44279094	27														
5	0	1.72303986	29.44279094	26														
6	0	6.01220635	29.44279094	25														
7	0	21.86669922	29.44279094	23														
8	0	37.49758468	29.44279094	22														
9	0	46.47723551	29.44279094	18														
10	0	58.35910322	29.44279094	15														
11	0	69.37192129	29.44279094	13														
12	0	74.80464232	29.44279094	8														
13	0	77.90905396	29.44279094	4														
14	0	72.74878332	29.44279094	11														
15	0	75.44453172	29.44279094	6														
16	0	72.8580565	29.44279094	10														
17	0	66.86359519	29.44279094	14														
18	0	75.99466206	29.44279094	5														
19	0	74.30665069	29.44279094	9														
20	0	77.92734127	29.44279094	3														
21	0	71.14285679	29.44279094	12														
22	0	53.60307041	29.44279094	16														
23	0	44.64048452	29.44279094	21														
24	0	45.73886827	29.44279094	19														
25	0	44.81558588	29.44279094	20														
26	0	51.38211788	29.44279094	17														
27	0	87.76895133	29.44279094	1														
28	0	85.60452527	29.44279094	2														
29	0	75.12943317	29.44279094	7														
30	0	17.82545493	29.44279094	24														
31	1	9.235161946	29.44279094	29														
32	1	16.15103029	29.44279094	24														
33	1	19.51700824	29.44279094	21														
34	1	22.32975616	29.44279094	17														
35	1	17.94410458	29.44279094	22														
36	1	12.03576916	29.44279094	28														

Phone model



4

Classes가 [[phone],[phone2], [person, phone]]라벨만 저장하는 구간

특정라벨을 지정하여 저장하는 방식으로 구현되어 있습니다.
스마트폰에 사람이 반사되었을 때 사람 이미지는 삭제하고 스마트폰 이미지만 저장하는 방식으로 구현되었습니다.

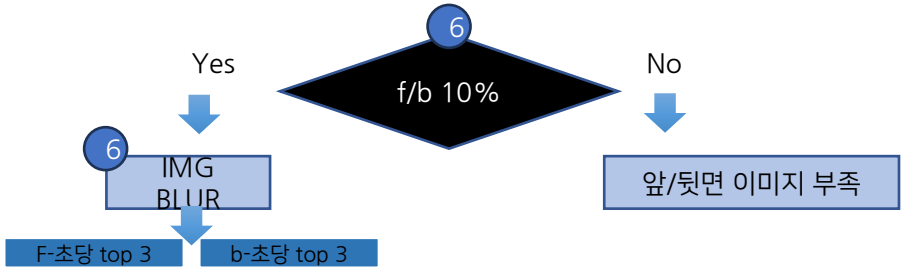
5

앞면 뒷면 계산 코드 및 데이터 저장 코드

영상을 촬영할 때 정면, 우측, 좌측, FLIP 등 자이로 값에 따라 움직이는 시간을 DB에 저장하여 원하는 시간 이미지와 FRONT_BACK구간 이미지를 저장하는 방식이다

IMG	IMG_NAME	VIDEO_TIME	DETECT	DETECT_NAME	FRONT_BACK	BLUR_TIME
1	1	1.69105E+12	1	cell phone	del_data	0
2	2	1.69105E+12	1	cell phone	del_data	0
3	3	1.69105E+12	1	cell phone	del_data	0
4	4	1.69105E+12	1	cell phone	del_data	0
5	5	1.69105E+12	1	cell phone	del_data	0
6	6	1.69105E+12	1	cell phone	del_data	0
7	7	1.69105E+12	1	cell phone	del_data	0
8	8	1.69105E+12	1	cell phone	del_data	0
9	9	1.69105E+12	1	cell phone	del_data	0
10	10	1.69105E+12	1	cell phone	del_data	0
11	11	1.69105E+12	1	cell phone	del_data	0
12	12	1.69105E+12	1	cell phone	del_data	0
13	13	1.69105E+12	1	cell phone	del_data	0
14	14	1.69105E+12	1	cell phone	del_data	0
15	15	1.69105E+12	0	person,cell phone	del_data	0
15	15	1.69105E+12	1	person,cell phone	del_data	0
16	16	1.69105E+12	0	person,cell phone	del_data	0
16	16	1.69105E+12	1	person,cell phone	del_data	0
17	17	1.69105E+12	1	cell phone	del_data	0
18	18	1.69105E+12	0	person,cell phone	del_data	0
18	18	1.69105E+12	1	person,cell phone	del_data	0
19	19	1.69105E+12	0	person,cell phone	del_data	0
19	19	1.69105E+12	1	person,cell phone	del_data	0
20	20	1.69105E+12	1	cell phone	del_data	0
21	21	1.69105E+12	1	cell phone	del_data	0
22	22	1.69105E+12	0	person,cell phone	del_data	0
22	22	1.69105E+12	1	person,cell phone	del_data	0
23	23	1.69105E+12	0	person,cell phone	del_data	0
23	23	1.69105E+12	1	person,cell phone	del_data	0
24	24	1.69105E+12	0	person,cell phone	del_data	0
24	24	1.69105E+12	1	person,cell phone	del_data	0
25	25	1.69105E+12	0	person,cell phone	del_data	0
25	25	1.69105E+12	1	person,cell phone	del_data	0
26	26	1.69105E+12	0	person,cell phone	del_data	0
26	26	1.69105E+12	1	person,cell phone	del_data	0
27	27	1.69105E+12	0	person,cell phone	del_data	0
27	27	1.69105E+12	1	person,cell phone	del_data	0
28	28	1.69105E+12	0	person,cell phone	del_data	0
28	28	1.69105E+12	1	person,cell phone	del_data	0
29	29	1.69105E+12	0	person,cell phone	del_data	0
29	29	1.69105E+12	1	person,cell phone	del_data	0
30	30	1.69105E+12	0	person,cell phone	del_data	1
30	30	1.69105E+12	1	person,cell phone	del_data	1
31	31	1.69105E+12	1	person,cell phone	del_data	1
31	31	1.69105E+12	0	person,cell phone	del_data	1

데이터 전처리



6

이미지 수량 체크 구간
전체 이미지에서 앞면 이미지가 평균 5%이상이면 2차 BLUR작업 진행한다.
전체 이미지에서 뒷면 이미지가 평균 10%이상이면 2차 BLUR작업 진행한다.

Blur작업 구간
이미지 초점이 맞으면 crack모델로 가져가는 방식이며, opencv2에서 제공하는 라이브러리로 이미지가 초점이 얼마나 맞는지 수치로 나타내는 기술이며, 평균값을 지정하여 초당 top3 이미지를 저장하는 방식입니다.

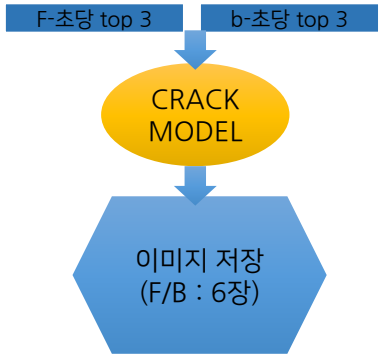
Front DataFrame

IMG	IMG_NAM	VIDEO_TIM	DETECT	DETECT_N	FRONT_BA	BLUR_TIM	BLUR	BLUR_RAN
226	226	1.69E+12	1	cell phone	front	9	3.633685	2
227	227	1.69E+12	1	cell phone	front	9	3.675212	1
228	228	1.69E+12	1	cell phone	front	9	3.579925	3
240	240	1.69E+12	1	cell phone	front	10	2.908086	1
241	241	1.69E+12	1	cell phone	front	10	2.77977	2
282	282	1.69E+12	1	cell phone	front	11	2.884842	1
305	305	1.69E+12	1	cell phone	front	12	2.863255	2
306	306	1.69E+12	1	cell phone	front	12	3.037133	1
307	307	1.69E+12	1	cell phone	front	12	2.82057	3
317	317	1.69E+12	1	cell phone	front	13	3.379746	2
318	318	1.69E+12	1	cell phone	front	13	3.274275	3
331	331	1.69E+12	1	cell phone	front	13	3.412793	1
337	337	1.69E+12	1	cell phone	front	14	3.499246	3
338	338	1.69E+12	1	cell phone	front	14	3.90104	2
339	339	1.69E+12	1	cell phone	front	14	4.02259	1
373	373	1.69E+12	1	cell phone	front	15	3.362393	2
378	378	1.69E+12	1	cell phone	front	15	3.713997	1
379	379	1.69E+12	1	cell phone	front	15	3.006247	3
417	417	1.69E+12	1	cell phone	front	17	3.099019	1
548	548	1.69E+12	1	cell phone	front	22	3.615588	2
549	549	1.69E+12	1	cell phone	front	22	3.690269	1
550	550	1.69E+12	1	cell phone	front	22	3.567448	3
565	565	1.69E+12	1	cell phone	front	23	3.728112	2
566	566	1.69E+12	1	cell phone	front	23	3.767294	1
572	572	1.69E+12	1	cell phone	front	23	3.673412	3
588	588	1.69E+12	1	cell phone	front	24	3.755799	1

Back DataFrame

	IMG	IMG_NAME	VIDEO_TIME	DETECT	DETECT_NAME	FRONT_BACK	BLUR_TIME	BLUR	BLUR_RANK
0	797	797	1.6914E+12	1	cell phone	back	33	2.738537072	1
1	798	798	1.6914E+12	1	cell phone	back	33	2.710819027	2
2	799	799	1.6914E+12	1	cell phone	back	33	2.654091332	3
3	829	829	1.6914E+12	1	cell phone	back	34	2.919197931	3
4	833	833	1.6914E+12	1	cell phone	back	34	3.265345568	1
5	836	836	1.6914E+12	1	cell phone	back	34	3.152633596	2
6	854	854	1.6914E+12	1	cell phone	back	35	3.466266875	2
7	857	857	1.6914E+12	1	cell phone	back	35	3.347603639	3
8	860	860	1.6914E+12	1	cell phone	back	35	3.830283815	1
9	864	864	1.6914E+12	1	cell phone	back	36	4.228622758	3
10	865	865	1.6914E+12	1	cell phone	back	36	4.368722197	2
11	866	866	1.6914E+12	1	cell phone	back	36	4.483851163	1
12	1016	1016	1.6914E+12	1	cell phone	back	42	3.696944324	3
13	1017	1017	1.6914E+12	1	cell phone	back	42	3.730098337	2
14	1030	1030	1.6914E+12	1	cell phone	back	42	3.739862268	1
15	1034	1034	1.6914E+12	1	cell phone	back	43	3.799642347	3
16	1036	1036	1.6914E+12	1	cell phone	back	43	3.819557896	2
17	1037	1037	1.6914E+12	1	cell phone	back	43	3.889389776	1
18	1056	1056	1.6914E+12	1	cell phone	back	44	3.815692227	2

Crack model



Crack model 추론 구간
초당 top3에 대한 데이터를 가지고 crack model를 사용한다.
이미지가 model에 들어가기 전에 image processing 작업을 진행한다.
(ios/android 기종마다 image processing 방식이 달라짐)
- 앞면 : 이미지 밝기를 높여 미세 기스를 잘 보이게 변화 후 추론을 시작한다.
- 뒷면 : 여러가지 색상을 가지고 있어 GRAYSCALE로 변화 후 추론을 시작한다.

image processing

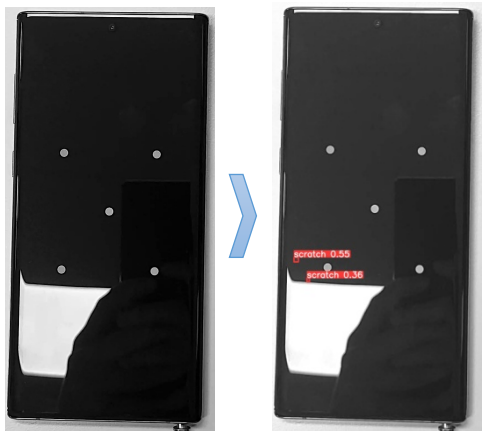


Image processing 전 / Image processing 후

Image processing 전 / Image processing 후

Crack model 추론 완료 후 저장되는 이미지 자료


FRONT_USER_CRACK


FRONT_USER_SCRATCH


FRONT_USER_SCRATCH_SIZE

BACK_USER_CRACK

BACK_USER_SCRATCH

BACK_USER_SCRATCH_SIZE

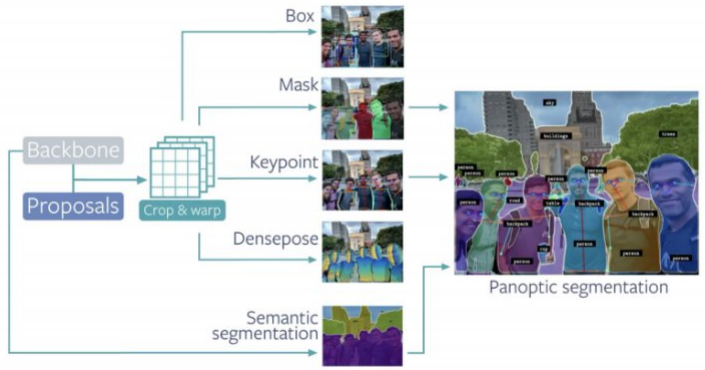
Crack model 추론 이미지 저장
원본 이미지에서 박스 좌표 값을 이용하여 박스를 그려주는 방식을 사용한다.
Classes 정보는 영어에서 한국어로 표현하는 방식은 PIL라이브러리를 사용하고 있다
{앞면/뒷면}_USER_CRACK : CRACK 제일 많은 이미지
{앞면/뒷면}_USER_SCRATCH : SCRATH 제일 많은 이미지
{앞면/뒷면}_USER_SCRATCH_SIZE : SCRATH 제일 큰 이미지
많은 이미지와 큰 이미지가 같을 경우 큰 이미지를 두번째 많은 이미지로 변경한다

새로 적용 예정 모델 정보
- 내부 협의 필요!!

Detectron2

Detectron2 설명
Facebook AI Research(FAIR)에서 개발한 오픈 소스 컴퓨터 비전 라이브러리입니다.
이 라이브러리는 객체 감지(Object Detection) 및 인스턴스 분할(Instance Segmentation)과 같은 컴퓨터 비전 작업을 위한 풍부한 기능을 제공합니다.
Detectron2는 기존의 Detectron 라이브러리를 대폭 개선한 버전으로, 더 빠르고 유연한 구조를 가지고 있습니다
Faster R-CNN, Mask R-CNN, RetinaNet 및 DensePose와 같은 원래 Detectron에서 사용할 수 있었던 모든 모델과
Cascade R-CNN, Panoptic FPN 및 TensorMask를 포함한 일부 최신 모델이 포함되었습니다.

DATA label 방식
Yolov5 처럼 박스 라벨 방식이 아닌 Create Polygons방식으로 데이터를 생성한다.



추가된 기술

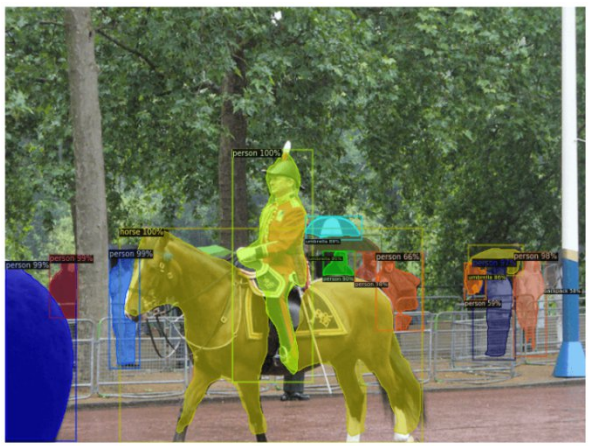
Detectron의 개선 사항
기계 학습 프레임워크: 원래 탐지는 Caffe2로 작성되었지만 Detectron2는 PyTorch로 전환했습니다. 이를 통해 개발자는 훨씬 더 직관적인 접근 방식을 사용하여 모델을 테스트하고 변경할 수 있습니다.

모듈식 설계: Detectron2의 모듈식 설계는 가능한 최대의 사용자 정의 가능성을 허용합니다. 개발자는 개체 감지 모델의 모든 부분에 대한 사용자 지정 구현을 작성할 수 있으므로 프로젝트 내에서 더 큰 유연성을 얻을 수 있습니다.

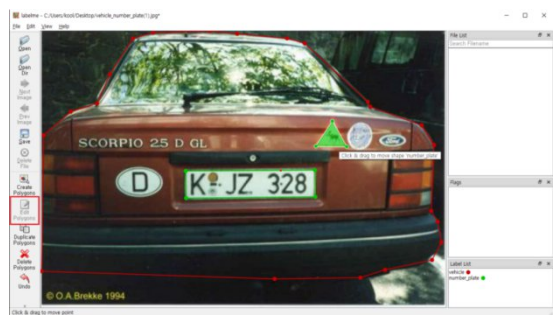
새로운 모델: Detectron2에는 Cascade R-CNN, Panoptic FPN 및 TensorMask와 같은 새로운 모델이 포함되어 있습니다.

모델을 프로덕션으로 가져오기: 이제 Detectron2에는 Detectron2go라는 추가 레이어가 포함되어 있습니다. 이를 통해 클라우드 및 모바일 장치에 대한 모델 변환, 네트워크 양자화 및 표준 교육 워크플로를 포함한 기능을 통해 최첨단 모델을 프로덕션에 더 쉽게 가져올 수 있습니다.

속도: Detectron2의 전체 훈련 파이프라인이 GPU로 이동되어 Detectron2가 훨씬 빨라졌습니다.



추론 결과



Create Polygons방식 예시

추가 정보 링크 입니다.
https://blog.roboflow.com/how-to-train-detectron2/

Detectron2 VS yolov5

Detectron2 와 YOLOv5 속도 비교

•Detectron2:
- Faster R-CNN 모델 기준으로, 일반적으로 20-30 FPS 정도의 속도를 가질 수 있습니다. 그러나 정확도가 높아 상대적으로 높은 계산 비용이 듭니다.

•YOLOv5:
- YOLOv5 모델은 상대적으로 가볍고 최적화되었기 때문에, 일반적으로 60 FPS 이상의 빠른 속도를 제공합니다. 이는 높은 프레임 속도로 실시간 객체 감지가 가능하다는 것을 의미합니다.

Detectron2 와 YOLOv5 정확도 비교

Detectron2:
• Detectron2는 다양한 모델을 제공하며, Faster R-CNN, Mask R-CNN, RetinaNet 등이 있습니다.
• 높은 정확도를 제공하지만, 비교적 높은 계산 비용이 들기 때문에 보다 강력한 하드웨어나 시간이 필요할 수 있습니다.
• COCO 데이터셋에서 Faster R-CNN의 mAP(평균 정밀도)는 40% 이상일 수 있습니다.

YOLOv5:
• YOLOv5는 경량화된 디자인을 가지고 있어 모델 크기가 작고 상대적으로 빠른 속도를 가집니다.
• COCO 데이터셋에서 YOLOv5x의 mAP는 50% 이상으로 예상됩니다.

※ 정확도는 매우 구체적인 상황과 사용된 모델 및 데이터에 따라 크게 다르므로, 실제 응용에서는 자신의 데이터셋과 요구 사항에 맞게 실험하여 결과를 확인하는 것이 좋습니다. 데이터 전처리, 학습 스케줄, 하이퍼파라미터 조정 등을 통해 정확도를 향상시킬 수 있습니다

※ 두 모델의 비교는 정확도와 성능 사이의 트레이드오프를 고려해야 합니다. 더 무거운 모델은 더 높은 정확도를 가질 수 있지만, 동시에 더 높은 계산 비용과 처리 시간을 요구할 수 있습니다. 따라서 사용 사례와 하드웨어 리소스에 맞게 모델을 선택하는 것이 중요합니다.