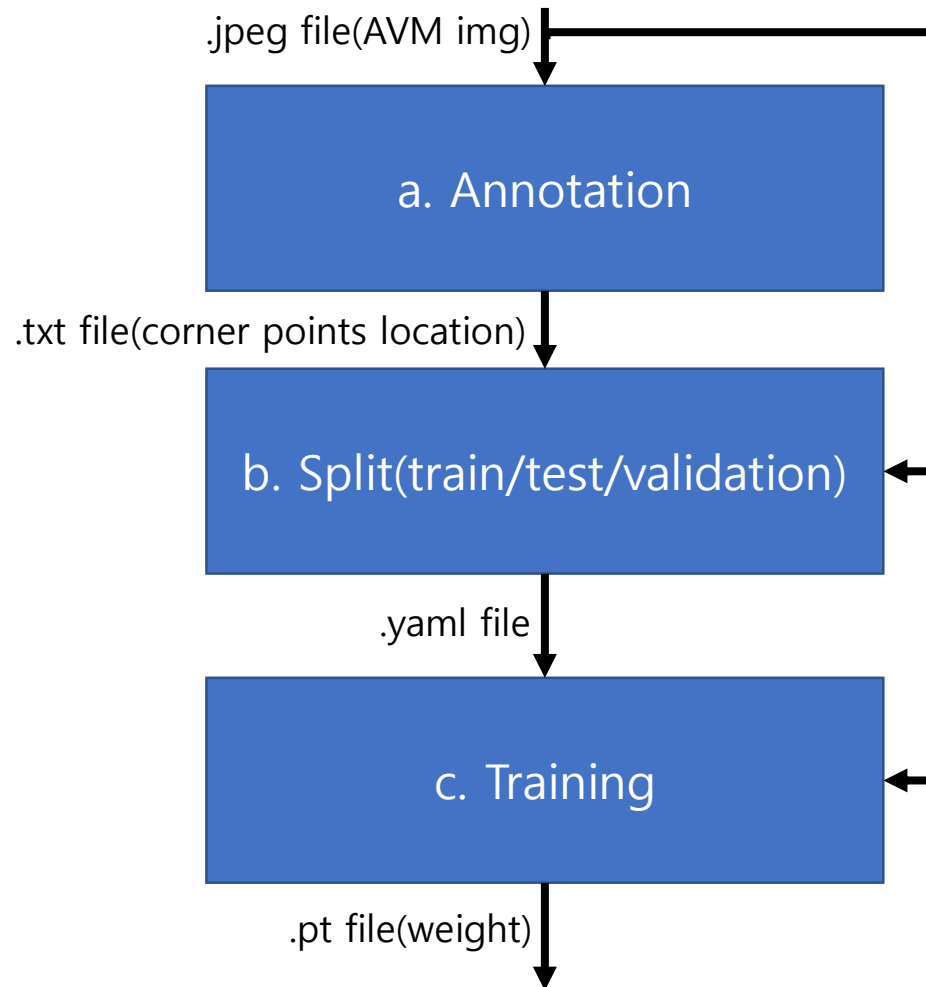


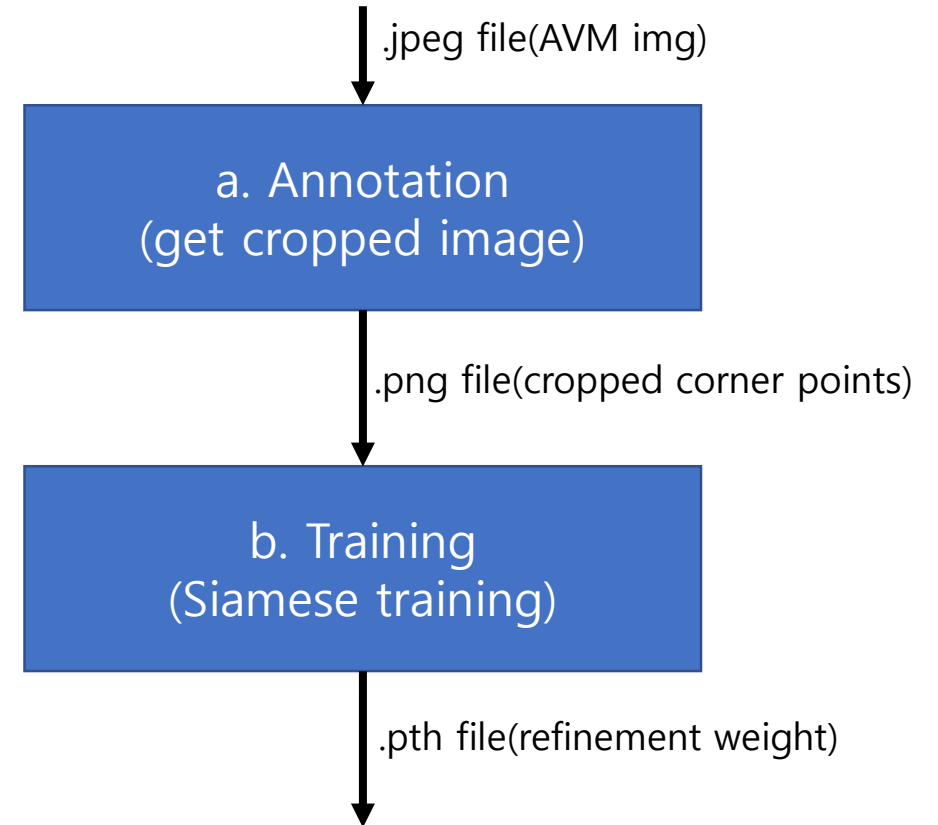
설명자료

YOLOv5 & DeepSORT Training

1. Yolo training



2. Feature extractor training (DeepSORT training)



Corner point training

1. Yolo training
2. DeepSORT refinement training
3. <https://github.com/JunhyeokRui/training> README 참고

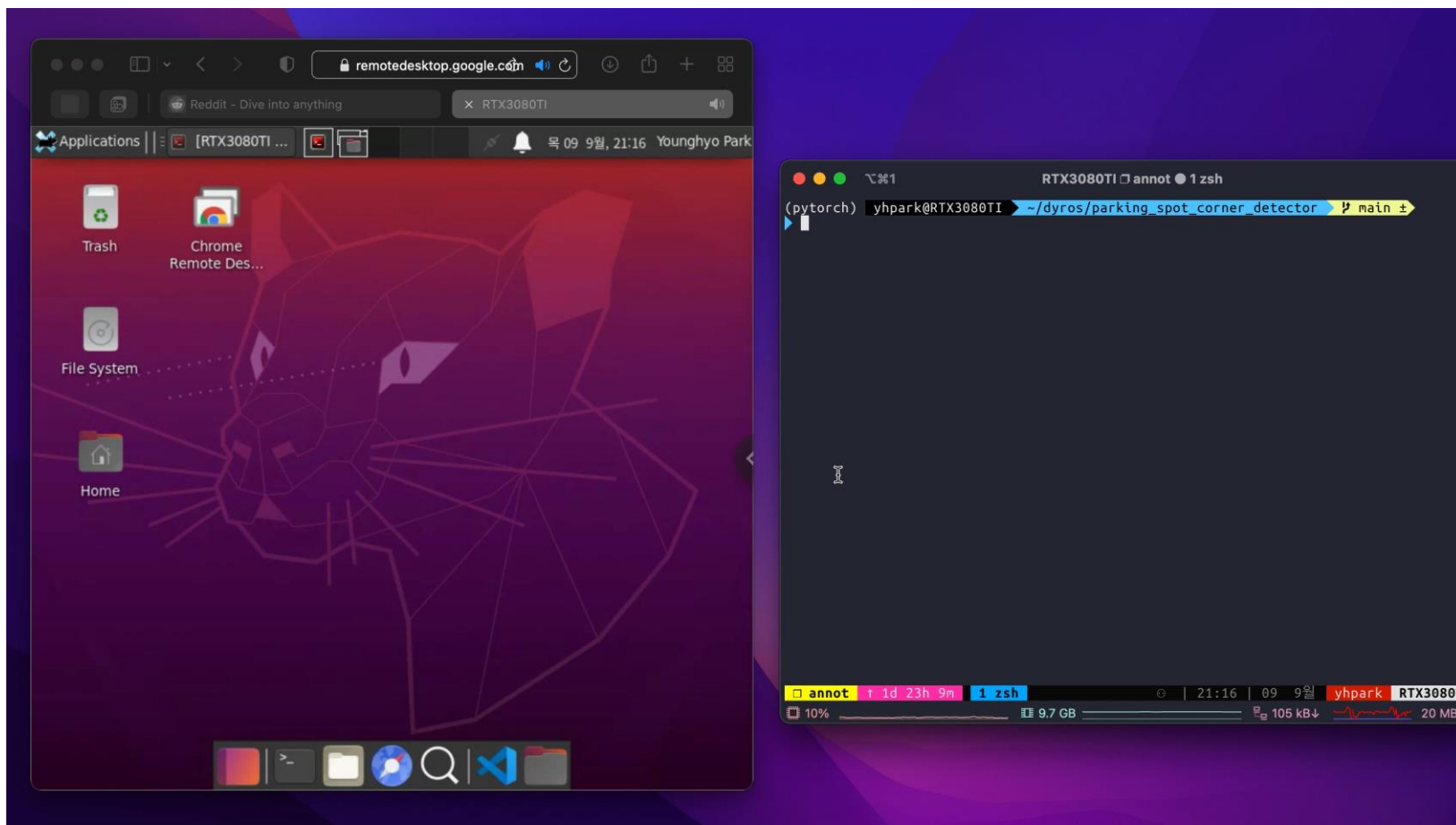
1. YOLO training

- 목적: YOLO performance 향상
- **Dataset structure**
 - **AVM_center_data_4class** → dataset (이름)
 - **trial_A, B, C...** → 각각 다른 주차 시나리오의 dataset
 - **Labels** → 이미지에서 corner point x y 좌표, corner point 타입
- **Training**
 - a. **Annotation**
 - b. **Split**(train/test/validation)
 - c. **Training**

```
..
├── AVM_center_data_4class
│   ├── Images
│   │   ├── trial_A
│   │   │   ├── image_001.jpeg
│   │   │   ├── image_002.jpeg
│   │   │   └── ...
│   │   ├── trial_B
│   │   ├── trial_C
│   │   └── ...
│   ├── Labels
│   │   ├── trial_A
│   │   │   ├── image_001.txt
│   │   │   ├── image_002.txt
│   │   │   └── ...
│   │   ├── trial_B
│   │   ├── trial_C
│   │   └── ...
│   └── README.md
```

1-a. Annotation - YOLO training

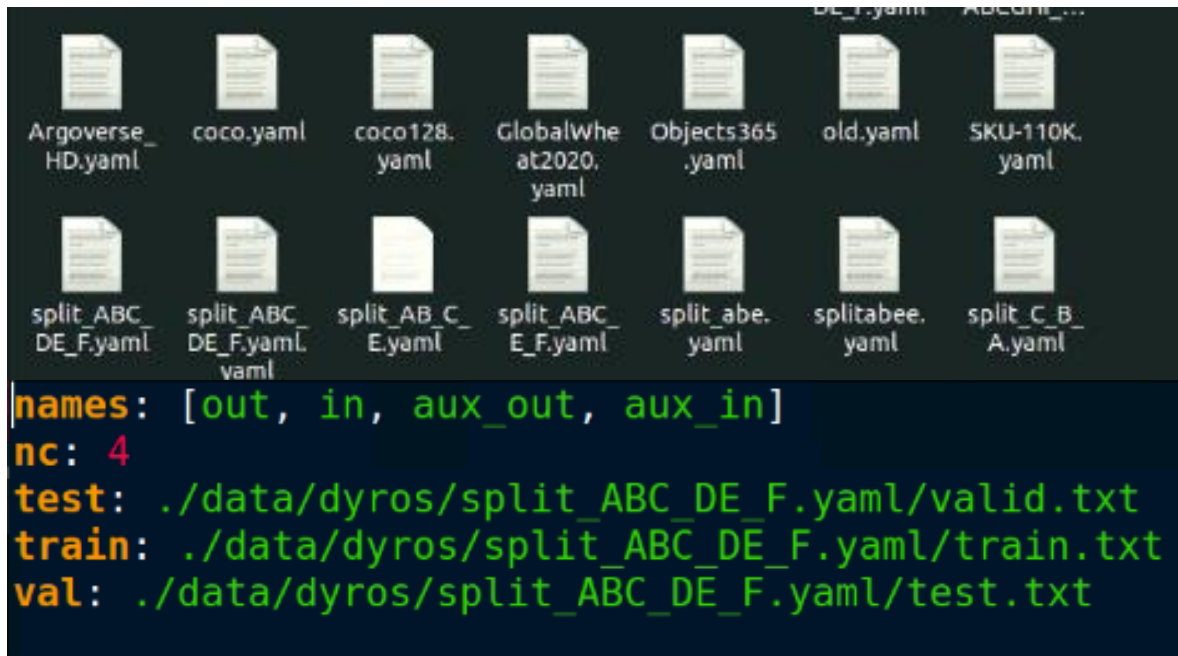
- AVM 이미지에 marking point 클릭 전, 키보드로 marking point 종류 선택 option(A = outer ,S = inner ,D = outer aux, F = inner aux)



1-b. Split - YOLO training

- 주차 실험 episode 별로 train / test / validation 분할
- parser(option) --train A,B,C --valid D,E --test F 입력 시 data 폴더에 split_ABC_DE_F.yaml 생성

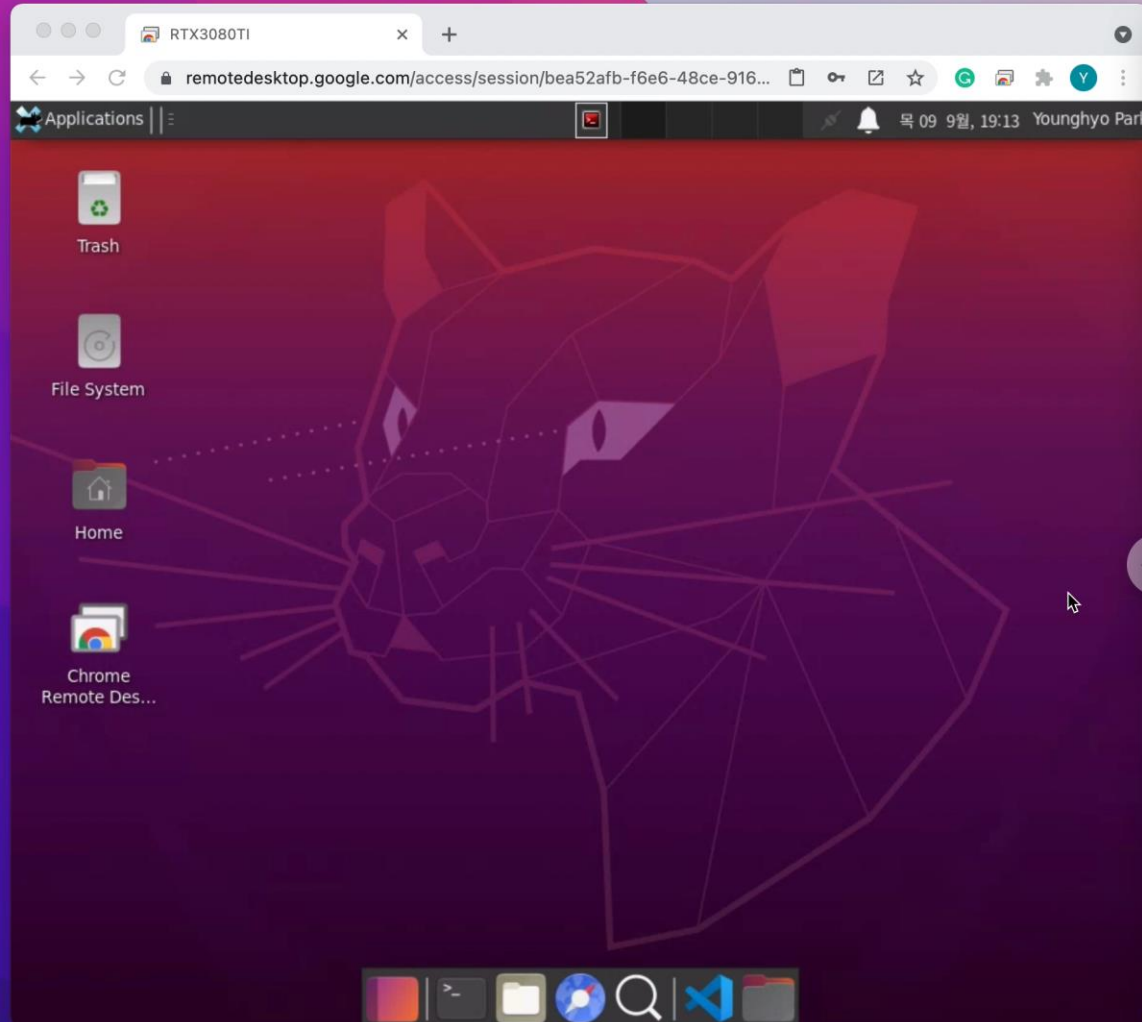
- 예시



1-c. Training - YOLO training

- trained weight 이 ./runs/train/yourexperiment/last.pt, best.pt 로 저장됨
- best.pt 트레이닝 동안의 best weights.
- last.pt last epoch weights.
- □-view-img parser(option) 로 opencv window 통해 live result 볼 수 있음

Marking Point Detection Workflow



```
RTX3080TI □ ros ● 1 zsh
yhpark@RTX3080TI ~ /HDD0/bagfile/new11

yhpark@RTX3080TI ~ /dyros/parking_spot_corner_detector ▸ main

yhpark@RTX3080TI ~ ▸

yhpark@RTX3080TI ~ ▸

RTX3080TI Thu Sep 9 19:13:11 2021 460.91.03
[0] GeForce RTX 3080 Ti | 39°C, 0%, 0%, 29 / 400 W | 18 / 12053 MB |
Xorg(9M) gnome-shell(6M)

1 [ ] 2.6% 7 [ ] 2.6% 13 [ ] 0.7% 19 [ ] 1.3%
2 [ ] 0.7% 8 [ ] 1.3% 14 [ ] 0.7% 20 [ ] 1.3%
3 [ ] 0.7% 9 [ ] 6.0% 15 [ ] 1.3% 21 [ ] 0.0%
4 [ ] 1.3% 10 [ ] 1.3% 16 [ ] 1.3% 22 [ ] 0.0%
5 [ ] 5.3% 11 [ ] 0.0% 17 [ ] 0.7% 23 [ ] 2.0%
6 [ ] 3.9% 12 [ ] 1.3% 18 [ ] 3.3% 24 [ ] 1.3%
Mem [ ] 2.92G/62.8G Tasks: 337, 856 thr: 1 running
Swp [ ] 0K/2.00G Load average: 0.06 0.29 0.44
Uptime: 1 day, 21:05:49

ros 1d 21h 5m 1 zsh 19:13 09 9월 yhpark RTX3080TI
10% 10 GB 191 kB↓ 25 MB↑
```

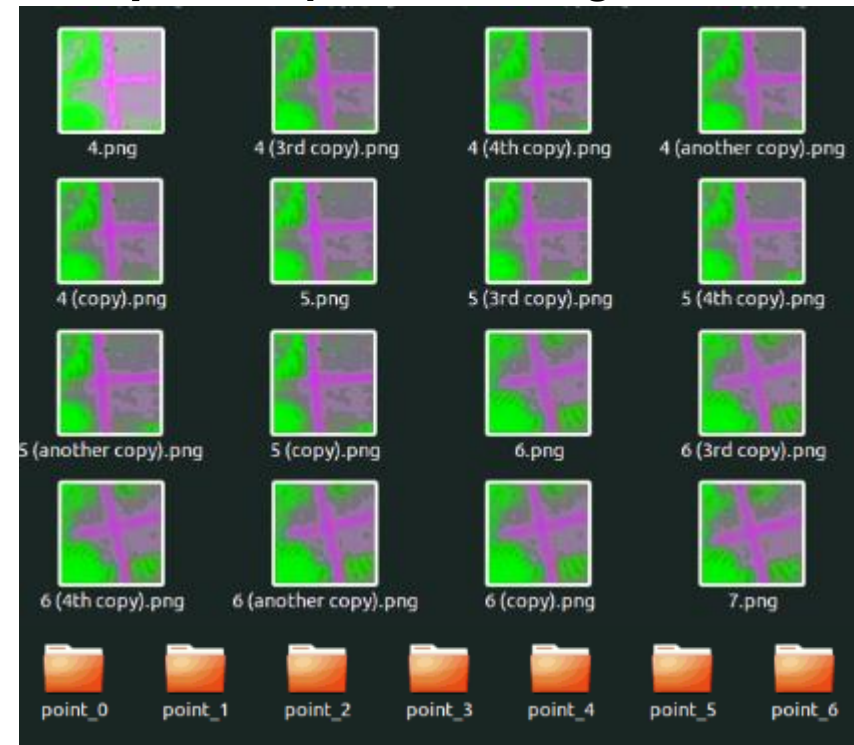

2. DeepSORT training

- 목적: corner point ID switching을 줄이기 위해서
- **Dataset structure**
 - **dyros_deepsort_dataset** → dataset (이름)
 - **cropped_30** → cropped 사이즈
 - **trial_A, B, C...** → 각각 다른 주차 시나리오의 dataset
 - **point_1, 2, 3...** → 각 corner point 분류
- **Training**
 - a. **Annotation** (cropped image of points)
 - b. **Training** (Siamese training)

```
..
├── dyros_deepsort_dataset
│   ├── cropped_30
│   │   ├── trial_A
│   │   │   ├── point_0
│   │   │   │   ├── 1.png
│   │   │   │   ├── 2.png
│   │   │   │   ├── 3.png
│   │   │   │   └── ...
│   │   │   ├── point_1
│   │   │   │   ├── 1.png
│   │   │   │   ├── 2.png
│   │   │   │   ├── 3.png
│   │   │   │   └── ...
│   │   │   ├── point_2
│   │   │   ├── point_3
│   │   │   ├── ...
│   │   │   └── ...
│   │   └── pixel_labels
│   │       ├── trial_A
│   │       │   ├── point_0
│   │       │   │   ├── 1.txt
│   │       │   │   ├── 2.txt
│   │       │   │   ├── 3.txt
│   │       │   │   └── ...
│   │       │   ├── point_1
│   │       │   │   ├── 1.txt
│   │       │   │   ├── 2.txt
│   │       │   │   ├── 3.txt
│   │       │   │   └── ...
│   │       │   ├── point_2
│   │       │   ├── point_3
│   │       │   ├── ...
│   │       │   └── ...
│   └── README.md
```

2-a. Annotation(cropped image of points) - DeepSORT training (refinement)

- Cropped image 예시 → corner points
- Annotation point by point 로 진행됨.
- AVM 이미지에서 corner point 선택 시 cropped corner point → dataset/cropped_(bb_size)/trial_A 폴더에 저장
- Corner point x y 좌표 → dataset/pixel_labels/trial_A 폴더에 저장
- --crop-only parser for cropping corner points only(corner point x y 좌표 생기지 않음)



2-b. Training(Siamese training) - DeepSORT training(refinement)

- 트레이닝 시작 전 siamese_training.py 65번째 줄 dataset 경로 확인
- 요구하는 성능, 환경에 따라 hyperparameter 튜닝 필요
- './deep_sort_pytorch/deep_sort/deep/experiment_name.pth' 경로에 .pth 파일 저장됨
- detect or ROS_track_marking_points.py 의 -refinements_weight parser 로 사용함.

□

□

□