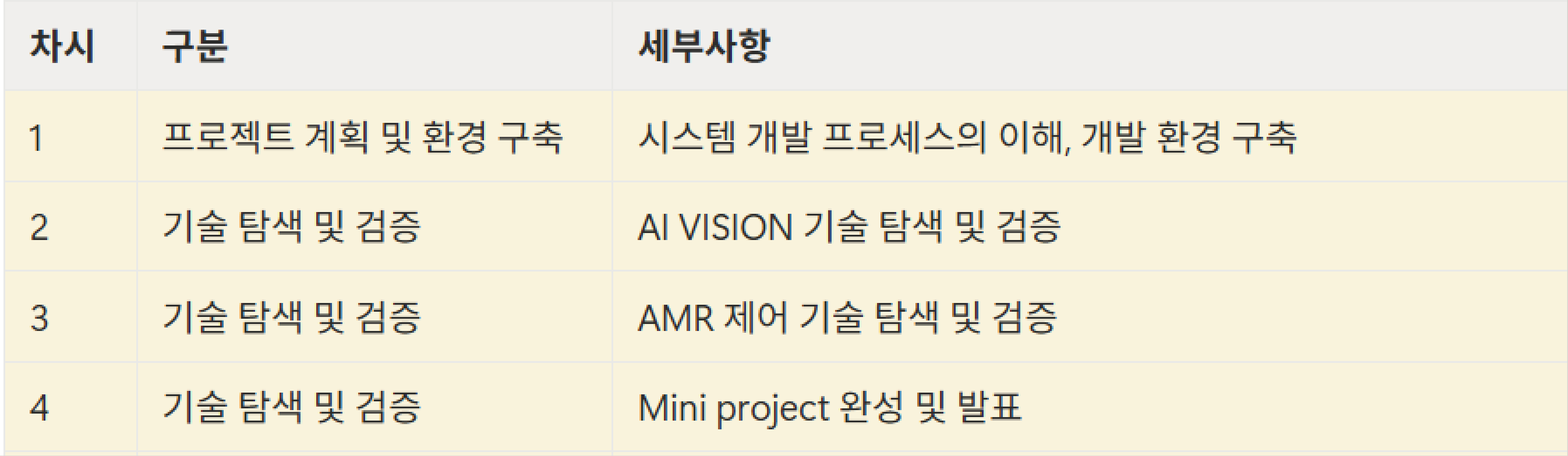
GOOD MORNING!
早上好!
안녕하세요!

DAY 2

DAY1 RECAP

2 PROJECTS

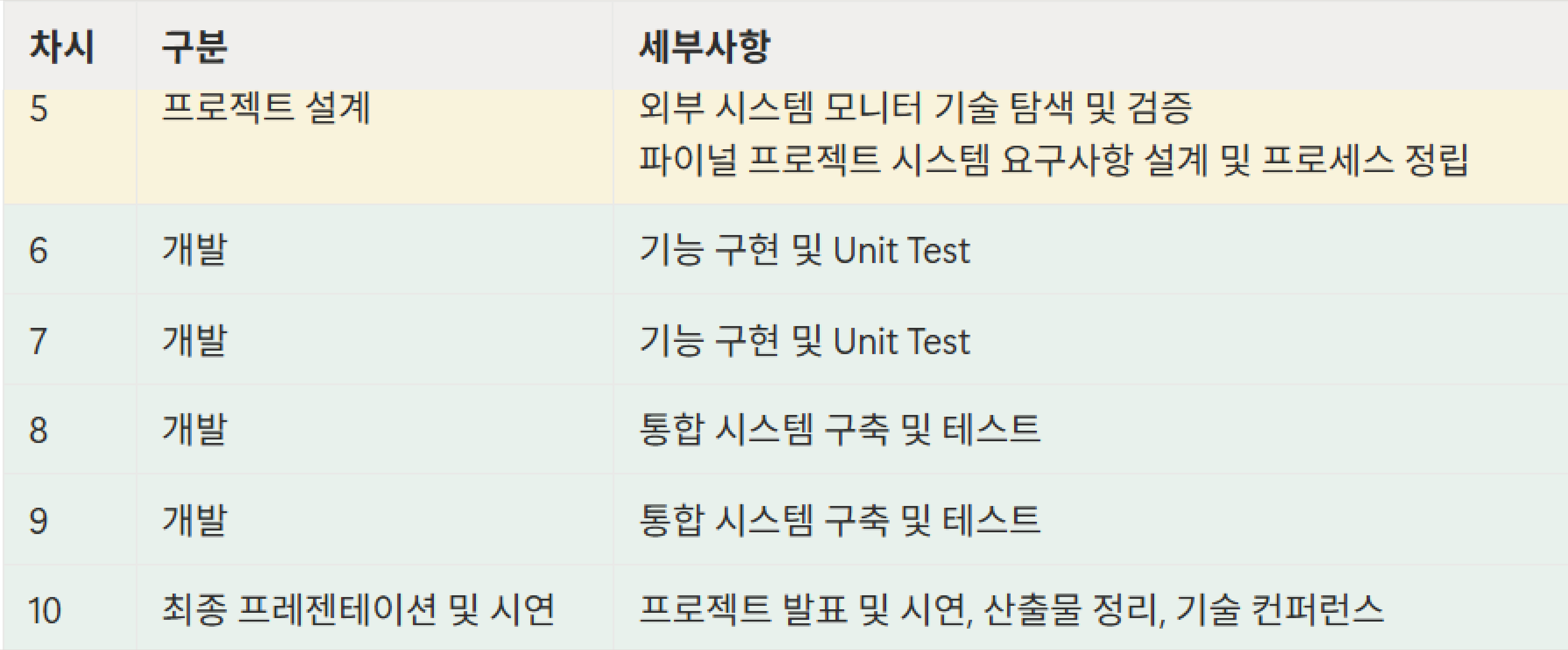
- Mini Project (Individual Team)
  - For learning techniques



차시	구분	세부사항
1	프로젝트 계획 및 환경 구축	시스템 개발 프로세스의 이해, 개발 환경 구축
2	기술 탐색 및 검증	AI VISION 기술 탐색 및 검증
3	기술 탐색 및 검증	AMR 제어 기술 탐색 및 검증
4	기술 탐색 및 검증	Mini project 완성 및 발표

2 PROJECTS

- Final Project (2 Teams in One)



차시	구분	세부사항
5	프로젝트 설계	외부 시스템 모니터 기술 탐색 및 검증 파이널 프로젝트 시스템 요구사항 설계 및 프로세스 정립
6	개발	기능 구현 및 Unit Test
7	개발	기능 구현 및 Unit Test
8	개발	통합 시스템 구축 및 테스트
9	개발	통합 시스템 구축 및 테스트
10	최종 프레젠테이션 및 시연	프로젝트 발표 및 시연, 산출물 정리, 기술 컨퍼런스

DAY 1

- Welcome
- Project Introduction
- Introduction to Project Development Process
- Business Requirement Development
- System Requirement Development
- System and Development environment Setup

DAY 2 (MINI PROJECT)

- Yolo객체 인식 모델 활용과 성능 평가 방법 이해
  - Custom Dataset과 Fine Tuning으로 자체 객체 인식 모델 구현 및 평가
  - (Optional)경량화 모델 등 개별 요구사항에 적합한 모델 탐색 및 성능 검증

DAY 2 (MINI PROJECT)

WEB-CAM 기반 객체 인식

- YOLOv8 기반 데이터 수집/학습/deploy (Detection Alert)
  - 감시용 데이터 수집(rc_car, dummy, 등)
  - 감시용 데이터 라벨링
  - YOLOv8 기반 학습
  - YOLOv8 Object Detection

AMR-CAM 기반 객체 인식

- AMR(Autonomous Mobile Robot) Turtlebot4 개발 환경 구축
- 로봇 개발 환경에 완성 모델 서빙 및 테스트 / 로봇 H/W, 제반 환경의 한계점 도출
  - Tracking 데이터 수집((rc_car, dummy, 등)
  - Tracking 데이터 라벨링
  - YOLOv8 기반 학습
  - YOLOv8 Object Tracking

DAY 3 (MINI PROJECT)

- Auto. Driving 시스템 학습
  - Digital Mapping of environment
  - Operate AMR (Sim. & Real)
  - Tutorial 실행
  - Detection, Depth and AMR 주행
  - 로봇 개발 환경에 적용 및 테스트 / 로봇 H/W, 제반 환경의 한계점 도출

TURTLEBOT4 시뮬레이션 DEMO

- SLAM과 AutoSLAM으로 맵 생성
- Sim. Tutorial 실행
- Detection, Depth and AMR 주행 example

DAY 3 (MINI PROJECT)

REAL ROBOT

- Manually operating the AMR (Teleops)
- autonomous driving 시스템 with obstacle avoidance
  - Digital Mapping of environment
  - Launching Localization, Nav2, and using Rviz to operate a robot
  - Goal Setting and Obstacle Avoidance using Navigation

TUTORIAL

- Turtlebot4 API를 활용한 Initial Pose Navigate_to Pose 구현
- Turtlebot4 API를 활용한 Navigate_Through_pose, Follow Waypoints 구현

HOW TO WORK TOGETHER

- Participate, Participate, Participate!!!
- No long emails or Kakaotalk, prefer face to face
- Be open to suggestions and idea
- Be proactive (적극적), take initiative (주도적)
- HOW is as important as WHAT
- Ask the right questions? (to **YOU, team** and me)
- Investigate/Research/Analyze

프로젝트 RULE

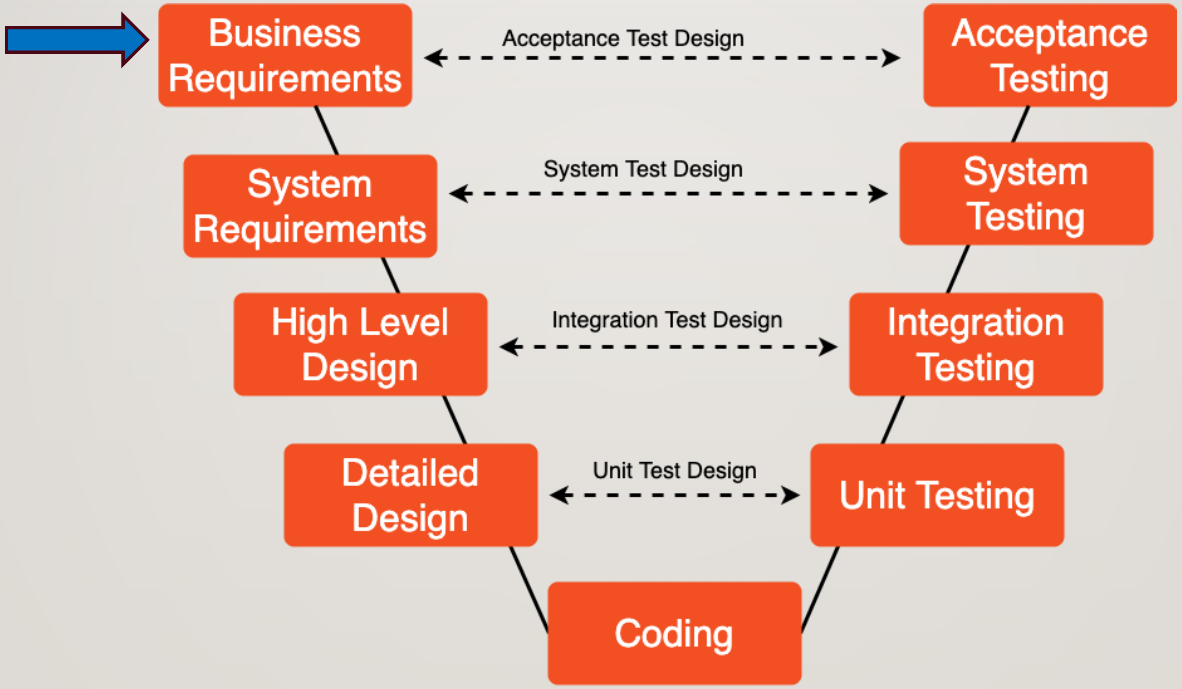
80/20 → 20/80

TEAMWORK AND PROJECT MANAGEMENT

PROJECT DEVELOPMENT IS A PROCESS

SW DEVELOPMENT PROCESS

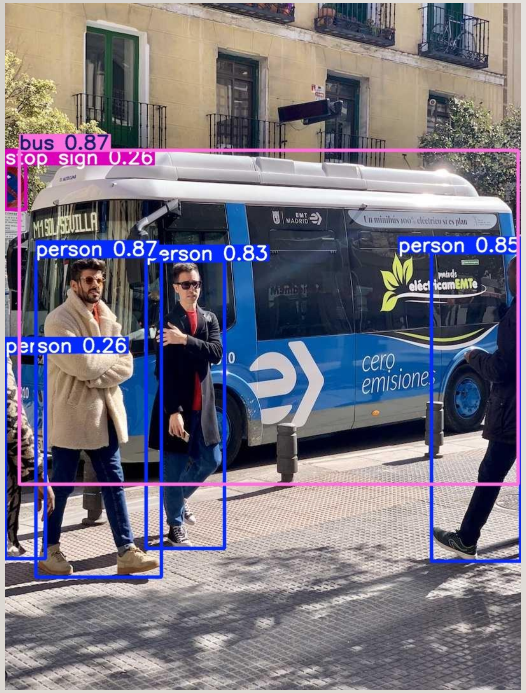
Why and What



SDLC - V Model - notepub.io

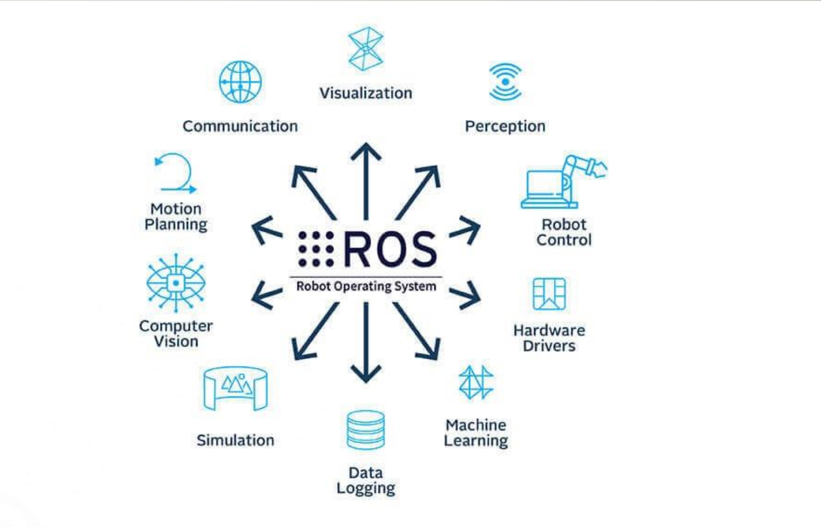
ADVANCED TECHNIQUES THAT WE HAVE

- AI Object Detection



- AMR
  - Navigation with obstruction avoidance
  - Sensors

- ROS2



BRAINSTORM A SITUATION THAT WILL BENEFIT FROM YOUR SOLUTION

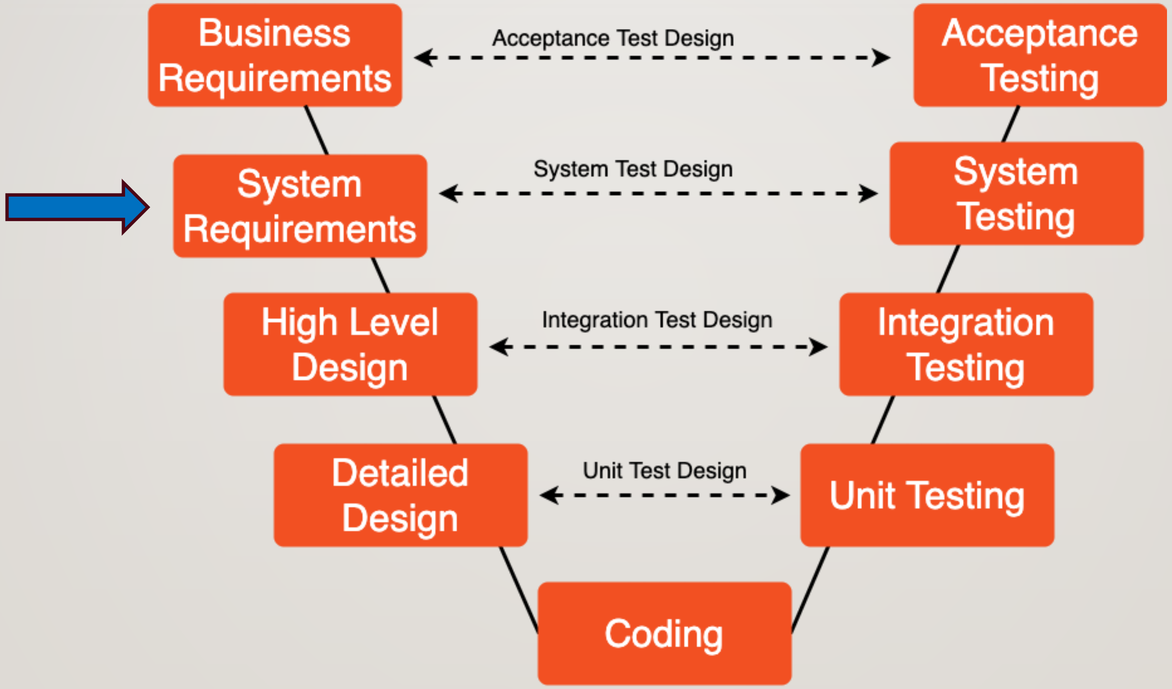
Must have measurable benefits. Search for them online

BRAINSTORMING RULES

- Every input is good input
- Do not critique inputs only seek to understand
- Organize inputs into logical groupings
- Sequence or show relationships as needed
- Use Posted Notes on Flip Chart

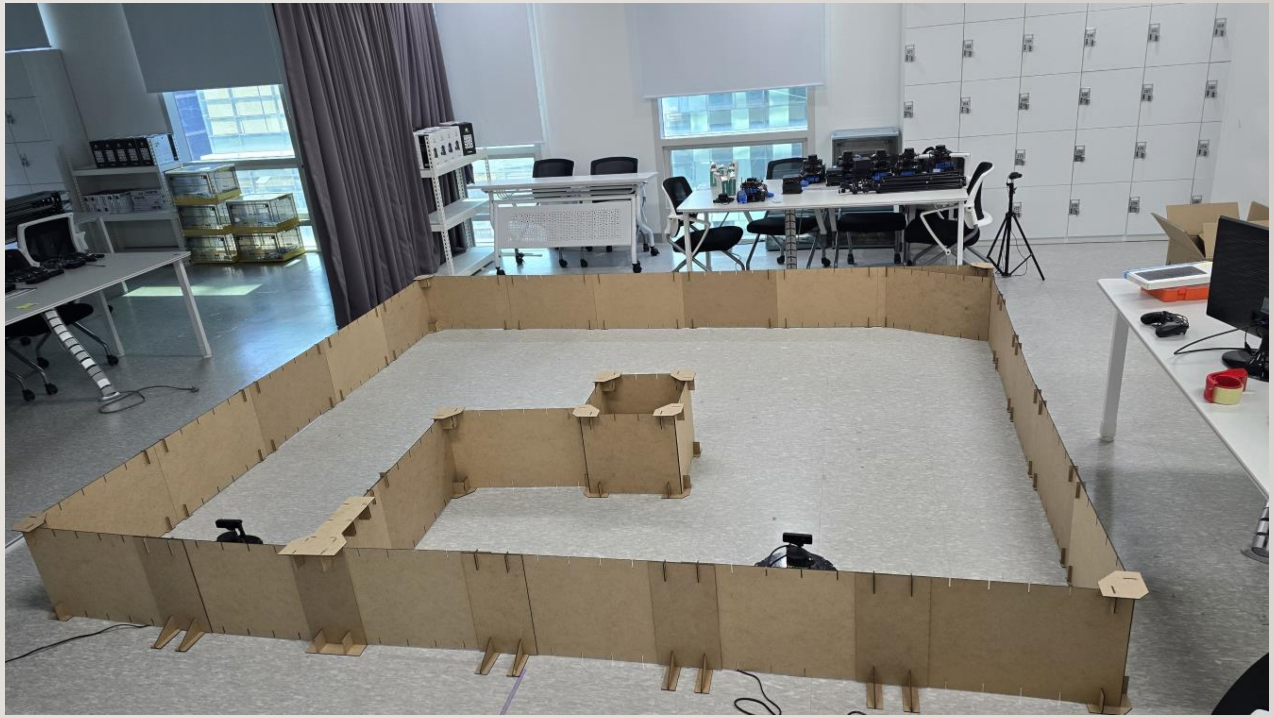
SW DEVELOPMENT PROCESS

With What
And
How Much

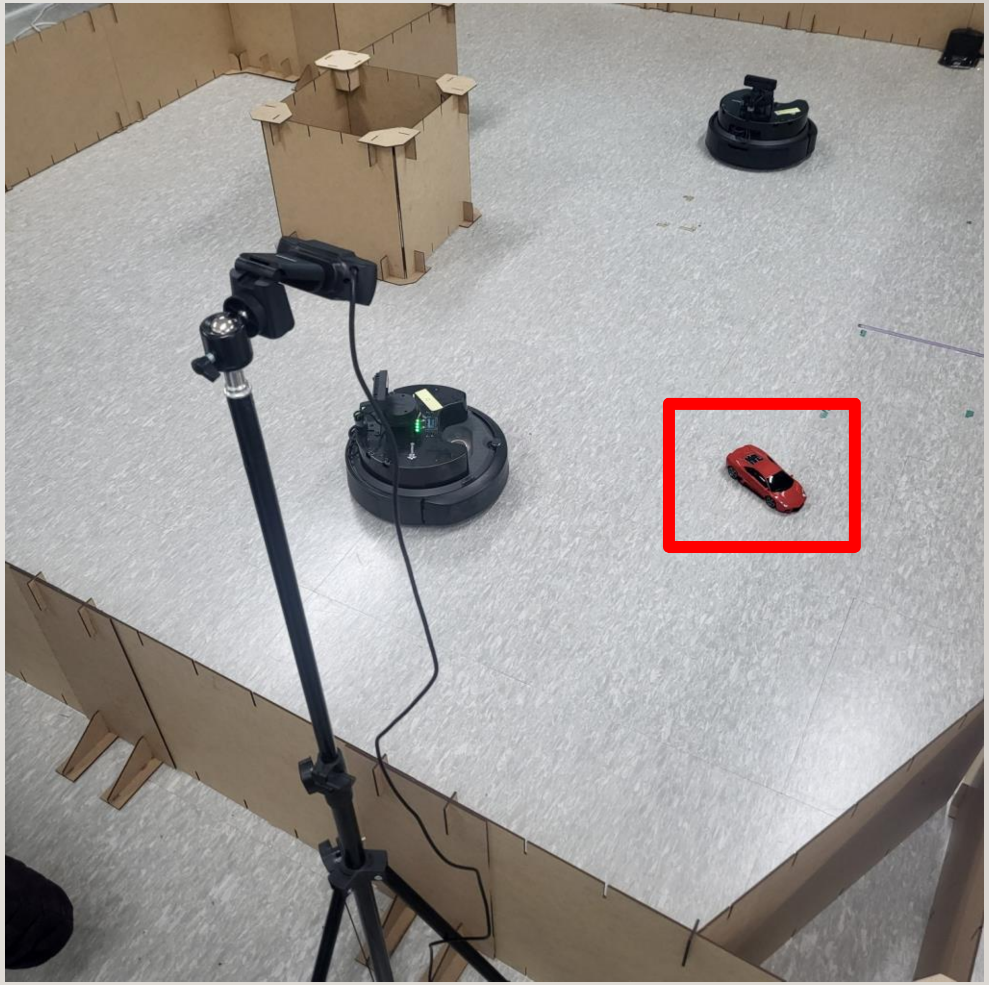


SDLC - V Model - notepub.io

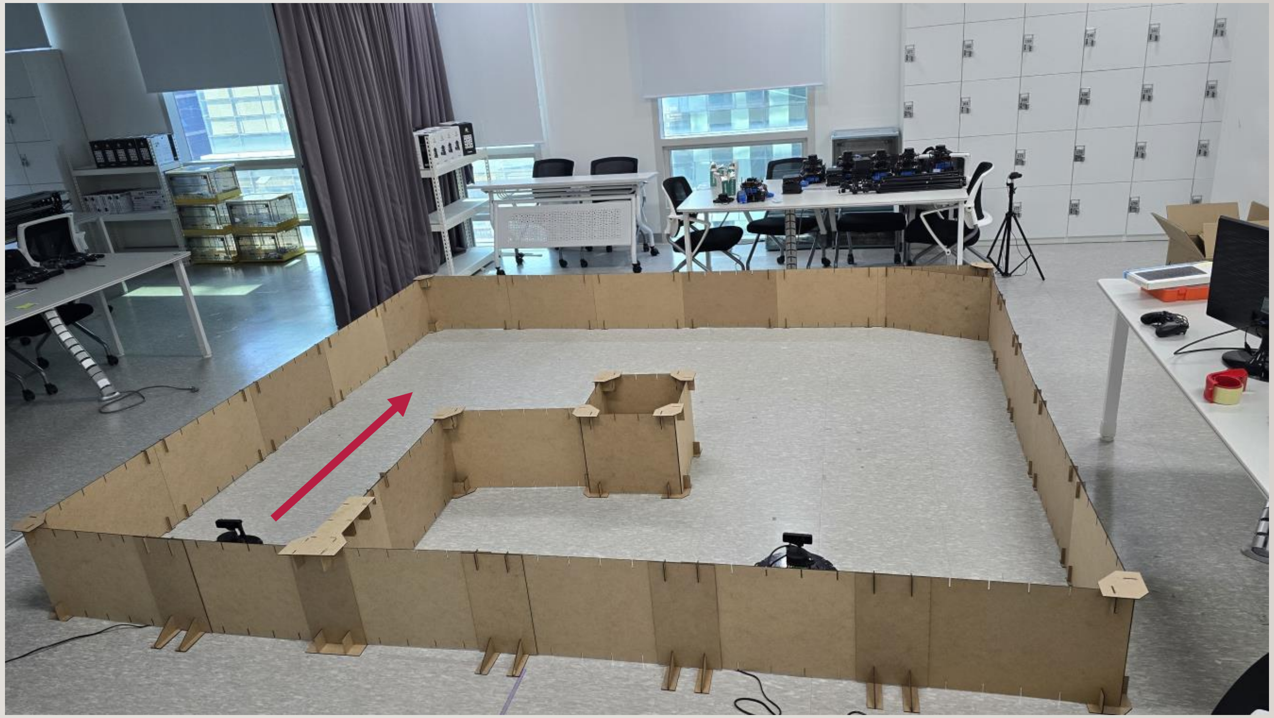
MINI PROJECT DESCRIPTION



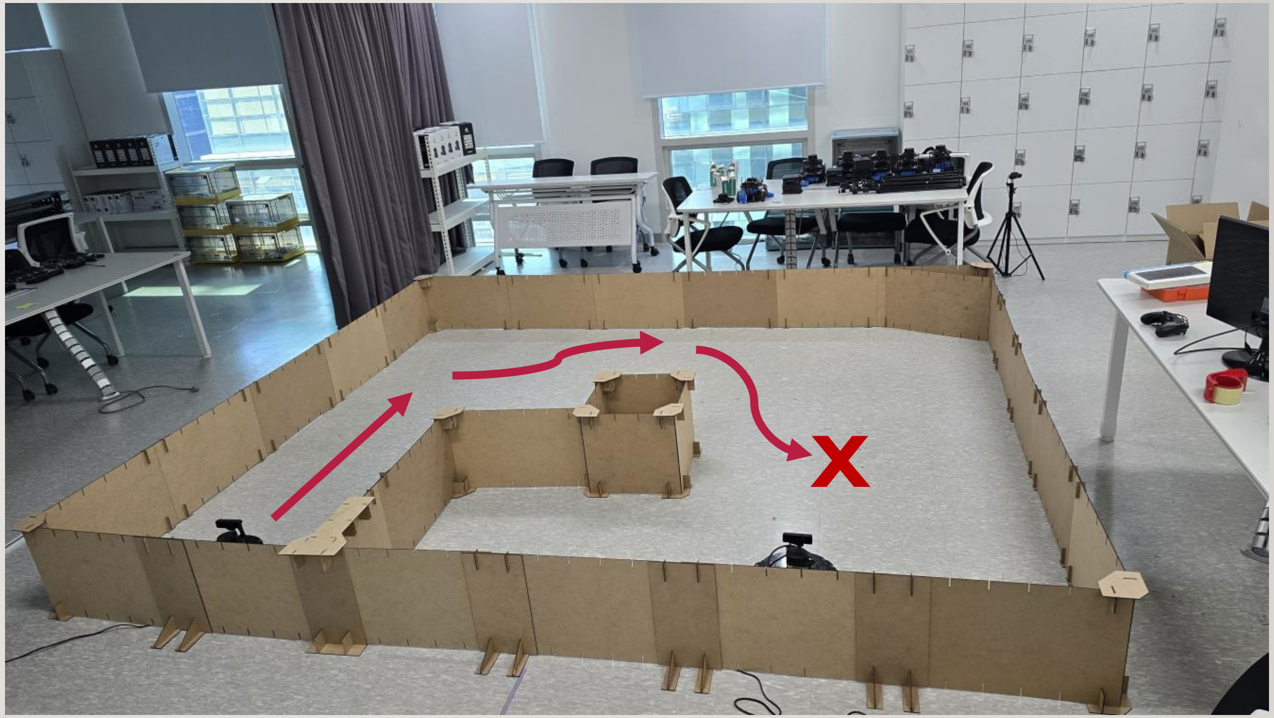
DETECTION ALERT



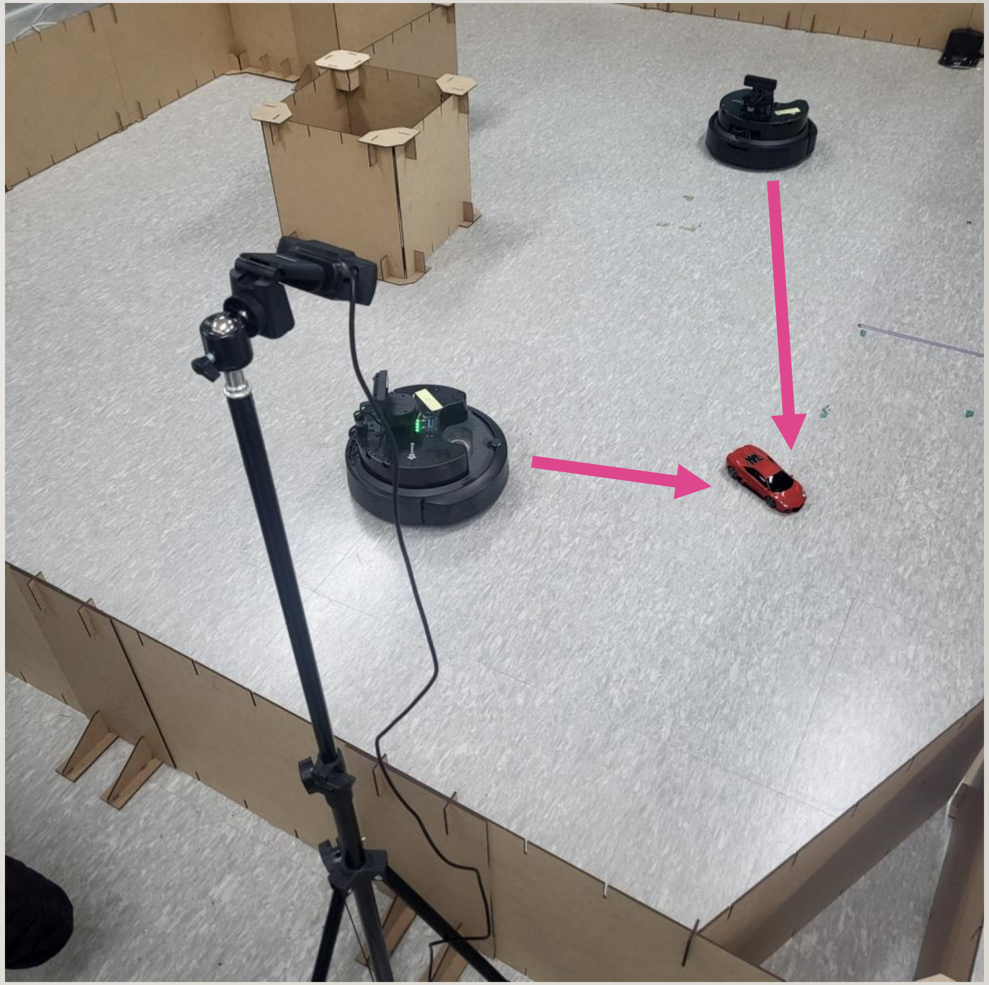
START



NAVIGATE TO A POSITION



TRACK & APPROACH

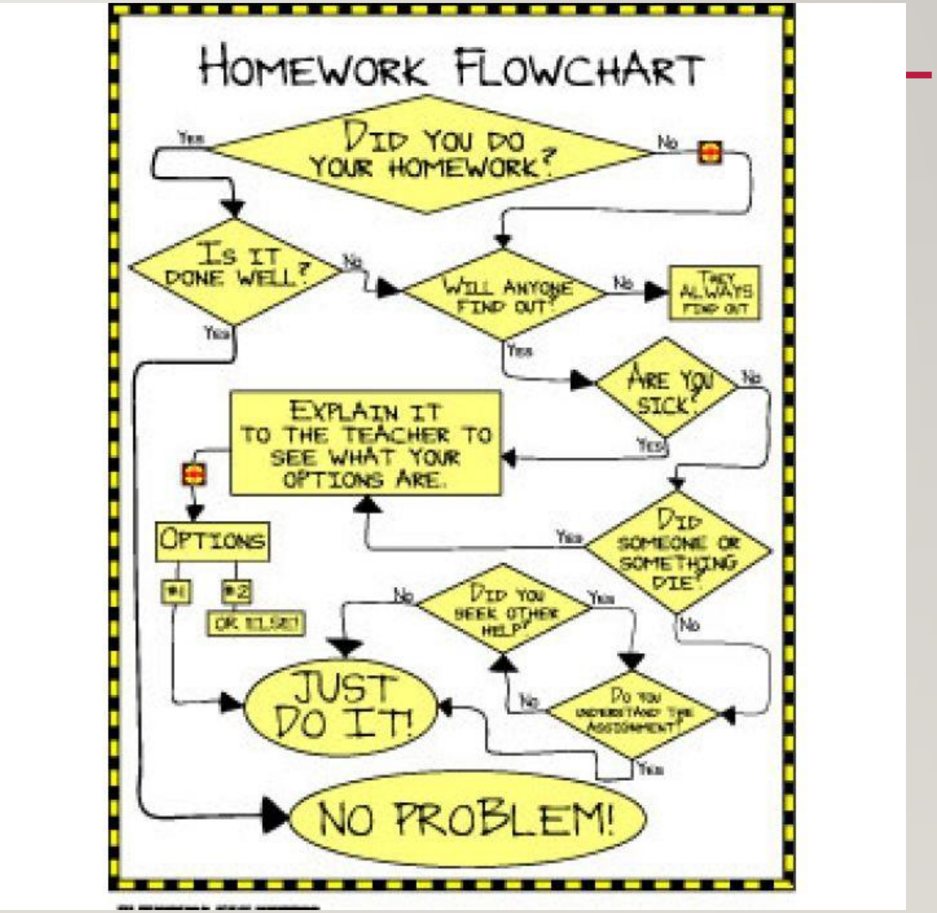


VISUALIZATION – SCENARIO PROCESS DIAGRAMS

- As-Is Functional Process Diagram
  - Current states

- To-Be Functional Process Diagram
  - Future states

- Untitled Diagram - draw.io
- https://app.diagrams.net/

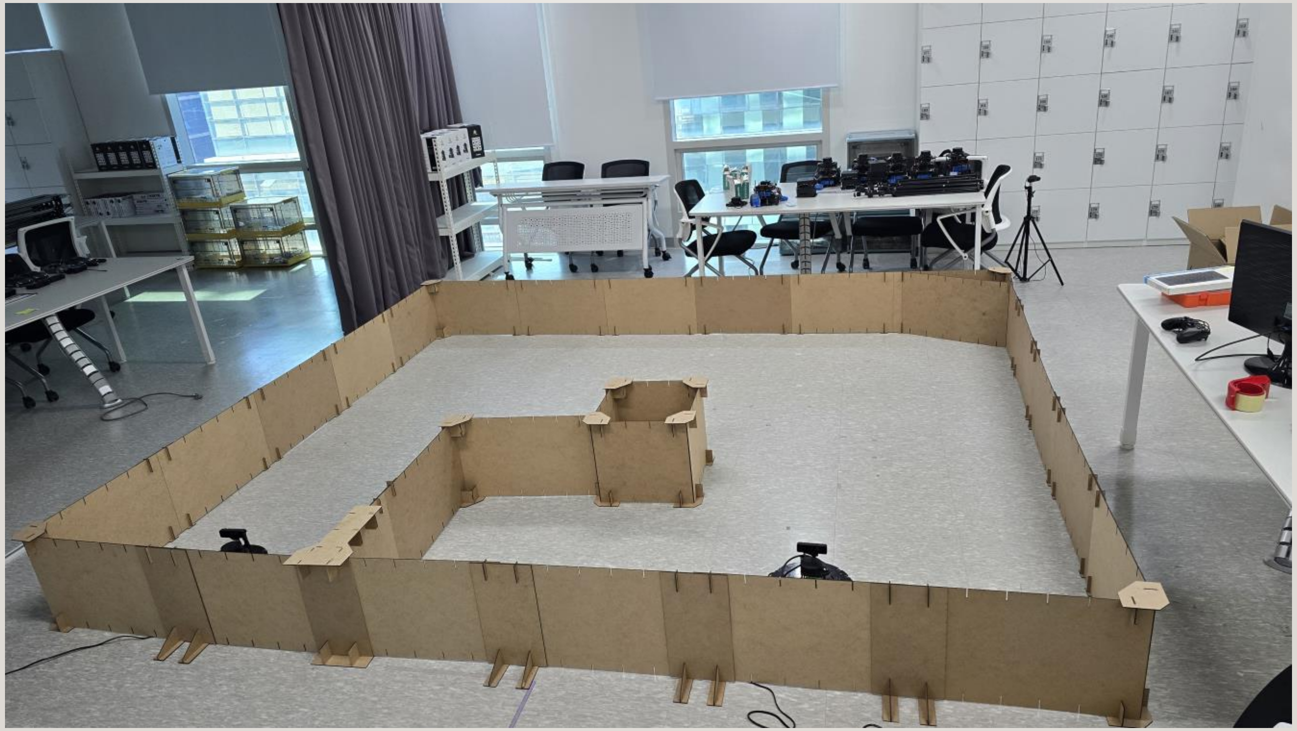


DEVELOP MINI PROJECT SCENARIO (USE-CASE) PROCESS DIAGRAM

Using the posted notes and flipchart as needed

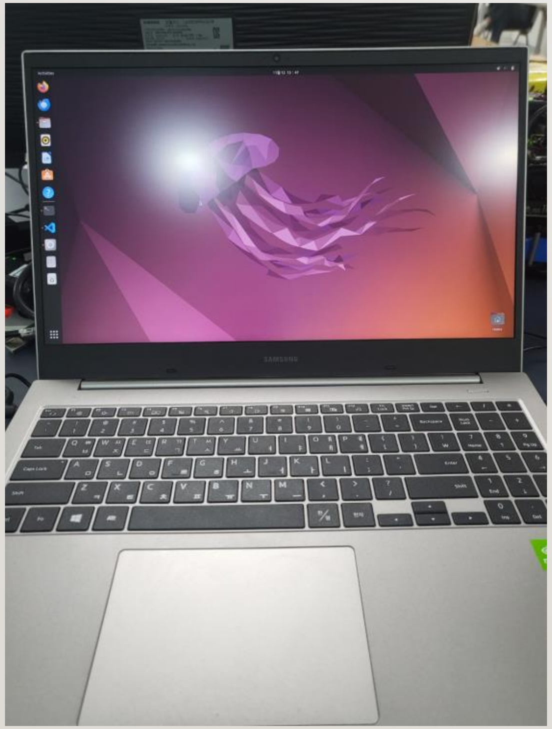
WITH WHAT

YOUR PROJECT ENVIRONMENT



BASE HW/OS

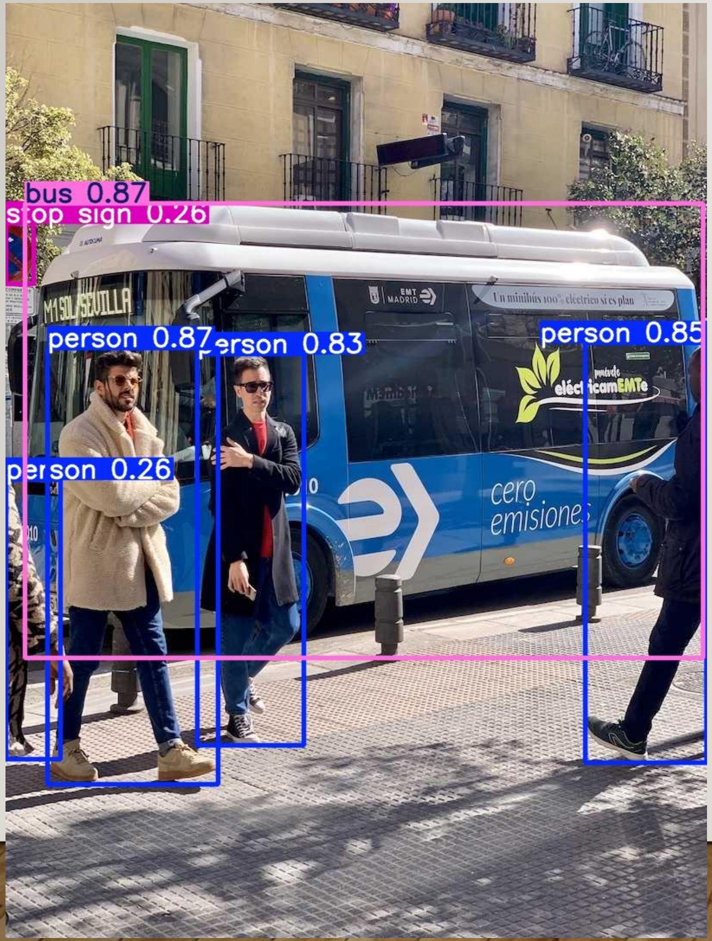
- PC
  - Ubuntu 22.04
  - USB Camera



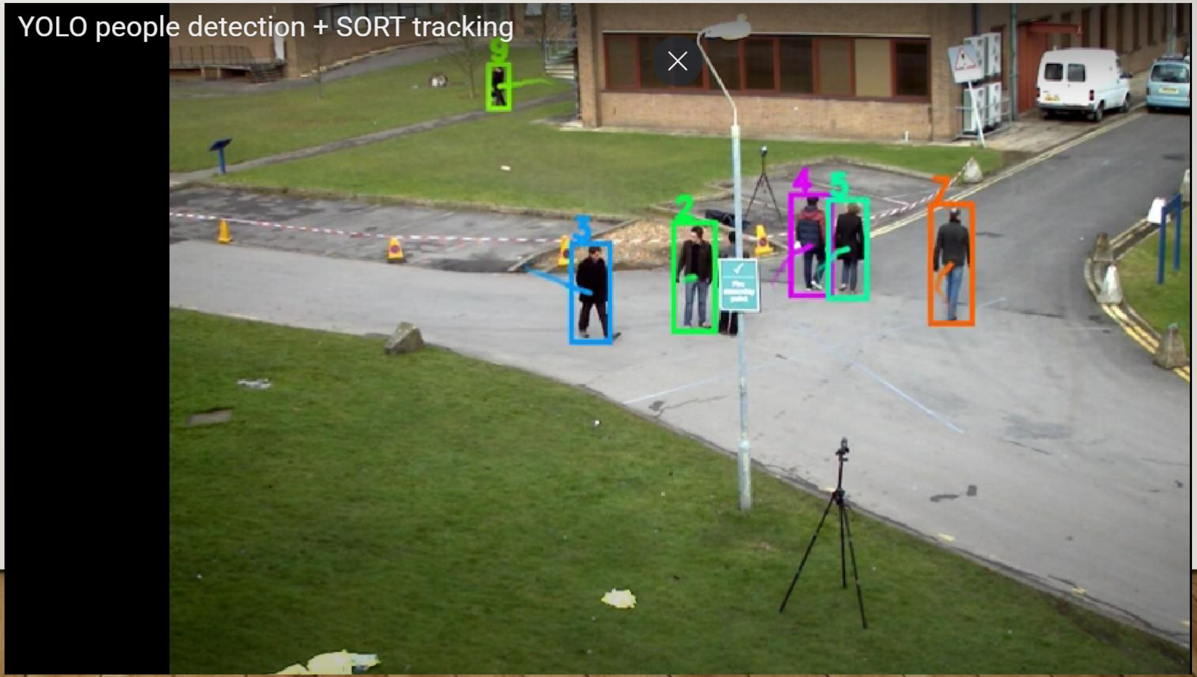
- Network
  - Wifi

- AMR
  - TurttleBot4
  - Ubuntu 22.04

YOLO OBJ. DET. VS. YOLO TRACKING



- Track - Ultralytics YOLO Docs
    - (469) YOLO people detection + SORT tracking – YouTube
    - Bing Videos



KEY SUBSYSTEM (MODULES) TO DEVELOP

- Detection Alert
  - Camera Capture
  - Object Detection
  - Send messages to other subsystems

- AMR Controller
  - Receive messages and act accordingly
  - Move using (SLAM) with Obstruction avoidance
  - Target Acquisition (Obj. Det.) and Tracking
  - Follow target using camera and motor control

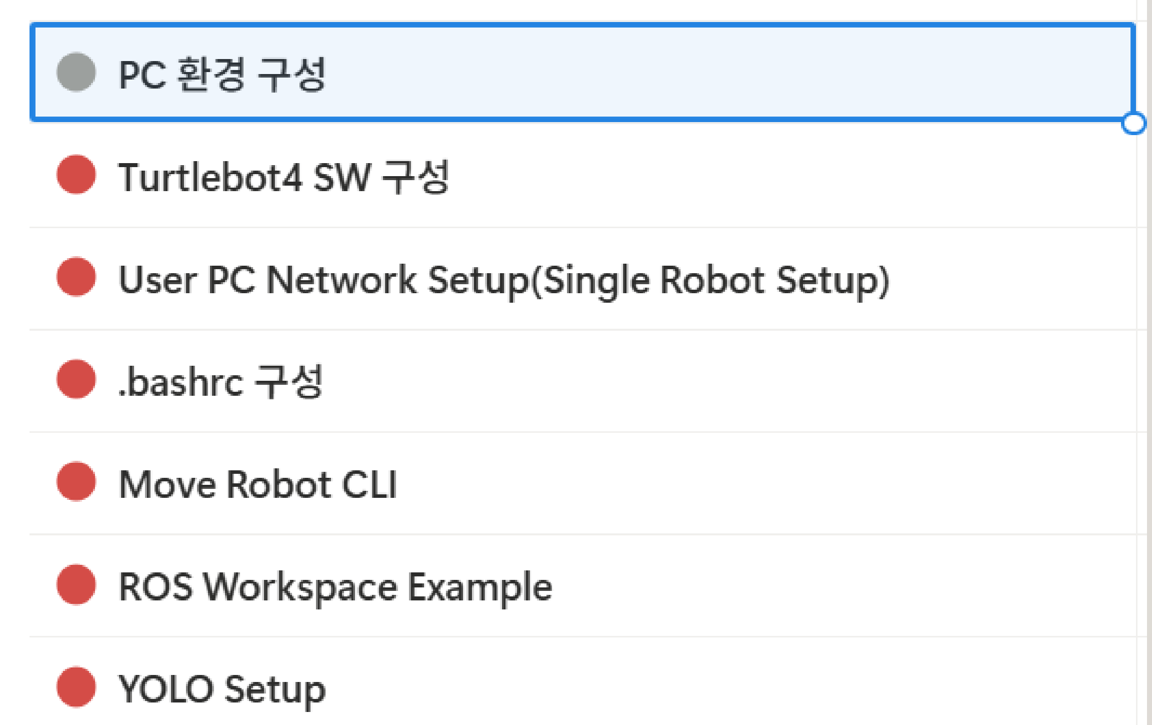
TEAM EXERCISE 2-2

Brainstorm mini-project System Requirement for the project and document

Using the posted notes and flipchart as needed

Include where, when, what will be used

SYSTEM AND DEVELOPMENT ENVIRONMENT SETUP



- ● PC 환경 구성
- ● Turtlebot4 SW 구성
- ● User PC Network Setup(Single Robot Setup)
- ● .bashrc 구성
- ● Move Robot CLI
- ● ROS Workspace Example
- ● YOLO Setup

HOMEWORK CHECK

REVIEW AMR (TURTLEBOT4) E-MANUAL

- Features · User Manual
- https://turtlebot.github.io/turtlebot4-user-manual/overview/features.html

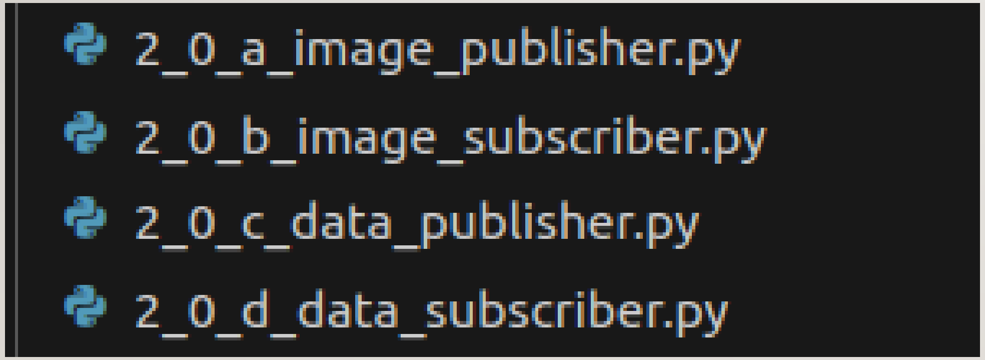
PLEASE REVIEW YOU WORK FROM EARLIER ONLINE CLASS

- Yolo obj. Det. Vs. Yolo Tracking
  - Object Detection - Ultralytics YOLO Docs
  - Track - Ultralytics YOLO Docs
  - Model Training with Ultralytics YOLO - Ultralytics YOLO Docs
- Yolo
  - Data Labelling (ex: LabelImg/roboflow)
  - Data pre-processing for YoloV8 Training
  - YoloV8 training to create .pt file
  - Using .pt file to predict/inference

- ROS
  - colcon build
  - Node, Topic, Service, Action, Interface, etc. coding

ROS EXERCISE 1

Create a ROS2 Package with these publisher and subscribers

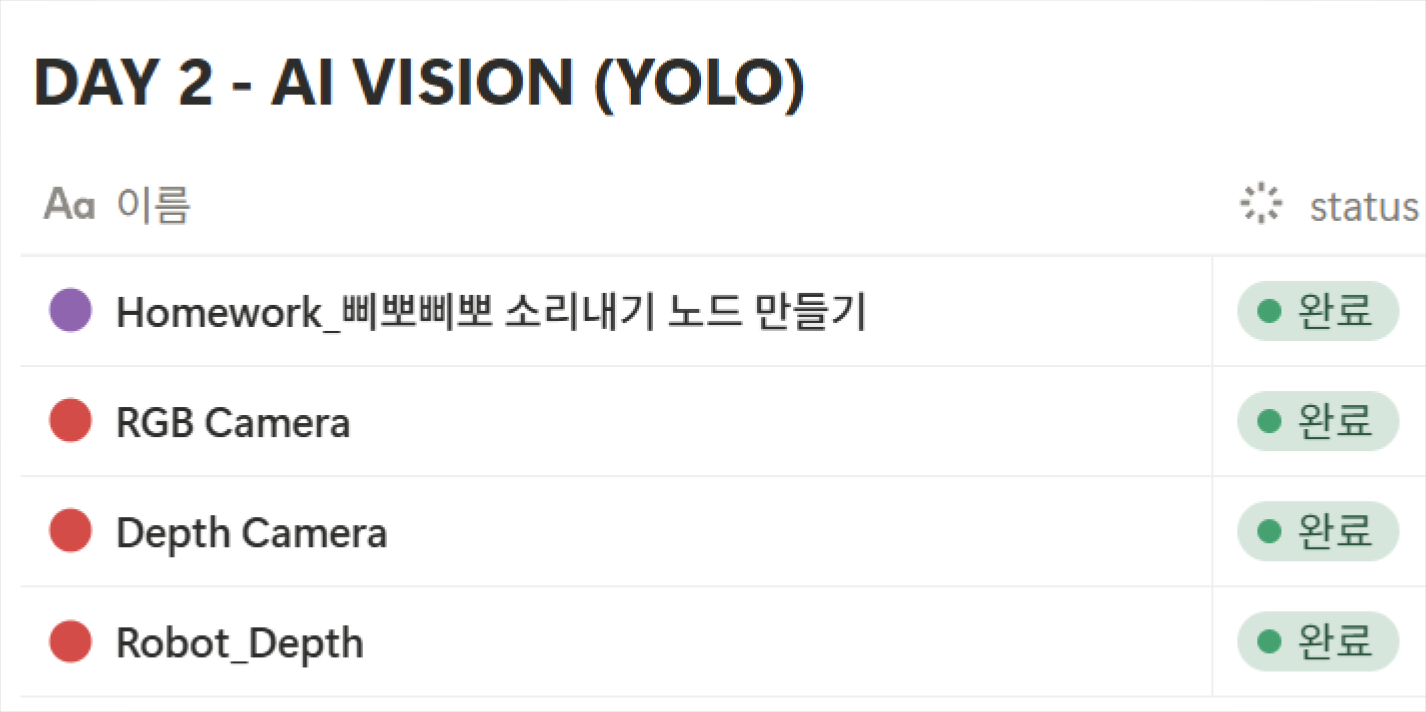


2_0_a_image_publisher.py
2_0_b_image_subscriber.py
2_0_c_data_publisher.py
2_0_d_data_subscriber.py

Try these CLI

```
$ ros2 run rqt_graph rqt_graph
$ ros2 node list
$ ros2 node info <node_name>
$ ros2 topic list
$ ros2 topic info <topic_name>
$ ros2 topic echo <topic_name>
$ ros2 interface list
$ ros2 interface show
  <package_name>/msg/<MessageName>
```

ROS EXERCISE 2



DAY 2 - AI VISION (YOLO)

Aa 이름	status
Homework_삐뽀삐뽀 소리내기 노드 만들기	● 완료
RGB Camera	● 완료
Depth Camera	● 완료
Robot_Depth	● 완료

프로젝트 RULE NUMBER ONE!!!

Are we having Fun???

DAY 2

KEY SUBSYSTEM (MODULES) TO DEVELOP

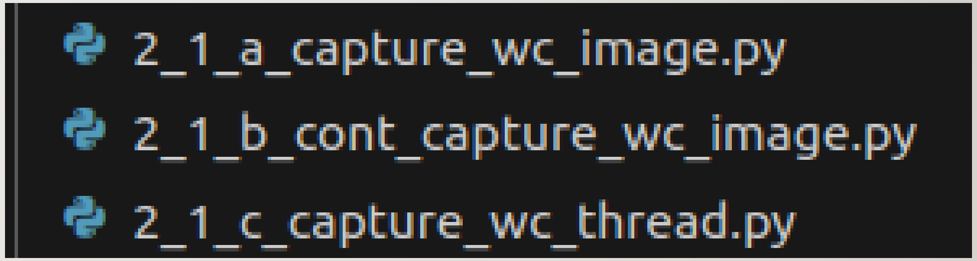
- Detection Alert
  - Camera Capture
  - Object Detection
  - Send messages to other subsystems

- AMR Controller
  - Receive messages and act accordingly
  - Move using (SLAM) with Obstruction avoidance
  - Target Acquisition (Obj. Det.) and Tracking
  - Approach target using camera and motor control

PERFORM DATA COLLECTION FOR DETECTION ALERT

COLLECTION IMAGES FROM WEBCAM

- Image Capture (WEBCAM)



2_1_a_capture_wc_image.py
2_1_b_cont_capture_wc_image.py
2_1_c_capture_wc_thread.py

COLLECTION IMAGES FROM AMR CAMERA

- Undock to see camera topics
- Are all camera topics available?
- Which image topic to use?
- Which dimensions and resolution?
- Are Depth and RGB pixel aligned?
- …

UNDOCK/DOCK AMR

UNDOCK

```
$ ros2 topic list
```

Check the list

```
$ ros2 action send_goal /robot<n>/undock irobot_create_msgs/action/Undock “{}”
```

```
$ ros2 topic list
```

Check the list and compare

DOCK

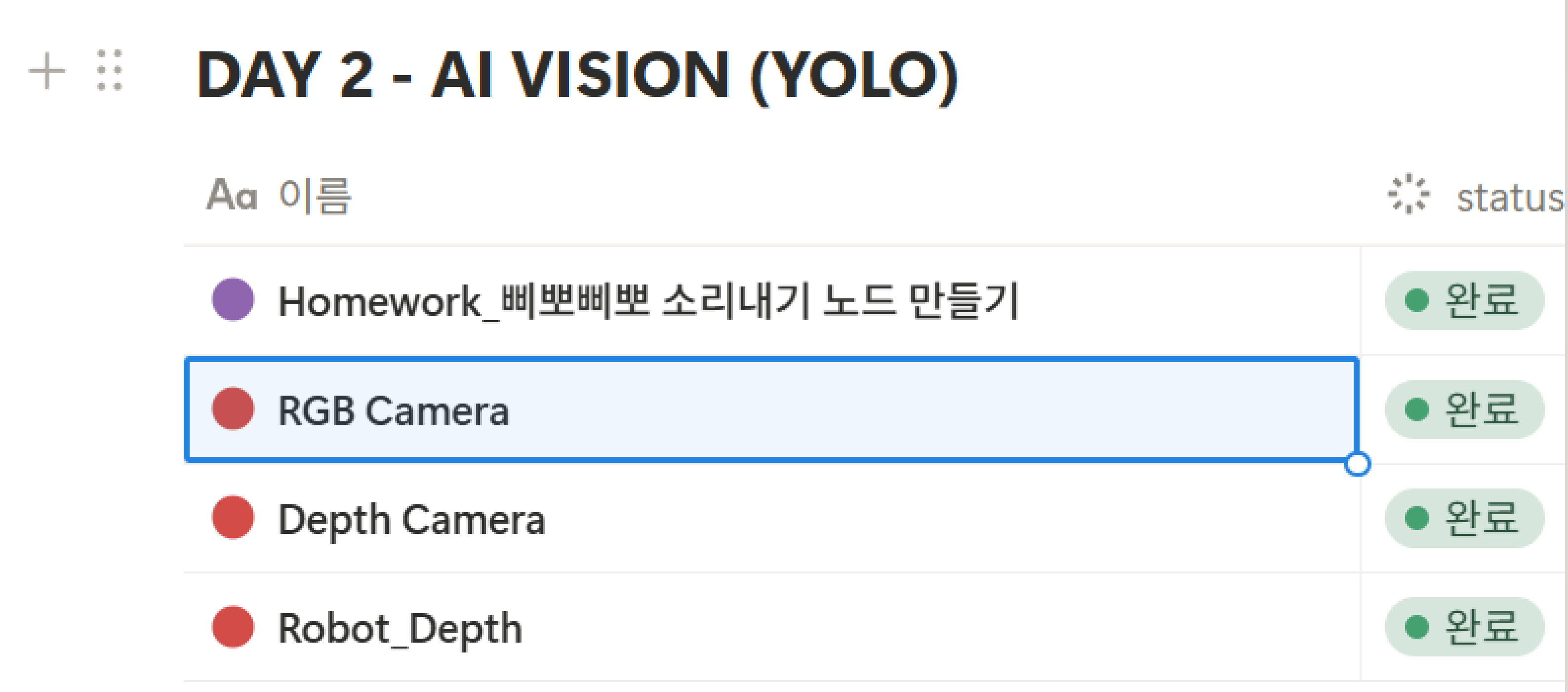
```
$ ros2 action send_goal /robot<n>/dock irobot_create_msgs/action/Dock “{}”
```

WHICH IMAGE TOPIC TO USE?

- /oakd/rgb/preview/image_raw
- /oakd/rgb/image_raw
- /oakd/rgb/image_raw/compressed
- /oakd/stereo/image_raw
- …
- ***** not all of the topics are visible, initially**

- EXERCISE
  - **Use rqt to view different image topics**
  - **Are all the topics viewable?**

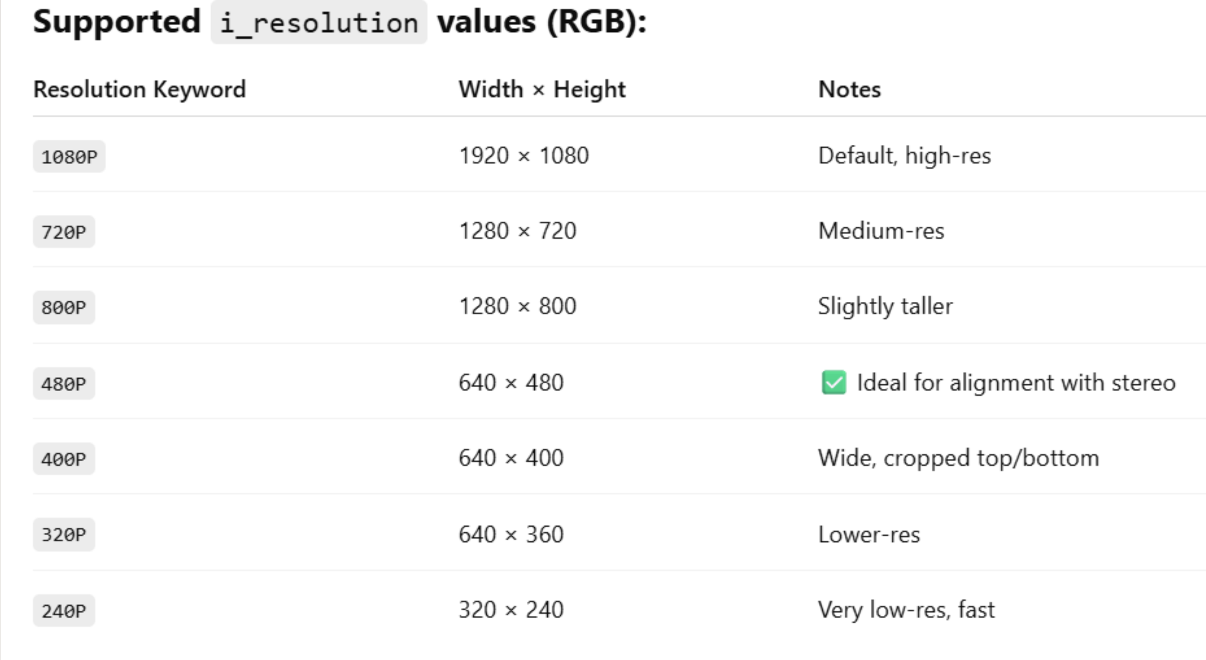
RGB CAMERA



DAY 2 - AI VISION (YOLO)

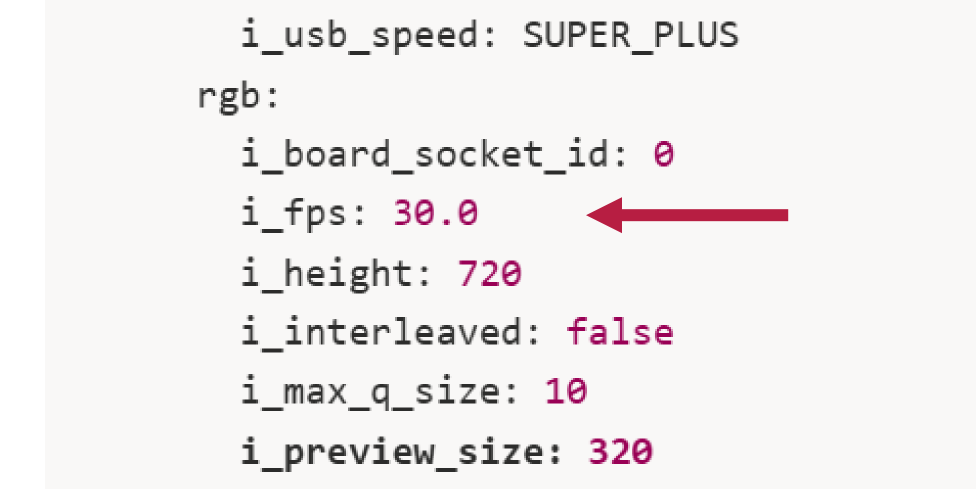
Aa 이름	status
● Homework_삐뽀삐뽀 소리내기 노드 만들기	● 완료
● RGB Camera	● 완료
● Depth Camera	● 완료
● Robot_Depth	● 완료

DIMENSIONS AND RESOLUTION AND FPS



Supported `i_resolution` values (RGB):

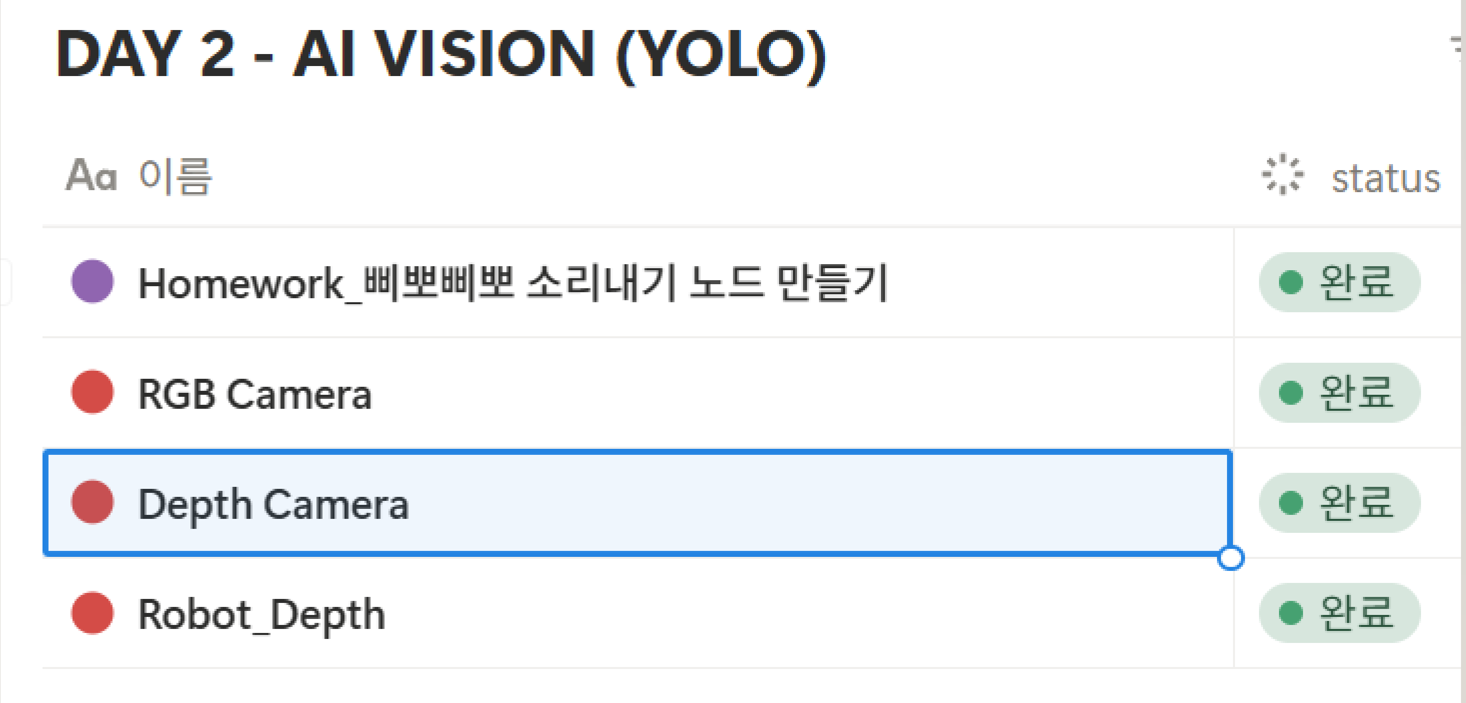
Resolution Keyword	Width × Height	Notes
`1080P`	1920 × 1080	Default, high-res
`720P`	1280 × 720	Medium-res
`800P`	1280 × 800	Slightly taller
`480P`	640 × 480	✅ Ideal for alignment with stereo
`400P`	640 × 400	Wide, cropped top/bottom
`320P`	640 × 360	Lower-res
`240P`	320 × 240	Very low-res, fast



```
    i_usb_speed: SUPER_PLUS
  rgb:
    i_board_socket_id: 0
    i_fps: 30.0
    i_height: 720
    i_interleaved: false
    i_max_q_size: 10
    i_preview_size: 320
```

Use rqt to check and compare the different image topics

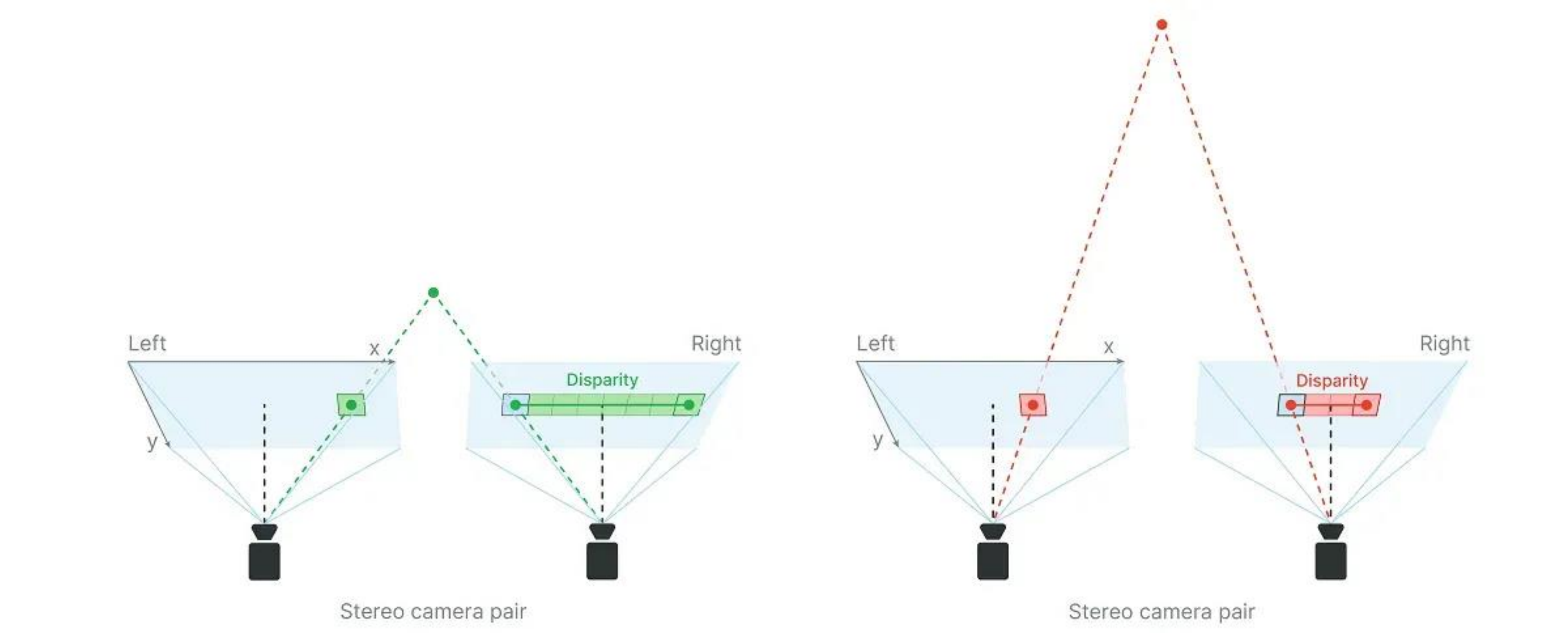
USING DEPTH



DAY 2 - AI VISION (YOLO)

Aa 이름	status
● Homework_삐뽀삐뽀 소리내기 노드 만들기	● 완료
● RGB Camera	● 완료
● Depth Camera	● 완료
● Robot_Depth	● 완료

DEPTH INTRO


Left
x
y
Right
Disparity
Stereo camera pair
Left
x
y
Right
Disparity
Stereo camera pair

UPDATING THE OAKD CONFIG (ROBOT)

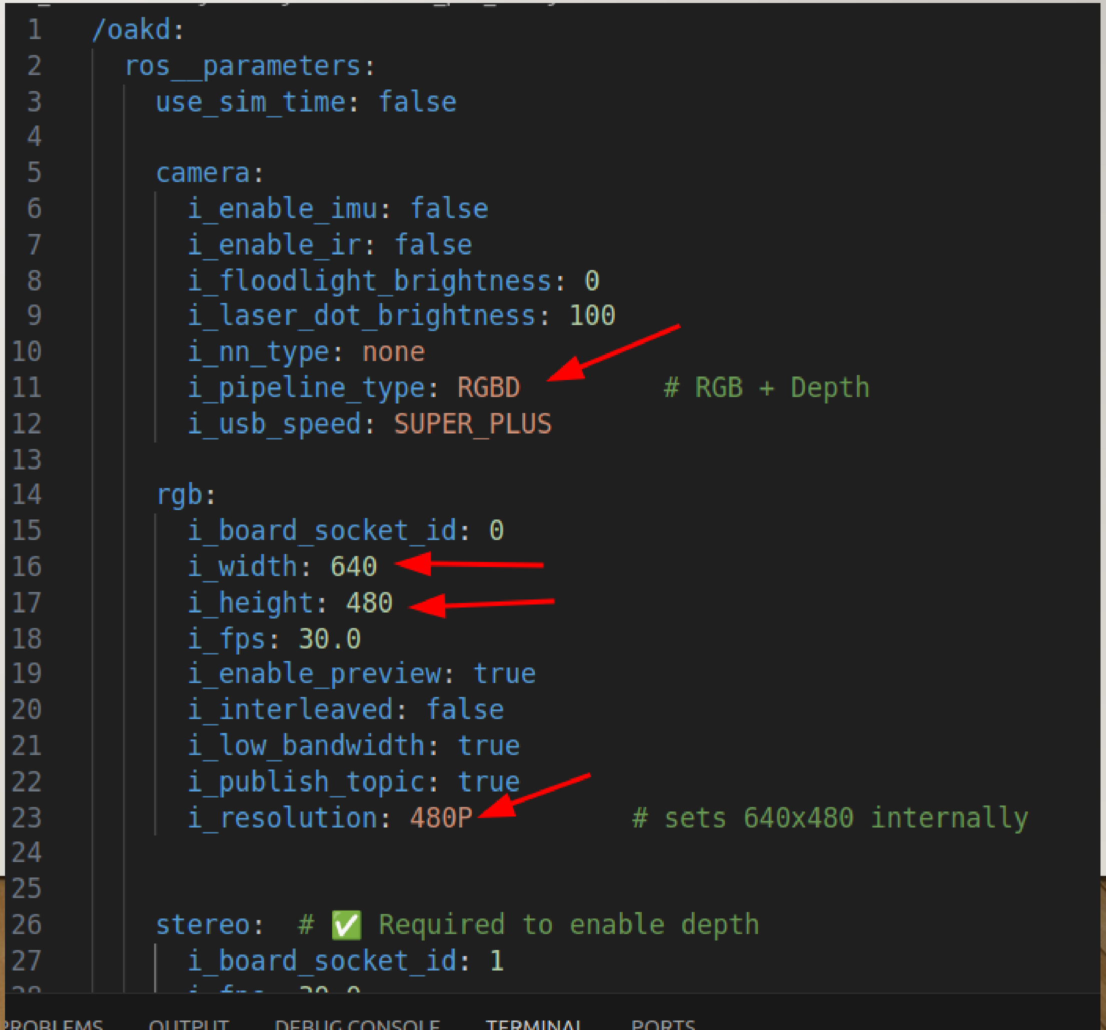
ON TURTLEBOT4:

```
$ cd /opt/ros/humble/share/turtlebot4_bringup/config
$ sudo cp oakd_pro.yaml oakd_pro_orig.yaml
$ sudo cp oakd_pro_new.yaml oakd_pro.yaml
```

```
$ turtlebot4-service-restart
```

Or

```
$ sudo reboot
```



```yaml
/oakd:
  ros__parameters:
    use_sim_time: false

    camera:
      i_enable_imu: false
      i_enable_ir: false
      i_floodlight_brightness: 0
      i_laser_dot_brightness: 100
      i_nn_type: none
      i_pipeline_type: RGBD          # RGB + Depth
      i_usb_speed: SUPER_PLUS

    rgb:
      i_board_socket_id: 0
      i_width: 640
      i_height: 480
      i_fps: 30.0
      i_enable_preview: true
      i_interleaved: false
      i_low_bandwidth: true
      i_publish_topic: true
      i_resolution: 480P          # sets 640x480 internally


    stereo:  # ✅ Required to enable depth
      i_board_socket_id: 1
```

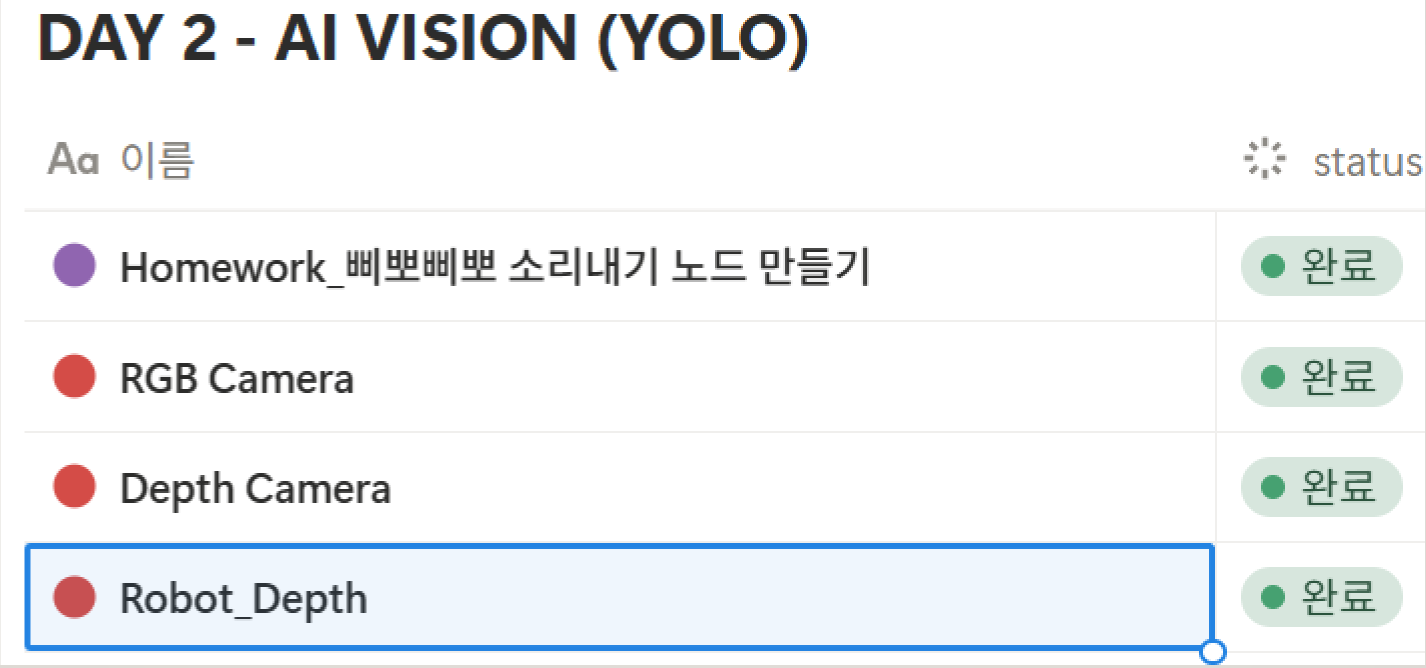
EXERCISE: ACHIEVE ALIGNED **FOV AND DIMENSION** FOR BOTH DEPTH AND RGB

Follow instruction on the notion to update the oakd config file.

Find config settings that will give same FOV for both Depth and RBG

Use rqt to view the topics

GETTING DISTANCE VALUE FROM DEPTH IMAGE



DAY 2 - AI VISION (YOLO)

Aa 이름	status
● Homework_삐뽀삐뽀 소리내기 노드 만들기	● 완료
● RGB Camera	● 완료
● Depth Camera	● 완료
● Robot_Depth	● 완료

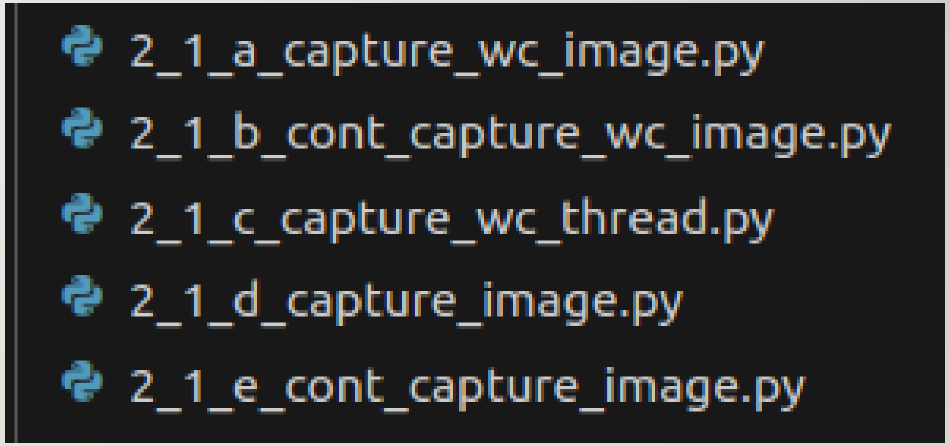
DATA COLLECTION FOR OBJ. DET.

AIM OF THE GOOD OBJ. DET. TRAINING SET

- Training images matches the real inference images as much as possible
  - Lighting
  - Dimensions
  - FOV
  - Backgrounds
  - Objects
  - …
- For RGB and Depth and …

CODING HINTS

- Image Capture
- Image Capture (AMR)

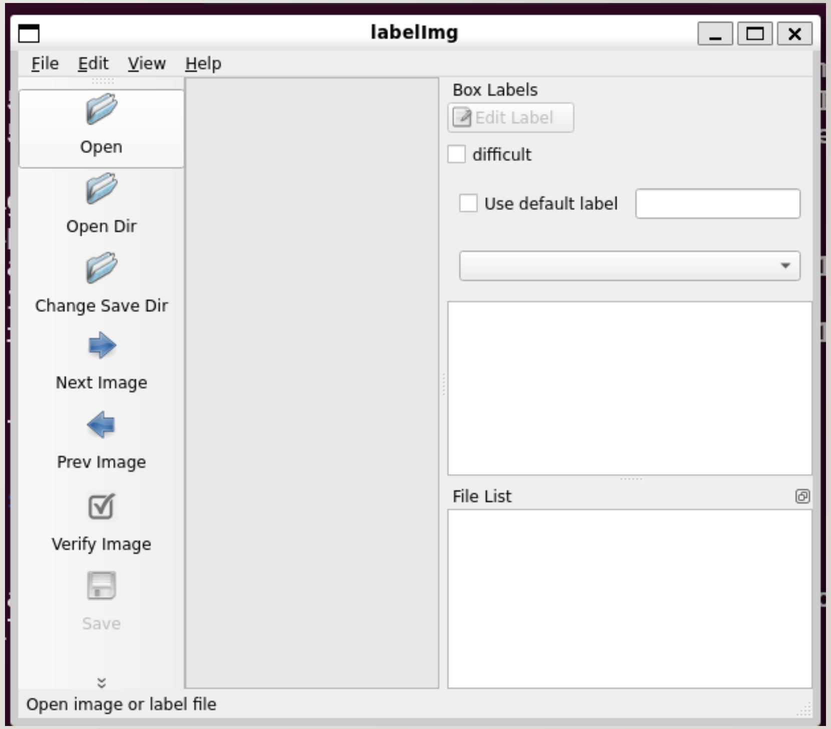


2_1_a_capture_wc_image.py
2_1_b_cont_capture_wc_image.py
2_1_c_capture_wc_thread.py
2_1_d_capture_image.py
2_1_e_cont_capture_image.py

CODING HINTS

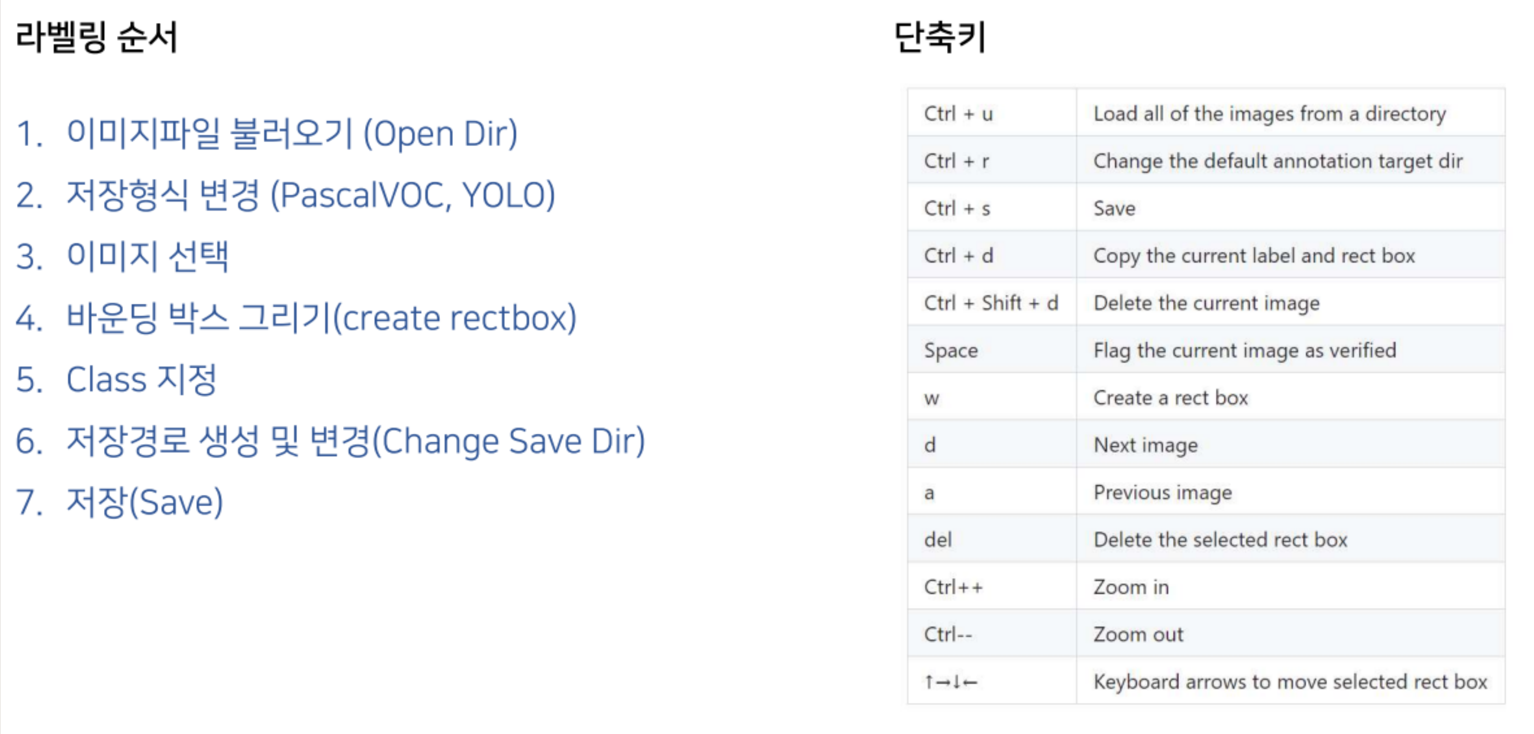
- Image Capture

- Data Labelling
  - Goto the /labelImg/data/ directory
  - Rename the predefined_classes.txt



CODING HINTS

- Data Labelling : LabelImg



라벨링 순서

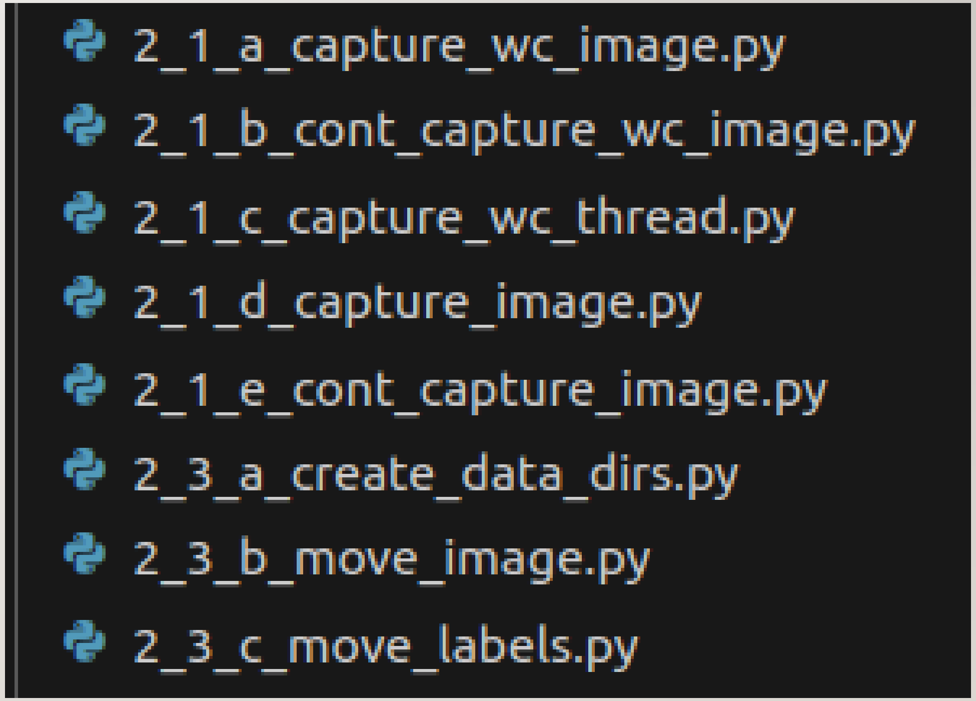
1. 이미지파일 불러오기 (Open Dir)
2. 저장형식 변경 (PascalVOC, YOLO)
3. 이미지 선택
4. 바운딩 박스 그리기(create rectbox)
5. Class 지정
6. 저장경로 생성 및 변경(Change Save Dir)
7. 저장(Save)

단축키

Ctrl + u	Load all of the images from a directory
Ctrl + r	Change the default annotation target dir
Ctrl + s	Save
Ctrl + d	Copy the current label and rect box
Ctrl + Shift + d	Delete the current image
Space	Flag the current image as verified
w	Create a rect box
d	Next image
a	Previous image
del	Delete the selected rect box
Ctrl++	Zoom in
Ctrl--	Zoom out
↑→↓←	Keyboard arrows to move selected rect box

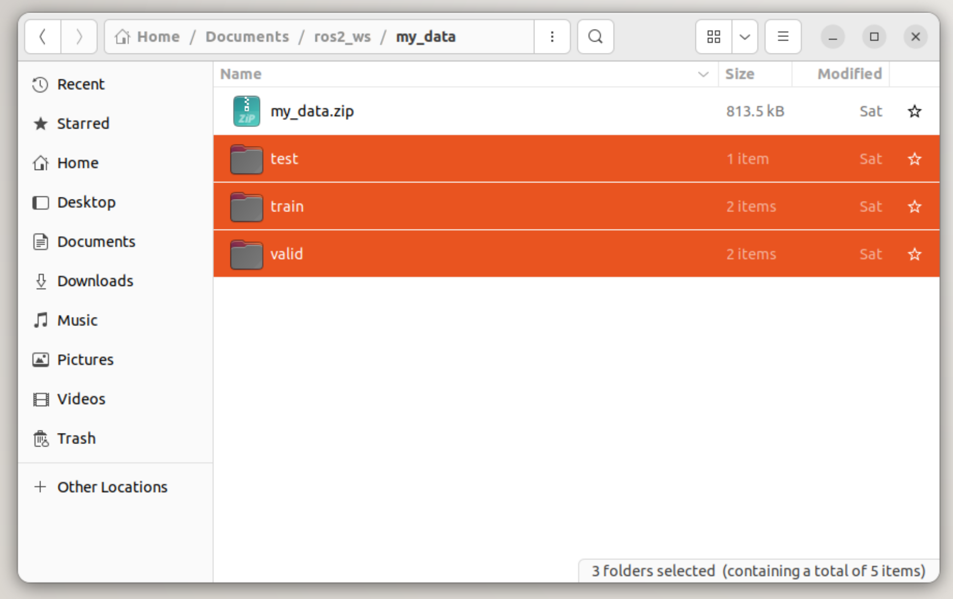
CODING HINTS

- Image Capture
- Data Labelling
- Data Preprocessing



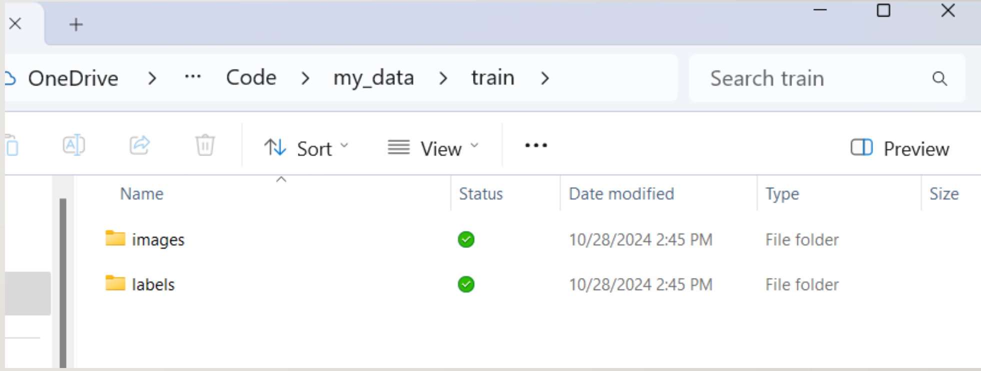
```
2_1_a_capture_wc_image.py
2_1_b_cont_capture_wc_image.py
2_1_c_capture_wc_thread.py
2_1_d_capture_image.py
2_1_e_cont_capture_image.py
2_3_a_create_data_dirs.py
2_3_b_move_image.py
2_3_c_move_labels.py
```

ZIP TRAIN DATA SET



Name	Size	Modified
my_data.zip	813.5 kB	Sat
test	1 item	Sat
train	2 items	Sat
valid	2 items	Sat

3 folders selected (containing a total of 5 items)

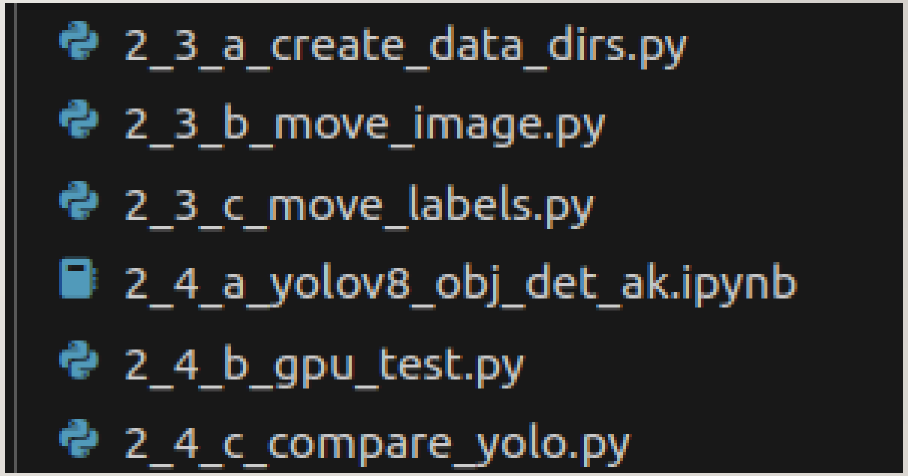


Name	Status	Date modified	Type	Size
images	✓	10/28/2024 2:45 PM	File folder	
labels	✓	10/28/2024 2:45 PM	File folder	

PERFORM YOLO TRAINING & INFERENCE

CODING HINTS

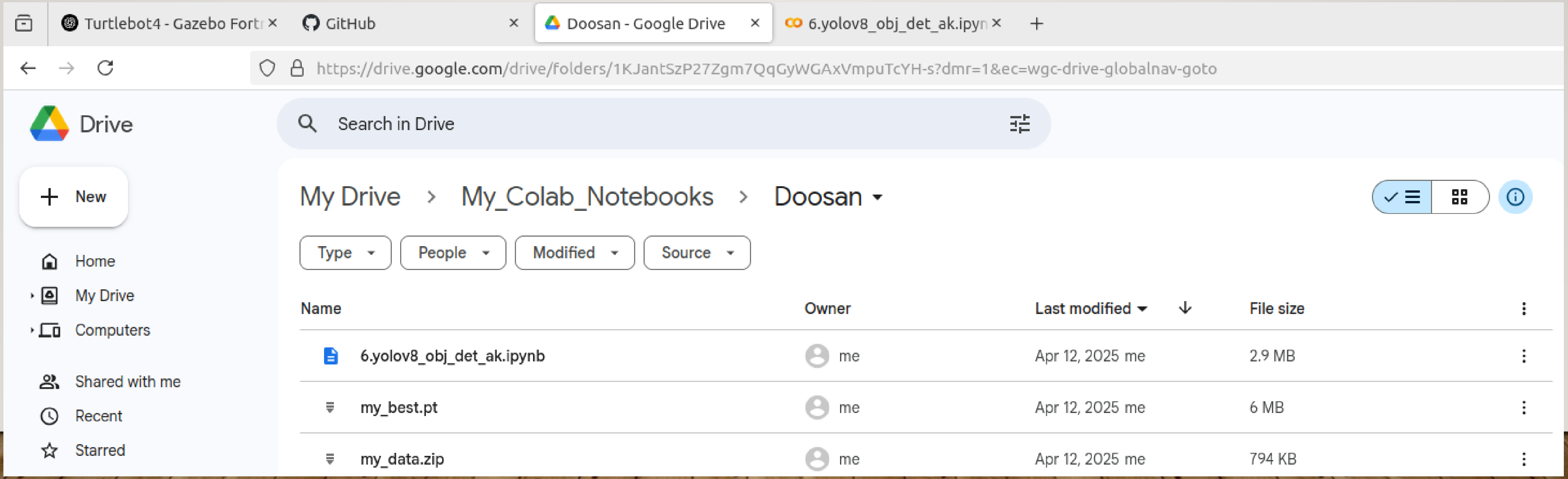
- Image Capture
- Data Labelling
- Preprocessing
- Yolo8 Object Det (Training)



2_3_a_create_data_dirs.py
2_3_b_move_image.py
2_3_c_move_labels.py
2_4_a_yolov8_obj_det_ak.ipynb
2_4_b_gpu_test.py
2_4_c_compare_yolo.py

USING GOOGLE COLLAB. TO CREATE CUSTOM MODEL

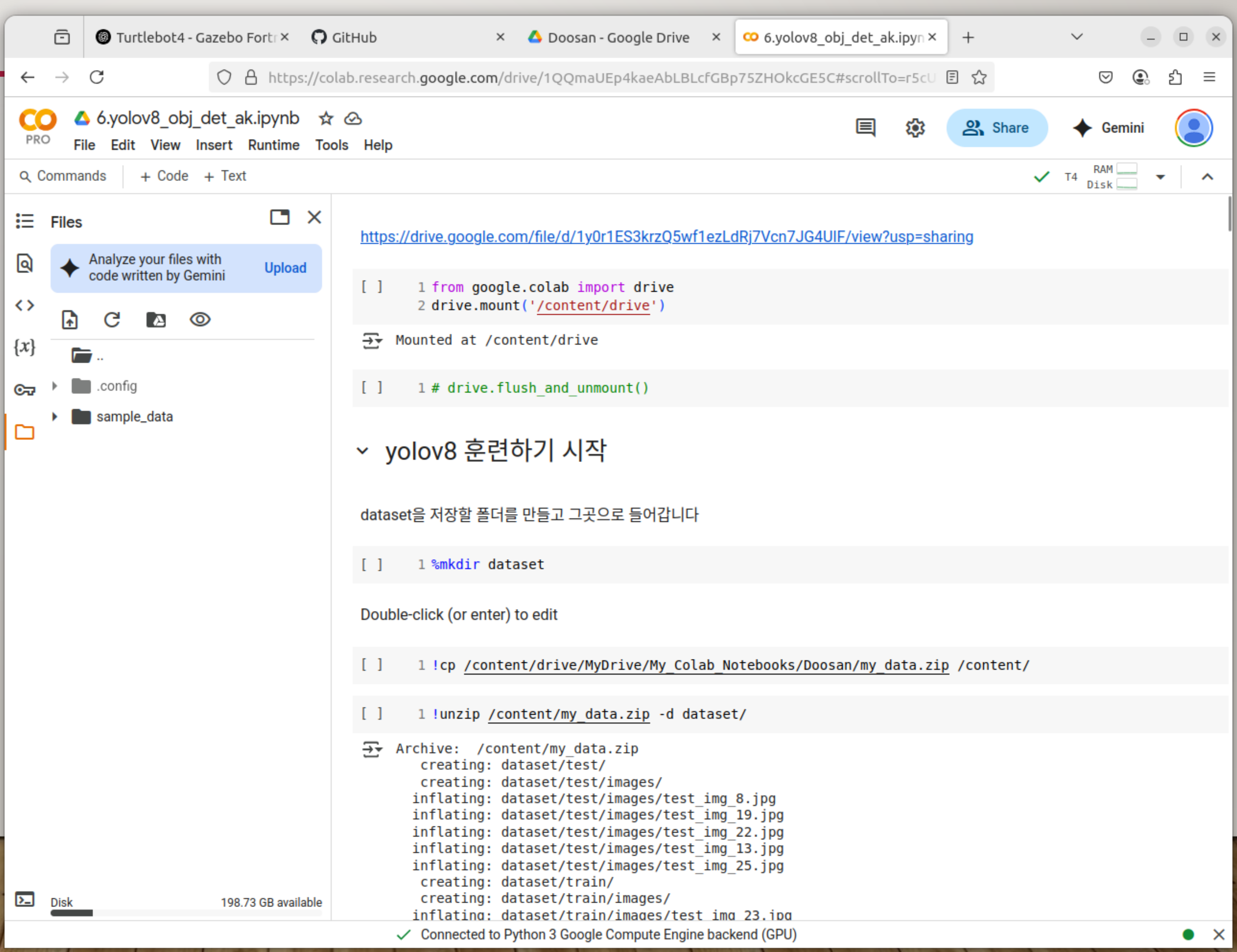
- Move the files to google drive
  - my_data.zip
  - yolov8.obj.det.ak.ipynb


Turtlebot4 - Gazebo Fortr
GitHub
Doosan - Google Drive
6.yolov8_obj_det_ak.ipyn
https://drive.google.com/drive/folders/1KJantSzP27Zgm7QqGyWGAxVmpuTcYH-s?dmr=1&ec=wgc-drive-globalnav-goto
Drive
Search in Drive
New
Home
My Drive
Computers
Shared with me
Recent
Starred
My Drive › My_Colab_Notebooks › Doosan
Type
People
Modified
Source

Name	Owner	Last modified	File size
6.yolov8_obj_det_ak.ipynb	me	Apr 12, 2025 me	2.9 MB
my_best.pt	me	Apr 12, 2025 me	6 MB
my_data.zip	me	Apr 12, 2025 me	794 KB

USING GOOGLE COLLAB. TO CREATE CUSTOM MODEL

- Move the training script to google collab. and execute line by line


Turtlebot4 - Gazebo Fort
GitHub
Doosan - Google Drive
6.yolov8_obj_det_ak.ipyn
https://colab.research.google.com/drive/1QQmaUEp4kaeAbLBLcfGBp75ZHOkcGE5C#scrollTo=r5cU
PRO
6.yolov8_obj_det_ak.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
T4
RAM
Disk
Files
Analyze your files with code written by Gemini
Upload
..
.config
sample_data
Disk
198.73 GB available

https://drive.google.com/file/d/1y0r1ES3krzQ5wf1ezLdRj7Vcn7JG4UIF/view?usp=sharing

```
[ ]  from google.colab import drive
     drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ]  # drive.flush_and_unmount()
```

yolov8 훈련하기 시작

dataset을 저장할 폴더를 만들고 그곳으로 들어갑니다

```
[ ]  %mkdir dataset
```

Double-click (or enter) to edit

```
[ ]  !cp /content/drive/MyDrive/My_Colab_Notebooks/Doosan/my_data.zip /content/
```

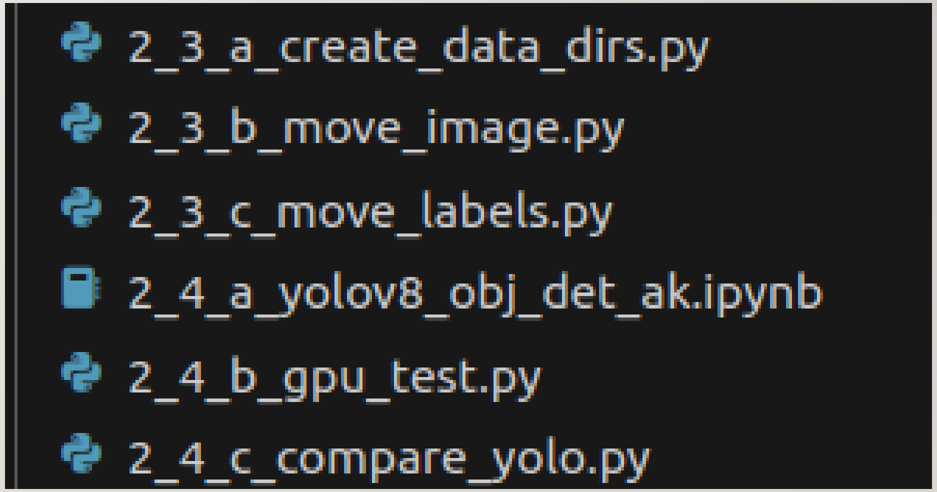
```
[ ]  !unzip /content/my_data.zip -d dataset/
```

```
Archive:  /content/my_data.zip
   creating: dataset/test/
   creating: dataset/test/images/
  inflating: dataset/test/images/test_img_8.jpg
  inflating: dataset/test/images/test_img_19.jpg
  inflating: dataset/test/images/test_img_22.jpg
  inflating: dataset/test/images/test_img_13.jpg
  inflating: dataset/test/images/test_img_25.jpg
   creating: dataset/train/
   creating: dataset/train/images/
  inflating: dataset/train/images/test_img_23.jpg
```

Connected to Python 3 Google Compute Engine backend (GPU)

CODING HINTS

- Image Capture
- Data Labelling
- Preprocessing
- Yolo8 Object Det (Analyze)



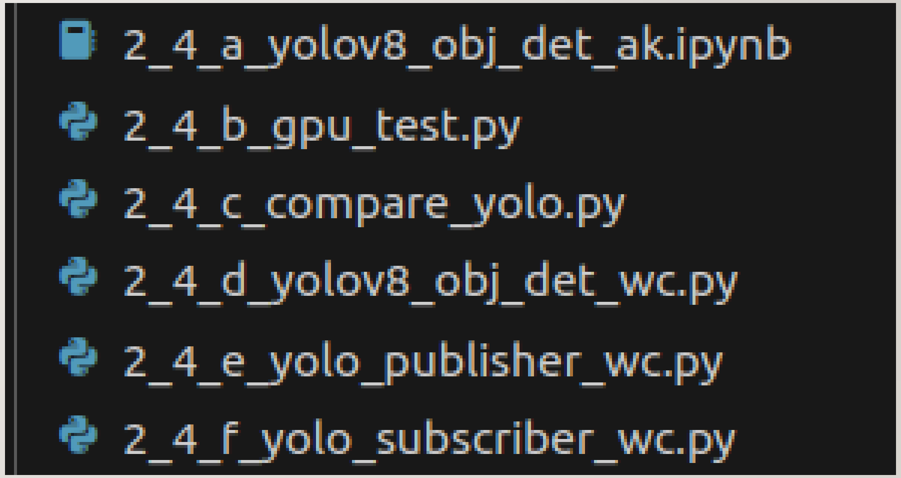
```
2_3_a_create_data_dirs.py
2_3_b_move_image.py
2_3_c_move_labels.py
2_4_a_yolov8_obj_det_ak.ipynb
2_4_b_gpu_test.py
2_4_c_compare_yolo.py
```

REQUIRED RESEARCH

1. **왜 yolov8n.pt 모델을 선정하였습니까?**
    - yolo 다른 버전과 비교 분석
    - mAP VS Inference speed
2. **객체 탐지 속도를 높이기 위한 최선의 전략은?**
    - 데이터 사이즈
    - processing 방식
    - 노드 구조
3. **다른 Pre-trained model(Huggingface)을 사용한다면?**
4. **object detection이 아닌 segmentation, pose, obb 등을 활용할 수 없을까?**

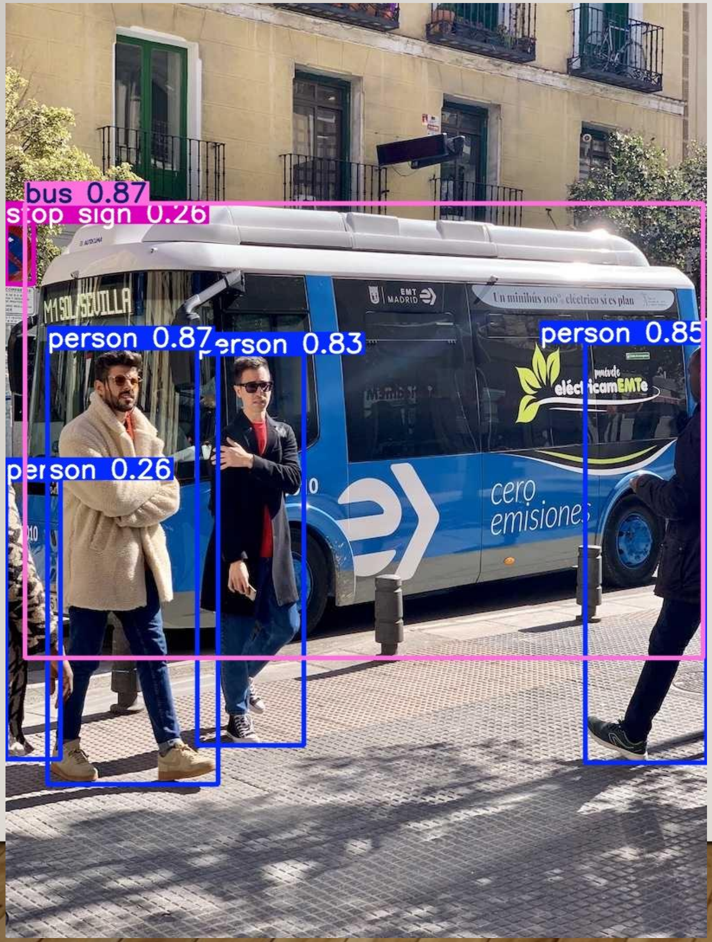
CODING HINTS

- Image Capture
- Data Labelling
- Preprocessing
- Yolo8 Object Det (WEBCAM)

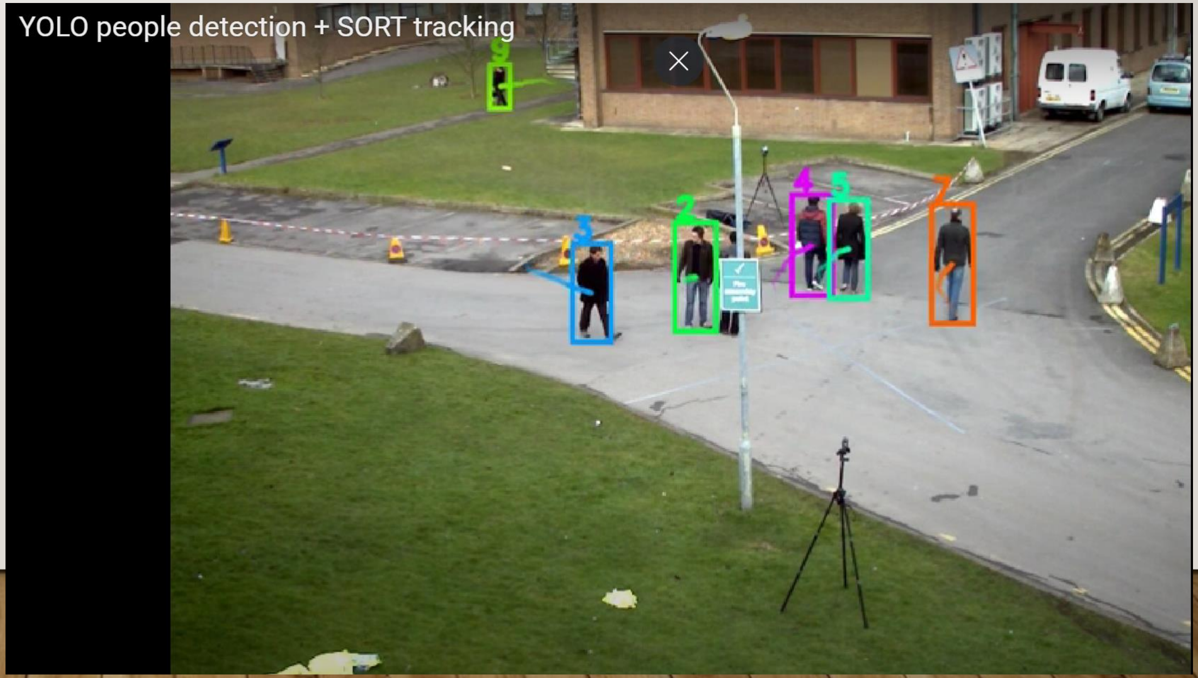


2_4_a_yolov8_obj_det_ak.ipynb
2_4_b_gpu_test.py
2_4_c_compare_yolo.py
2_4_d_yolov8_obj_det_wc.py
2_4_e_yolo_publisher_wc.py
2_4_f_yolo_subscriber_wc.py

YOLO OBJ. DET. VS. YOLO TRACKING

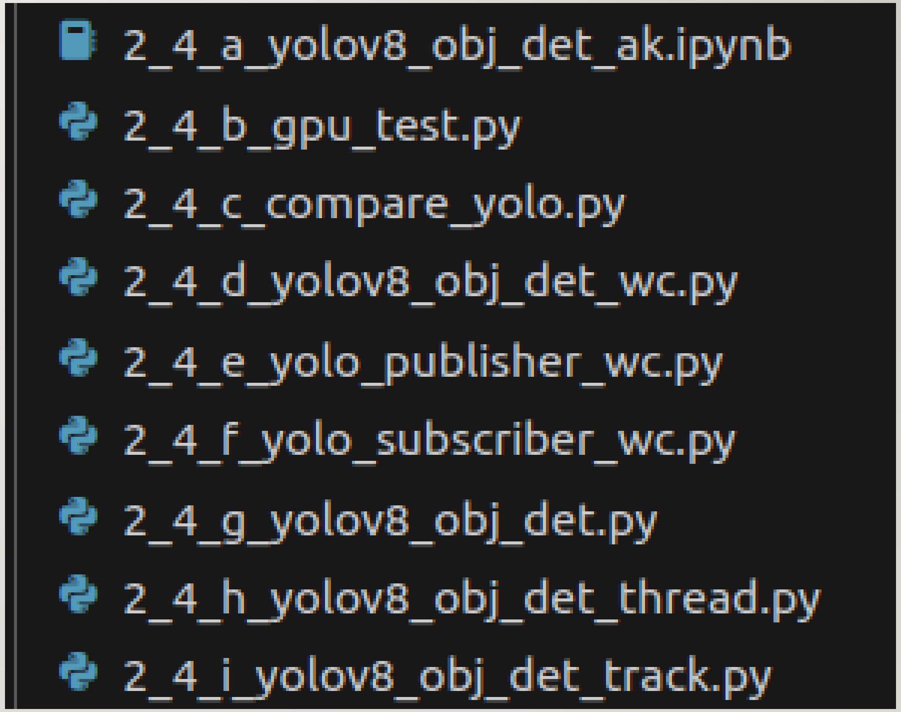


- Track - Ultralytics YOLO Docs
  - (469) YOLO people detection + SORT tracking – YouTube
  - Bing Videos



CODING HINTS

- Image Capture
- Data Labelling
- Preprocessing
- Yolo8 Object Det (AMR)



```
2_4_a_yolov8_obj_det_ak.ipynb
2_4_b_gpu_test.py
2_4_c_compare_yolo.py
2_4_d_yolov8_obj_det_wc.py
2_4_e_yolo_publisher_wc.py
2_4_f_yolo_subscriber_wc.py
2_4_g_yolov8_obj_det.py
2_4_h_yolov8_obj_det_thread.py
2_4_i_yolov8_obj_det_track.py
```

OBJECT DETECTION EXERCISE

Create a version of real time yolo inference code that **publish a ROS topic with annotated image of detection results and view it with rqt or rviz**

HOMEWORK

- **Achieve aligned RGB & Depth FOV**
- Object Detection
  - Collect various datasets (i.e. different topics/images sizes)
  - Create various models (i.e. v5, v8, v11, etc; arg: Epoch, Batch, Imgsz, augmentation, etc)
  - Analyze the results
  - Determine using key metrics which model best fit your solution
  - Using .pt file to predict/inference on pc
  - **Successfully publish the annotated image topic**
- Depth
  - **Find and display the distance to the center of the detected objects**
- Update System Requirement

프로젝트 RULE NUMBER ONE!!!

Are we still having FUN!