

Signal Masters

교통 수신호 인식

시스템

Team. Signal Masters 서창민, 박준수, 김도하

목 차

01 프로젝트 개요

02 팀 구성 및 역할

03 시스템 개발

04 결과 및 시연

05 마무리

1. 프로젝트 개요

1

문제 인식

2

개발 목표

3

개발 환경

1. 문제 인식

'차량 고장' 수신호 하다 참변...뒷차에 치여 사망

자동차전용도로서 '차 고장 수신호' 하던 60대, SUV에 치여 사망

입력 2024.09.18 | 송고시간 | 2024-03-1

임채우 기자

용담터널 내 차량 고장 수신호 하던 40대 차량에 치여 사망

기사입력 : 2024년09월18일 20:44 | 최종수정 : 2024년09월18일 21:19

가+ 가- 프린트

[안산=뉴스핌] 박승봉 기자 = 18일 오전 4시쯤 경기 안산시 상록구 서해안고속도로 목포 방향 용담터널 내에서 차량 고장 수신호를 하던 40대 남성이 차량에 치여 숨지는 사고가 발생했다.

(남원=연합뉴스)
자를 차로 치어 숨
2일 밝혔다.

A씨는 전날 오후 6시경 차량 고장 수신호를 하다가 차량에 치여 숨졌다.



경찰로고 [사진=뉴스핌DB]

뉴스핌

2. 개발 목표



- 자율주행 RC카 구현
- 초음파 센서로 장애물 회피



- 포즈 인식 모델을 이용하여 수신호 인식
- 수신호에 따라 RC카 제어

3. 개발 환경

개발 도구



PoseC3D



영상 관리

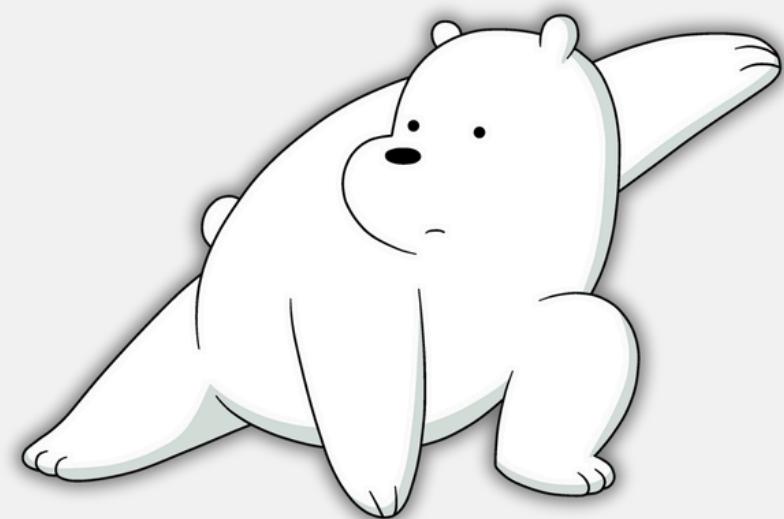


2. 팀 구성 및 역할

1

팀원 및 역할

1. 팀원 및 역할



서창민
조장

- 프로젝트 총 관리
- H/W 개발 및 제작
- Dataset 라벨링



박준수

- 개발환경 구현
- AI Modeling
- S/W 개발 및 연동
- 모델 최적화



김도하

- AI Modeling
- Dataset 구축

3. 시스템 개발

1

개발 일정

2

H/W 제작

3

S/W 개발

4

문제점 / 해결

5

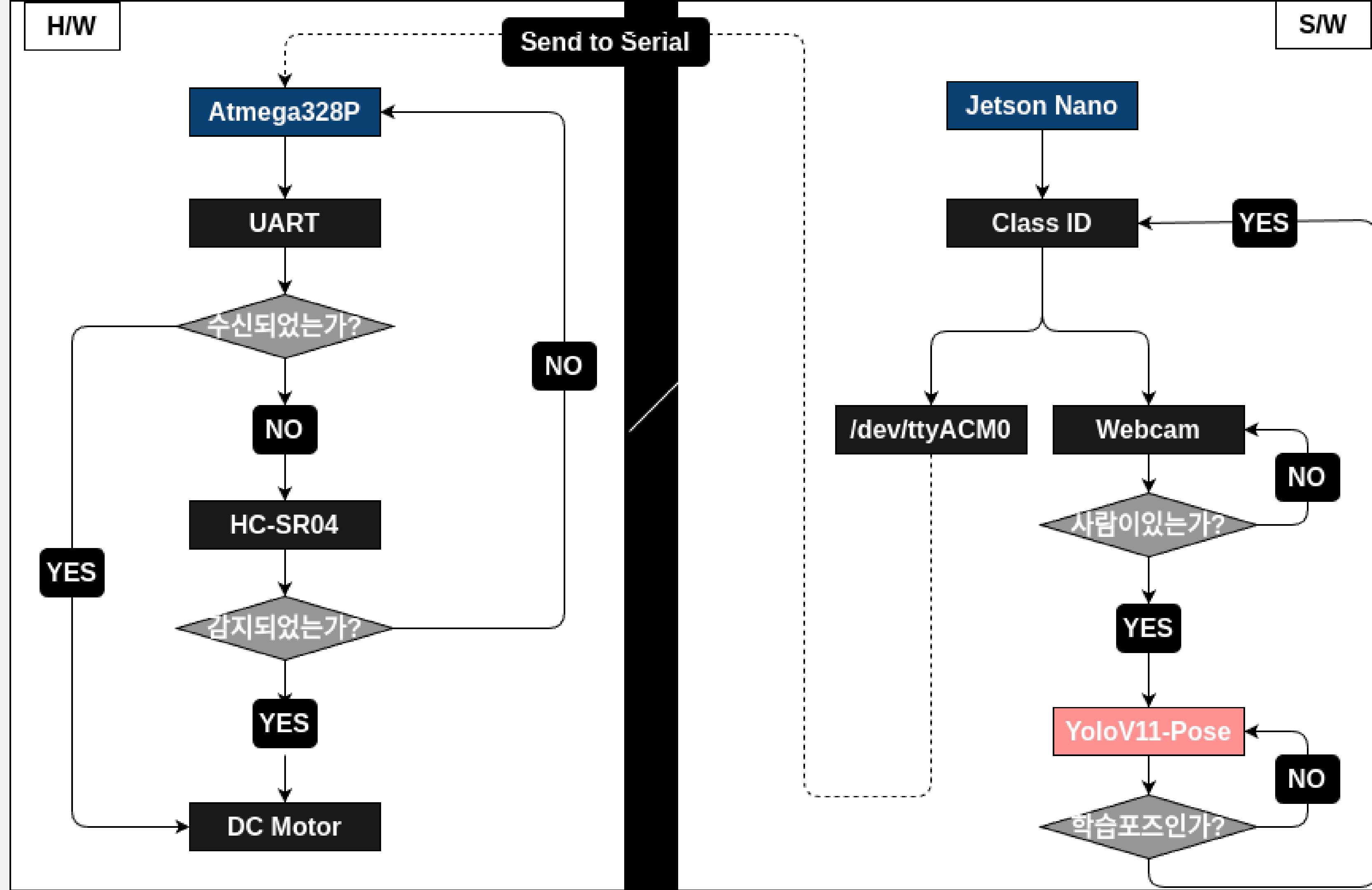
시스템 구조

1. 개발 일정

9월 23일 ~ 10월 23일

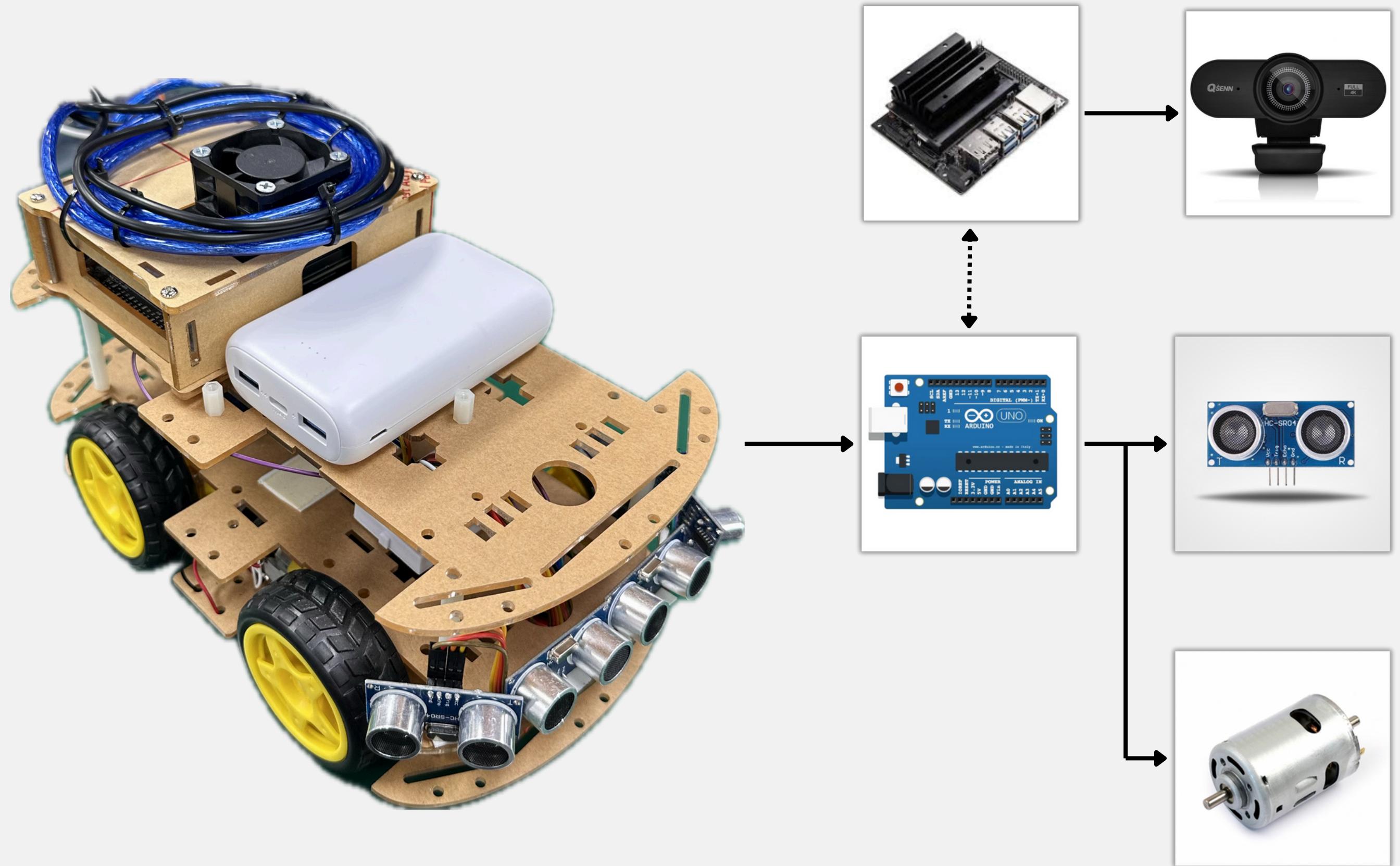
내용	개발 일정																			
계획	Dataset 찾기 및 모델 학습 가능 여부 분석 딥러닝 모델 조사 및 기자재 리스트업																			
	정보 수집						Human Detection AND Pose Estimation Action Recognition 추론 AND Model Test						Model Training AND Model Test RC Car Build UP							
메인 개발													소스코드 병합 및 작동 테스트 모델 양자화 및 연동							
																Github 및 PPT 준비 발표 준비				
세부 날짜	9월 23일	9월 24일	9월 25일	9월 26일	9월 27일	9월 30일	10월 2일	10월 4일	10월 7일	10월 8일	10월 10일	10월 11일	10월 14일	10월 15일	10월 16일	10월 17일	10월 18일	10월 21일	10월 22일	10월 23일

FlowChart



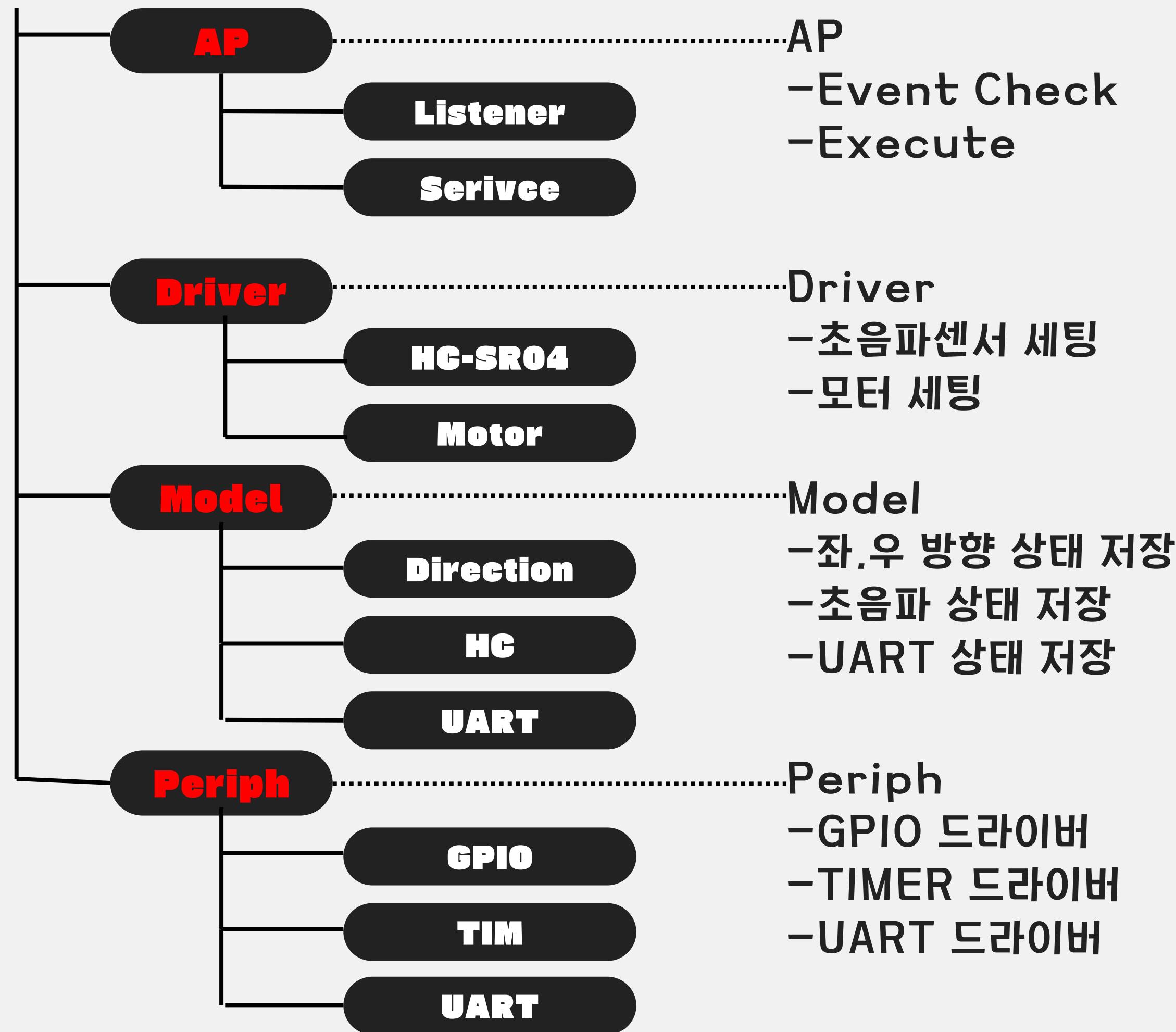
2.H/W 개발

H/W 설계도



2. H/W 개발

H/W 아키텍쳐



AP

- Event Check
- Execute

Driver

- 초음파센서 세팅
- 모터 세팅

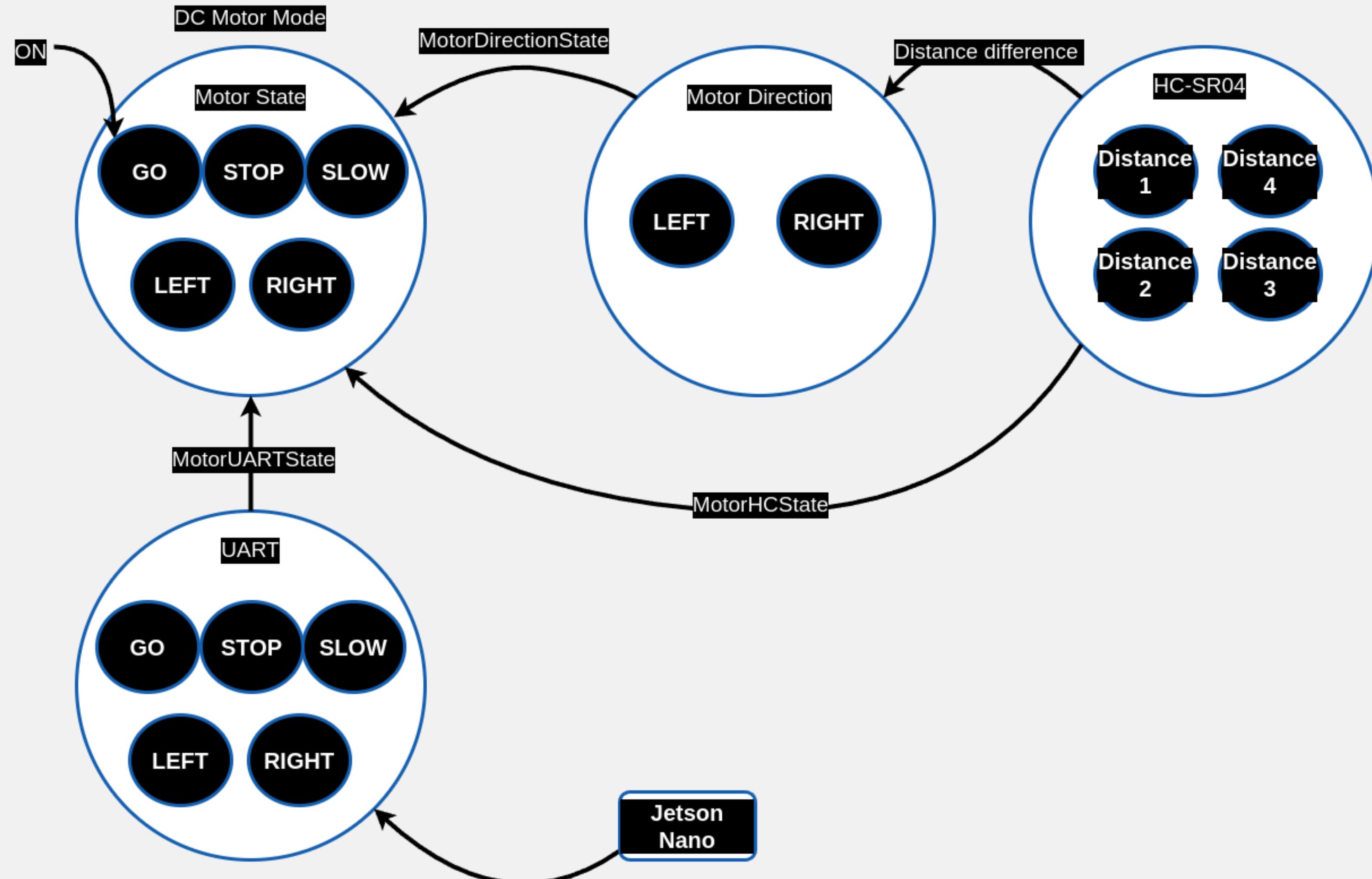
Model

- 좌,우 방향 상태 저장
- 초음파 상태 저장
- UART 상태 저장

Periph

- GPIO 드라이버
- TIMER 드라이버
- UART 드라이버

H/W FSM



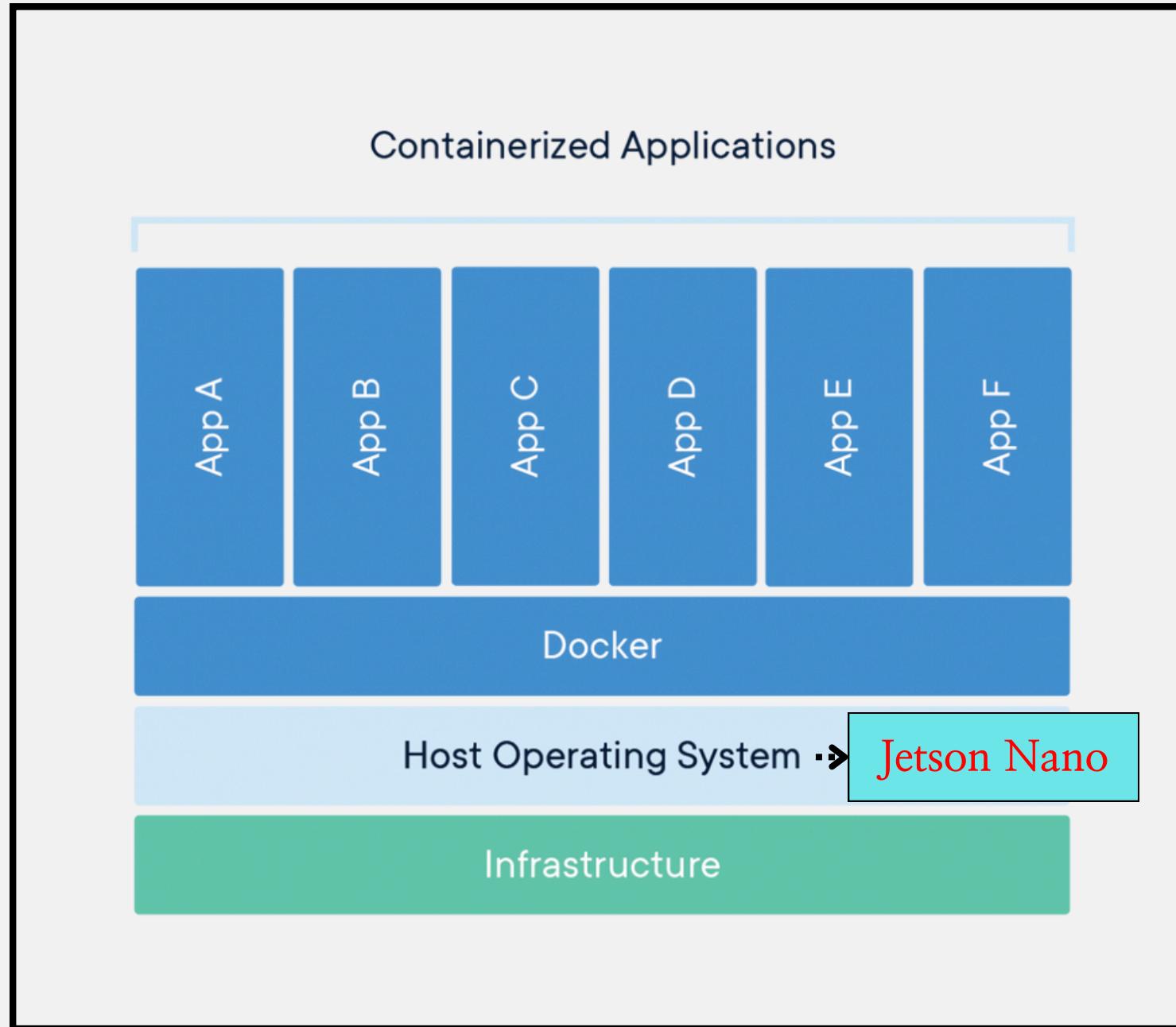
3.S/W 개발

* Challenges: 개발/배포 환경

- 현재 사용중인 **개발 환경**과 **배포 환경**이 동일한가?
 - Arm64 (AArch64) vs Amd64 (x86-64)
- 작성한 프로그램과 배포 환경에서의 시스템 패키지와의 **호환성**
- **Code Quality Tools**: Formatter, Linter
- **Reproducibility**
 - 팀원들의 PC 를 포함한 다른 PC 에서도 동일하게, 정상적으로 작동해야 함

* Reproducibility

Container

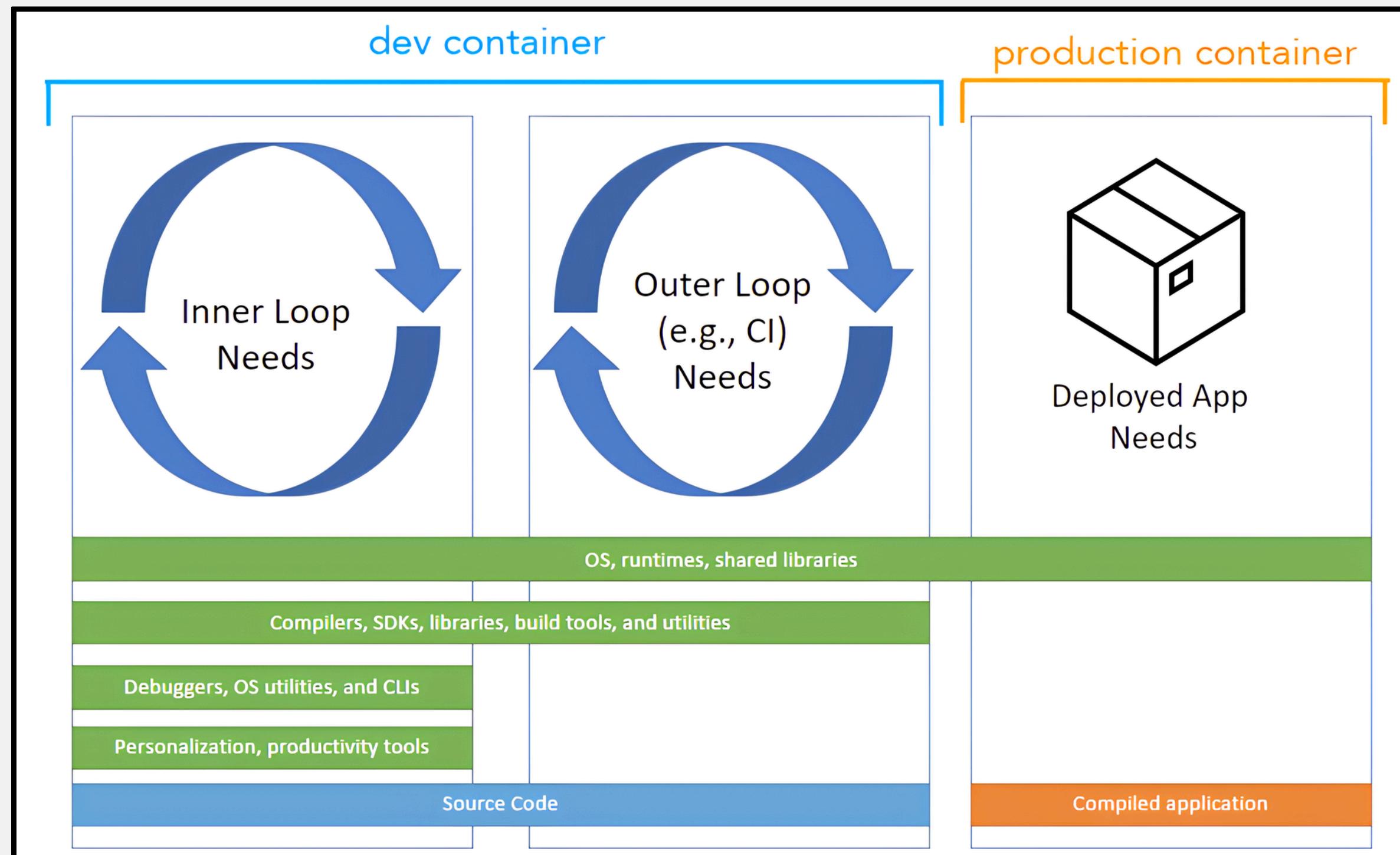


* Challenges

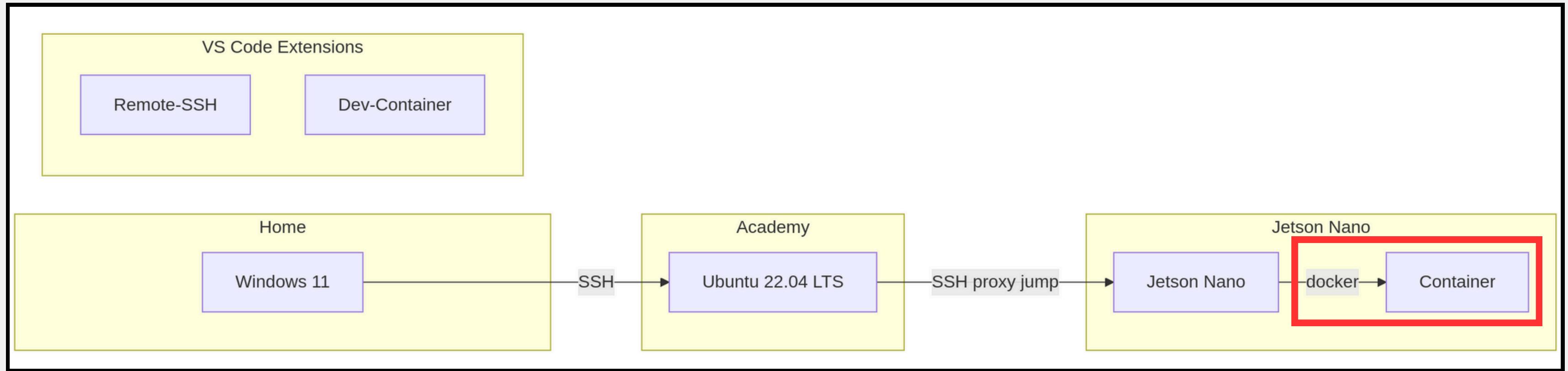
- 컨테이너에서 **GPU 장치**를 사용할 수 있는가?
→ NVIDIA 컨테이너 런타임 사용. GPU 접근 허용 및 할당
- 컨테이너에서 모델 추론 시 **GPU 장치**를 사용할 수 있는가?
 - version of **cuda**
 - version of **torch, ultralytics (YOLO) dependency**
→ **.whl** 패키지 설치
- 컨테이너에서 Host OS 의 **주변장치 (비디오, MCU)** 를 사용할 수 있는가?
- 컨테이너에서 root 가 아닌 사용자가 **주변장치 (비디오, MCU)** 를 사용할 수 있는가?
→ 사용자가 속한 group 추가: **video, dialout**
- 도커 네트워크와 외부 프로세스와 통신

* Reproducibility

Development Containers Stages



원격 개발 흐름도



* Challenges: ML 모델 파이프라인

- 모델 입력/출력 간의 Shape 일치
 - Object Detection
 - 사람 인식
 - Pose Estimation
 - 탐지된 사람의 포즈 추정
 - Multiple Object Tracking (**MOT**)
 - 동일한 사람에 대해 고유한 id 부여
 - Action Recognition
 - 동일한 id 별로 행동 인식
- 하드웨어 제약사항
 - Jetson nano: **472 GFLOPS** (0.472 TFLOPS)

* Pose Estimation

초기 계획

- Yolo-NAS
 - Object Detection
- Mediapipe - Object Detection + MMPose (Toolbox)
- Mediapipe - Pose Estimation
 - Output: 3x5 array corresponding to (x, y, z, visibility, presence)
 - for Single-person

- ▶ API 문서화가 잘 되어 있는 라이브러리
- ▶ 여러 명을 한 번에 동시 탐지할 수 있는 모델
- ▶ 포즈 추정 모델로부터 나온 포즈 데이터를 행동 인식 모델의 입력에 그대로 사용이 가능한가?
- ▶ 하드웨어 제약사항을 고려하였는가?

* Yolo Pose v11 n

Performance Metrics

Performance

Detection (COCO) Segmentation (COCO) Classification (ImageNet) **Pose (COCO)** OBB (DOTAv1)

See [Pose Estimation Docs](#) for usage examples with these models trained on [COCO](#), which include 1 pre-trained class, 'person'.

Model	size (pixels)	mAP _{pose} 50-95	mAP _{pose} 50	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n-pose	640	50.0	81.0	52.4 ± 0.5	1.7 ± 0.0	2.9	7.6
YOLO11s-pose	640	58.9	86.3	90.5 ± 0.6	2.6 ± 0.0	9.9	23.2
YOLO11m-pose	640	64.9	89.4	187.3 ± 0.8	4.9 ± 0.1	20.9	71.7
YOLO11l-pose	640	66.1	89.9	247.7 ± 1.1	6.4 ± 0.1	26.2	90.7
YOLO11x-pose	640	69.5	91.1	488.0 ± 13.9	12.1 ± 0.2	58.8	203.3

* Fine-tuning

1. Dataset 구성

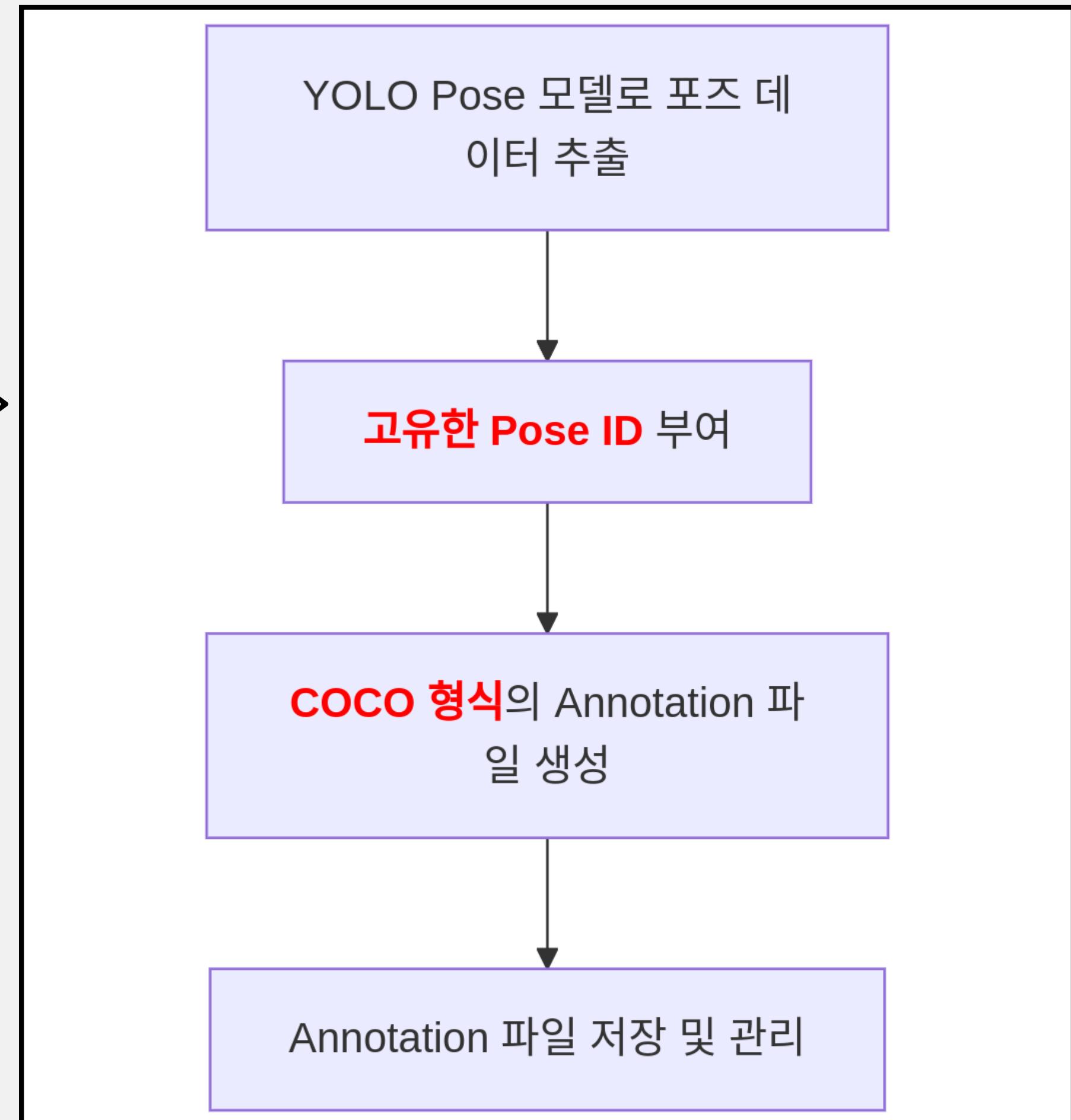
a. Labeling →

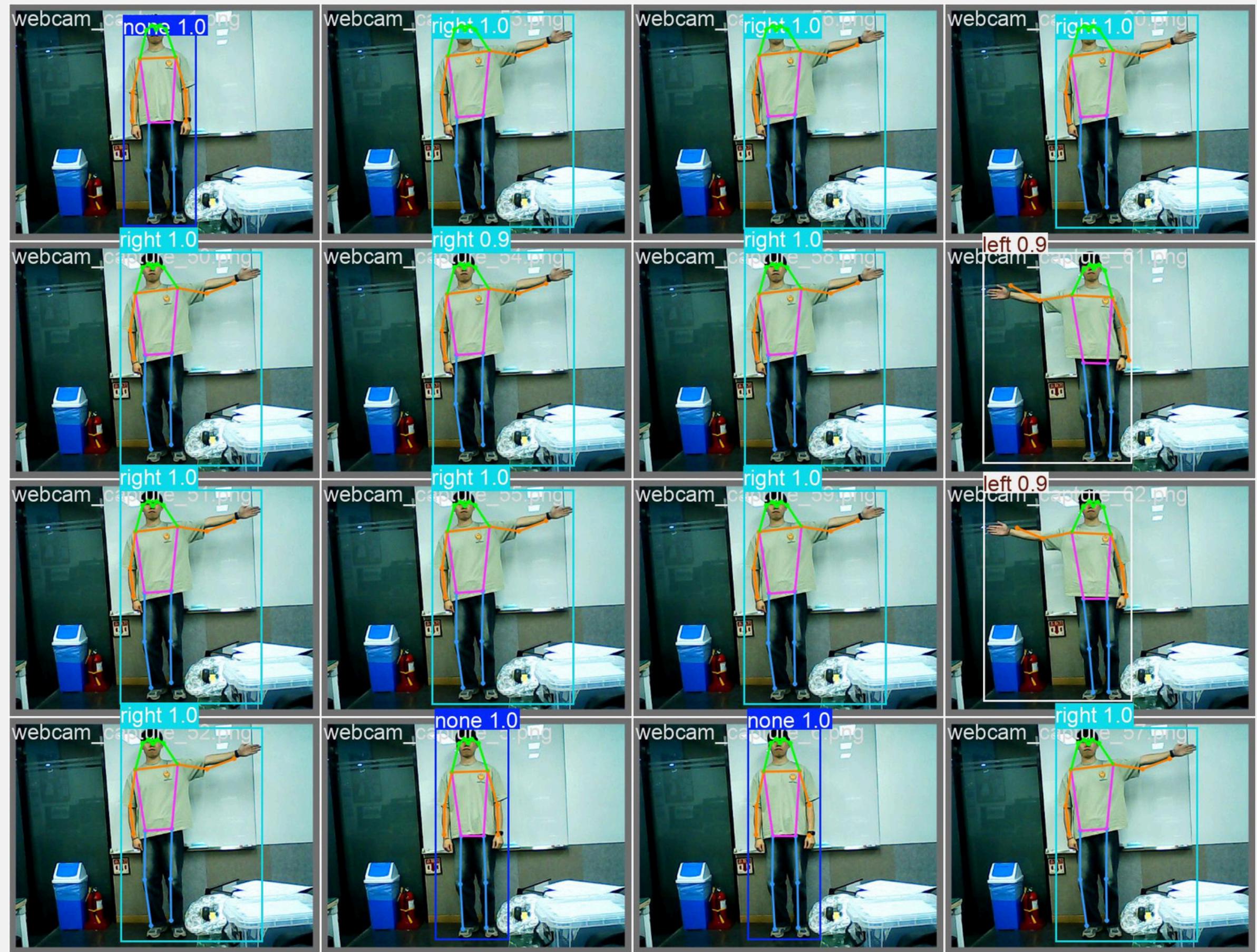
0: “Go”, 1: “Left”, 2: “Right”, 3: “Stop”, 4: “Slow”

1. Train

2. Test

3. Optimization





교통 수신호 인식 시스템

* Optimization

* Action Recognition

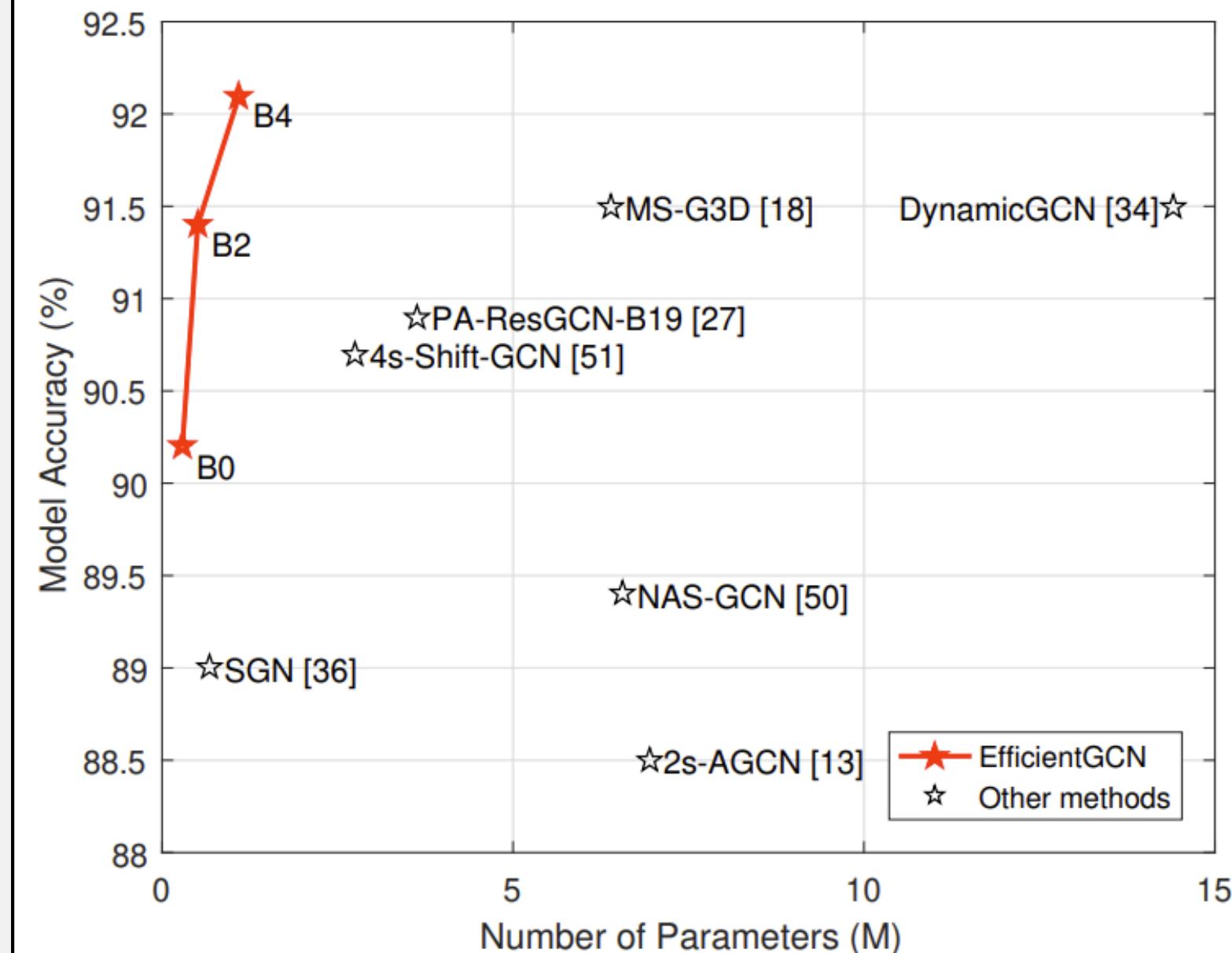
초기 계획

- EfficientGCN

- API 문서화가 잘 되어 있는 라이브러리
- 하드웨어 제약사항을 고려하였는가?
비전 기반이 아닌, 골격 기반
(Joint, Velocity, Bone) 3차원.

TABLE 7
Comparisons with SOTA methods on X-sub benchmark in accuracy (%), FLOPs ($\times 10^9$) and parameter number ($\times 10^6$). The models in three parts are compared with EfficientGCN-B0, B2, B4, respectively.

Model	Acc.	FLOPs	Ratio	# Param.
EfficientGCN-B0	90.2	2.73	1×	0.29
EfficientGCN-B2	91.4	4.05	1×	0.51
EfficientGCN-B4	92.1	8.36	1×	1.10



* PoseC3D (X3D)

- 골격 기반
- 노이즈 강인성
- 경량화된 X3D 백본 사용
- MMAction2 (Toobox)
 - pyskl (Toolbox)
- ; Models Zoo

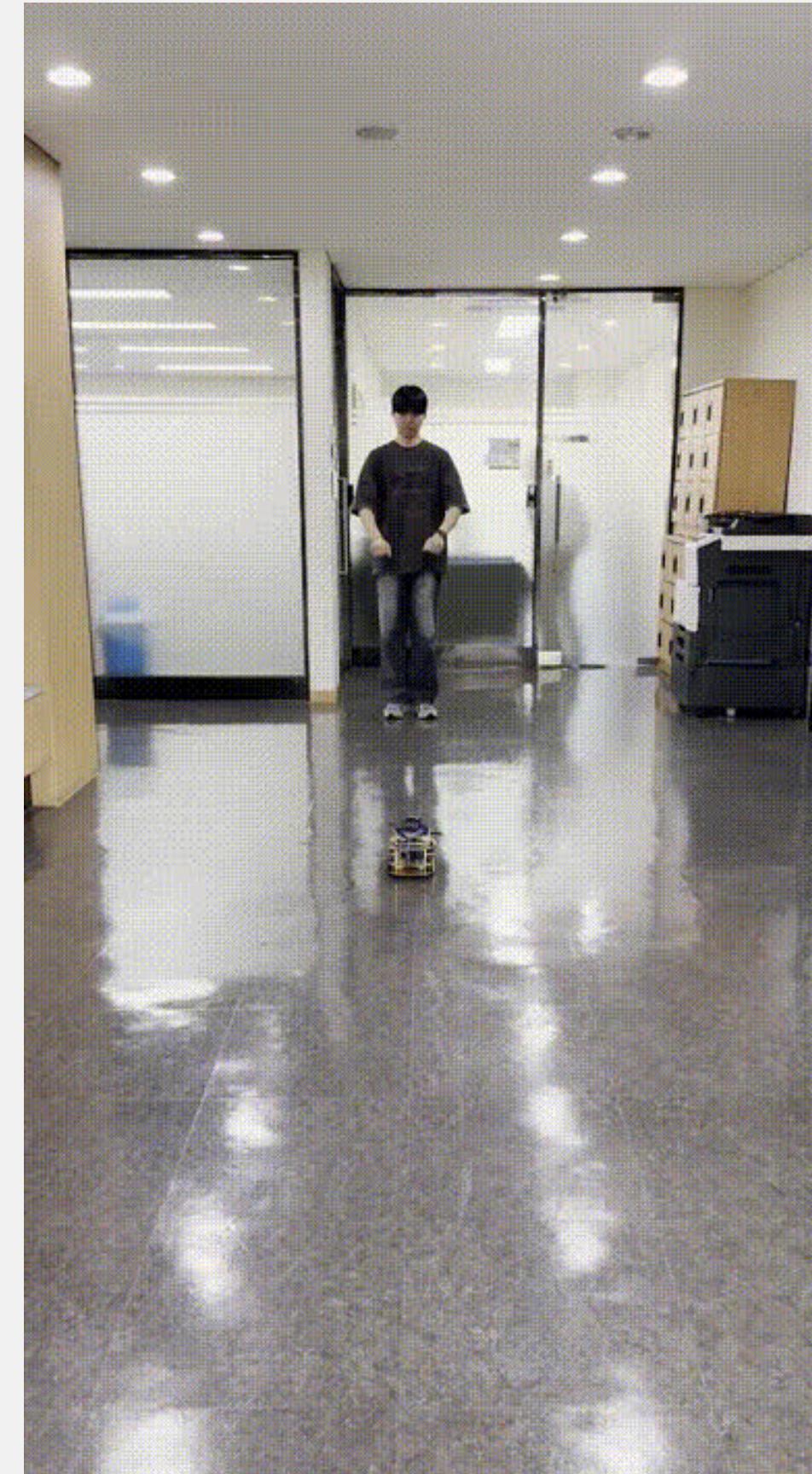
NTU RGB+D 60 Dataset

Table 2. **Evaluation of PoseConv3D variants.** ‘s’ indicates shallow (fewer layers); ‘HR’ indicates high-resolution (double height & width); ‘wd’ indicates wider network with double channel size.

Backbone	Variant	NTU60-XSub	FLOPs	Params
SlowOnly	-	93.7	15.9G	2.0M
SlowOnly	HR	93.6	73.0G	8.0M
SlowOnly	wd	93.7	54.9G	7.9M
C3D	-	93.0	25.2G	6.9M
C3D	s	92.9	16.8G	3.4M
X3D	-	92.6	1.1G	531K
X3D	s	92.3	0.6G	241K

시연영상

Pose:Stop → Motor: Stop

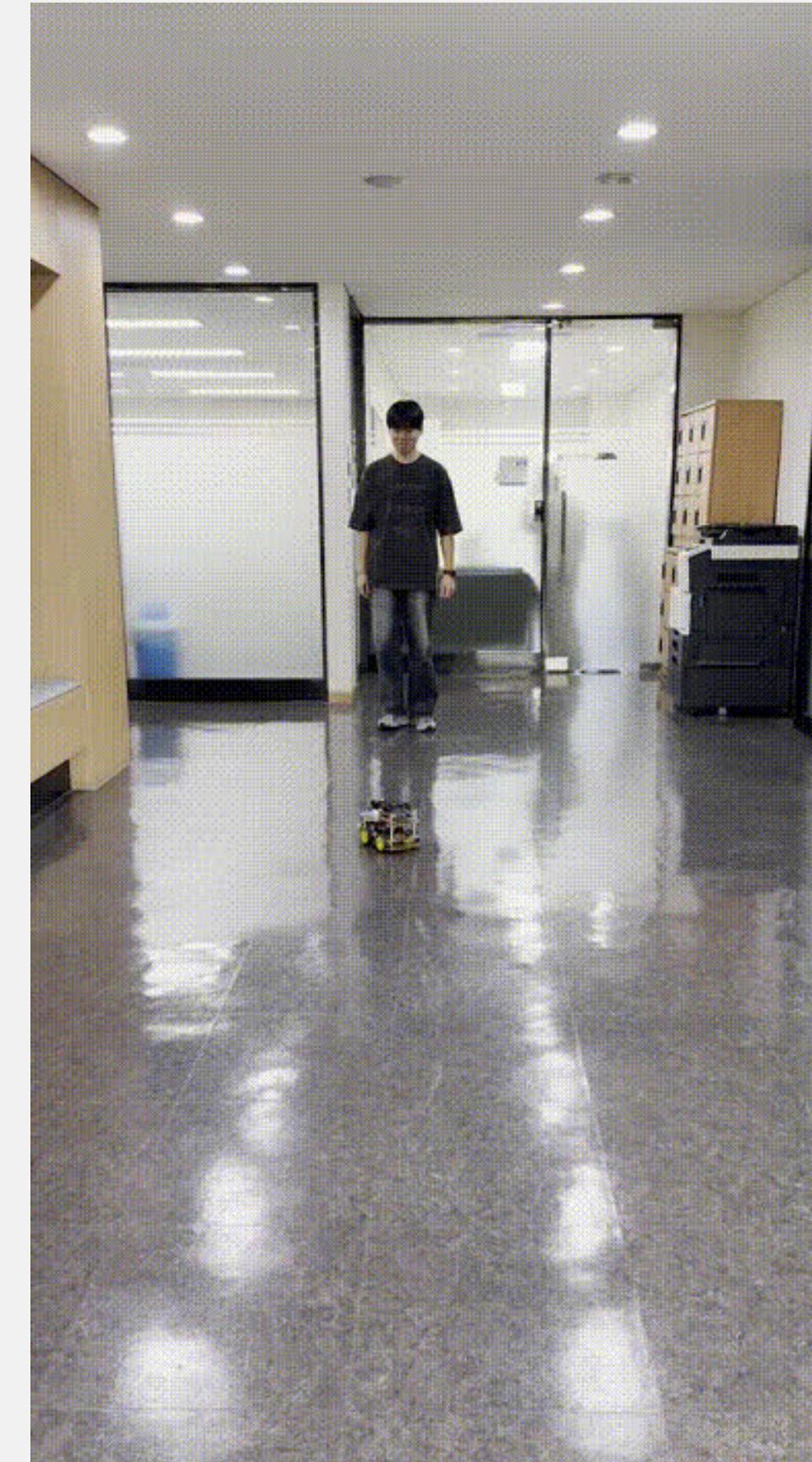


Pose:Slow → Motor: Slow

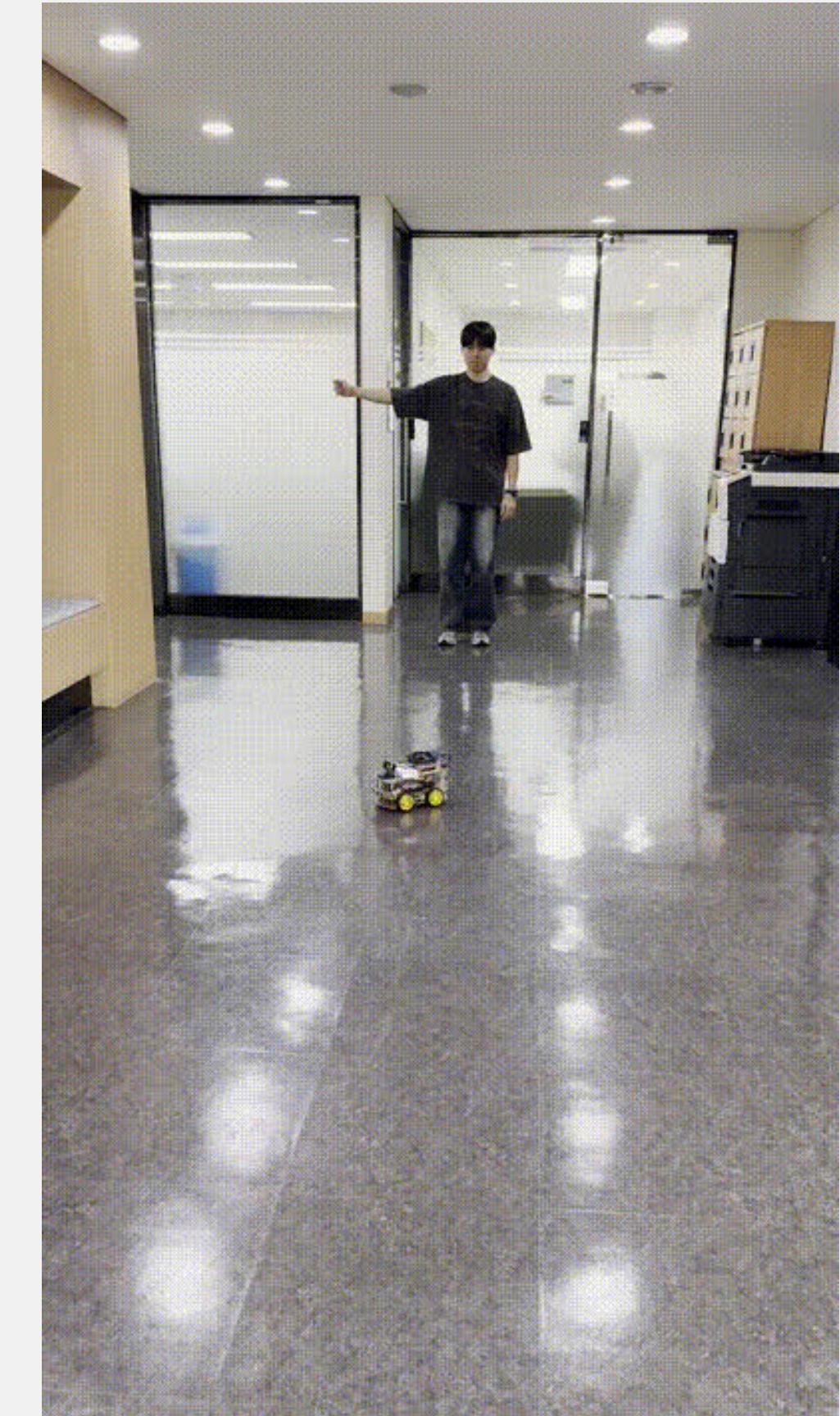


시연영상

Pose:Right → Motor: Right

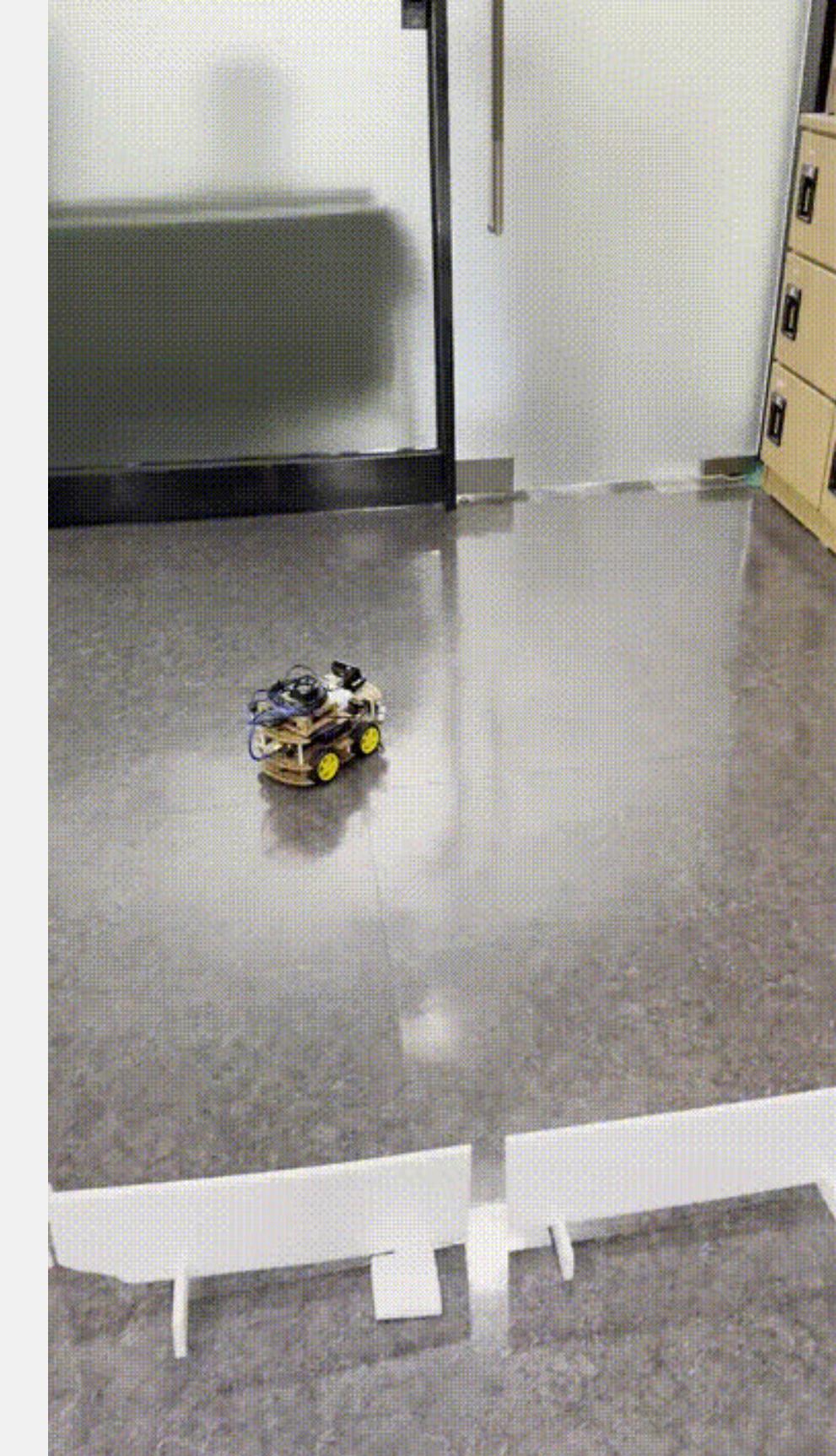


Pose:Left → Motor: Left



시연영상

HC-SR04 → Motor: Direction



마무리

아쉬운 점

H/W

* HC-SR04 거리 측정 정확도 ↓

좌, 우 거리 차이가 미세한 공간에서 좌, 우 결정이 올바르지 못함

해결방안

- Direction이 결정되지 않으면 강제적으로 한방향 결정 (1주 요소 예상)
- RIDAR 방식을 이용한 자율주행 구현 (2주 소요 예상) → HC-SR04는 보조장치

* Serial 송-수신 속도 ↓

- 모델 추론 및 송신의 딜레이 500ms 발생
- 포즈 구현 후 0.5~1초 뒤 수신이 완료됨

해결방안

- 카메라 해상도에 맞춰서 학습 → 불필요한 resize 제거 → 딜레이 ↓ 예상
- 양자화는 성공했지만 사용 과정에서 class id 오류가 발생 → 조사 필요 (3일 소요 예정)

마무리

아쉬운 점

S/W

* 행동 인식 모델 PoseC3D 오작동

NTU RGB+D 60 Dataset 학습/테스트는 정확도 정상적. 실제 비디오 프레임으로부터 모델 추론에서 오작동

해결방안

- 모델이 어떤 데이터와 형식을 주는지 pyskl 스켈레톤 기반 행동 인식 모델 프레임워크 파악 필요.
(1주 소요 예상)

* Jetson nano - container에서 Git 사용.

동일한 jsnano 계정, 동일한 workspace에서 팀원들 간에 접속한 세션에 따라 git config 및 일회성 인증 토큰의 필요성

해결방안

- (임시) rsync에서 .gitignore에 있는 목록을 제외한 파일을 읽어서 백업.

마무리

아쉬운 점

S/W

* 양자화된 Yolo Pose 모델 predict 오류.

양자화된 yolo pose 모델에서 임시로 사용한 사진에 대해 성공적으로 추론하였으나, 현재 젯슨 나노에 있는 코드에서는 반환되는 클래스 id 에러 발생.

해결방안

- 모델이 어떻게 양자화되는지, 왜 오류가 발생하는지 tensorrt 로 변환된 .engine YOLO 모델파일의 데이터 구조 확인 필요. (3일 소요 예상)

마무리

아쉬운 점

S/W

The screenshot shows a GitHub search interface with the query `is:issue is:open` entered in the search bar. The results are filtered by status: **5 Open** and **4 Closed**. The results list five open issues:

- Yolo-Pose로 17개 관절 좌표 추출 후 데이터셋 및 라벨 만들기, 모델 학습하여 테스트**
#18 opened yesterday by opmaksim 2 tasks done ↗ 동작 인식 기반 ...
- Python Jupyter Interactive Kernel crashed when use cv2.imshow()**
#14 opened last week by abcde111112 ↗ 동작 인식 기반 ...
- Guthub README and PPT 준비**
#12 opened last week by opmaksim 3 tasks ↗ Github README ...
- 행동 인식 모델 PoseC3D (X3D-s backbone) 파이프라인 구현**
#10 opened last week by abcde111112 2 of 4 tasks ↗ 동작 인식 기반 ...
- RC카 시리얼 통신 완료**
#9 opened last week by opmaksim 1 of 4 tasks ↗ RC카 제작

감사합니다

Thank you