



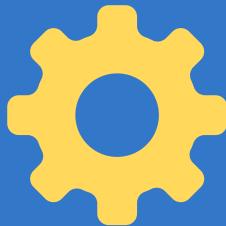
[두산로보틱스] 지능형 로보틱스 엔지니어

Face Arm (서비스 협동 로봇)

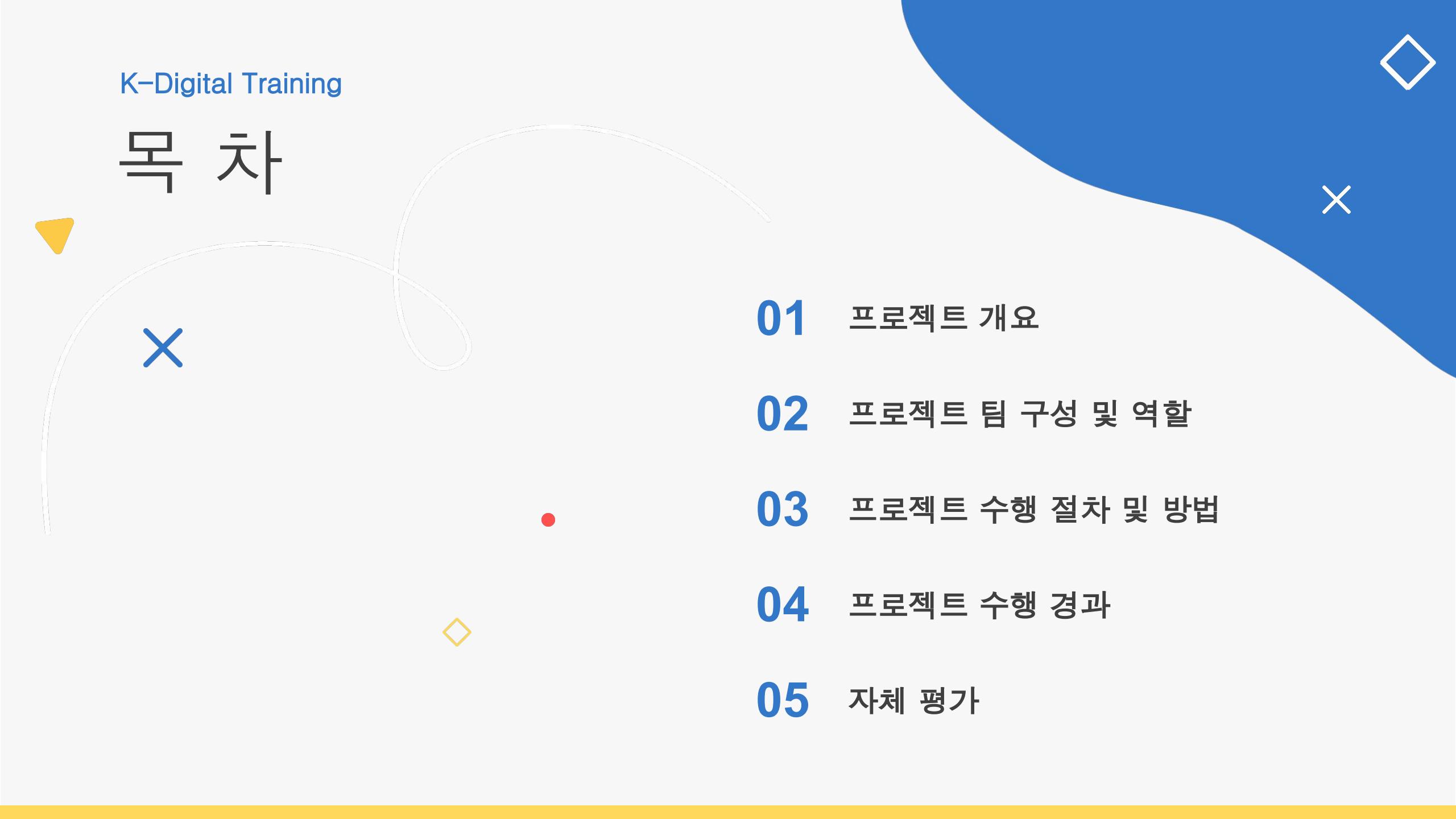
TEAM B-3조 가디구디

최정호, 이세현, 이하빈, 홍진규

[멘토] 이충현 튜터님



목 차

- 
- 01** 프로젝트 개요
 - 02** 프로젝트 팀 구성 및 역할
 - 03** 프로젝트 수행 절차 및 방법
 - 04** 프로젝트 수행 경과
 - 05** 자체 평가

01 프로젝트 개요

1

주제 선정 배경

고령화

2

프로젝트 내용

서비스 로봇

3

활용 장비 및 개발환경

활용 장비 및 개발환경



4

프로젝트 구조

객체인식과 음성을 활용한
서비스로봇

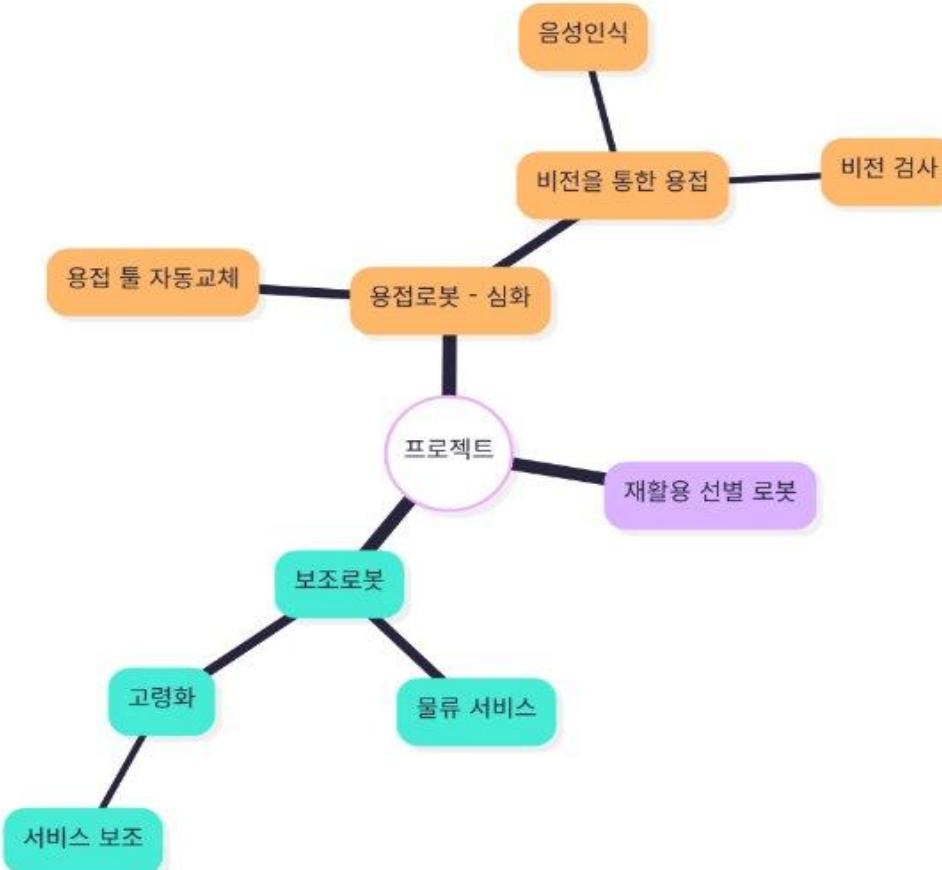
5

활용방안 및 기대 효과

기대 효과와
비즈니스 실무 활용성

01 프로젝트 개요

▶ 주제 선정 - 브레인스토밍



01 K-Digital Training 프로젝트 개요

▶ 주제 선정 배경

• 급격한 고령화 진행

- 2019년 대비 2050년 노인 약 2.5배 증가

• 고령사회 -> 초고령사회 전환

- 2020년대 이미 고령사회
- 2030년 초고령사회로 진입

• 2030 ~ 2050 노인 급증

- 2030년 이후 20년간 약 600만명 증가
- 이 시기에 노인복지 부담 크게 증가할 것으로 예측
(EX: 국민연금상승)



사진출처

:https://www.etnews.com/tools/image_popup.html?v=bjQ5VkJYY2dLZTZMZFBUbmNzandvWmpGcjBTK1pHYm1CUVRXR0pTMIdlYVZkdE5JeGJVUFVVNURJN04rbmhKTVICSy9QaVRaMWpERkc1TE4rcXgzXpmcDQvWEhaeEVjOCT0d1ZWN1NINGQxS1Vssts9MTUVrQ2d0aTRGZINYQIVZHBYmIGQ3FkWFo4aHJHL0RMK2x3YUloSjhyaDdNTDObzJUM05QK1A0cEFjQ1hvL293TWNuSUZkSDhHNVRNcmhRRI9RZU8xa281aU9WZC9aaU04WDhvNVZRHZDUjUwdk0wVThIM05PckR0TWxHWitSWFIIZExSLVIVpovTg

01 프로젝트 개요

▶ 주제 선정 배경

• 삼성 노블카운티

- 주거, 의료서비스, 요양, 문화, 스포츠 등, 풍요로운 노후 생활 제공
- 지역 주민들을 위한 어린이집, 스포츠 문화센터 개방
- 3세대가 공존하며 더불어 함께 생활

• 벤치마킹

- 아직 두산에서의 노인관련 사업 X
- 이에 노블카운티를 벤치마킹하여, 프리미엄 시니어 타운
- 해당 시설에서 사용할 시니어 보조 로봇



사진출처

https://www.etnews.com/tools/image_popup.html?v=bjQ5VkJYY2dLZTZMZFBUbmNzandvWmpGcjBTK1pHYm1CUVRXR0pTMldlYVZkdE5JeGJVUFVNURJN04rbmhKTVICSy9QaVRaMWpERkc1TE4rcXgzXpmcDQvWEhaeEVjOCT0d1ZWN1NINGQxS1Vs9MTUVrQ2d0aTRGZINYQIVZHBYmIGQ3fkWFo4aHJHL0RMK2x3YUloSjhyadNTFObzJUM05QK1A0cEFjQ1hvL293TWNuSUZkSDhHNVRNcmhRRI9RZU8xa281aU9WZC9aaU04WDhvNVZRHZDUjUwdk0wVThIM05PckR0TWxHWitSWFIIczExSLVIVpovTg

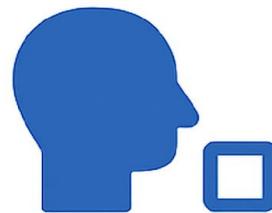
01 프로젝트 개요

▶ 프로젝트 내용

🔍 핵심 기능



음성 명령
로봇 동작 트리거



사용자 &
물체 위치 인지



로봇이 물체를
집어 사용자에 전달



목표 사람과 로봇 간의 자연스러운 상호작용 구현

01 프로젝트 개요

▶ 활용 장비 및 개발환경



Doosan M0609



Logitec C270



Rg2 gripper



d435i

ROS2™



Visual Studio Code

colab

python™

Notion

NAVER
CLOVA

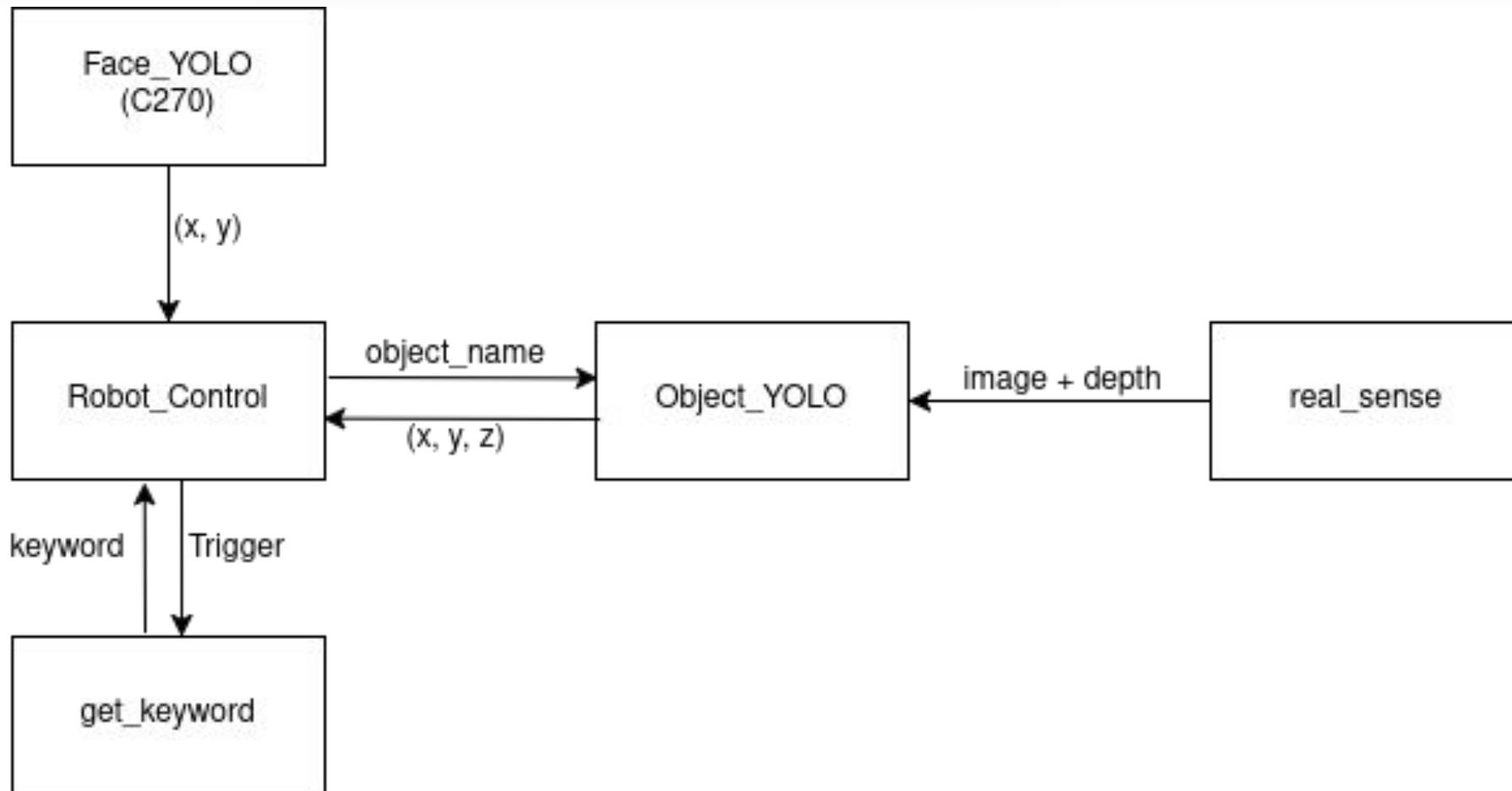
OpenAI
Whisper

ultralytics
YOLO

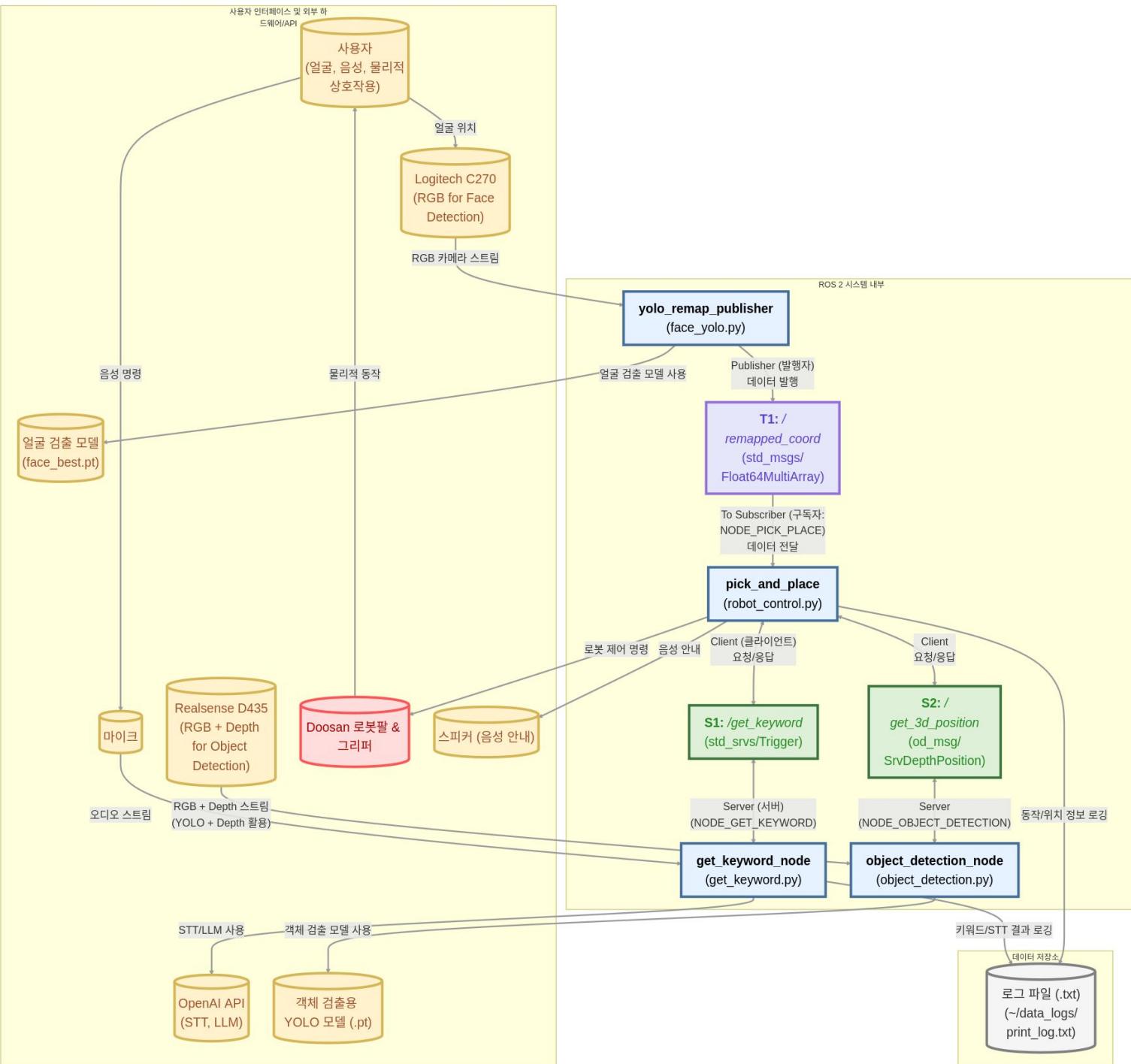
roboflow

01 프로젝트 개요

▶ 프로젝트 구조



▶ 아키텍처 구조



01 프로젝트 개요

- ▶ Face Arm 프로젝트의 기대효과 및 장점

1. 사회적 기여 및 복지 증진

- 고령자 및 장애인을 위한 생활 보조 기능 향상
- 자립 생활 지원을 통한 삶의 질 향상

2. 비즈니스 및 산업 활용성

- 보조 인력 부족 해소 및 자동화 대체 가능성
- 비접촉·음성 기반 인터페이스 구현
- 비대면 환경 최적화로 다양한 공간 적용 가능

프로젝트 팀 구성 및 역할

담당 업무와 역할

B-3	역할	주요 담당 업무
최정호	팀장	 객체 인식 모델 학습 및 발표
이하빈	팀원	 얼굴 인식 모델 학습 및 좌표 리매핑
이세현	팀원	 프롬프트 코딩 및 로봇 동작부 코딩
홍진규	팀원	 예외처리 코드 작성

03

K-Digital Training

프로젝트 수행 절차 및 방법

구분	기간	활동	비고
사전 기획	5/23(금) ~ 5/26(월)	🚩 프로젝트 기획 및 주제 선정	프로젝트 주제선정
정보 수집	5/27(화)	🚩 관련 자료 조사	서비스 로봇 관련 자료 조사
1차 개발	5/28(수) ~ 5/29(목)	🚩 개별파트 코딩 🚩 데이터셋 제작 및 학습	부분 실행 가능한 코드 작성
중간 점검	5/30(금)	🚩 파트별 코딩 테스트	팀원간 중간코딩 점검
2차 개발	6/1(토) ~ 6/3(화)	🚩 개별 코드 통합 및 테스트	최적화 및 오류 수정
최종 테스트	6/4(수)~6/5(목)	🚩 최종 테스트 🚩 발표자료 제작	프로젝트 완성
총 개발기간	5/23(금) ~ 6/5(목)(총 10일)	🚩 조원들의 적극적인 참여와 활동을 통해 획기적인 해결책 도출	

04 프로젝트 수행 경과

▶ 프로젝트 구현을 위한 핵심 코드 구조

1. 특정 물품과 사용자의 얼굴을 인식

- 1) Face_Yolo 파일을 만들어 얼굴 인식
- 2) 객체 인식 데이터셋 생성하여 객체 인식

2. 입력된 음성 명령에 따른 다양한

행동

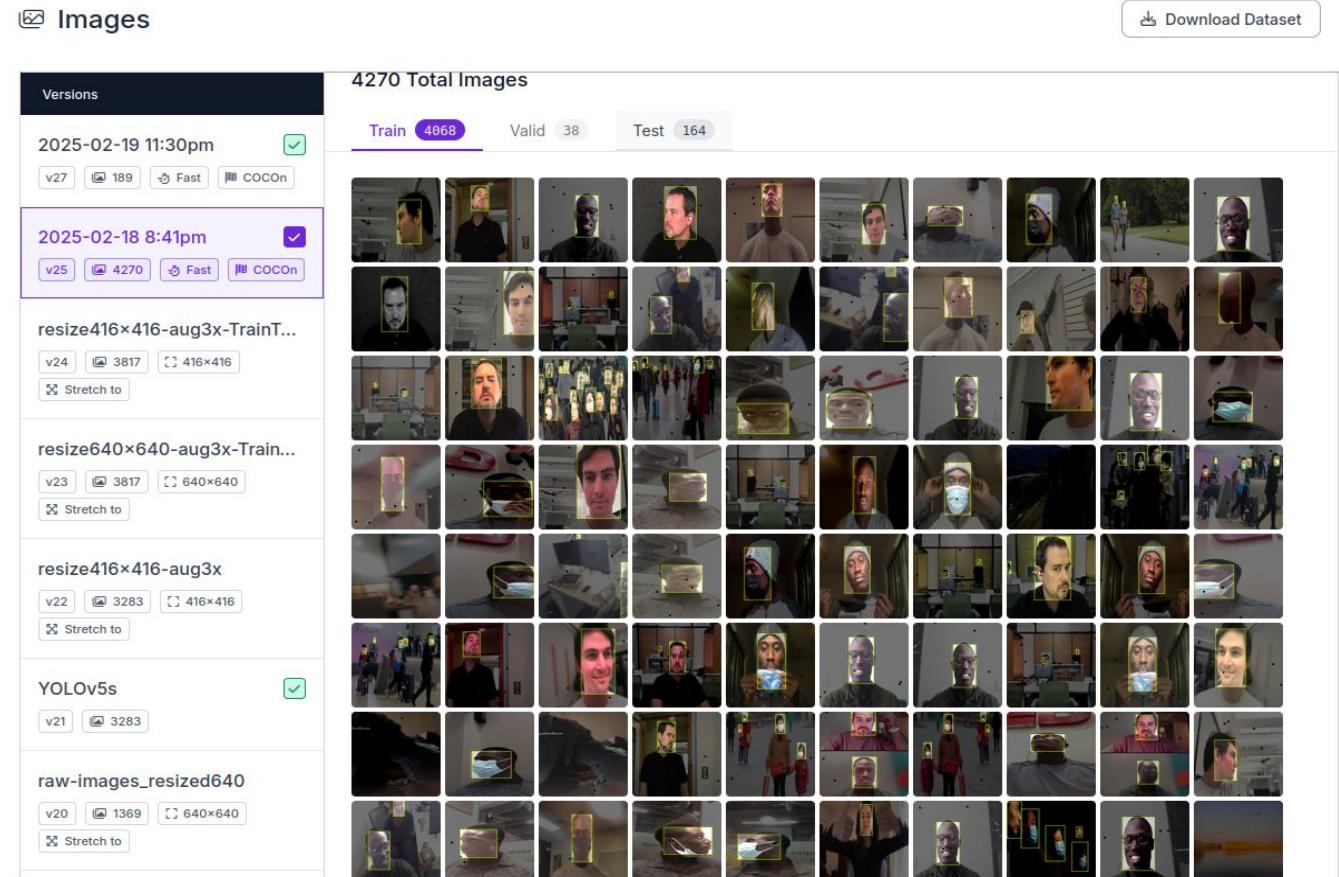
- 1) 갖다 줘
- 2) 돌아가
- 3) 내려놔, 갖다놔
- 4) 여기로 와
- 5) 위/아래/오른쪽/왼쪽/앞/뒤

04

프로젝트 수행 경과

▶ Face detect dataset

- 얼굴 인식 YOLO모델 학습하기 위해 Roboflow Universe에서 Face Detection 데이터셋 사용
- 원본 이미지 1장당 3장의 Data Agumentation 진행



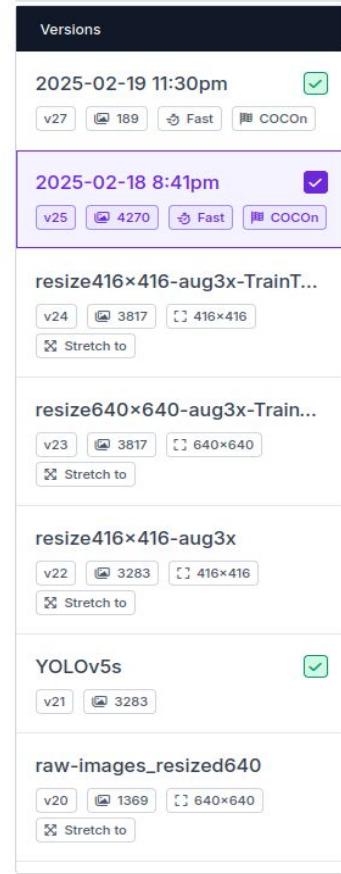
Roboflow – Face Detection 데이터셋 사용(4270장)

04

프로젝트 수행 경과

▶ Data Augmentation

- 선택한 데이터셋의 다양한 버전 중, 모델의 일반화 성능을 향상시키기 위해 Data Augmentation이
진행된 v25 버전 선택
- 수평 반전
- 최대 25% 확대
- 전체 이미지 중 5% Grayscale 변환
- 색조: -20 ~ +20
- 채도: -10% ~ +10%
- 밝기: -22% ~ +22%
- 노출: -10% ~ +10%
- 최대 1픽셀 정도의 Blur
- 이미지에 무작위로 2% 크기의 box를 5개
마스킹



Augmentations



Outputs per training example: 3

Flip: Horizontal

Crop: 0% Minimum Zoom, 25% Maximum Zoom

Grayscale: Apply to 5% of images

Hue: Between -20° and +20°

Saturation: Between -10% and +10%

Brightness: Between -22% and +22%

Exposure: Between -10% and +10%

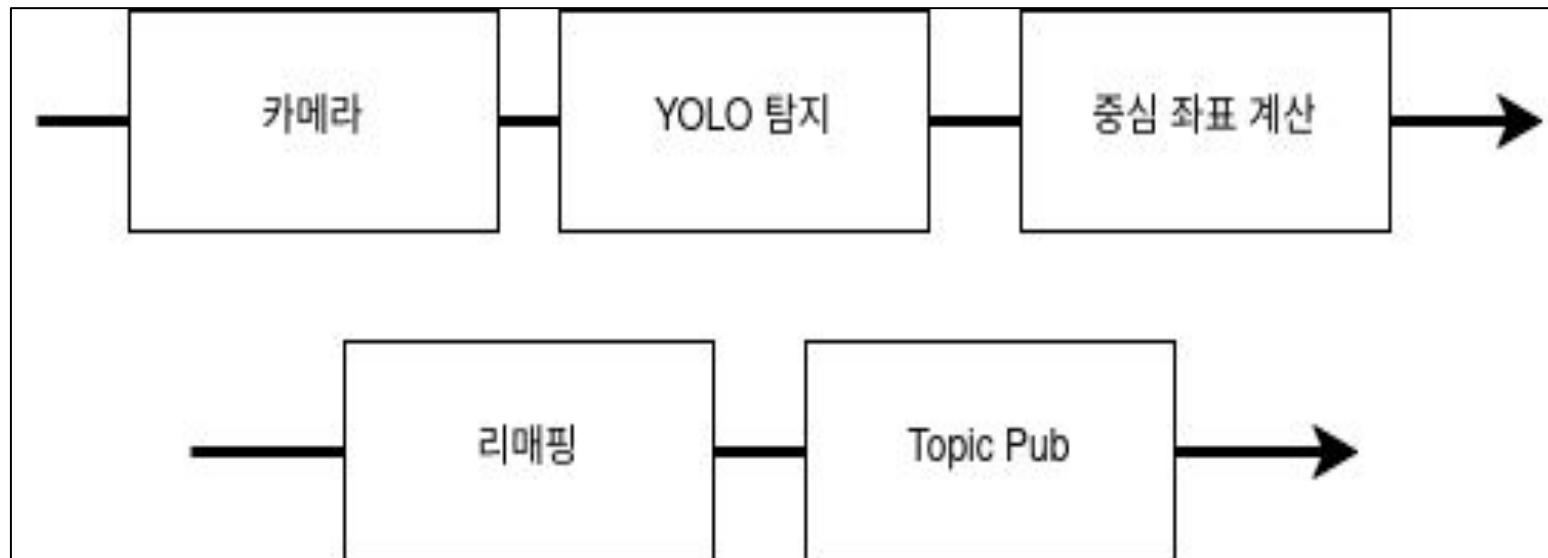
Blur: Up to 1px

Cutout: 5 boxes with 2% size each

04 프로젝트 수행 경과

▶ Face_YOLO.py

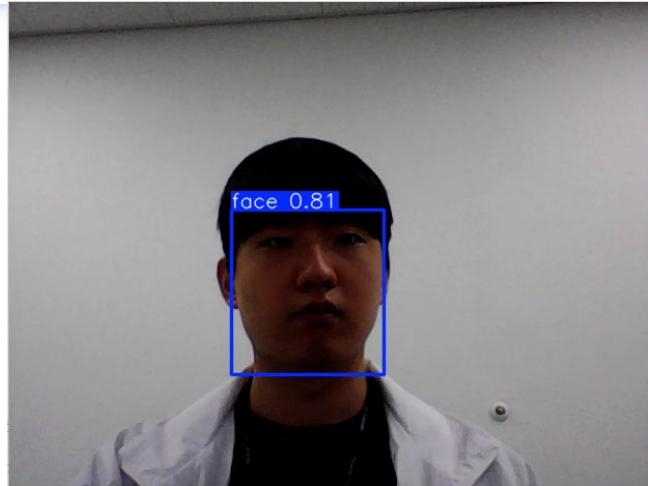
- YOLO 모델로 얼굴을 인식하여 좌표를 실제 작업 공간에 맞게 리매핑 후, ROS2 토픽으로 발행
- 다음 과정으로 작업 수행
 - 카메라 이미지 촬영
 - YOLO모델로 얼굴 인식
 - Bounding Box의 중심 좌표 계산
 - World 좌표계로 리매핑
 - ROS2 토픽 발행



04 프로젝트 수행 경과

▶ Face_YOLO.py

- 카메라 화면의 x,y 좌표를 world 좌표계의 x,z로 리매핑
- Cv 좌표계와 world 좌표계의 축 방향 다른 점 고려
- 카메라 해상도에 따라 같은 위치라도 cv좌표 달라지는 점 고려
- 카메라 좌표계의 (0,0)~(MAX, MAX) 가 World 좌표계의 (50, 660)~(330, 630)을 벗어나지 않도록 리매핑



카메라 좌표(288.7, 282.1)



리매핑 후 (384.9, 453.7)

```
[INFO] [1748913149.706830684] [yolo_remap_publisher]: [face] 원본 좌표: (288.7, 282.1)
[INFO] [1748913149.707538971] [yolo_remap_publisher]: [face] 퍼블리시 좌표: (384.9, 453.7)
```

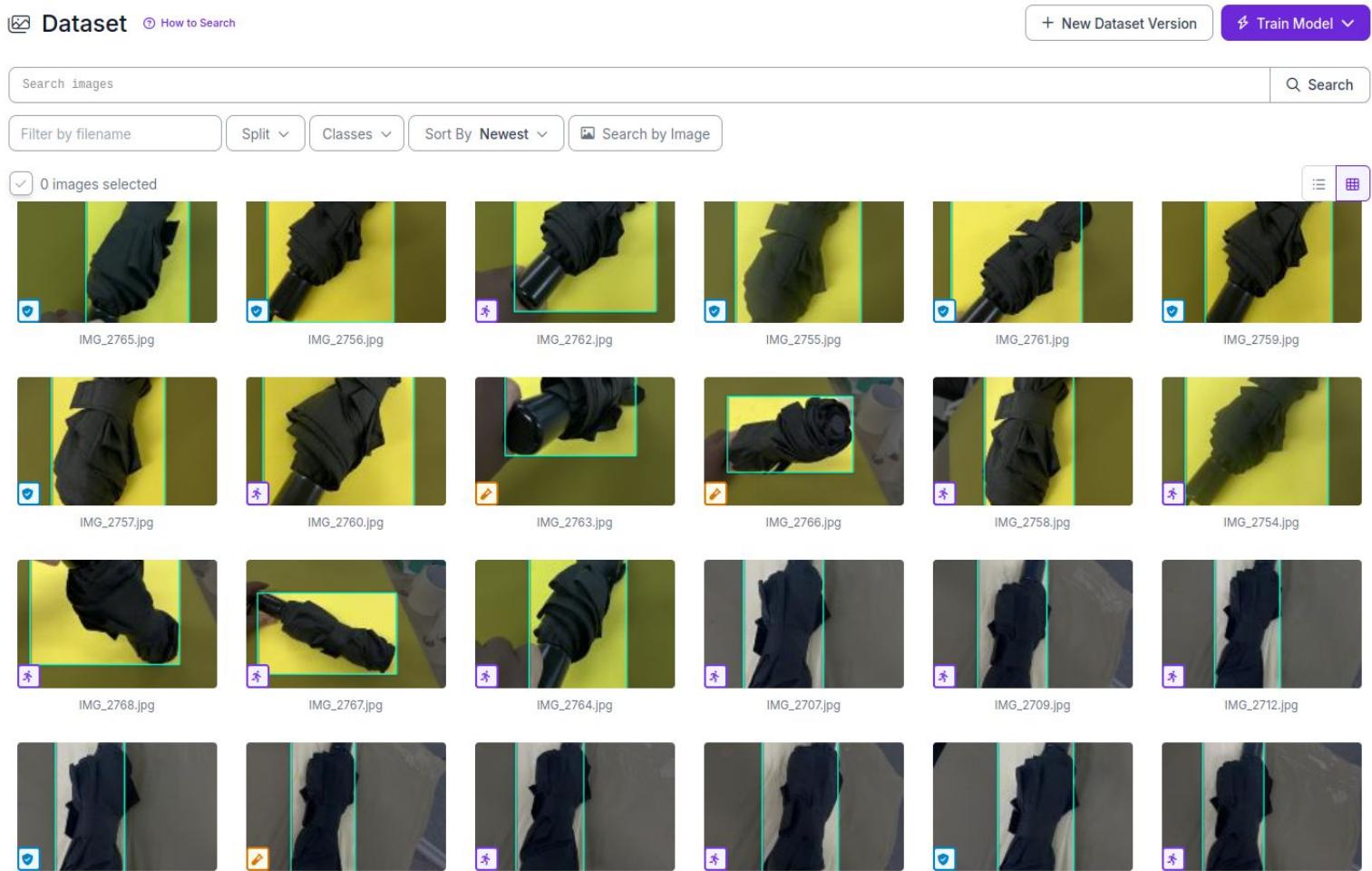


```
def remap(val, old_min, old_max, new_min, new_max):
    return new_min + ((val - old_min) / (old_max - old_min)) * (new_max - new_min)
```

04 K-Digital Training 프로젝트 수행 경과

▶ 3가지 클래스의 데이터셋 생성

- 객체 인식 YOLO모델 학습하기 위해
직접 사진을 찍고 데이터셋 생성
- 180장의 사진을 각 클래스마다
Train : Validation : Test = 7:2:1 분류
- 원본 이미지 1장당 3장의 Data
Augmentation 진행



04

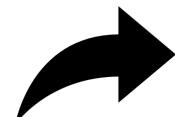
프로젝트 수행 경과

▶ Object_detection.py — 핵심 코드

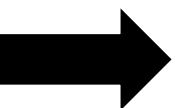
1. YOLO 탐지 + 멀티 프레임 집계

(Yolo.py)

- `_aggregate_detections`를 통해 IoU 기반 병합
- 다수결(label voting)으로 노이즈를 제거



```
def get_best_detection(self, img_node, target):  
    ...  
    results = self.model(frames, verbose=False)  
    detections = self._aggregate_detections(results)  
    ...
```



```
cx, cy = map(int, [(box[0] + box[2]) / 2, (box[1] + box[3]) / 2])  
cz = self._get_depth(cx, cy)
```

2. 객체 중심 좌표 + Depth 추출

- YOLO가 제공하는 박스에서 중심좌표(cx, cy)를 추출
- Realsense Depth Map을 활용해 해당 좌표의 실제 거리(cz) 계산



```
def _pixel_to_camera_coords(self, x, y, z):  
    return (  
        (x - ppx) * z / fx,  
        (y - ppy) * z / fy,  
        z  
    )
```

3. 픽셀 → 카메라 좌표계 변환

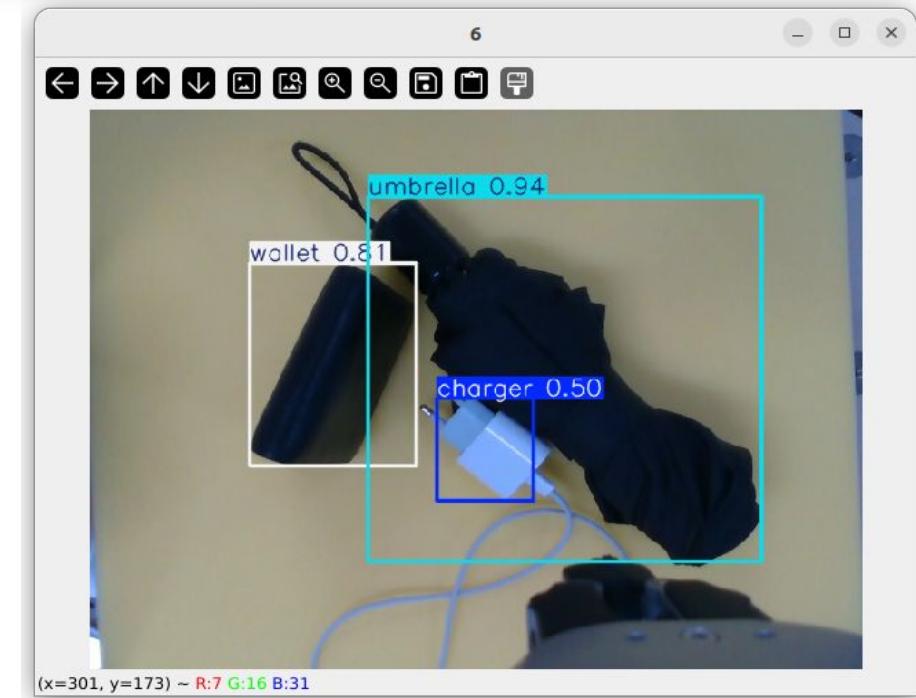
04

프로젝트 수행 경과

▶ Object_detection.py



```
0: 480x640 1 charger, 1 umbrella, 1 wallet, 78.8ms  
0: 480x640 1 charger, 1 umbrella, 1 wallet, 94.1ms  
0: 480x640 1 charger, 1 umbrella, 1 wallet, 83.3ms
```



```
0: 480x640 1 charger, 1 umbrella, 1 wallet, 74.0ms  
0: 480x640 1 charger, 1 umbrella, 1 wallet, 74.8ms  
0: 480x640 1 charger, 1 umbrella, 1 wallet, 80.8ms
```

04 프로젝트 수행 경과

▶ NAVER CLOVA

- 사용자와 상호작용 하기 위해 스피커에서 안내멘트 재생
- Naver Clova 더빙 서비스를 활용하여 다음과 같은 안내 음성들을 제작
 - 이동을 시작합니다
 - 물건을 갖다 드릴게요
 - 물건을 제자리에 돌려놓을게요
 - 잘 못 알아들었어요
 - 말씀하세요
 - 물건을 잡아주세요
 - 대기모드로 전환합니다

**NAVER
CLOVA**



04 프로젝트 수행 경과

▶ Get_keyword

- Prompt_content 추출 키워드

- 추출 대상
 - 일반 도구
 - wallet, umbrella, charger
 - 행동 또는 목표 상태
 - place
 - return
 - right, left, forward, backward, up, down
 - Come_here
 - retry

- Prompt_content 추출 규칙

1. 핵심 키워드 추출
2. 호출어 처리

(예: 로키 → 최종 키워드 포함X)
3. 문맥상 관련 도구 추출
4. 키워드 미발견 시

(retry 반환)
5. 공백과 도구

(공백과 도구 동시 추출시. 최종 키워드 → 도구)
6. 기타 언어 필터
7. 한국어/영어 키워드만 반환

04 프로젝트 수행 경과

▶ Data_logs – 텍스트 기록

- 로봇 동작 과정에서 발생하는 중요 데이터를

시간별 텍스트화를 통해 기록

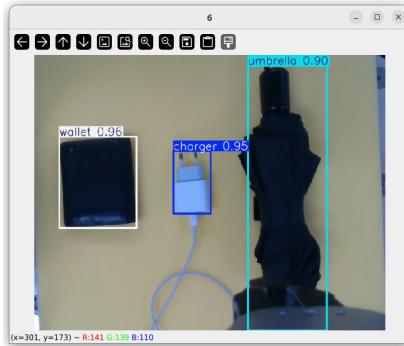
- **stt** → 음성 인식 텍스트 저장
- **get_keyword** → 키워드 추출 텍스트 저장
- **robot_control** → 도구 좌표 텍스트 저장

```
# --- 경로 설정 시작 ---
log_file = os.path.expanduser("~/data_logs/print_log.txt")
os.makedirs(os.path.dirname(log_file), exist_ok=True)
# --- 경로 설정 끝 ---
with open(log_file, "a", encoding="utf-8") as f:
    f.write(datetime.now().isoformat())
```

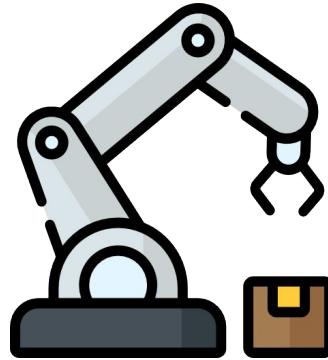
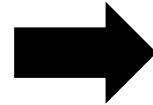
04

프로젝트 수행 경과

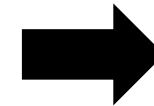
▶ 명령어 - 000 가져와



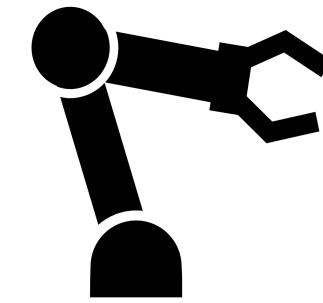
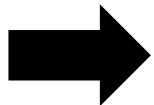
Realsense
카메라로
물품 좌표 확인



해당 좌표로 move 명령 후
그리퍼 grip



Face yolo로 사람 위치 sub



안내 멘트 이후
그리퍼 외력 5초간 감지

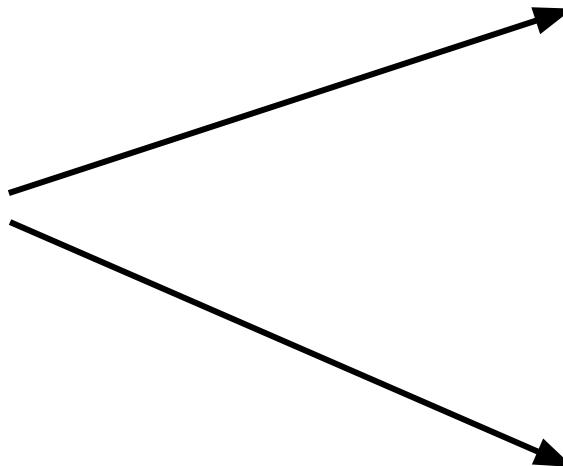
04

프로젝트 수행 경과

▶ 명령어 - 000 가져와



안내 멘트 이후
그리퍼 외력 5초간 감지



그리퍼 release 후 다음
wakeup 명령어 대기



계속 물품을 잡은 채로
다음 wakeup 명령어 대기

04

프로젝트 수행 경과

▶ 명령어 – 정리해, 갖다놔



그리퍼를 닫아 물품을 집음



이전에 저장한 물품의 좌표로 이동

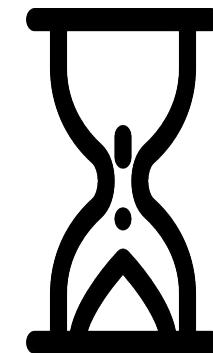
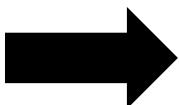


그리퍼 release

※ 사용자가 물품을 집지 않았을 경우
생략※



홈 위치로 이동

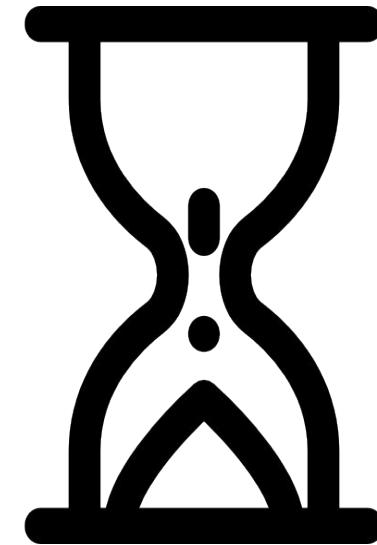


다음 wakeup 명령어 대기

프로젝트 수행 경과

▶ 명령어 - 돌아가

※realsense 와 로지텍 카메라가 로봇팔에 의해 시야를 가릴 경우※



기본 위치로 설정된 조인트각도로 movej 실행

다음 wakeup 명령어 대기

04 K-Digital Training 프로젝트 수행 경과

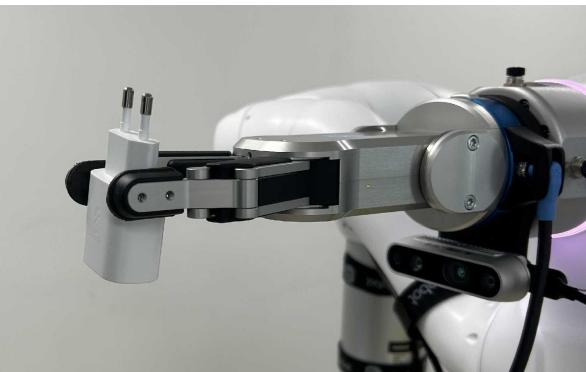
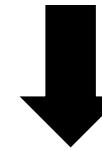
▶ 명령어 – 여기로 와



C270 카메라로 인식한 사람의
좌표로 이동



5초 안에 사람이 물건을 집으면, 그리퍼를 **release** 후,
그 자리에서 다음 **wakeup** 명령어 대기



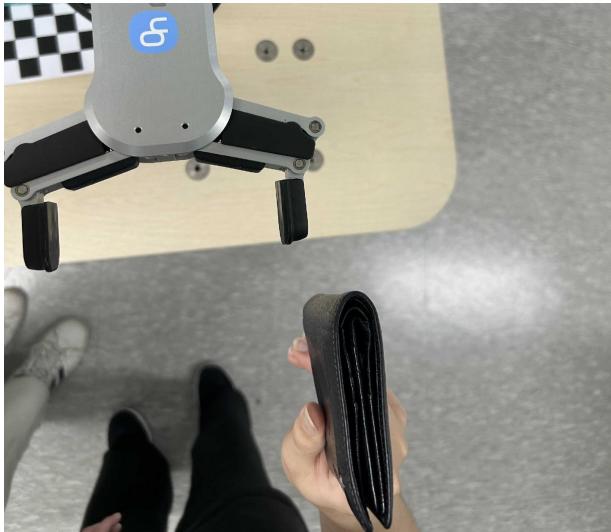
만약 그리퍼에 물체가 있는
경우, 안내 멘트 이후 그리퍼
외력 5초간감지

5초 안에 사람이 물건을 집지
않으면, 계속 물품을 잡은
채 다음 **wakeup** 명령어 대기

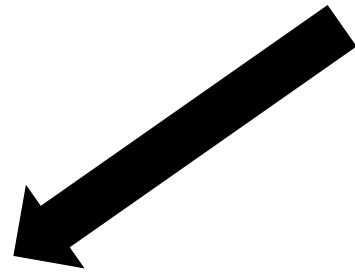
04

프로젝트 수행 경과

▶ 명령어 – 위/아래/오른쪽/왼쪽/앞/뒤



해당 위치로 일정 거리만큼
이동



만약 그리퍼에 물체가 있는
경우, 안내 멘트 이후 그리퍼
외력 5초간감지



5초 안에 사람이 물건을
집으면, 그리퍼를
release 후, 그 자리에서
다음 **wakeup** 명령어 대기

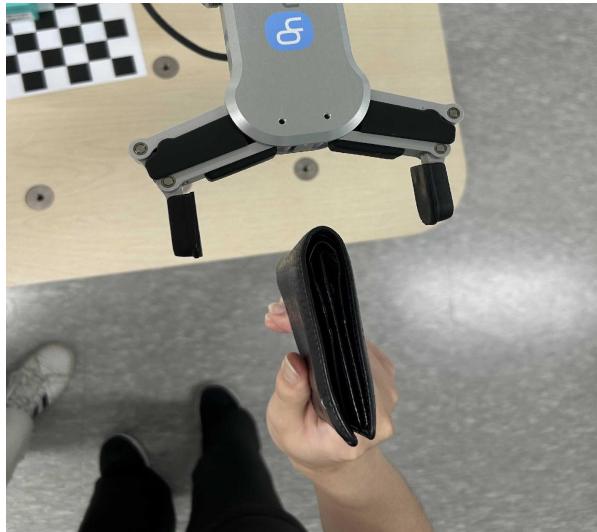


5초 안에 사람이 물건을 집지
않으면, 계속 물품을 잡은 채로
다음 **wakeup** 명령어 대기

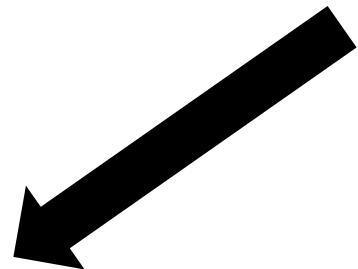
04

프로젝트 수행 경과

▶ 명령어 – 위/아래/오른쪽/왼쪽/앞/뒤



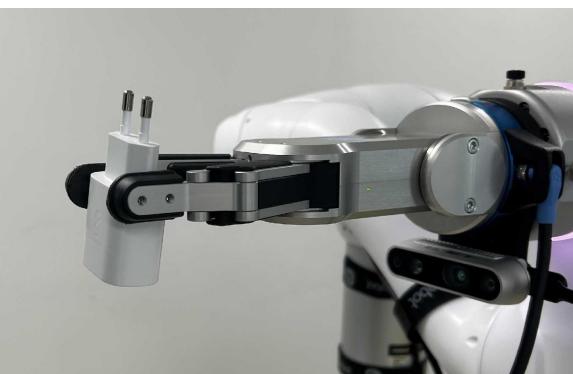
해당 위치로 일정 거리만큼 이동



만약 그립에 물체가 있는 경우, 안내 멘트 이후 그립에 외력 5초간감지



5초 안에 사람이 물건을
집으면, 그립에
release 후, 그 자리에서
다음 **wakeup** 명령어 대기



5초 안에 사람이 물건을 집지
않으면, 계속 물품을 잡은 채로
다음 **wakeup** 명령어 대기

04 프로젝트 수행 경과

▶ 예외처리코드 부분

[예외]	[조건]	[결과]
객체 인식 실패	5초간 YOLO 인식 안될시	로봇 정지 및 카메라/모델 이상 로그 출력 → 오작동 방지
좌표 수신 실패	10초간 좌표 미수신 시	로봇 이동 중단 및 정지
비정상 좌표 수신	($0 < x, y < 700$) 범위 외 좌표 필터링	'작동불가범위' 로그 출력 및 정지
로봇 이동 중 예외	Movel 중 오류 발생 시	긴급 정지 및 실패 로그 출력
충돌 감지	Force Sensor 값이 임계치 초과 시	로봇 경고음 발생 및 즉시 정지
음성 명령 오류	정의되지 않은 음성 명령어 입력 시	로봇 작동 X 및 사용자에게 피드백 출력
사용자 다중 감지	감지 인원이 2명 이상일 경우	정확도가 더 높은 사용자를 최우선시하며 작동 및 로그 출력
도달 불가능한 위치	inverse kinematics 실패	로봇 작동 중지 및 '작동불가'로그 출력

05 K-Digital Training 자체 평가

▶ 평가 의견과 느낀 점

프로젝트 결과물에 대한 완성도 평가

9.9/10

잘한 부분과 아쉬운 점

잘한 부분 : 구상했던 부분을 모두 구현했다

아쉬운 부분 : 물체 이름이나 명령어를 호출할 때 사투리나 비슷한 발음으로 인해 로봇이 정확히 인식하지 못한 경우가 있어 아쉬웠다

보완할 점

1. 실제 사용자 환경에서의 테스트가 부족해 다양한 상황 대응에는 한계가 있다
2. 음성 명령을 인식하고 동작을 하는 부분이 느린거 같아 보완해야 할 것 같다

느낀 점이나 경험한 성과

느낀점 : 데이터 셋 형성의 원리와 효과적인 객체인식 방법 확인하였다.
우리의 일상생활을 도와주는 로봇 출시가 얼마 남지 않은 것 같다