Cig-Bee

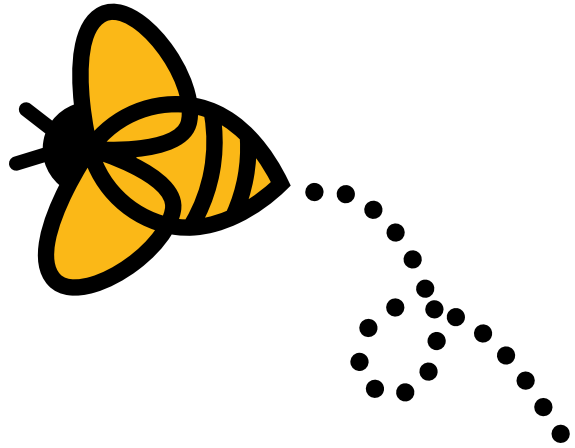
탐지 로봇을 이용한 담배 꽁초 탐지

[Intel] 엣지 AI SW 아카데미 4기

1조 STACK

권오준, 김정대, 장혜원, 차창섭, 최재혁

목차



01 프로젝트 개요

1. 주제 선정 배경
2. 문제 분석
3. 구현 목표

1. 주제 선정 배경

담배 꽁초로 인한 사회 환경 문제

봄철 화재 재산 피해 가장 커…담배꽁초 무단투기 단속 강화

입력 : 2023.02.20 12:23 | 수정 : 2023.02.20 12:25 박용필 기자

2020년 4월 30일 경기도 이천 한익스프레스 물류창고 화재 현장에서 이천소방서 긴급구조반 대원들이 아슬하게 걸쳐있던 철근구조물 제거작업을 하고있다. / 이준헌 기자 ifwedont@

(...)
소방청이 최근 5년 간의 화재발생 건수 등을 빅데이터 분석한 결과, 최근 5년간 봄철(3~5월)에 5만4485건의 화재가 발생한 것으로 나타났다. 인명피해는 2743명(사망자 458명, 부상자 2285명) 발생했으며, 재산피해는 1조4208억원에 달했다.
(...)
화재 원인은 **담배꽁초**와 쓰레기 소각 등 **부주의로 인한 화재**가 전체의 **55.6%로 가장 많았다**. 그 다음으로 전기적요인(22.2%), 기계적요인(9.1%) 순이었다.
(...)

기사 출처(경향신문): https://www.khan.co.kr/national/national-general/article/202302201223001

1. 주제 선정 배경

담배 꽁초로 인한 사회 환경 문제

[꽁초혁명①] 과소평가된 담배꽁초의 유해성

조은비 기자 | 2023.03.27 17:05

(...)
담배꽁초가 오염시킨 토양과 물이 어떤 영향을 받는지 조사하는 연구도 수차례 진행됐다.

미국환경보호청(EPA)은 **담배꽁초 한 개비**를 96시간 동안 넣어둔 물 1L에서 민물고기, 해수어 **모두 50% 이상 폐사**했다는 연구 결과를 발표했다.
(...)

기사 출처(뉴스펭귄): https://www.newspenguin.com/news/articleView.html?idxno=13729

1. 주제 선정 배경

담배 꽁초로 인한 사회 환경 문제

무단 투기 꽁초 52억 개...처리비용 딴 데 쓰는 환경부

2023년 09월 18일 11시 19분

(...)
정부는 해마다 담배회사에서 폐기물 부담금 890억 원 이상을 걷고 있지만, 꽁초를 따로 회수하거나 처리하는 데는 이 부담금을 사용하지 않고 있습니다.
(...)
"담배회사에서 걷은 돈으로 왜 꽁초 회수나 처리 관련 예산을 따로 편성하지 않느냐"는 질문에 환경부는 "답변하기 어렵다"고 했습니다.
(...)

기사 출처(YTN 사이언스): https://science.ytn.co.kr/program/view.php?mcd=0082&key=202309181119432182

2. 문제 분석

문제 범위 지정 :
공공 장소 외 일상 생활 공간에서 매일 버려지고 있는 담배 꽁초가 불법 투기로 인한환경 오염 및 안전 문제 발생을 자동 담배 꽁초 감지하여 수거하는 해결이 필요함

WHAT	• 국내에서 매일 버려지는 담배 꽁초로 인해 환경 오염과 안전 문제가 발생하고 있음
WHERE	• 도로, 공원, 거리 등 공공장소와 일상 생활 공간에서 담배 꽁초가 버려지고 있음
WHEN	• 담배 꽁초 불법 투기는 매일 발생하고 있으며, 매일 약 1,246만개가 버려지는 것으로 추정
HOW MUCH	• 담배 꽁초 불법 투기로 인해 화재가 발생하고 있으며, 배수로 막힘과 빗물 역류로 인한 침수 피해가 발생함 • 담배 꽁초 필터에는 미세 플라스틱이 포함되어 있으며, 바다로 유입되는 담배 꽁초로 인해 해양 환경 오염이 발생하고 있음

3. 구현 목표

담배 꽁초를 탐지하고 수거하는 로봇 개발

1

담배 꽁초를 탐지하는 모델 제작

화재 예방을 위해 불씨가 남은 담배 꽁초와 일반 담배 꽁초로 구분

2

도로를 따라 자율 주행하는 기능 구현

라인을 따라 이동하며 꽁초를 탐지하는 기능 수행

02 프로젝트 계획

1. 팀 구성
2. 수행 계획
3. 소프트웨어
4. 하드웨어
5. Flow Chart

1. 팀 구성

조 이름

STACK(스택)

선입 후출(FILO) 방식으로 저장되는 스택처럼
가장 먼저 나와서 마지막에 퇴근하자는 의미이다.

프로젝트명

Cig-Bee(시그비)

담배(Cigarette)와 **벌(Bee)**를 합친 단어로,
꽃을 찾아가는 벌처럼 담배 꽁초를 찾아가는
탐지 로봇의 모습을 담았다.

1. 팀 구성

권오준

모델 학습 및 최적화
보고서 및 발표 자료 작성

장혜원

데이터 수집
모델 학습 및 최적화
자료 정리

차창섭

데이터 수집
모델 학습 및 최적화
자료 정리

김정대

하드웨어 구현
이동 및 탐지 기능 구현
자료 정리

최재혁

하드웨어 구현
이동 및 탐지 기능 구현
발표

2. 수행 계획

NO	업무	5월												
		10	11	12	13	14	15	16	17	18	19	20	21	22
1	**프로젝트 구성**													
1.1	팀 구성 및 팀 빌딩													
1.2	프로젝트 기획													
2	**담배 꽁초 탐지 모델**													
2.1	학습 데이터 수집													
2.2	모델 학습													
2.3	모델 최적화													
3	**탐지 로봇 제작**													
3.1	HW 구현													
3.2	원격 주행 기능 구현													
3.3	주행 데이터 수집													
3.4.	자율주행 구현													
4	**탐지 로봇에 기능 추가**													
4.1.	자율주행과 담배 꽁초 탐지 병렬 처리													
4.2.	이동 및 피드백 구현													
4.3.	테스트 및 수정													
5	**보고서 작성 및 종료**													
5.1	보고서 작성													
5.2	발표													

3. 개발 환경 - 소프트웨어

개발 도구

python™

ANACONDA.

jupyter

모델 학습 및 탐지

TensorFlow

ultralytics
YOLO

OpenCV

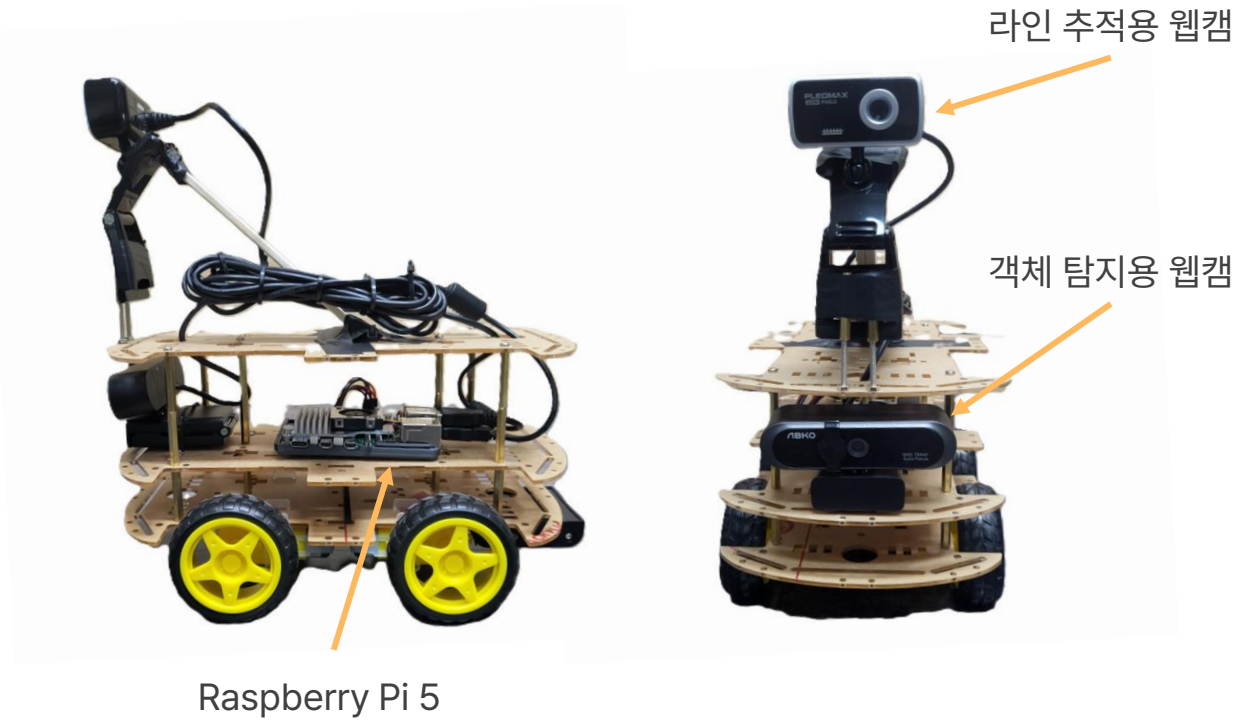
협업 도구

Notion

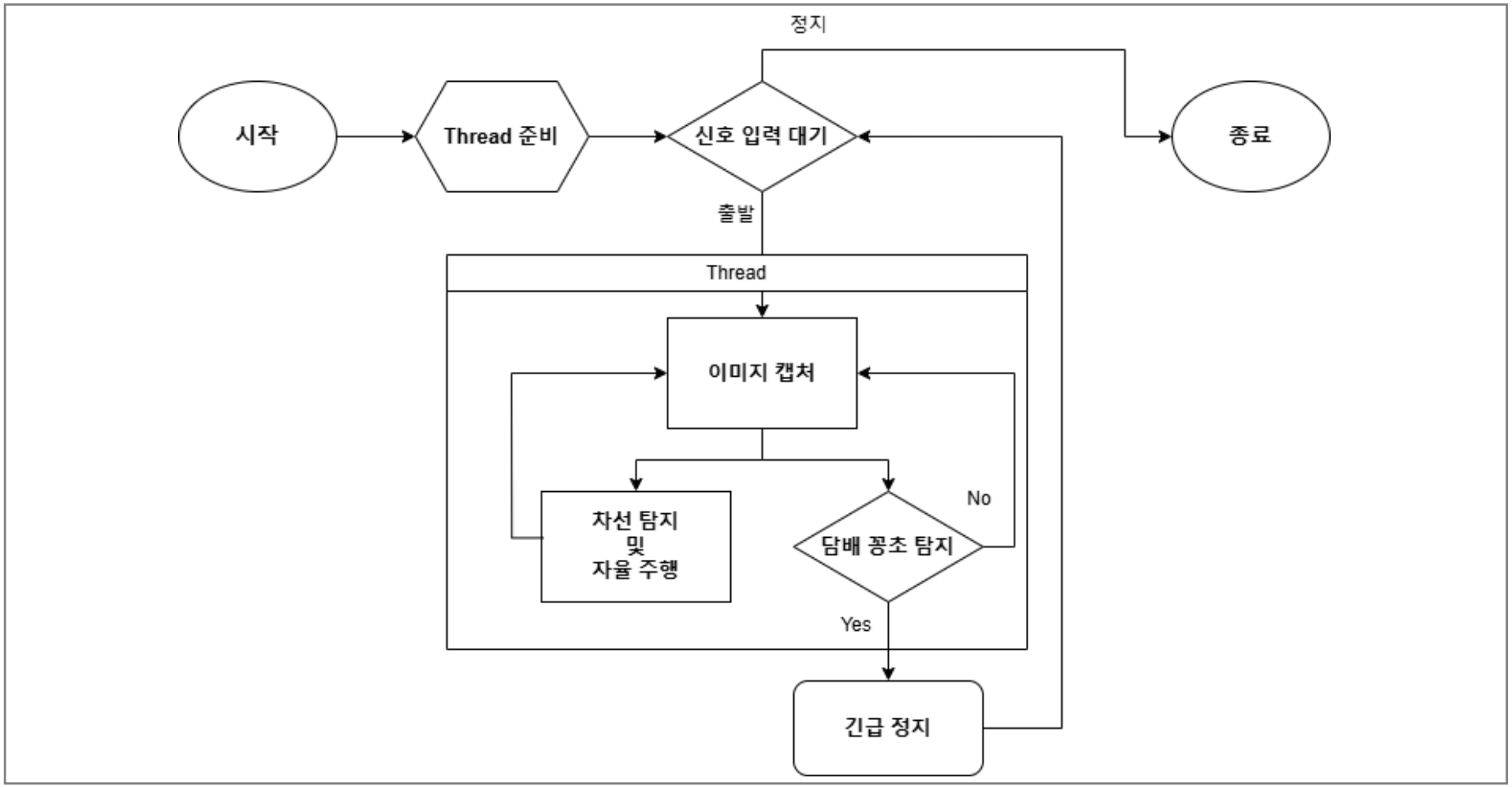
Discord

4. 개발 환경 - 하드웨어

1. Raspberry Pi 5 8GB
2. 웹캠 * 2
   (라인 추적용, 담배 꽁초 탐지용)
3. L298N 모터 드라이버
4. DC 모터 * 4
5. 6V 전원



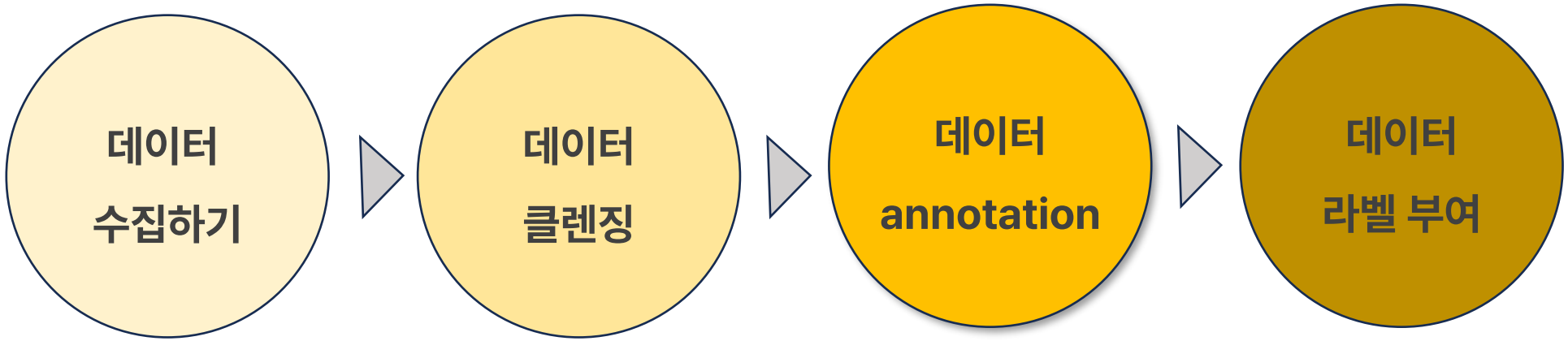
5. Flow Chart


정지
시작
Thread 준비
신호 입력 대기
종료
출발
Thread
이미지 캡처
차선 탐지
및
자율 주행
No
담배 꽁초 탐지
Yes
긴급 정지

03 프로젝트 수행 과정

1. 학습 데이터 셋 구축
2. 꽁초 탐지 모델 제작
3. 하드웨어 제어 구현
4. 자율주행 구현
5. 탐지 기능 추가
6. 통합 테스트

1. 학습 데이터 셋 구축



학습에 이용할 공개 데이터셋 수집하기

학습에 이용할 공개 데이터셋 수집하기

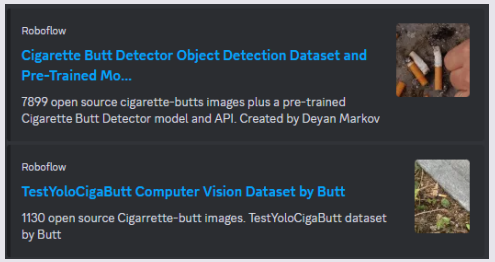
추가 수집된 데이터 annotation 작업

이미지 분류 후 데이터 라벨 부여

1. 학습 데이터 셋 구축

1) 학습에 이용할 데이터 수집하기

공개 데이터 셋 공유 포탈인 **Roboflow**에서 담배 데이터 셋을 찾음



Cigarette Butt Detector
총 7,899개 이미지
- train: 7,029
- valid: 789
- test: 1,578

클래스: 2개 (‘0’, ‘cig_butt)

데이터 셋 링크: https://universe.roboflow.com/deyan-markov/cigarette-butt-detector-0jv4z

1. 학습 데이터 셋 구축

2) 데이터 클렌징 수행

I. 해당 데이터 셋은 클래스가 2개로 되어 있었음. ‘cig_butt’으로 통일함

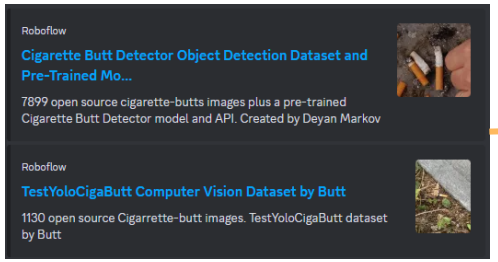
II. 학습에 방해가 될 수 있는 데이터들을 선별하여 제거 (중복 데이터, 꽁초가 아닌 이미지)

```
{
  0: ‘0’,
  1: ‘cig_butt’
}
```

→

```
{
  0: ‘cig_butt’
}
```

라벨을 하나로 통일



→

담배 꽁초와 관련 없는 데이터 제거

1. 학습 데이터 셋 구축

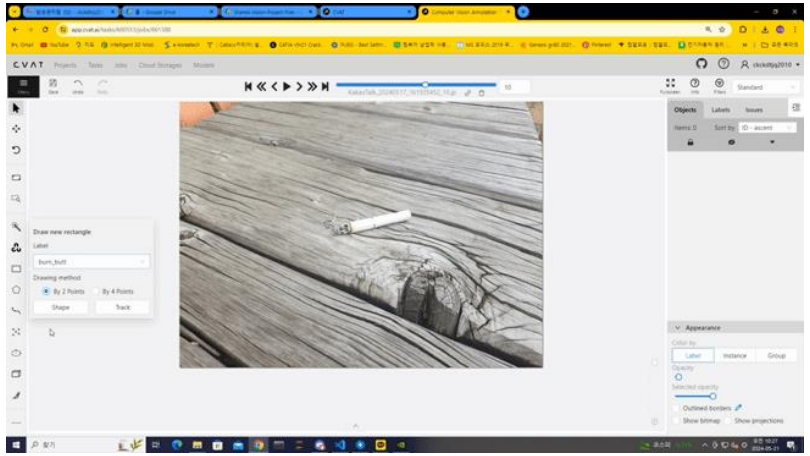
3) 이미지 추가로 수집 후 annotation 작업 수행

I. 모델의 정확도를 높이기 위해 Bing에서 담배 꽁초 이미지를 크롤링함

II. 실제 담배 꽁초 사진도 촬영

III. 추가로 수집한 이미지는 annotation 도구 사이트인 CVAT에서 작업 수행

실제로 촬영한 이미지



CVAT에서 Annotation 작업 수행

1. 학습 데이터 셋 구축

4) 이미지 분류 후 라벨 부여

I. 데이터를 불이 붙은 꽁초와 일반 꽁초로 나눠 두 개의 폴더에 저장

II. 일반 꽁초의 라벨은 {0: 'cig_butt'}, 불이 붙은 꽁초의 라벨은 {1: 'burn_butt'}으로 수정

III. 두 개의 폴더를 합쳐 데이터 셋 완성

2. 꽁초 탐지 모델 학습

순서

data.yaml
파일 수정

YOLOv8
모델을 기반으로
커스텀 데이터
재학습

2. 꽁초 탐지 모델 학습

1) data.yaml 파일 수정

I. 데이터 셋의 정의가 저장되어있는 파일

II. 데이터 경로와 클래스를 구축한 데이터 셋에 맞게 수정함

```
{
  0: ‘0’,
  1: ‘cig_butt’
}
```

→

```
{
  0: ‘cig_butt’
  1: 'burn_butt'
}
```

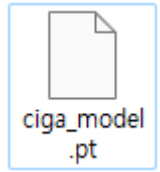
2. 꽁초 탐지 모델 학습

2) YOLOv8 모델을 기반으로 커스텀 데이터 재학습

I. Ultralytics YOLOv8 모델을 사용

II. 구축한 커스텀 데이터를 이용해 모델을 재학습함

```
device = torch.device("cuda" if
torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")

model = YOLO("yolov8n.pt")

model.train(data=data_yaml_path,
            epochs=50,
            patience=20,
            batch=32)
```

생성된 담배 꽁초 탐지 모델

3. 탐지 로봇 제작 및 구동 테스트

순서

1) 탐지 로봇 제작

2) GPIO를 통한 모터 제어 테스트

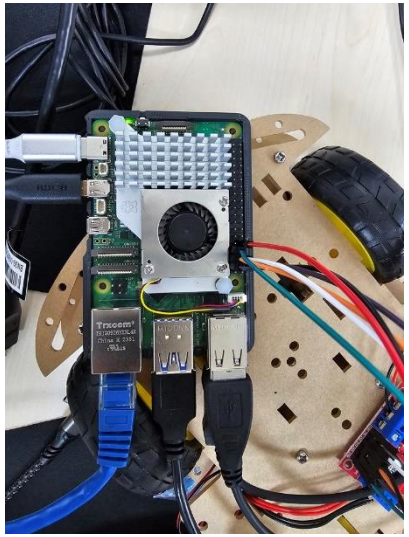
3) 키보드를 이용한 탐지 로봇 조작 기능 구현

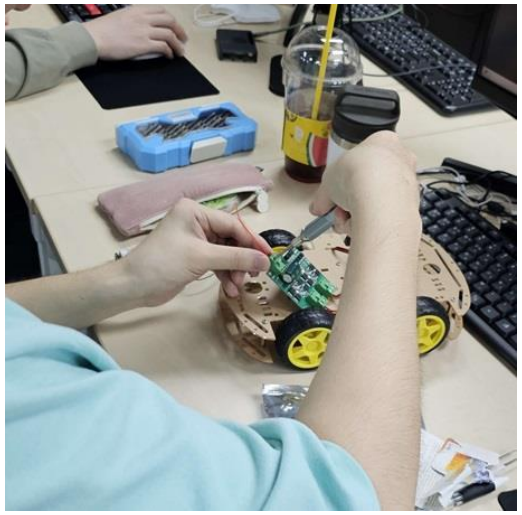
3. 탐지 로봇 제작 및 구동 테스트

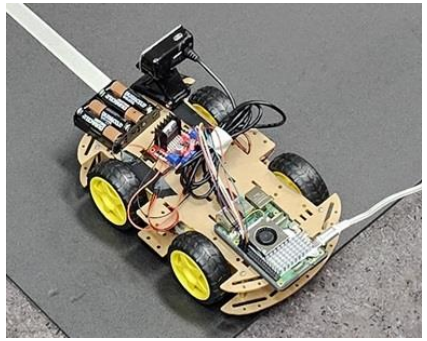
1) 탐지 로봇 제작

I. RC카 키트 위에 라즈베리 파이와 기타 장치들을 장착하여 탐지 로봇을 제작함

II. AA건전지 * 4로 모터 드라이버에 전원 공급, 라즈베리 파이는 5V 5A 충전기 사용







3. 탐지 로봇 제작 및 구동 테스트

2) GPIO를 이용한 모터 제어 테스트

I. 라즈베리 파이의 GPIO 제어는 파이썬 gpiozero 모듈을 사용했음

II. 모터 드라이버를 GPIO로 연결 후 모터 제어 코드를 작성하여 테스트

```python
from gpiozero import PWMOutputDevice, DigitalOutputDevice
from time import sleep

PWMA, AIN1, AIN2 = 12, 16, 20
PWMB, BIN1, BIN2 = 13, 19, 26

# 모터 객체 생성
L_Motor = PWMOutputDevice(PWMA)
R_Motor = PWMOutputDevice(PWMB)

# 방향 제어 핀 설정
AIN1_pin = DigitalOutputDevice(AIN1)
AIN2_pin = DigitalOutputDevice(AIN2)
BIN1_pin = DigitalOutputDevice(BIN1)
BIN2_pin = DigitalOutputDevice(BIN2)
```

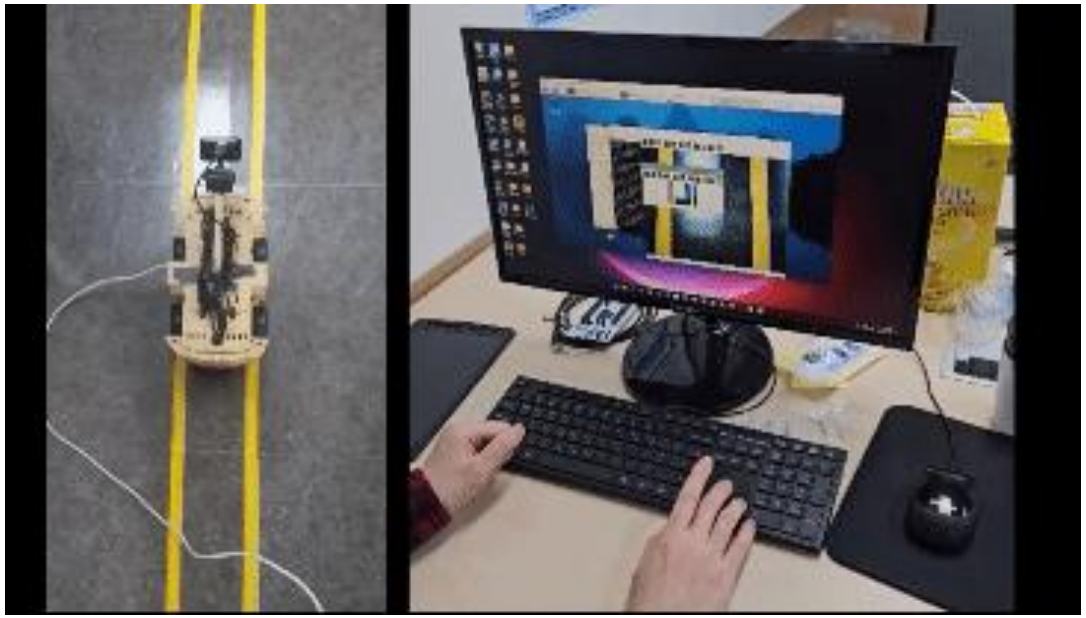
3. 탐지 로봇 제작 및 구동 테스트

3) 키보드를 이용한 탐지 로봇 조작 기능 구현

I. 키보드 입력 시 모터가 작동하여 움직이게 하는 기능 구현

II. 좌, 우측 회전과 직진 가능

```
keyValue = cv2.waitKey(10)

if keyValue == ord('q'):
    break
elif keyValue == 82:
    print("go")
    carState = "go"
    motor_go(speedSet)
elif keyValue == 84:
    print("stop")
    carState = "stop"
    motor_stop()
elif keyValue == 81:
    print("left")
    carState = "left"
    motor_left(speedSet)
elif keyValue == 83:
    print("right")
    carState = "right"
    motor_right(speedSet)
```



4. 자율주행 구현

순서

1) 주행 데이터 수집 전 카메라에 전처리 적용

2) 키보드를 이용한 수동 조작으로 주행 데이터 수집

3) 수집한 주행 데이터로 자율주행 모델 생성

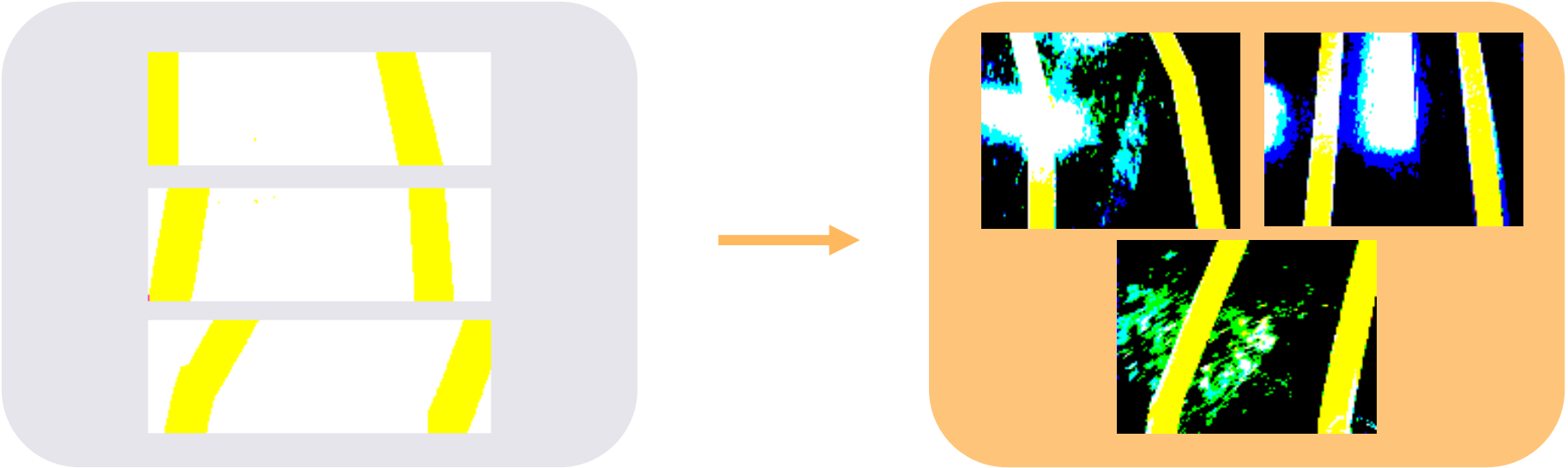
4) 자율주행 테스트

4. 자율주행 구현

1) 주행 데이터 수집 전 카메라에 전처리 적용

I. 트랙의 특징을 더욱 잘 보기 위해 BGR2YUV 변환 후 Threshold를 주어 이진화 하여 전처리 작업 수행

II. 학습시킨 model이 실제 주행 환경에서의 반사광 등을 반영하지 못하여 data 새로 수집

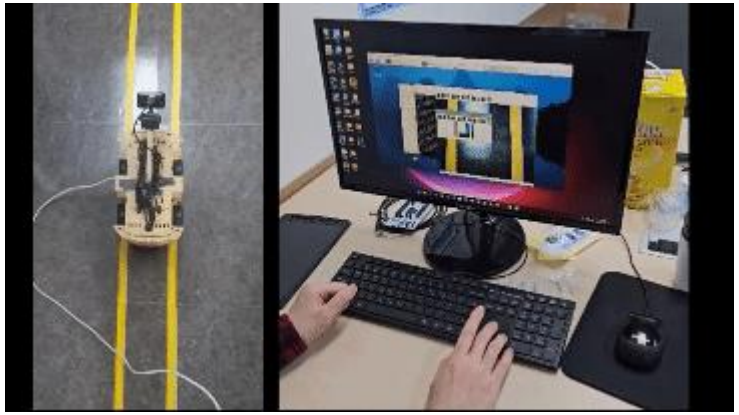


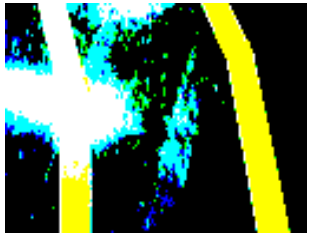
4. 자율주행 구현

2) 키보드를 이용한 수동 조작으로 주행 데이터 수집

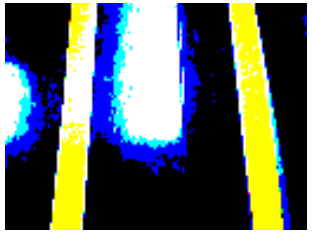
I. 키보드 방향키 입력 시 전처리된 카메라 이미지를 저장함

II. 좌측, 상단, 우측 방향키 입력 시 각각 “left”, “go”, “right” 폴더에 이미지 저장

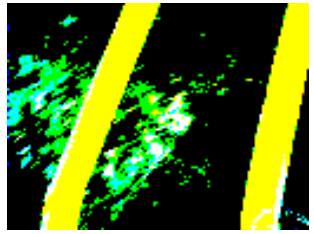




좌측 방향키 입력



상단 방향키 입력



우측 방향키 입력

4. 자율주행 구현

3) 수집한 주행 데이터로 자율주행 모델 생성

I. 앞서 수집한 주행 데이터를 이용하여 자율 주행 모델을 생성함

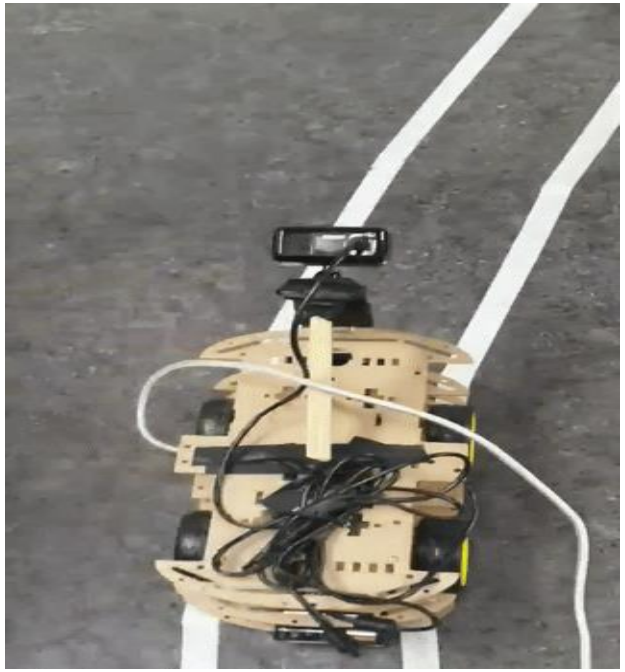
II. 텐서플로우 Keras 사용

```
    model = tf.keras.Sequential()

    model.add(Conv2D(24, (5, 5), input_shape=(96, 128, 3), activation='elu',
padding="same"))
    model.add(Conv2D(36, (5, 5), activation='elu', padding="same"))
    model.add(Dropout(0.2))
    model.add(Conv2D(64, (3, 3), activation='elu', padding="same"))

    model.add(Flatten())
    model.add(Dropout(0.2))
    model.add(Dense(50, activation='elu'))
    model.add(Dense(10, activation='elu'))

    model.add(Dense(1))

    optimizer = Adam(learning_rate=1e-3)
    model.compile(loss='mse', optimizer=optimizer)
```

4. 자율주행 구현

4) 자율주행 테스트

I. 트랙을 만들어 학습한 데이터로 자율 주행 테스트 실시



5. 탐지 기능 추가

순서

1) 두 가지 모델을 이용하여 동시 추론 수행 테스트

2) 꽁초 탐지 모델 파일을 다른 포맷으로 변환

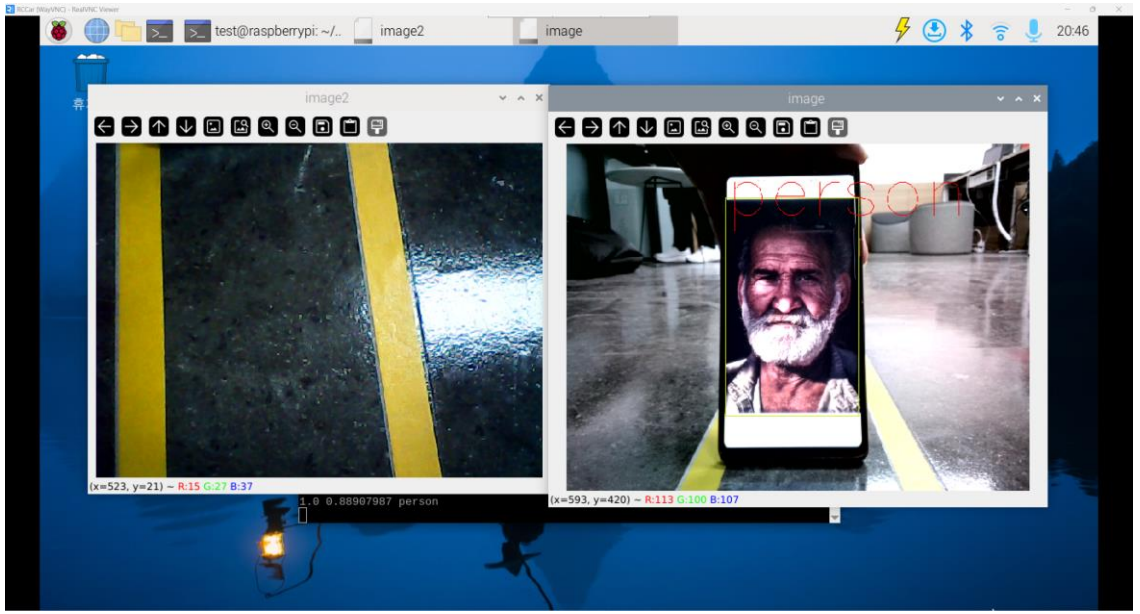
3) 변환한 탐지 모델을 라즈베리 파이에 적용

5. 탐지 기능 추가

1) 두 가지 모델을 이용하여 동시 추론 수행 테스트

I. 담배 꽁초 탐지 모델을 적용하기 전, COCO 탐지 모델을 이용하여 자율주행과 동시에 수행 시 정상적으로 작동하는 지 테스트를 수행함

II. 정상적으로 작동하는 것을 확인했으나 추론 속도가 매우 떨어지는 것을 확인함 (성능 문제)

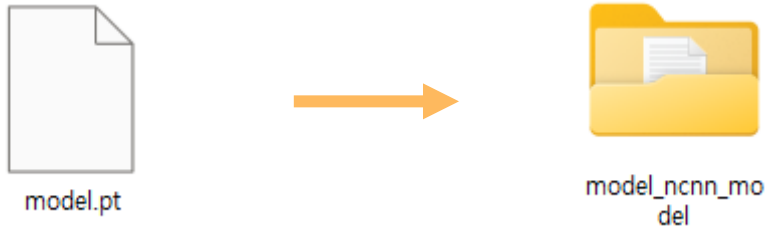


5. 탐지 기능 추가

2) 꽁초 탐지 모델 파일을 다른 포맷으로 변환

I. 성능을 높이기 위해 PyTorch 형식으로 만든 꽁초 탐지 모델을 NCNN 형식으로 변환

```
# Load the exported NCNN model
model_name = 'C:/Users/USER/Documents/vision-AI/project/7899/models/ccs-model_ncnn_model'

ncnn_model = YOLO('./models/ccs-model_ncnn_model', task="detect")

# 이미지에 대한 추론 수행
results = ncnn_model.predict(source='./test_img1.jpg', save=True)
```

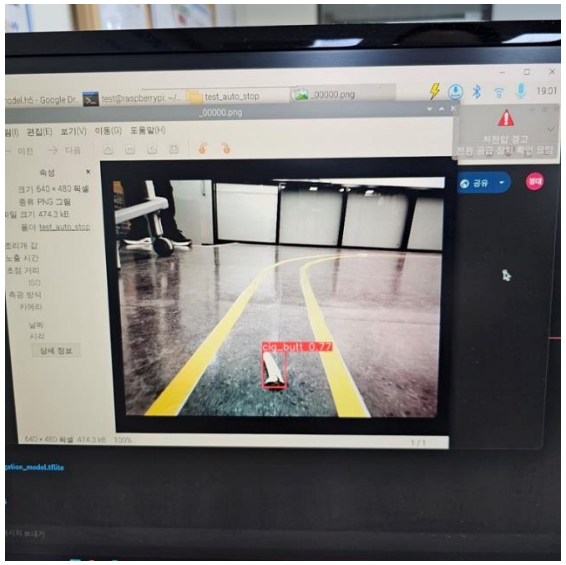


5. 탐지 기능 추가

3) 변환한 탐지 모델을 라즈베리 파이에 적용

I. 변환한 담배 꽁초 탐지 모델을 라즈베리 파이에 적용한 후 추론 테스트를 수행

II. 이전보다 추론 속도가 향상되었고, 담배 꽁초도 제대로 인식하는 것을 확인함



6. 통합 테스트

순서

1) 구현한 기능 테스트

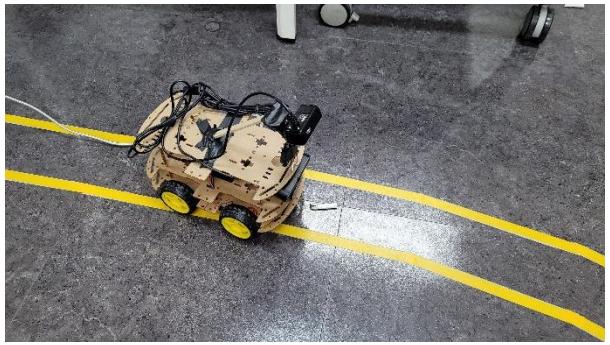
6. 통합 테스트

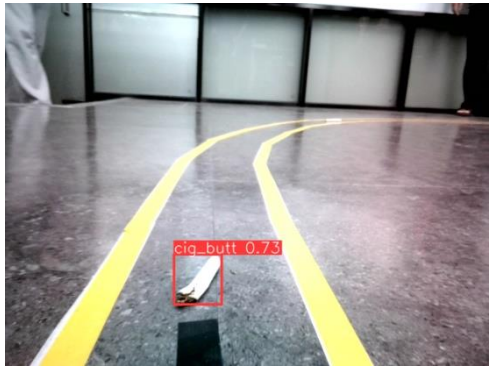
1. 앞서 만든 모든 기능들의 테스트 수행

I. 자율 주행 및 담배 꽁초 탐지 테스트를 수행함

II. 트랙을 따라 자율 주행하다 담배 꽁초를 발견할 경우 이미지 캡쳐 후 정지

III. 테스트하며 발견한 문제점들을 보완함





04 문제점 및 해결 방안

1. 학습한 모델의 테스트 중 발생한 문제
2. 자율주행 데이터 수집 중 발생한 문제
3. 주행 시 모터가 정상 작동하지 않는 문제
4. 웹캠 성능으로 인한 문제
5. 라즈베리 파이의 성능 부족으로 인한 문제
6. 객체 탐지 시 오인식하는 문제

1. 학습한 모델의 테스트 중 발생한 문제

1) 이미지 크기 오류로 인한 과탐지

- 이미지 추론 시 탐지가 과도하게 일어나는 현상 발생

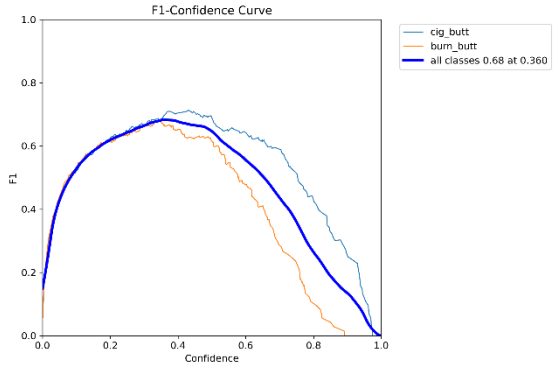
  ⟹ 학습 이미지 크기 변경 (416x416→640X640)

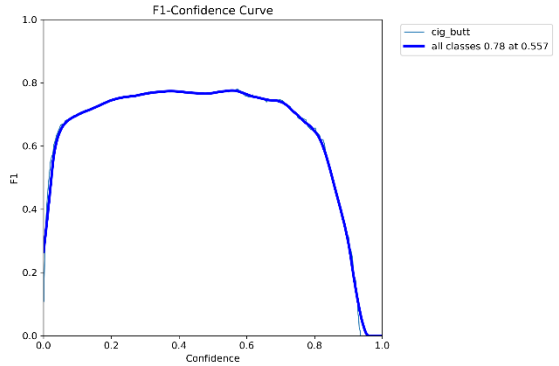
1. 학습한 모델의 테스트 중 발생한 문제

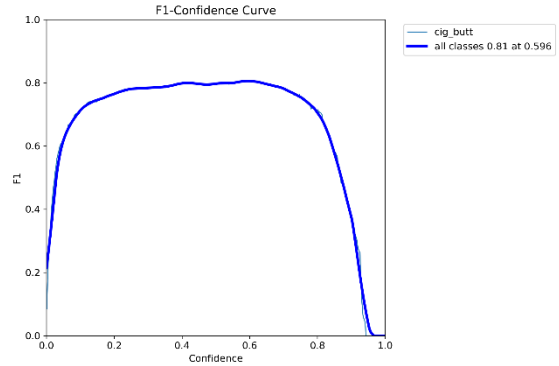
2) 모델의 정확도가 높지 않음

⇒ 학습률에 관련이 있는 변수(Hyperparameter)를 조정

	1차	2차	3차
epoch	15	100	200
patience	10	10	20
batch	8	16	32
f1	0.68~0.360	0.72~0.480	0.81~0.596





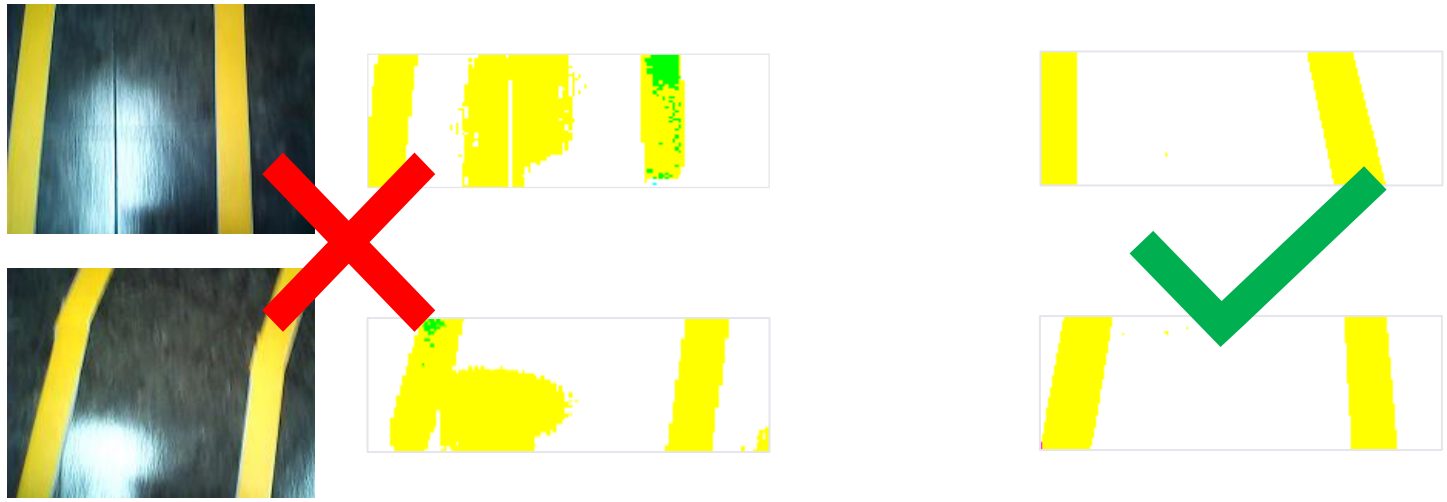


2. 자율주행 데이터 수집 중 발생한 문제

트랙의 상태가 좋지 못함

- 기존에 준비한 폼보드는 마찰력이 약해 주행이 어려움
- 타일은 조명의 영향을 크게 받음

⇒ 타일에서 주행하며 수집한 데이터 중 조명의 영향을 받지 않은 이미지만 학습함



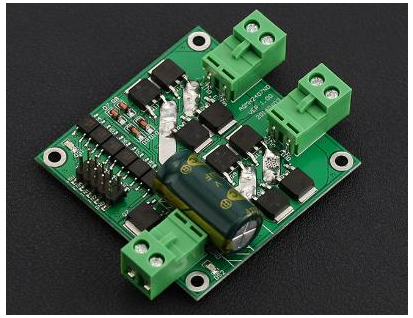
3. 주행 시 모터가 정상 작동하지 않는 문제

1) 모터 드라이버의 최소 요구 전압

- 기존에 사용하려던 모터 드라이버 AQMH2407DN는 데이터 시트 확인 결과 6.5V이상부터 작동
- 공급 전원은 1.5V * 4 = 6V 전압이기 때문에 모터 드라이버의 요구치에 못미침
  ⇒ 모터 구동 입력 전압이 4.7V 이상인 L298N 모터 드라이버로 교체

Table 1.1 AQMH2407ND motor driver module electrical parameters

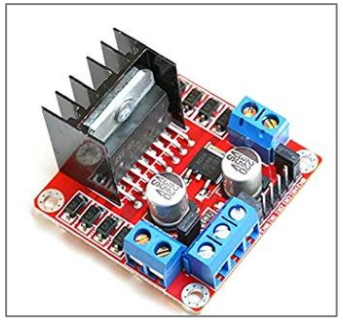
project	parameter	Remark
Input voltage range	DC6.5V ~ 27V	The power supply must not be connected reversely. It is recommended to connect a 15A fuse in series with the power supply, and the voltage should not exceed 27V, otherwise the module may be burned.



AQMH2407DN

Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver



L298N

3. 주행 시 모터가 정상 작동하지 않는 문제

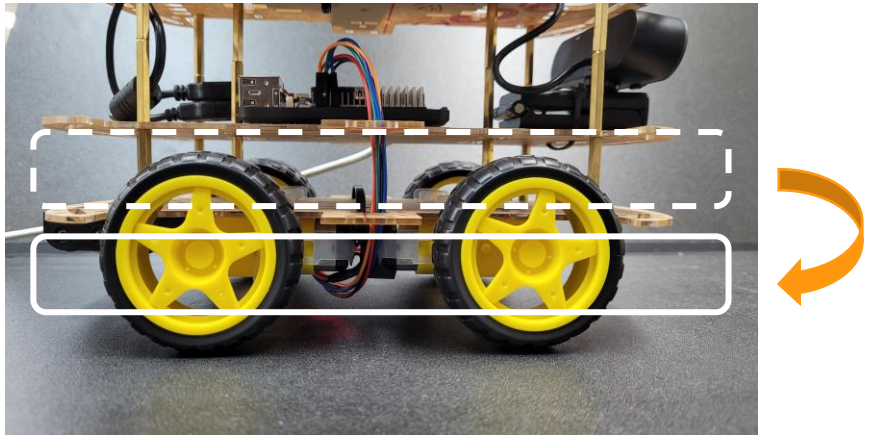
2) 무게 중심 쏠림으로 인해 바퀴가 헛도는 문제

I. 탐지 로봇의 무게 중심이 한쪽으로 쏠려 바퀴마다 가해지는 하중이 일정하지 않았음

II. 모터가 탐지 로봇의 하단판 위에 위치하여 하중을 직접적으로 받지 못함

III. 충분하지 못한 마찰력과 모터 노후화로 인해 바퀴가 헛돎

⇒ 모터 교체 및 차체 하중의 전달을 위해 모터의 위치를 하단판 밑으로 이동



4. 웹캠 성능으로 인한 문제

빛 번짐으로 인해 꽁초를 탐지하는데 어려움이 발생함

I. 주행 시 카메라가 흔들려 빛 번짐이 발생

II. 입력 이미지에 노이즈가 발생하여 꽁초 탐지 불가

⇒ 꽁초 탐지 카메라를 빛 번짐이 덜한 모델로 교체

PLEOMAX W-210

ABKO APC930U

5. 라즈베리 파이의 성능 부족으로 인한 문제

1) 모델 중첩으로 인한 연산 과부하 및 지연

I. SBC(Single Board Computer) 특성 상 연산량에 한계가 있음

II. 꽁초 감지와 트랙 감지를 동시 진행 시 연산이 지연되거나 잘못된 연산 결과가 출력

⇒ 다중 쓰레드 및 뮤텍스를 통해 연산 안정성 확보

```python
if __name__ == '__main__':
    capture_thread = threading.Thread(target=capture_and_preprocess)
    dnn_thread = threading.Thread(target=opencvdnn_thread)

    capture_thread.start()
    dnn_thread.start()

    main()

    capture_thread.join()
    dnn_thread.join()
    camera.release()
    camera2.release()
    cv2.destroyAllWindows()
```

```python
def capture_and_preprocess():
```

```python
def opencvdnn_thread():
```

```python
def main():
```

5. 라즈베리 파이의 성능 부족으로 인한 문제

2) YOLO 모델 자체의 무거운 구조

I. Object Detection 특성 상 과도한 연산이 발생

II. 해당 연산 처리로 인해 추론시간이 길어짐

⇒ 라즈베리 파이에 적절한 모델 형식으로 변환 후 사용

성능

YOLOv8n RPi5에서 | YOLOv8s RPi5에서 | YOLOv8n RPi4에서 | YOLOv8s RPi4에서

형식	상태	디스크 크기(MB)	mAP50-95(B)	추론 시간(ms/im)
PyTorch	✅	6.2	0.6381	508.61
TorchScript	✅	12.4	0.6092	558.38
ONNX	✅	12.2	0.6092	198.69
OpenVINO	✅	12.3	0.6092	704.70
TF SavedModel	✅	30.6	0.6092	367.64
TF GraphDef	✅	12.3	0.6092	473.22
TF Lite	✅	12.3	0.6092	380.67
PaddlePaddle	✅	24.4	0.6092	703.51
NCNN	✅	12.2	0.6034	94.28

```
model.pt

0: 480x640 (no detections), 545.1ms
Speed: 9.0ms preprocess, 545.1ms inference, 0.7ms postprocess per image at shape
 (1, 3, 480, 640)
```

```
model.onnx

0: 640x640 (no detections), 205.3ms
Speed: 8.4ms preprocess, 205.3ms inference, 1.2ms postprocess per image at shape
 (1, 3, 640, 640)
```

```
model.ncnn

0: 640x640 (no detections), 123.9ms
Speed: 9.2ms preprocess, 123.9ms inference, 2.0ms postprocess per image at shape
 (1, 3, 640, 640)
```

6. 객체 탐지 시 오인식하는 문제

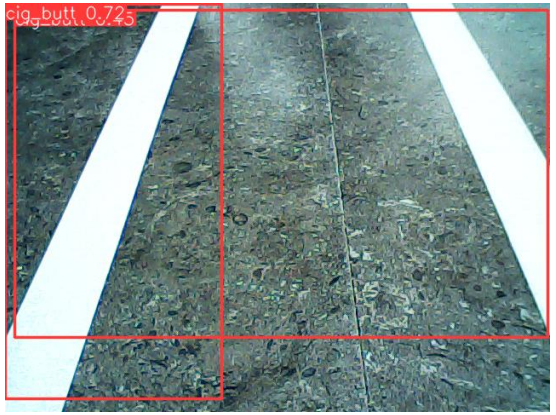
1) 꽁초 탐지 모델이 라인을 담배로 인식

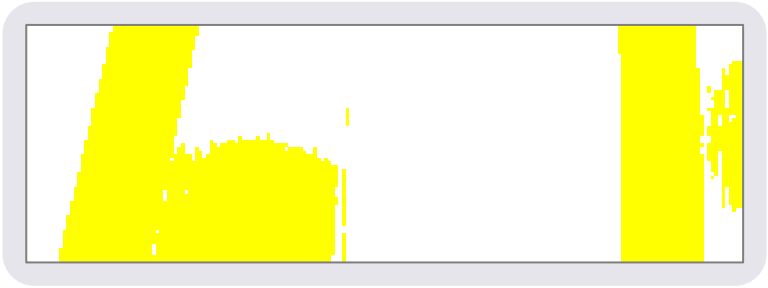
- 트랙을 꽁초로 인식하여 의도치 않은 긴급정지 발생

2) 트랙 탐지 시 조명 노이즈

- 학습 데이터에 노이즈를 제외하여 실제 환경의 노이즈에 대해 오작동 문제 발생

  ⟹ 트랙 색상 변경 및 학습 데이터에 노이즈를 포함한 RGB 이미지 학습





6. 객체 탐지 시 오인식하는 문제

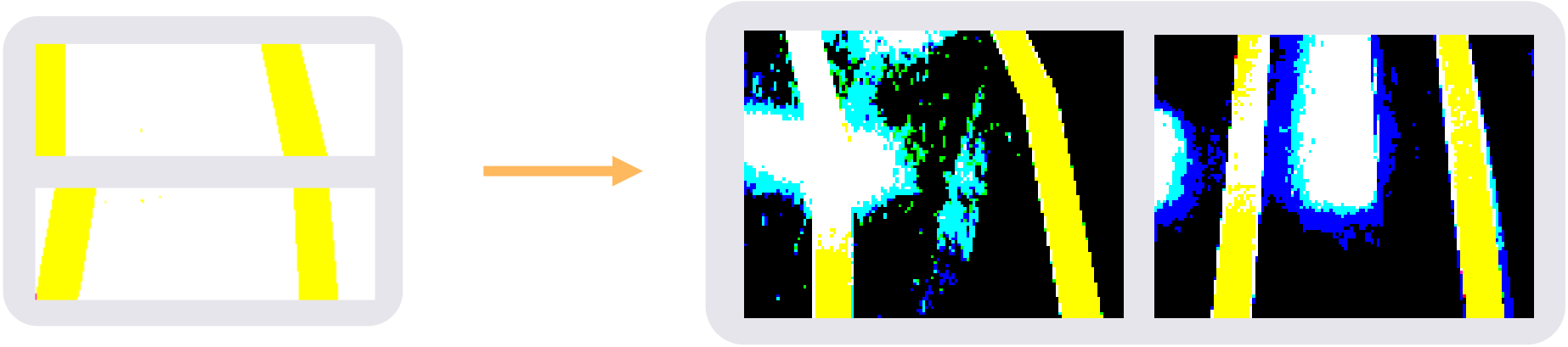
1) 꽁초 탐지 모델이 라인을 담배로 인식

- 트랙을 꽁초로 인식하여 의도치 않은 긴급정지 발생

2) 트랙 탐지 시 조명 노이즈

- 학습 데이터에 노이즈를 제외하여 실제 환경의 노이즈에 대해 오작동 문제 발생

  ⇒ 트랙 색상 변경 및 학습 데이터에 노이즈를 포함한 RGB 이미지 학습



05 결과물

1. 시연 영상

1. 시연 영상


Cig-Bee
탐지 로봇을 이용한 담배 꽁초 탐지
[Intel] 엣지 AI SW 아카데미 4기
1조 STACK
권오준, 김정대, 장혜원, 차창섭, 최재혁

시연영상

감사합니다