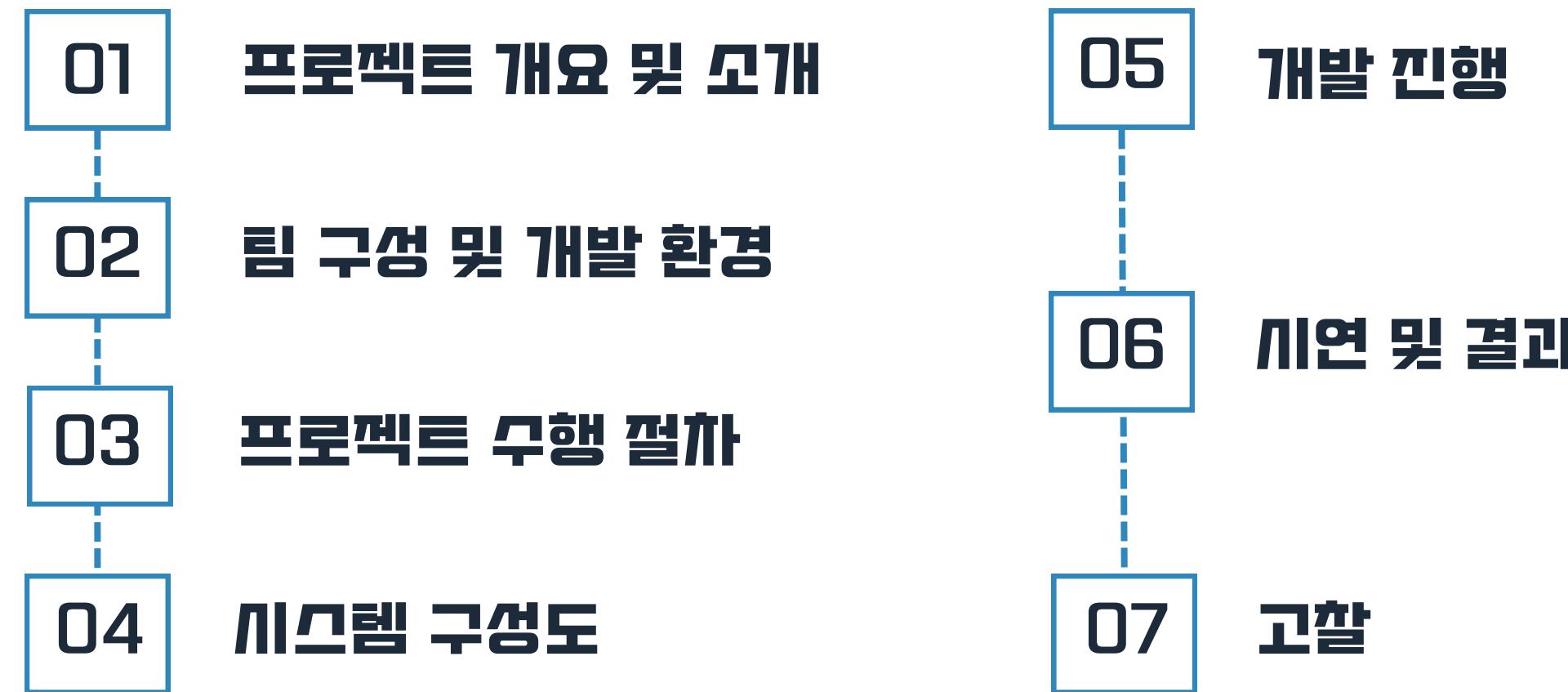




(Drone Tracking System)

Date. 2024.07.02 | Team. 4조 D-Gaja | 이성우, 김민정, 김윤우, 허찬욱

목 차



01

프로젝트 개요 및 소개

- 문제 정의
- 프로젝트 목표

드론 날려 나체 촬영한 30대 징역 8개월 법정구속

| 드론에 장착된 카메라로 불법 촬영

입력 : 2022-01-13 11:21

한밤중 드론 띄워 10쌍 성관계 불법촬영 40대 男 실형

동아닷컴 | 업데이트 2021-02-02 16:25



늘어난 드론에 각종 범죄 활개…불안한 시민들

입력 2023-06-18 15:00:28 수정 2023.06.18 15:00:28 정유민 기자

드론이 상용화되면서 불법 촬영 등 각종 범죄에 악용됨에 따라 불안함을 호소하는 시민들이 늘고 있다. 지난 5일 새벽 서울 마포구 한 고층건물에서 드론을 띄워 여성의 나체를 촬영한 혐의를 받는 남성이 논란을 낳은 데 이어 서울 시내 상반기(2월 22일~6월 10일) 미확인 드론 신고 건수가 144건으로 집계됐다. 구체적인 신고 처리 내용은 △승인 44건 △불발견 42건 △오인 9건 △훈방 16건 △통보 33건으로 확인됐다.

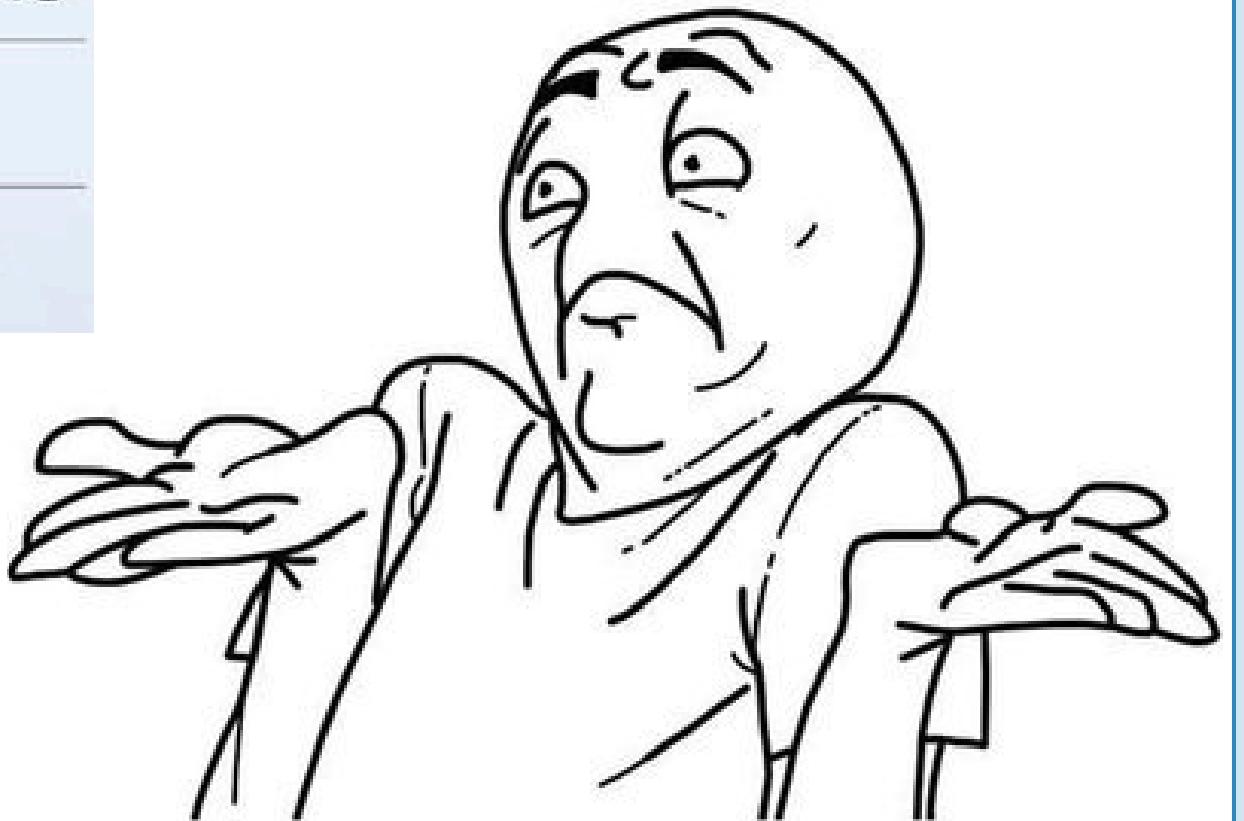
01 문제 정의(1)

DTS (Drone Tracking System)

• Anti-Drone System의 종류

종류	특징	효과	문제점
전파방해	드론이 쓰는 통신 및 위성항법(GPS) 신호를 파악, 더 강한 세기의 전파를 발신	드론 비행 불가능	주변에서 사용하는 다른 전자기기도 무력화
스푸핑 (Spoofing)	거짓 GPS 신호를 보내 드론을 나포하거나 다른 방향으로 비행도록 유도	지상 낙하 위험 피하면서 드론 제압	주변 지역 전자기기도 영향
그물	지상 또는 공중에서 그물을 펼쳐 드론 포획	쿼드콥터 포획에 유용	유효거리가 매우 짧음. 고속 드론에는 적용 불가능
맹금류	독수리 등을 훈련시켜 드론 포획에 투입	지상 낙하 위험 등 낮아	고속 드론에는 적용 불가능
대공포	기관포나 지대공 미사일 등으로 공격·파괴	신뢰성과 위력이 강해 드론 요격 성공률 높음	지상 낙하시 2차 피해 우려
레이저	레이저 무기로 드론을 공격	빛의 속도로 발사, 다수 표적 대응 가능	날씨 등에 영향을 받음

현재 군용 Anti-Drone System만 존재



민간용은 ?

01 프로젝트 목표(1)

DTS (Drone Tracking System)

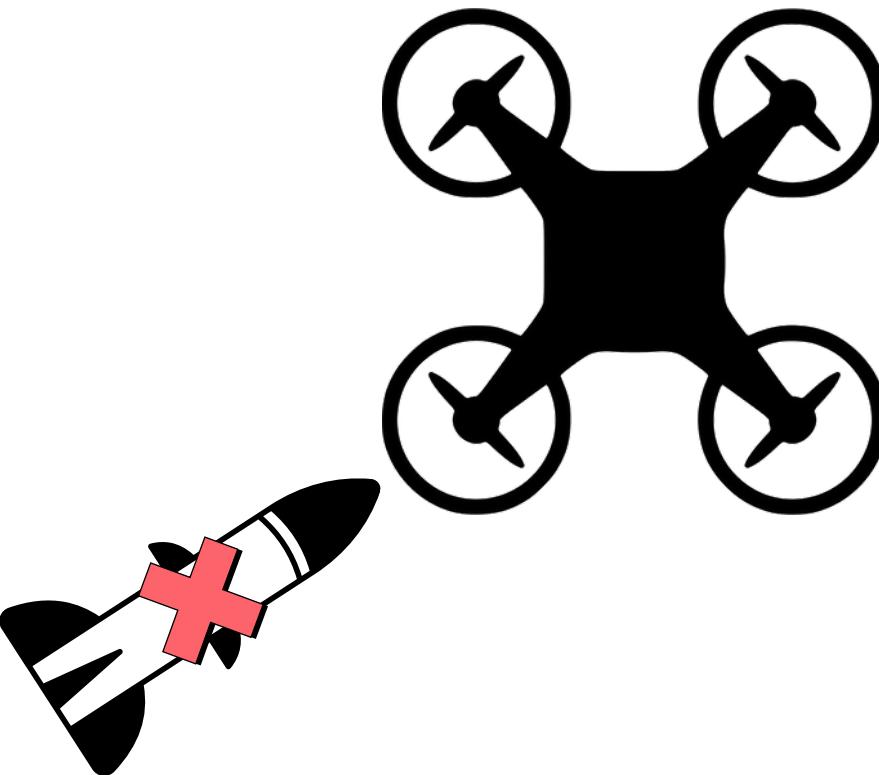


주요 드론 관련 범죄 종류

- 비행 금지구역 불법 비행
- 불법촬영
- 마약 거래



깨물곤고!



'경찰 드론' 운영 관서 35개소로 확대…634명 자격증 보유

실종자 등 91명 구조…전문인력 확대 배치

신형 드론 시범운영…탐조등 등 장비 도입

(서울=뉴스1) 송상현 기자 | 2023-06-27 12:00 송고

전남자치경찰, 치안 드론을 활용한 범죄 예방 체계 구축

기사승인 2024.01.03 17:31:22

- 행안부 특별교부세 1억 8천만원 확보, 이동식 드론 관제 시스템 구축



01 프로젝트 목표(2)

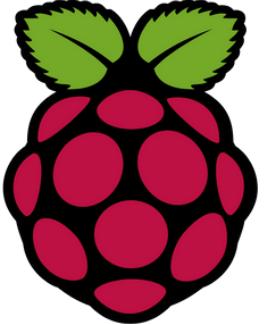
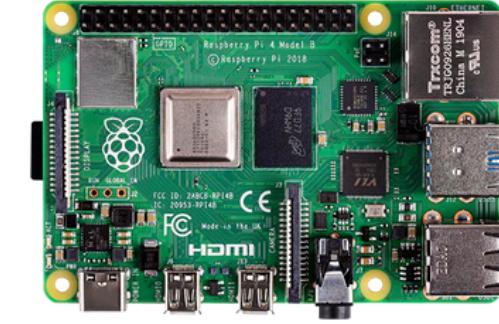
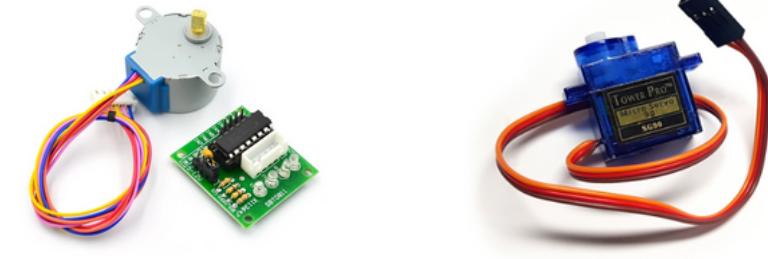
DTS (Drone Tracking System)



02

팀 구성 및 개발 환경

- 개발 환경
- 팀원 소개 및 역할 분담

언어	Python 3.11		
보드	Raspberry Pi (Ram : 2GB, 4-Core)	 	
모듈	Camera – Logitech(VU0040) Motor – Servomotor, Steppermotor Library – OpenCV-Python, Socket		
모델	OpenVINO(Object-Detection) Training – OTX	OS	Ubuntu 22.04.4 LTS (64-bit)

〈 D-gaja 〉

협업툴 :



Leader



이성우

- 보드 성능 개선
- 드론 객체 인식 및 추적
코드 개발



허찬욱

- AI 모델 학습
- H/W 설계 및 제작



김윤우

- 보드 개발환경 구축
- 모터 케이블 개발
- 원격통신 개발



김민정

- 카메라 구동 확인
- 마이크 모듈 개발

*D-gaja(드가까) : 드론으로부터 가드하는 짜들

03

프로젝트 수행 절차

- 프로젝트 수행 일정

03 프로젝트 수행 일정

DTS (Drone Tracking System)

	6월 18일	19일	20일	21일	24일	25일	26일	27일	28일	7월 1일	2일
팀빌딩											
기획 및 설계											
구현 및 개발											
테스트, 기제품 완성											
발표자료 제작, 시연 연습											
프로젝트 발표											

04

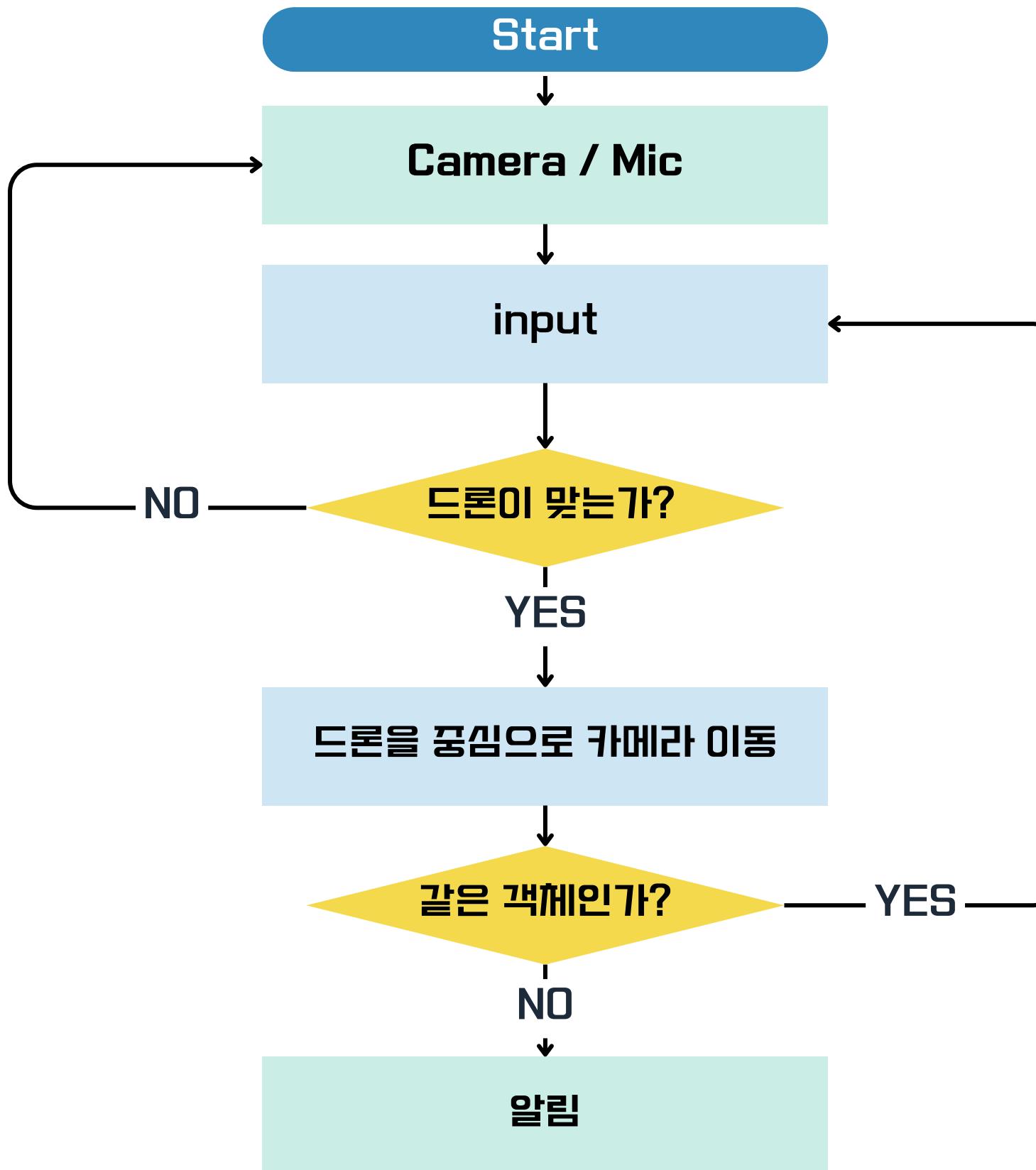
시그널 구성도

- **Process Diagram**
- **Circuit Diagram**
- **System Diagram**

04 Process Diagram

Flow Chart

DTS (Drone Tracking System)



04 System Diagram

DTS (Drone Tracking System)

Input

보드



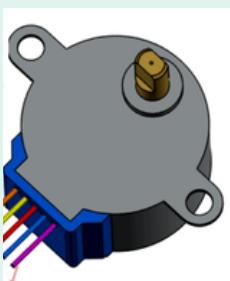
PC



UI 구성

Main.py

보드



좌우
360° 회전



상하
180° 회전

Stepper Motor

Servo Motor

모터 짹어로 카메라 회전

제어 통신

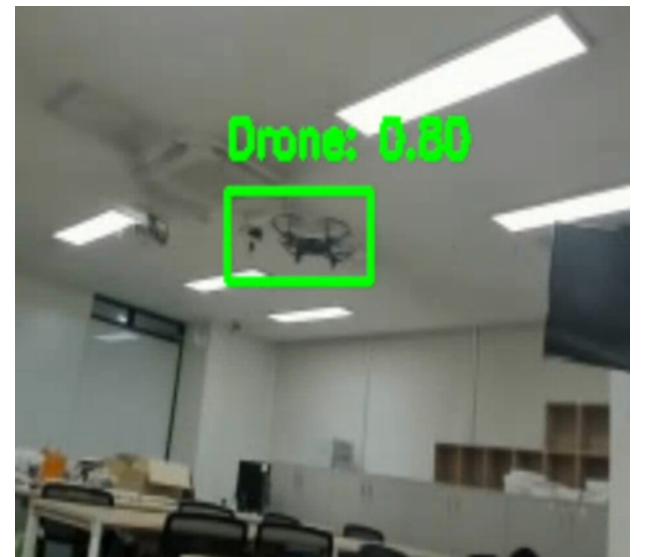
PC



드론 탐지 및 좌표값 계산

박스의 X축, Y축 좌표 값을
제어로 전송

Output



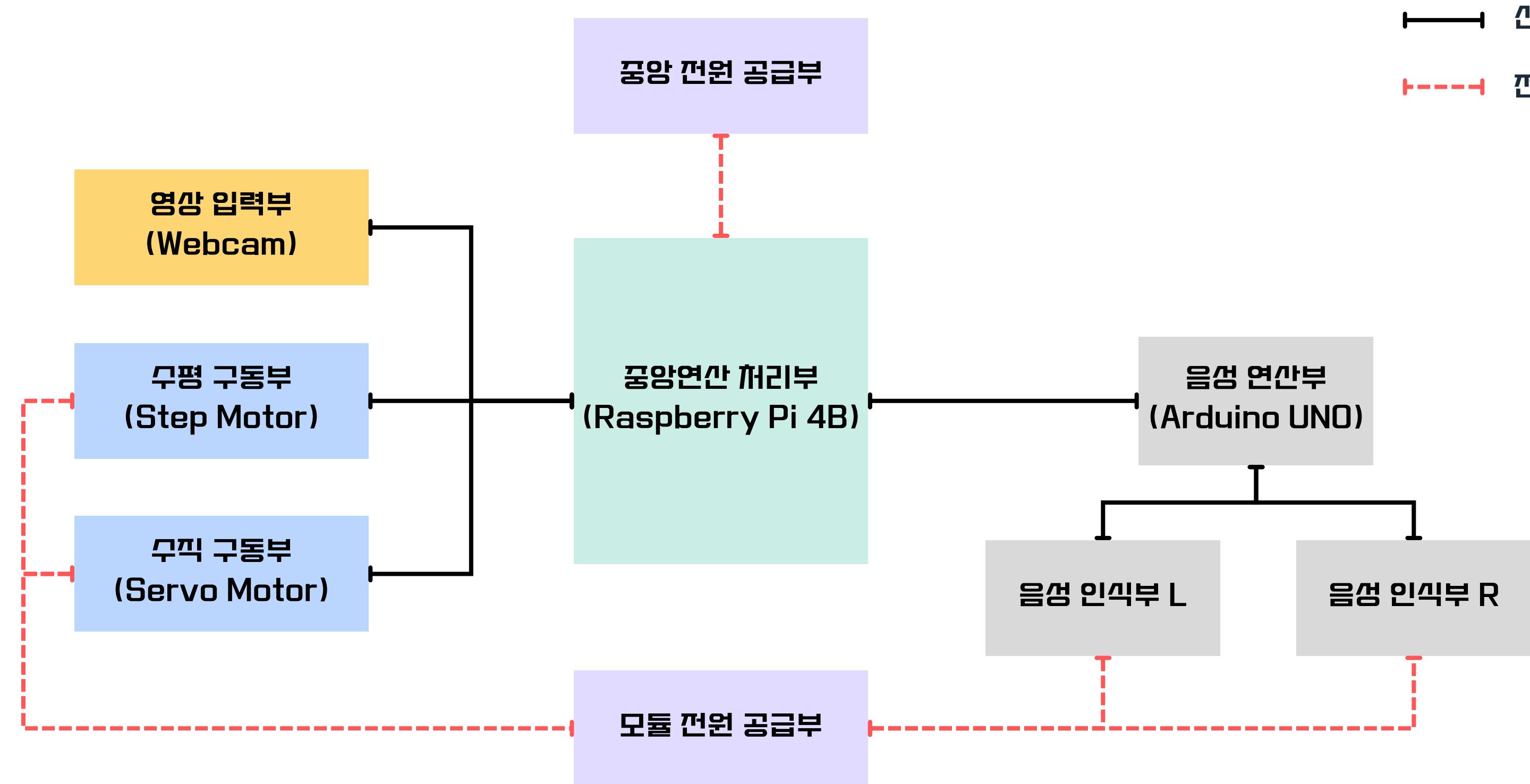
실시간 드론 탐지



사용자에게 알림

04 Circuit Diagram

DTS (Drone Tracking System)



05

개발 진행

- Project Issue
- 개발 진행 과정
- 회용 하드웨어 구성

05 Project Issue

DTS (Drone Tracking System)

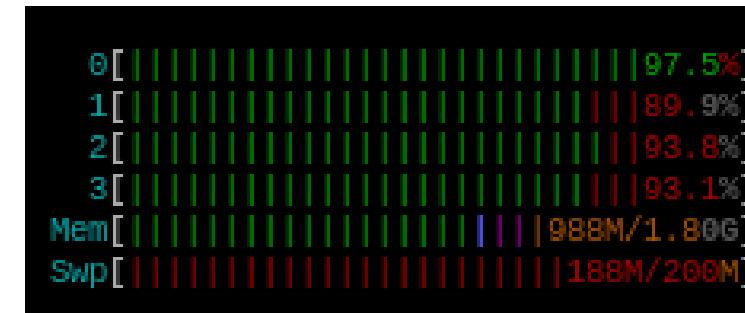
모델 선택

- 낮은 인식률
- 느린 inference time



보드 성능

- 과도한 메모리 사용
- 낮은 FPS



트래커

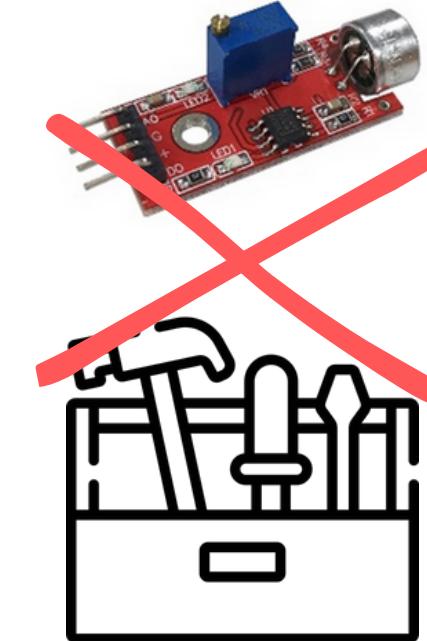
- 데이타 포맷 호환 문제
- 적합한 추적기 탐색 끽연



DELAY

하드웨어

- 마이크 성능 문제
- 내구성 부족



1. 모델 선택

1) 데이터 수집

Roboflow



444장의 공유 데이터셋 드론 이미지 사용

드론 데이터셋 확장



982장의 Roboflow 공유 데이터셋
드론 이미지로 교체

2) 모델 학습

OTX - Object Detection 모델 중 MobileNetV2 모델 사용

Datumaro를 활용, COCO Json 파일을 변환

OTX에서 모델 빌드, 학습, 평가, 투출, 최적화, 배포 과정 진행

모델 변경



MobileNetV2보다 성능이
가장 높은 YOLOX-X로 변경

```
otx find --template --task DETECTION
```

ID	NAME
Custom_Object_Detection_Gen3_ATSS	MobileNetV2 ATSS
Object_Detection_ResNeXt101_ATSS	ResNeXt101-ATSS
Custom_Object_Detection_Gen3_SSD	SSD
Object_Detection_YOLOX_L	YOLOX-L
Object_Detection_YOLOX_S	YOLOX-S
Custom_Object_Detection_YOLOX	YOLOX-TINY
Object_Detection_YOLOX_X	YOLOX-X



드론 객체 인식의 accuracy가
0.57에서 0.89로 향상된 것을 확인

2. 보드 성능

1) 모델의 부동 소수점 표현을 FP32에서 FP16으로 전환

2) input size 384x384 > 256x256으로 수정

3) OpenVINO의 비동기 실행(Async) 모드를 사용하여 여러 프레임 동시에 처리



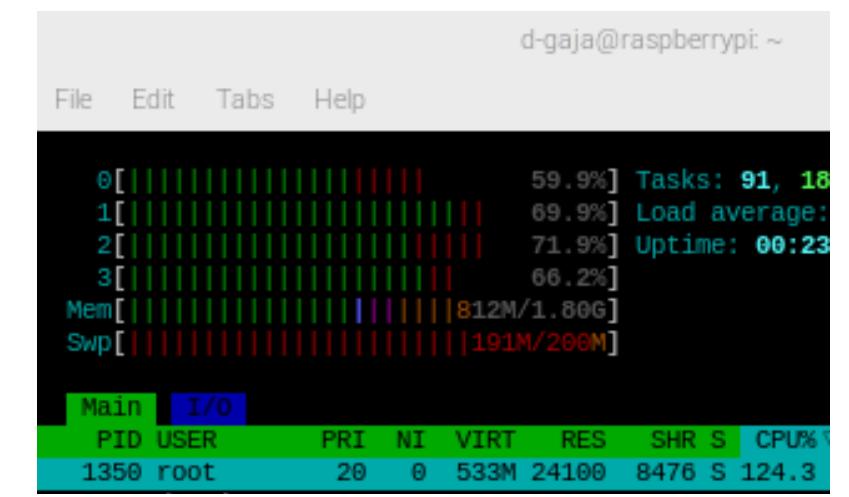
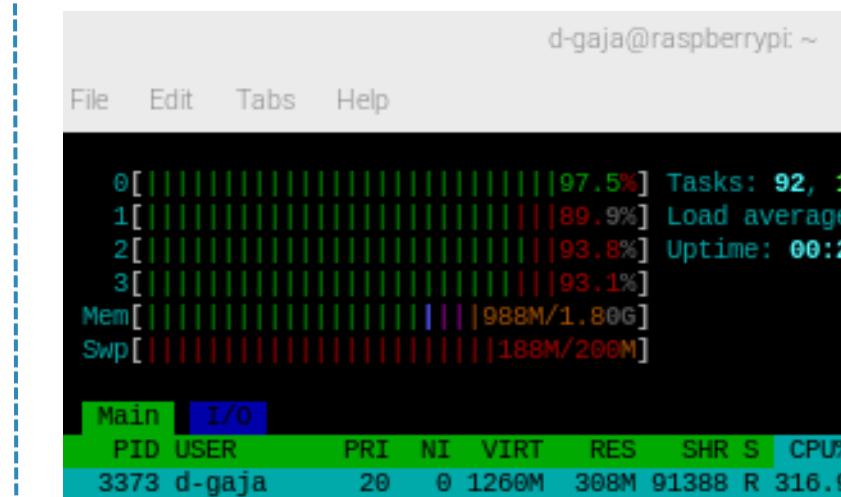
Sync 사용 시



Async 사용 시

10번의 레그스트로 평균값을 추출

FPS가 4.3에서 4.76으로 10.70% 증가



CPU 사용량 93.6 %에서 77%로 감소

드론 인식 및 후작 속도 개선

3. 트래커 - 객체 추적 알고리즘

1) 트래커 방식 변경

Deepsort에서 TrackerCSRT로 전환

Deepsort는 otx의 데이터 포맷과 짜리 방식의 차이로 인해 통합이 복잡
otx의 데이터 포맷과 잘 맞는 TrackerCSRT 선택

2) 트래커 코드

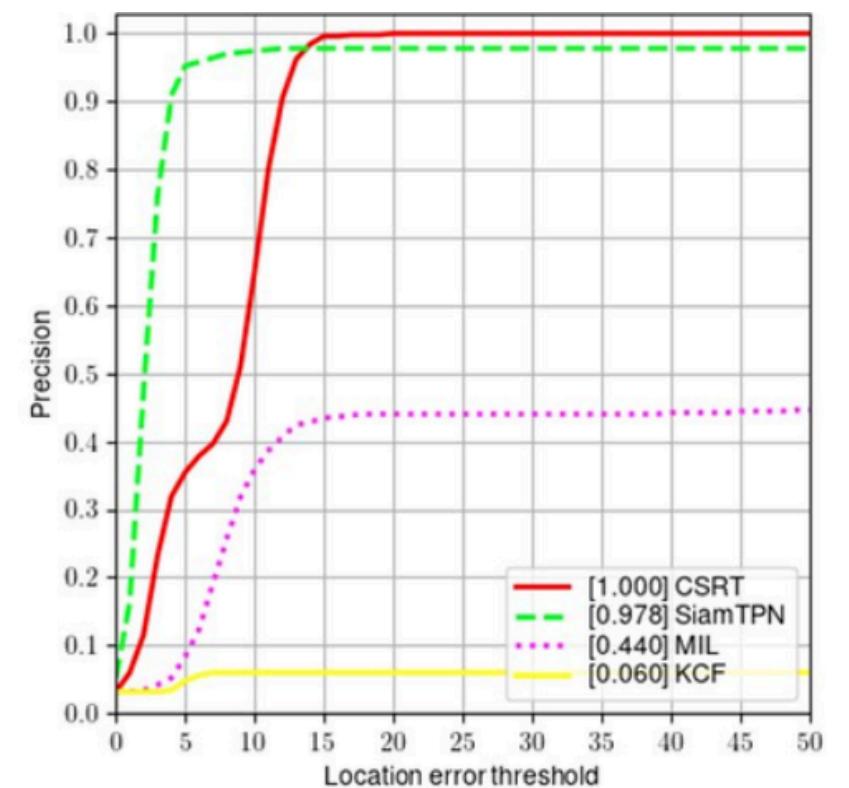
객체를 추적하기 위해 트래커를 초기화하고 객체의 새 위치를 업데이트 후.
추적된 객체의 경계를 사각형으로 그리는 작업 수행

```
# 추적기 초기화
tracker = cv2.TrackerCSRT.create()

# 추적 업데이트
bbox_tracker = tracker.update(frame)

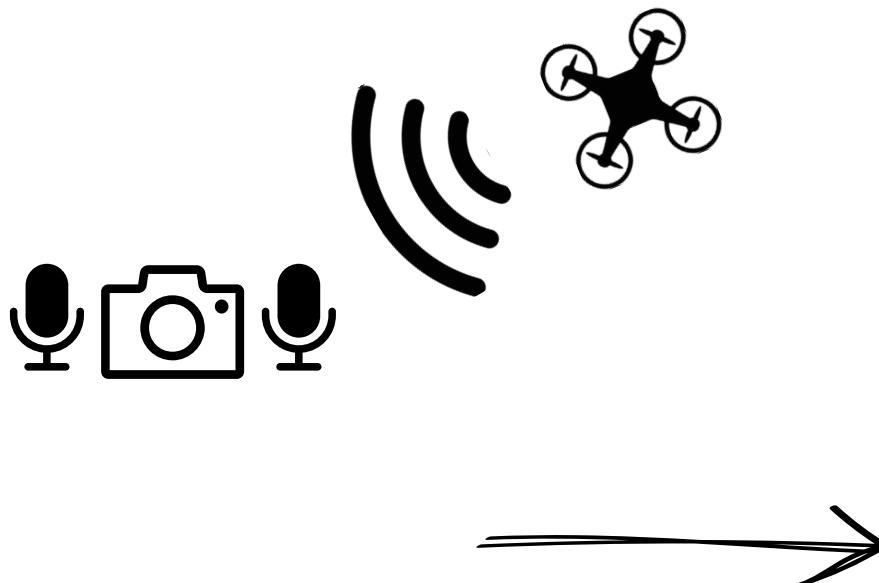
# 추적된 객체 사각형
p1 = (int(bbox_tracker[0]), int(bbox_tracker[1]))
p2 = (int(bbox_tracker[0] + bbox_tracker[2]), int(bbox_tracker[1] + bbox_tracker[3]))
cv2.rectangle(frame, p1, p2, (0,255,0), 2, 1)
```

CSRT 추적기의 정확도가 매우 높음



*Deepsort 방식
: 미각적 특징 벡터를 입력으로 사용
[feature_vector]

4. 하드웨어 - 마이크



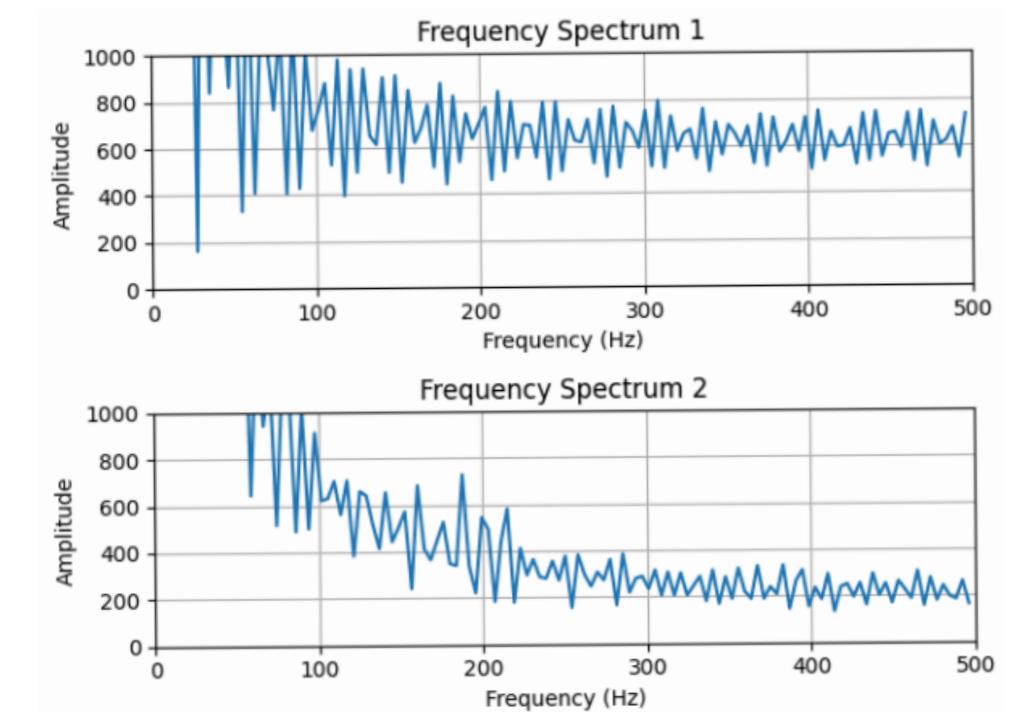
1) 모리 크기

모듈을 이용하여 모리 크기를 감지해 카메라 이동
→ 엔진가 모리가 감지하는 모리의 크기 변화가 없어 드론 감지 불가

L : 595, R : 602
L : 596, R : 603
L : 595, R : 607
L : 594, R : 603
L : 596, R : 605
L : 596, R : 606
L : 595, R : 604
L : 593, R : 602

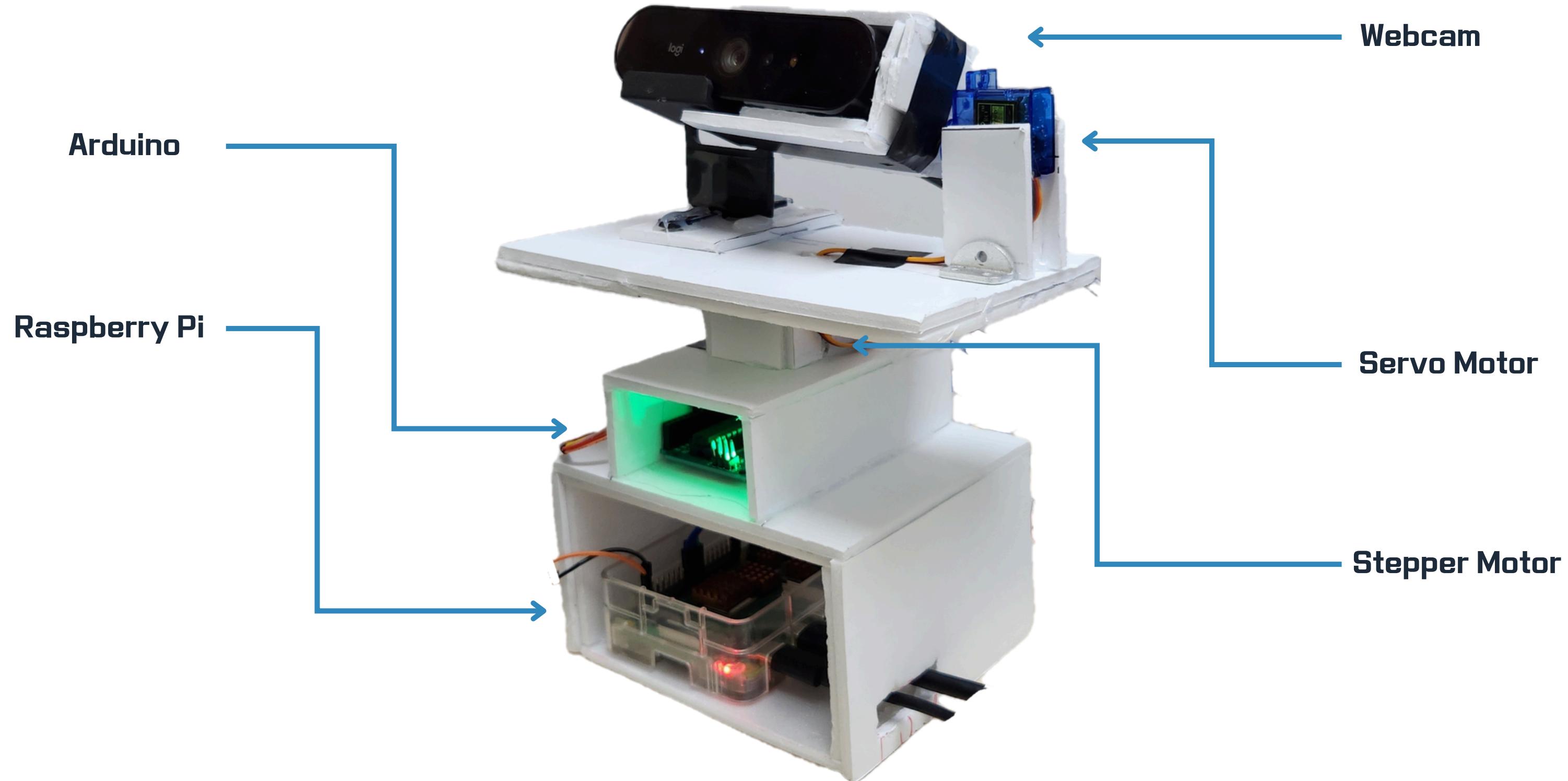
2) FFT 푸파수 기각화 분석

드론 모리 실시간 감지 및 모터 푸파수 영역 기각화
→ 엔진가 모리를 불규칙한 패턴으로 감지해 정확한 파형 특정 불가



05 회로 하드웨어 구성

DTS (Drone Tracking System)



06

프로젝트 개요 및 결과

- 프로젝트 개요



개연영상



구행영상



07

고찰 및 후후 개선 과정

- 고찰 및 후후 개선 과정
- 느낀점

고찰 및 후후 개선 과정

- 1) 방위각 기반 알람 기능을 추가하여 실시간 알람 전송 로직을 구축한 상태이고. 경찰서 신고 통신 채널이 후후 개설되면, 해당 채널을 통해 신고를 진행할 예정
- 2) 트래킹 및 인식 프로세스를 멀티 스레딩을 사용해 징업을 병렬로 처리해 CPU의 짜원 활용도를 높여 CPU 사용량을 줄일 예정
- 3) 성능이 좋은 마이크 모듈로 업그레이드하여 학습된 딥러닝 모델을 적용. 드론 모리를 보다 정확히 탐지하여 드론의 위치를 정확하게 추적하는 기능을 추가할 예정
- 4) TV, 형광등, 책상 같은 배경 속에서 객체를 더 잘 추적할 수 있도록 영상 알고리즘을 개선할 예정
- 5) 3D 프린터나 철을 활용해 하드웨어 성능을 업그레이드



코드가 예상대로 작동하지 않고, 작은 변수에도 민감하게 반응하는 것이 답답했다.
한 줄씩 코드를 수정하는 작업이 힘들지만 보람찼다.
또한 팀원들과의 분업 덕분에 프로젝트를 성공적으로 마무리할 수 있어 감사하다.



OTX를 활용해 모델을 Train 하는 과정이 꽤 오래 걸려서 힘들었지만, 오류들을 해결해 나가면서 보람과 즐거움을 느꼈다.
내 자신에게 많은 부족함을 느꼈고, 보다 더 많이 공부하고 노력해야겠다.

OpenVINO 모델을 제한된 환경에서 구동을 하는데 어려움을 겪었고,
모델 효율화, 원격 통신 등을 통해 문제를 해결
할 수 있어서 좋았다.



라끄베리 환경 설정하는 게 생각보다 시간이 오래 걸렸고 프로젝트 시간 내에 마이크 모듈이 제대로 동작하지 않아 아쉬움이 남지만 팀원들과 협력하여 프로젝트를 잘 마무리한 거 같다.



감사합니다