

# 지능형 무인 경계 로봇 구현



팀명	T1_민영대
지도교수	탁성우
팀원	김대영 박민재 조영진

---

# 목차

1. 요구조건 및 제약 사항 분석에 대한 수정사항 .....	3
1.1 요구조건 .....	3
1.2 제약 사항 분석에 대한 수정사항 .....	3
2. 설계 상세화 및 변경 내역 .....	4
2.1 설계 상세화.....	4
2.1.1 초기 단계에서 제안된 시스템 구조 .....	4
2.1.2 현재 프로젝트의 시스템 구조 .....	4
2.2 변경 내역.....	5
3. 보고 시점까지의 과제 수행 내용 및 중간 결과.....	6
3.1 중간 결과 구조도 .....	6
3.2 테스트 결과.....	7
4. 구성원별 진척도 .....	9
5. 갱신된 과제 추진 계획 .....	10

---

## 1. 요구조건 및 제약 사항 분석에 대한 수정사항

### 1.1 요구조건

- 물체 인식 기술을 사용하여 객체를 추적하고, 해당 객체가 사유지를 침범한 것을 판별한다.
  - YOLO 모델을 사용하여 객체를 감지하고, 감지된 객체를 DeepOCSORT 모델을 사용해 추적한다.
- 객체가 사유지를 침범했는지 판별한다.
  - OpenCV 라이브러리를 사용해 화면에 가상의 선을 그리고, 추적하는 객체와 가상의 선의 좌표를 비교하여 객체가 사유지를 침범했는지 판별한다.
- 사유지를 침범한 객체를 내쫓을 수 있는 기능을 제공한다.
  - 객체가 사유지를 침범했을때, 실시간 영상 데이터를 사용자 핸드폰에 전송

### 1.2 제약 사항 분석에 대한 수정사항

- 라즈베리 파이 기반 개발
  - 라즈베리 파이의 카메라 모듈의 주문이 누락되는 등 배송이 늦어져, 윈도우 환경에서 개발 및 테스트 진행
  - 이후 라즈베리 파이 부품들이 도착한다면, 라즈베리 파이 환경에서 테스트 진행 예정
- 카메라 영상 화면에서 사유지 경계 판단
  - 카메라가 이동하거나 회전할때, 화면 상에서 사유지의 경계를 판별할 수 없음
  - 따라서 고정식 카메라를 가정하고, 고정된 화면에서 OpenCV 라이브러리 화면을 사용해 경계선을 그리고 판단함.

---

## 2. 설계 상세화 및 변경 내역

### 2.1 설계 상세화

#### 2.1.1 초기 단계에서 제안된 시스템 구조

- 머신러닝 모델 학습 구조
  - 다양한 환경에서 수집한 데이터를 기반으로 CNN, MobileNet, Tiny YOLO 등의 경량화된 알고리즘을 사용하여 물체 인식 모델을 설계하고, 이를 라즈베리 파이에 탑재하여 실시간 영상 데이터를 처리하는 구조.
- 서버 및 Application 구조
  - Spring Framework 기반의 서버와 Flutter로 제작된 모바일 애플리케이션을 사용하여, 실시간 경고 메시지 및 알림 시스템을 구축하는 구조.

#### 2.1.2 현재 프로젝트의 시스템 구조

##### a. 라즈베리 파이와 머신러닝 모델 통합

- YOLO 모델 기반 객체 감지
  - 라즈베리 파이에 YOLO v8n 모델을 탑재하여, 실시간으로 웹캠에서 수신한 영상에서 객체를 감지함
- DeepOCSORT를 활용한 객체 추적
  - 감지된 객체는 DeepOCSORT 알고리즘을 통해 지속적으로 추적되어, 객체의 위치 및 움직임을 추적함
- OpenCV로 가상 경계선 설정
  - 감지된 객체가 지정된 가상의 경계선을 넘을 경우를 감지하기 위해, OpenCV를 사용하여 실시간 영상에 경계선을 그림
- 라인 크로싱 감지
  - 객체가 설정된 경계선을 넘었을 때, 이를 감지하고 비디오 녹화가 시작되며, 서버와의 연결이 시도됨

---

## b. 서버 및 Application 구조

- **WebSocket 기반 서버 통신**

- 경계선 크로싱이 감지되면, 라즈베리 파이는 WebSocket 서버에 연결을 요청
- 서버는 요청을 승인하면 h264 비디오 스트림을 수신하여 모바일 애플리케이션으로 전송

- **모바일 애플리케이션 통합**

- Flutter로 제작된 애플리케이션은 WebSocket 서버와 연결되어 h264 비디오 스트림을 수신하고, 이를 실시간으로 화면에 출력
- 현재는 텍스트 형식으로 출력되지만, 향후 실제 영상 스트림을 재생할 수 있도록 개선할 예정입니다.

## 2.2 변경 내역

- **YOLO v8n 모델 채택**

- 다양한 모델 중 경량화와 정확도를 고려하여 YOLO v8n 모델을 사용하기로 결정하였습니다.

- **DeepOCSORT로의 객체 추적**

- 초기에는 단순 객체 감지에 중점을 두었으나, 더 높은 추적 정확도를 제공하는 DeepOCSORT 알고리즘으로 객체 추적 방식을 변경했습니다.

- **WebSocket 통신 프로토콜 채택**

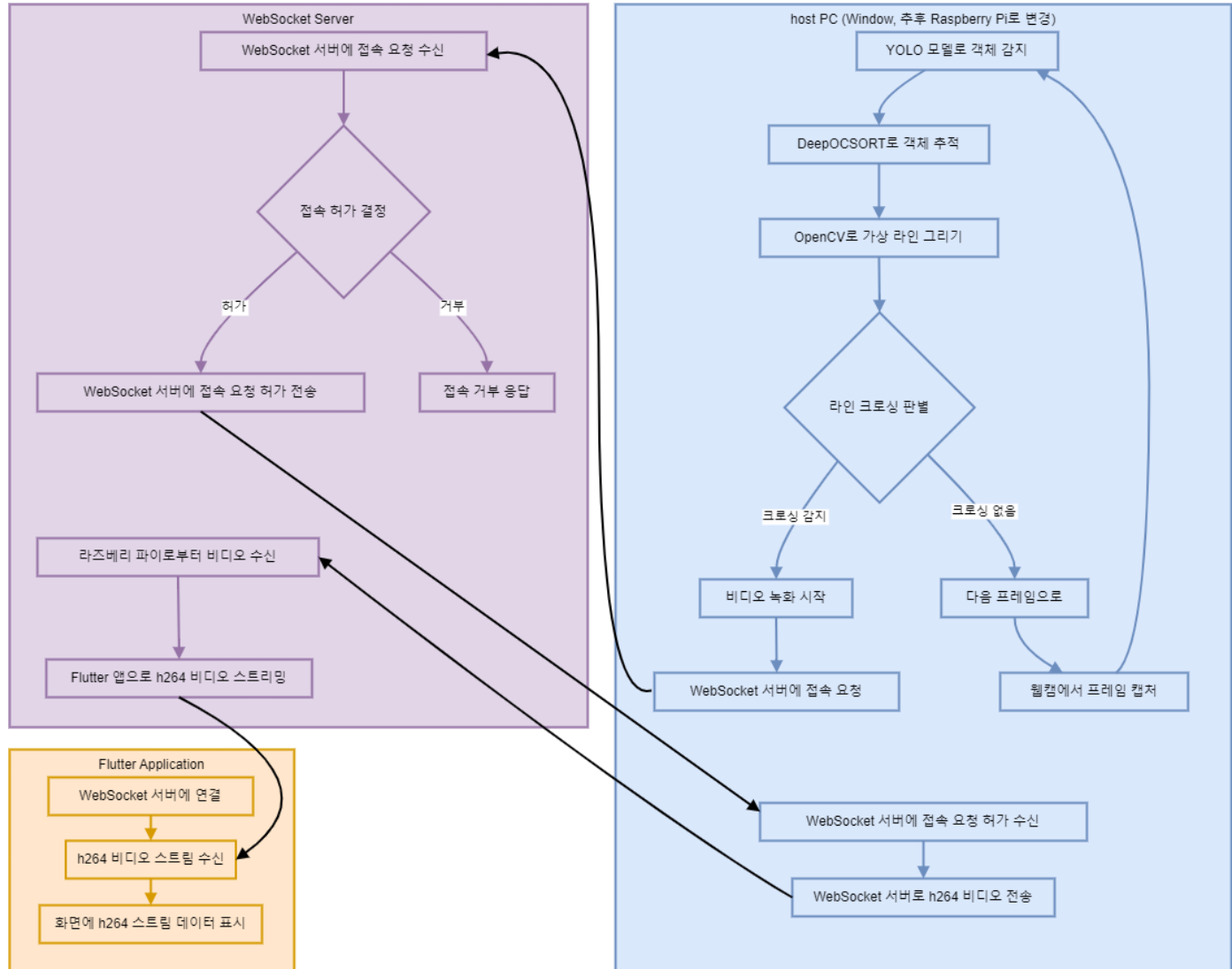
- 초기 설계에서는 HTTP를 통한 통신이 고려되었으나, 실시간 데이터 전송의 효율성을 높이기 위해 WebSocket 프로토콜로 변경되었습니다.

- **Flutter 애플리케이션으로의 데이터 전송**

- 초기에는 경고 메시지를 중심으로 한 간단한 알림 시스템이 계획되었으나, 실제 h264 비디오 스트림을 전송하고 화면에 출력할 수 있도록 설계가 확장되었습니다.

### 3. 보고 시점까지의 과제 수행 내용 및 중간 결과

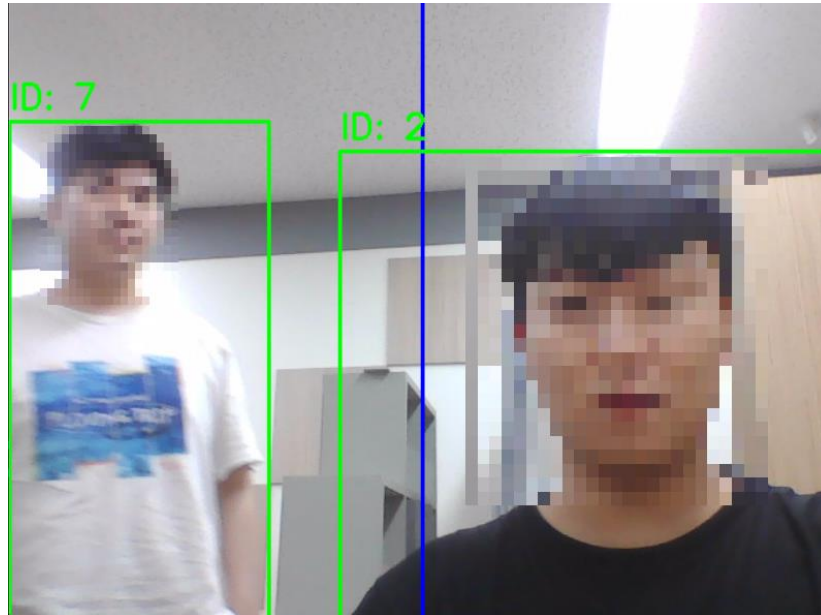
#### 3.1 중간 결과 구조도



---

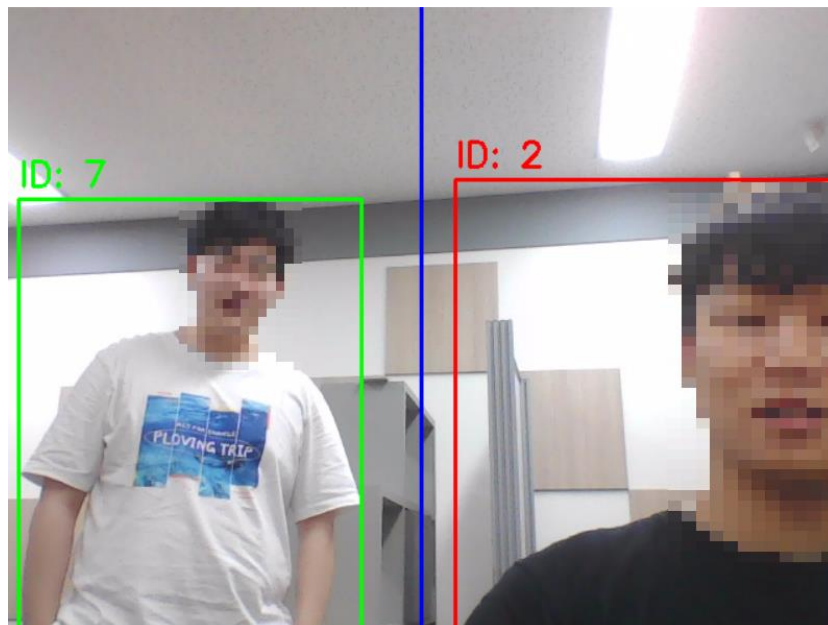
## 3.2 테스트 결과

### 3.2.1 웹캠 화면 1: 탐지된 객체가 경계선을 넘지 않았을 때:



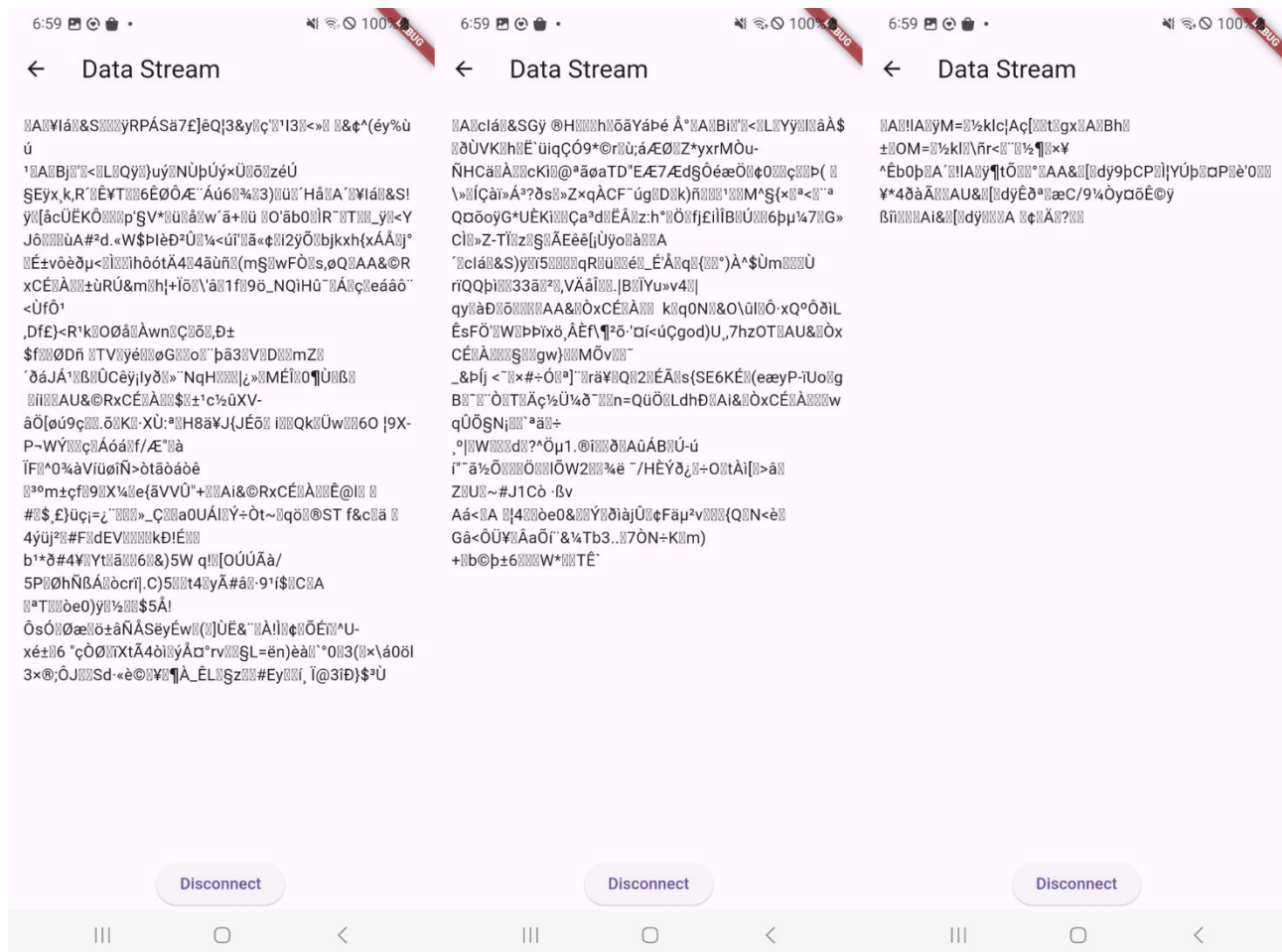
→ 해당 사진에서 보이는 것처럼, 객체가 녹색 테두리로 표시됨

### 3.2.2 웹캠 화면 2: 탐지된 객체가 경계선을 넘었을 때:



→ 해당 사진에서 보이는 것처럼, 객체가 빨간색 테두리로 표시되고, 웹소켓 서버에 실시간 영상 전송 시작

### 3.2.3 Flutter 앱 화면 :



→ WebSocket 서버로부터 받은 h264 데이터를, 텍스트 형식으로 화면에 출력

(서버로부터 받는 h264 데이터 파싱에 실패하여, 텍스트 형식으로 표시 중. 추후 실시간 영상으로 나오도록 수정)



---

#### 4. 구성원별 진척도

이름	역할
김대영	- 웹소켓 서버와 연결하여 실시간으로 h264 데이터를 수신하고, 이를 텍스트 형식으로 화면에 출력하는 Flutter 애플리케이션을 제작
박민재	- YOLO v8n 모델을 사용하여 실시간 영상에 물체 인식 기능을 구현 - 웹캠을 통해 수집된 실시간 영상에 가상의 경계선을 설정하고, 인식된 객체가 해당 경계선을 넘을 경우 웹소켓 서버로 데이터를 전송하는 프로그램을 작성
조영진	- Spring Framework를 사용하여 웹소켓 서버를 제작 - 파이썬 프로그램에서 접속 요청을 승인하고, 실시간으로 h264 영상 스트림을 수신하여 Flutter 애플리케이션으로 전송하는 기능 구현

## 5. 갱신된 과제 추진 계획

5월		6월				7월				8월				9월				10월
3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1
착수 보고서 작성, 지도확인서 제출																		
		사용 기술 조사																
					모델, 서버, 앱 개발													
									시스템 테스트 및 중간보고서 작성									
												모델 정확도 테스트						
													라즈베리파이 기반 로봇 제작					
														라즈베리파이 기반 시스템 테스트				
																	최종 보고서 작성 및 발표 준비	