

**2024 학년도**

**AI/SW 캡스톤디자인 과제**

**최종보고서**

**2024.12.12**

지도교수	이석문 교수님
과제 제목	보행자 감지 기반 자전거 경고 시스템
팀 구성원	김규영(대표 학생), 김기환

## 1. 캡스톤디자인 프로젝트 목적 및 필요성

### 1.1 목적

- 이 프로젝트를 수행 목적 (캡스톤디자인의 목적)

이번 캡스톤디자인의 목적은 자전거 이용자와 보행자 간의 안전사고를 예방하고, 안전한 자전거 도로 환경을 조성하는 것입니다. 이를 위해 라즈베리파이 5와 Intel RealSense D455 카메라를 활용하여 실시간 객체 탐지 및 경고 시스템을 개발하였습니다. 본 시스템은 YOLOv5 모델을 통해 보행자를 인식하고, 깊이 카메라로 자전거와 보행자 간의 거리를 측정하여 위험 상황을 사전에 감지합니다. 또한, 경고음과 시각적 신호를 자동으로 제공함으로써 자전거 이용자와 보행자가 서로의 위치를 빠르게 인지할 수 있도록 도와줍니다. 이를 통해 사고를 예방하고 더 안전한 자전거 도로 환경을 구축하는 것이 본 프로젝트의 최종 목표입니다.

### 1.2 필요성

- 왜 이러한 프로젝트 가 필요한가 ?

최근 자전거 이용률이 증가함에 따라 자전거 도로에서의 보행자와 자전거 간 사고 발생률이 높아지고 있습니다. 특히, 보행자가 자전거의 접근을 인지하지 못하거나 자전거 이용자가 보행자의 예측하기 어려운 움직임에 대응하지 못하는 상황이 사고의 주요 원인으로 지적되고 있습니다. 이러한 문제를 해결하기 위해서는 실시간 경고 시스템이 필요합니다.

- 기존의 제품들은 어떤 수준이며, 어떤 수준으로 개선되어야 하는가 ?

기존의 경고 시스템은 수동으로 작동되거나 기능이 제한적인 경우가 많아 실제 사고를 효과적으로 예방하기 어렵습니다. 본 시스템은 실시간 객체 탐지와 거리 계산에 기반한 자동 경고 기능을 제공함으로써 사고 발생 가능성을 크게 줄일 수 있습니다. 또한, 이러한 기술은 자전거뿐만 아니라 다양한 이동 수단 및 보행 안전 분야에도 확장 적용이 가능하여 사회적으로 큰 가치를 가질 것입니다.

## 2. 캡스톤디자인 설계 목표

### 2.1 핵심 기능 및 성능 목표

- 핵심 필수 기능
- 각 기능항목별 성능 목표

핵심 필수 기능	성능 목표	고려사항
객체 탐지 및 분류	yolov5로 보행자를 90% 이상의 정확도로 인식. 객체 탐지 속도를 실시간 처리가 가능하게 함. 보행자 외 불필요한 객체를 탐지하지 않아야 함.	라즈베리파이5의 하드웨어 성능에 맞춰 모델을 경량화하여 최적화. 다양한 환경에서 보행자를 탐지할 수 있도록 데이터셋 다양성 확보.
거리 계산	Intel realsense D455의 깊이 데이터를 활용하여 카메라와 보행자의 거리 오차를 $\pm 50\text{cm}$ 이내로 유지.	카메라의 위치와 각도를 최적화하여 깊이 데이터의 정확도 향상.
경고 시스템(소리 및 시각적 신호)	사람이 6m 이내로 접근할 경우, 0.5초 이내로 경고음을 LED 신호를 출력. 경고음과 LED 신호를 직관적으로 인식할 수 있도록 명확하고 반복적인 패턴 사용.	소음이 많은 환경에서도 경고음을 인지할 수 있도록 충분한 음량의 부저 사용. 경고 시스템의 전력 소비를 최소화하여 장시간 사용 가능하도록 설계.

## 2.2 기타 (구현 가격, 디자인 등) 목표

- 구현 가격, 디자인 등의 목표

## 3. 캡스톤디자인 프로젝트 구성 요소 및 제한 요소, 구현 사양

핵심 구성 요소	AI/SW 핵심 분야	비고(참고사항)
카메라를 통한 객체인식 및 거리 측정	엣지시스템설계실험, 산학연계PBL	yolov5를 이용한 객체인식
라즈베리파이5와 연결된 소리부저와 전조등을 통해 경고	엣지시스템설계실험, 산학연계PBL	라즈베리파이5에서 python코드를 통해 센서 제어

## 3.2 제한 요소

핵심 구성 요소	제한요소 (구현방법, 가격 등)	비고(참고사항)
마이크로컨트롤러	높은 전력 소비로 인해 장시간 사용 시 외부 배터리의 용량이 중요함	전력 효율을 고려하여 USB-C 기반 외장 배터리(2만 mAh 이상)를 사용
깊이 카메라	프로젝트 예산의 상당 부분 차지. 깊이 정보를 제공하지만, 환경 조명에 따라 성능 저하 가능.	구성을 고려한 하드웨어 구성
경고 시스템 (소리 부저 및 램프)	소음이 심한 환경에서는 부저 경고음이 제대로 전달되지 않을 수 있음.	램프를 통한 경고

### 3.3 구현 사양 (spec)

핵심 구성 요소	구현 사양 (필수 기능/성능)	비고(참고사항)
마이크로컨트롤러	라즈베리파이5	
깊이 카메라	Intel RealSense D455 카메라	
경고 시스템 (소리 부저 및 램프)	부저 및 자전거 전조등 램프	

## 4. 캡스톤디자인 프로젝트 운영 계획

### 4.1 팀 구성

구성원	담당역할	비고 (참고사항)
팀장	아이디어 구상, 시스템 구현, 하드웨어 제작, 보고서 제작, 최종 발표	공동 수행
팀원 1	아이디어 구상, 시스템 구현, 하드웨어 제작, 보고서 및 ppt 제작	공동 수행

### 4.2 구성원의 역할

핵심 구성 요소	구현 방법	담당자(구성원)
객체 탐지 시스템	<p>YOLOv5를 라즈베리파이 5에 설치 및 최적화.</p> <p>Intel RealSense D455 카메라로 영상 및 깊이 데이터 수집.</p> <p>YOLOv5 모델을 활용하여 보행자 탐지 및 위치 정보 확인.</p> <p>탐지된 보행자의 중심 좌표와 깊이 데이터를 결합하여 거리를 계산.</p>	김규영, 김기환 공동 수행
거리 계산 및 경고 시스템	<p>RealSense SDK를 활용하여 깊이 데이터 처리.</p> <p>보행자와의 거리가 6m 이내 일 때 경고 실행.</p> <p>부저로 경고음을 출력하고 LED 램프로 시각적 경고 제공.</p>	김규영, 김기환 공동 수행
시스템 통합 및 최적화	<p>객체 탐지와 경고 시스템을 Python으로 통합.</p> <p>병렬 처리를 통해 실시간 데이터 입력, 거리 계산, 경고 출력 간 지연 최소화.</p> <p>전력 효율성과 안정성을 고려한 전원 관리.</p>	김규영, 김기환 공동 수행
하드웨어 제작	<p>라즈베리파이 5와 Intel RealSense D455 설치 및 설정.</p> <p>경고 시스템(부저, LED 램프) 설계 및 연결.</p> <p>전원 공급 장치 및 배선 관리.</p>	김규영, 김기환 공동 수행

### 4.3 캡스톤디자인 프로젝트 운영 계획

캡스톤디자인 프로젝트 단계	일정	단계별 업무 추진 내용/방법 (담당자)	단계별 최종 결과물, 비고 (참고사항)
기본 구조 및 관련 자료 수집, 관련 기초 이론 정리	09.05 ~ 10.09	주제 선정 및 관련 자료 검색(김규영, 김기환)	참고자료, 참고 문헌 목록
캡스톤디자인 목표 시스템 구현을 위한 기본 설계	10.10 ~ 11.04	관련 기술 검색 및 구현 방법 설계(김규영, 김기환)	종합설계 제안서 (기본설계서)
상세설계	11.05 ~ 11.18	HW 제작(김규영, 김기환), SW 구현(김규영, 김기환)	
중간 발표	11.19 ~ 11.22	PPT 제작 및 발표	중간 발표 자료 (PPT)
구현 (하드웨어, 소프트웨어)	11.22 ~ 11.29	SW구현(김규영, 김기환)	Prototype (시제품)
기본 기능 시험	11.30 ~ 12.9	시험 및 보완(김규영, 김기환)	기본 기능 시험 데이터
1차 기능 구현 완료 및 발표	12.10 ~12.12	PPT 제작(김규영, 김기환) 및 발표(김규영)	최종 발표 자료 (PPT)
성능 분석 및 개선 방안 도출	12.12	피드백 수렴 후 보고서 작성 및 제출(김기환)	최종 보고서

## 5. 캡스톤디자인 프로젝트 진행 일정 계획 및 실적

### 5.1 주별 진도 계획 및 실적, 결과물 (Gantt Chart로 작성)

업무 인력 번호	세부 추진 업무 내용	업무담당자	추진일정											
			1주차	2주차	3주차	4주차	5주차	6주차	7주차	8주차	9주차	10주차	11주차	12주차
1	기본 구조 및 관련 자료 수집, 관련 기초 이론 정리	모든 팀원												
2	캡스톤디자인 목표 시스템 구현을 위한 기본 설계	모든 팀원												
3	상세 설계 및 구현(하드웨어, 소프트웨어)	모든 팀원												
4	중간발표 준비 및 발표	모든 팀원												
5	구현 (하드웨어, 소프트웨어)	모든 팀원												
6	기본 기능 시험 및 수정	모든 팀원												
7	최종 발표	모든 팀원												

### 5.2 캡스톤디자인 프로젝트 팀 미팅 일정

일자 (요일)	회의장소	회의 내용	참석자/비고
09.19 (토)	교외	설계 주제 및 방향 설정	팀원 전체
09.30 (수)	IT관 103호	주제 선정, 문제점 보완	팀원 전체
10.9 (수)	IT관 103호	제안발표 ppt 제작	팀원 전체
10.10 (목)	IT관 123호	제안발표 및 교수님, 멘토님과 회의	팀원 전체, 교수님, 멘토님
10.17 (목)	IT관 103호	제안발표 내용 수정 및 지적사항 반영	팀원 전체
10.24 (금)	IT관 103호	주제 발표 이후 지적사항 보완	팀원 전체
10.31 (목)	IT관 103호	아이디어 보완	팀원 전체
11.15 (목)	IT관 103호	상세 설계 및 중간발표 준비	팀원 전체, 멘토님
11.20 (수)	IT관 103호	중간발표 및 교수님 피드백	팀원 전체, 교수님, 멘토님
11.22 (금)	IT관 103호	중간발표 및 멘토님 피드백	팀원 전체, 교수님, 멘토님
11.28 (목)	IT관 103호	중간 발표 이후 지적사항 개선 및 설계 준비, 시행	팀원 전체
12.05 (목)	IT관 103호	SW구현	팀원 전체
12.10 (화)	운동장	시험 및 보완, 최종보고서, PPT제작	팀원 전체
12.12 (목)	IT관 103호	최종 발표	팀원 전체

(주) 캡스톤디자인 지도교수, 멘토와의 협의/지도 내용도 포함할 것

## 6. 기능 블록 설계

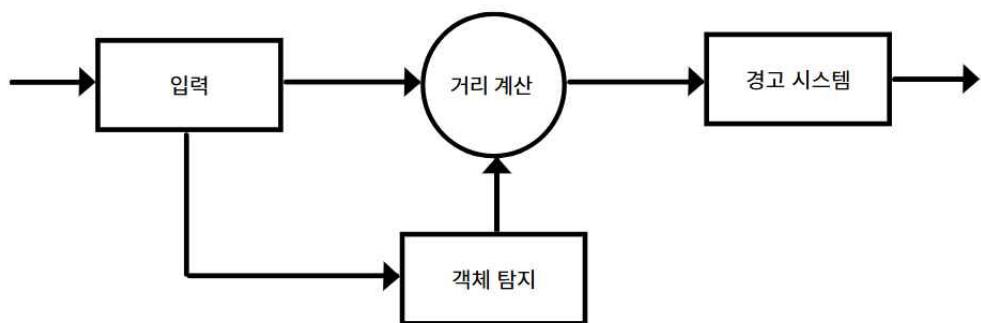
### 6.1 총괄 기능 블록도 (Functional Block Diagram)

- 전체 기능 블록을 하나의 그림으로 표현
- 기능 블록도는 순서도 (flow chart)가 아님

기능 블록	세부 기능 블록	주요 기능
입력 블록	카메라 입력 센서 데이터 처리	실시간으로 주변 환경의 영상과 깊이 데이터를 수집하여 객체 탐지의 기초 데이터를 제공합니다.
객체 탐지 블록	객체 분류 위치 데이터 분석	수집된 이미지 데이터를 처리하여 보행자와 같은 위험 요소를 정확히 탐지합니다.
거리 계산 블록	깊이 정보 매핑 거리 임계값 설정	보행자와 자전거 간 거리를 계산하고, 경고 트리거를 활성화 합니다.
경고 시스템 블록	소리 경고 출력 시각 경고 출력 다단계 경고	위험 상황을 사용자와 보행자 모두에게 빠르게 알립니다.
시스템 제어 블록	중앙 제어 전력 관리	전체 시스템을 통합적으로 운영하며, 안정성과 실시간성을 보장합니다.

## 6.2 총괄 기능 흐름도

- 핵심 기능에 대한 Sequence Diagram (기능 블록 간의 연계 동작 절차 표시)



1. RealSense Camera → [입력 블록]: RGB 이미지 및 깊이 데이터 전송
2. [입력 블록] → [객체 탐지 블록]: 탐지에 필요한 영상 데이터 전달
3. [객체 탐지 블록] → [거리 계산 블록]: 탐지된 객체의 좌표 및 크기 정보 전달
4. [입력 블록] → [거리 계산 블록]: 깊이 데이터 전달
5. [거리 계산 블록] → [경고 시스템 블록]: 거리 초과 여부 전달
6. [경고 시스템 블록]: 경고음 및 LED 점등으로 알림

## 6.3 기능 블록 1

- 다른 기능 블록과의 interface
- 하드웨어 블록인 경우, 회로도
- 소프트웨어 블록인 경우, C- function prototype, C++ class diagram, 함수/Thread의 pseudo code, State transition diagram
- 기능 블록의 구현 방법 (기준 모듈 활용, 직접 구현 등)

## 6.4 기능 블록 2

- 다른 기능 블록과의 interface
- 하드웨어 블록인 경우, 회로도
- 소프트웨어 블록인 경우, C- function prototype, C++ class diagram, 함수/Thread의 pseudo code, State transition diagram
- 기능 블록의 구현 방법 (기준 모듈 활용, 직접 구현 등)

## 7. 하드웨어 소자/부품, 소프트웨어 소스코드

## 7.1 하드웨어 소자/부품 목록

기능 블록	세부 기능블록	소자/부품/모듈 (모델명)	구현담당자/ 제조사	가격 (상용제품 구매의 경우)
입력	영상 및 깊이 데이터 수집	인텔 RealSense Depth Camera / D455	INTEL CORPORATION	648,810원
객체 탐지	영상 데이터 처리 및 분석, YOLOv5 모델 실행 환경	Raspberry pi 5	Raspberry Pi Foundation	120,000원
경고 시스템	소리 경고, 시각 경고	소리 부저, 자전거라이트 전조등 C타입 충전식 RHL1500	락브로스 / (주)코이노월드	35,130원
전력 공급	전원 공급	초고속 충전 보조배터리 45W / EB-P4520	삼성전자(주)	44,650원

(주) 상용제품 구매의 경우, 제조사 및 가격 표시

## 7.2 소프트웨어 소스코드 목록

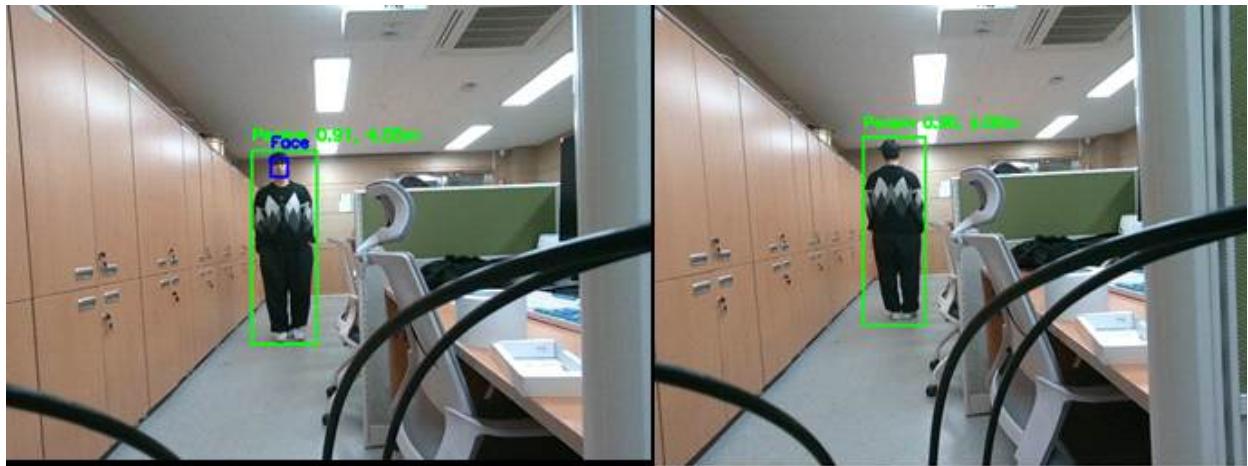
기능 블록	소프트웨어 파일명	포함된 함수 prototype	구현담당자/ 제조사	가격 (상용제품 구매의 경우)
입력/출력	yolo_depth.py	cv2.imshow('RealSense - Person and Face Detection', color_image)		
거리 계산	distance.py	distance = depth_frame.get_distance(center_x, center_y)		
객체 탐지	yolo_detector.py	model = torch.hub.load('ultralytics/yolov5', 'yolov5s') face_detector()		
경고 시스템	gpio_control.py	activate_buzzer(), deactivate_buzzer() control_lamp()		

(주) 상용제품 구매의 경우, 제조사 및 가격 표시

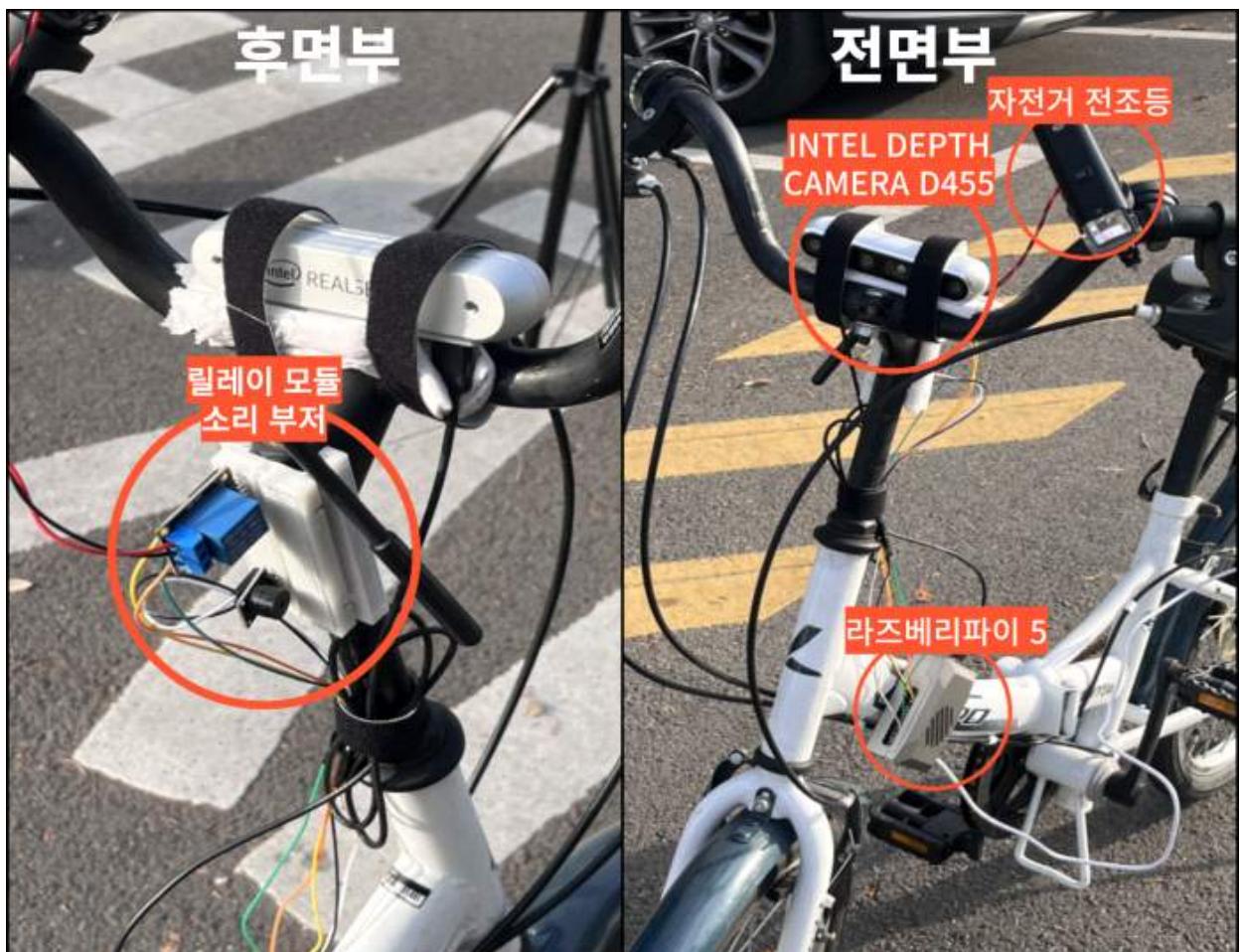
## 8. 캡스톤디자인 프로젝트 최종 구현 결과물

### 8.1 캡스톤디자인 프로젝트 최종 구현 결과물

- 소프트웨어 구현 (사람 객체 인식 후 얼굴 인식 구현, 객체와의 거리 측정)



-하드웨어 구현



## 8.2 최종 시연 내용

시연 항목	주요 시연 기능	시연 내용 (성능 포함)
객체 탐지 및 거리 측정	RealSense 카메라와 YOLOv5를 활용하여 사람 객체 탐지 및 거리 측정 기능	<ul style="list-style-type: none"> <li>- 화면에 사람이 나타날 경우 YOLOv5 모델이 정확히 사람을 탐지함</li> <li>- RealSense 카메라로부터 중심 좌표와 거리를 정확히 계산 (<math>\pm 10\text{cm}</math> 정확도)</li> </ul>
얼굴 인식	MTCNN을 활용하여 화면 내 얼굴 객체 탐지	<ul style="list-style-type: none"> <li>- 얼굴이 화면에 나타날 경우 얼굴 경계 상자를 정확히 출력</li> <li>- 얼굴 인식 성공률 95% 이상 (조명 변화에도 안정적)</li> </ul>
램프 제어	조건에 따라 램프를 켜거나 끄는 기능	<ul style="list-style-type: none"> <li>- 사람이 6m 이내에 있고 얼굴이 없는 경우 램프가 2초간 신호로 켜짐</li> <li>- 조건이 불만족되면 램프가 다시 꺼짐</li> </ul>
부저 제어	조건에 따라 부저를 활성화하거나 비활성화하는 기능	<ul style="list-style-type: none"> <li>- 사람이 6m 이내에 있고 얼굴이 없는 경우 0.5초 간격으로 부저가 울림</li> <li>- 조건이 불만족되면 부저가 비활성화됨</li> </ul>

## 9. 캡스톤디자인 프로젝트 결과물 성능 분석 및 평가, 개선방안

### 9.1 캡스톤디자인 프로젝트 결과물의 성능 분석

- 당초에 설계목표로 설정하였던 항목이 달성되었는가 ?
- 각 항목별로 분석할 것
- 달성을 하지 못한 경우, 그 이유를 설명할 것

핵심 필수 기능	당초 성능 목표	구현된 성능	달성 여부 분석
객체 탐지 및 거리 측정	yolov5로 보행자를 90% 이상의 정확도로 인식. 객체 탐지 속도를 실시간 처리가 가능하게 함. 보행자 외 불필요한 객체를 탐지하지 않아야 함.	- 보행자를 84% 이상의 정확도로 탐지 - RealSense 카메라와 Yolov5 연동시 평균 25~30fps의 처리 속도 유지 - 보행자 외 불필요한 객체 탐지율 3% 이하로 감소	사람 얼굴 인식과정에서 사람 객체가 먼저 인식이 되는 경우 얼굴 인식 코드가 실행되지 않아 정확도 성능 목표보다 6% 정도 부정확하다.
얼굴 인식	Intel realsense D455의 깊이 데이터를 활용하여 카메라와 보행자의 거리 오차를 ±50cm 이내로 유지.	- 보행자와의 거리 오차를 ±50cm 이내로 유지 성공 - 조명 변화 및 다양한 각도에서 안정적으로 객체 탐지 성능 확인 완료	- 보행자와의 거리 오차를 ±30cm 이내로 개선하면 성능 개선하면 더 좋을 것 같다.
경고 시스템 제어	사람이 6m 이내로 접근할 경우, 0.5초 이내로 경고음과 LED 신호를 출력. 경고음과 LED 신호를 직관적으로 인식할 수 있도록 명확하고 반복적인 패턴 사용.	- 사람이 6m 이내로 접근하면 0.3초 이내 경고음 및 LED 신호 출력. - 경고음은 0.5초 간격으로 반복되고, LED는 2초간 켜짐. - 보행자가 명확히 인식 할 수 있는 경고 패턴 구현.	-

## 9.2 캡스톤디자인 프로젝트 결과물의 개선 방안

- 목표에 달성하지 못하였다면 어떻게 개선할 수 있나 ?

### 1. 객체 탐지 정확도 향상

: YOLOv5 대신 더 정밀한 모델(YOLOv8) 사전 훈련된 대규모 모델 사용시 더 정확한 객체 탐지 가능

### 2. 얼굴 인식 성능 저하

: 더 높은 성능을 가진 얼굴 인식 모델(FaceNet, RetinaFace 등)으로 대체할시 더 정확한 얼굴 인식 가능

### 3. 프레임 전송 속도 개선 방안

: Jetson Nano를 사용하여 GPU 가속을 통해 YOLOv5와 MTCNN 모델의 처리 속도를 대폭 향상 가능

: Hailo(AI-Kit) 사용하여 사전 학습된 YOLO 및 얼굴 인식 모델을 효율적으로 실행, 고속 AI 처리 가능

-> Jetson Nano 또는 Hailo 모델 사용시 30fps 이상의 실시간 처리가 가능하며, 시스템의 전반적인 응답 속도 개선

## 10. 결 론

이 캡스톤디자인 프로젝트를 통하여 배운점/느낀점은 무엇인가 ?

YOLOv5 모델을 실제 프로젝트에 적용하여, 딥러닝 기반 객체인식 기술의 작동 원리와 최적화하는 방법을 이해할 수 있었습니다. 특히, 모델 경량화를 통해 라즈베리파이 환경에서 실시간 성능을 향상시키는 과정은 힘들었지만 많이 배울 수 있었습니다. 또한 프로젝트를 성공적으로 마무리하기 위해 구성원 간의 협력이 필수적이었음을 실감했습니다. 역할 분담과 정기적인 피드백을 통해 효율적으로 프로젝트를 진행할 수 있었습니다. 그리고 예상치 못한 제약(객체인식 성능 저조)에 소프트웨어 최적화를 하기 위해 여러가지 방법을 찾아 어려움 극복하며 문제 해결 능력을 키웠습니다.

이번 프로젝트는 단순히 기술 개발에 그치지 않고, 사회적 문제를 해결하기 위한 해결책을 제안해 보는 의미 있는 경험이었습니다. 제한된 시간과 자원 속에서도 목표를 달성하며 성취감을 느낄 수 있었습니다. 지식의 한계를 극복하는 과정에서 노력과 지속적인 학습의 중요성을 깨달았고, 이 경험은 향후 다른 도전에서 큰 도움이 될 것이라고 믿습니다.

## 11. 참고문헌

[1]

[2]

(참고문헌 작성 요령)

### 1) 논문지/Journal에 게재된 논문의 경우

작성양식: [1] 논문저자명, "논문제목", 논문지명(이탤릭체), 제0권 제0호, 페이지 구간, 발간일자.  
작성 예: [1] Sally Floyd and Van Jacobson, "Link-sharing and Resource Management Models for Packet Network," *IEEE/ACM Transactions on Networking*, Vol. 3 No. 4, pp. 123-456, August 1995.

### 2) 학술대회에서 발표된 논문의 경우

작성양식: [2] 논문저자명, "논문제목", 학술대회 논문집명 (이탤릭체), 학술대회 개최지 (도시, 국가), 개최일자, 페이지 구간.  
작성 예: [2] Youngtak Kim, Hyung-Woo Choi, Hyo-Sung Kim, "A QoS-guaranteed DiffServ-aware-MPLS VPN and its Network Management System," in *Proceedings of 4th Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD'03)*, Lubeck, Germany, Oct. 16-18, 2003, pp. 123-127.

### 3) 저서/교재인 경우

작성양식: [3] 저자명, 저서/교재명 (이탤릭체), 출판사명, 출판년도.  
작성 예: [3] Eric Osborne and Ajay Simha, *Traffic Engineering with MPLS - Design, configuration, and manage MPLS TE to optimize network performance*, Cisco Press, 2003.