

# 프로젝트 계획서

## ◎ 작성 시 유의사항

- ▶ 내용작성과 관련한 설명내용 “작성요령(제출 시 삭제)”은 제거하고 내용 기술
- ▶ 폰트 및 글자 크기에 대한 제한은 없으며, 가독성 등을 고려하여 자유롭게 설정
- ▶ 주의: 여백, 정렬, 폰트 및 글자 크기, 표, 그림 등 전체적으로 통일성 맞출 것
- ▶ 주의: 비전공자가 본 프로젝트 계획서를 본다는 생각으로 작성할 것
- ▶ 주의: 그림을 많이 활용할 것
- ▶ 주의: 인터넷에서 다양한 프로젝트 계획서 (특히, 내용) 참고해서 작성할 것

프로젝트명	원격 AI 기반 시각장애인 보행 보조 기술		
작성일	2025년 03월 12일	버전	0.0.1
변경 이력			
팀장	AI 팀: 임석범, 시스템 팀: 차호현		
팀원	김병하, 이승재, 임석범, 차호현		

< 요약서 >

1. 프로젝트명	원격 AI 기반 시각장애인 보행 보조 기술		
2. 총수행기간	2025년 03월 12일 - 2025년 05월 31일	3. 총 투입인원	총 4명
4. 프로젝트 목표	<div>o 프로젝트 최종 목표</div> <div>- 내용: 원격 클라우드 환경에서 동작하는 AI와 어플리케이션을 통한 시각 장애인의 보행 보조 기술</div> <div>- 내용</div> <div>o 프로젝트 최종 산출물</div> <div>- SaaS클라우드 인프라 환경 구축</div> <div>- 시각 장애인 보행 보조 시각 AI 모델</div> <div>- 다중 사용자에게 AI 시각 보조 서비스를 위한 서비스 시스템</div> <div>- 사용자의 주행 보조를 위한 카메라 사용이 가능한 어플리케이션</div>		
5. 주요내용	<div>핵심 구성 요소:</div> <div><div>• 클라우드 서버 연동:<div><div>o Kubernetes 클러스터 (Master, Worker, GPU 노드)를 통해 AI 모델 추론을 수행</div><div>o MetalLB와 Nginx Ingress로 외부 트래픽을 처리하고, SSL/TLS 보안을 적용</div></div></div><div>• 다중 사용자 지원:<div><div>o REST API와 JWT 기반 회원 관리로 사용자를 인증 및 권한 부여</div><div>o WebSocket 기반 세션 관리와 HPA를 활용하여 실시간 AI 추론 부하를 분산</div></div></div><div>• 유연한 접근성:<div><div>o 스마트폰 앱에서 0.05초 간격으로 이미지를 촬영 및 암호화 전송</div></div></div></div>		

- 음성 명령과 진동 피드백 등 시각 장애인 친화적 UI/UX 제공

#### AI 관련 내용:

- 객체 탐지 단계: YOLOv11 모델을 통해 실시간으로 보행 객체를 탐지하고, 탐지 데이터를 시간 순서의 Queue에 저장
- 충돌 예측 단계: 저장된 시계열 데이터를 기반으로 2~3초 후 발생할 충돌 가능성을 예측하여 사용자에게 경고

### 6. 기대효과

#### 보행 안전성 향상:

시각장애인의 안전한 보행을 지원하여, 기존 보조기기에 대한 의존도를 줄이고 자율적 이동 능력을 강화합니다.

#### 사회적 포용 및 독립성 강화:

안정적인 이동 지원으로 시각장애인의 심리적 부담을 경감하고, 적극적인 사회 활동 참여를 유도합니다.

이는 교통약자, 고령자 등 취약계층 지원 기술 발전에도 기여합니다.

#### 확장성 및 실생활 응용:

스마트 글라스, AI 안내기, 진동 경고 장치 등 다양한 웨어러블 및 보조 장치에 적용 가능하며, 실내 자율주행 로봇, 스마트시티 인프라 등 다양한 분야로 확장이 기대됩니다.

#### 공공 안전 및 복지 기술 발전:

실생활 밀착형 AI 활용 사례로, 공공안전, 복지 기술, 사회적 약자 지원 분야에서 실용적 가치를 창출할 수 있습니다.

## 1. 프로젝트 수행의 필요성

### 1. 프로젝트 수행의 필요성

장애인들이 일상 생활에서 겪는 큰 어려움 중 하나로, 안전하고 독립적인 보행을 하는 것이다. 시각 장애인이 이용하는 이동 수단은 장애인 택시가 가장 많으며, 다음으로 도보로 이동하는 비율이 20.3%로 많은 비율을 차지한다. 세계 보건 기구의 발표에 따르면 시각 장애인이 도보로 이동함에 따라서, 매년 15%의 시각 장애인이 보행 중 사고로 인해 신체적 부상을 입는다. [2] 국내에서도 매년 시각 장애인 중 약 60%가 보행 시 환경적 장애물로 인해 불편함과 불안을 경험함에 따라 독립적인 이동의 가장 큰 장애 요인이 된다.

안전한 보행 환경을 저해하는 문제점과 환경적 요인은 다음과 같다.

첫째, 보행 중 예측 불가능한 장애물로 인한 신체적 손상 위험이 존재한다. 보도 위에 무단으로 주차된 차량, 불법 적치물, 공사로 인한 가설물 등이 보행자의 이동을 방해할 수 있으며, 시각장애인이거나 거동이 불편한 노약자의 경우 이러한 장애물을 피하기 어려워 낙상 등의 사고로 이어질 가능성이 높다. 또한, 노면이 고르지 않거나 보도블록이 파손된 경우, 보행자가 발을 헛디뎈 넘어질 위험이 커진다.

둘째, 보행 보조 도구의 한계로 인해 안전한 이동이 어려운 경우가 많다. 시각 장애인의 경우 점자 블록과 흰 지팡이, 안내견을 활용해 이동하지만, 점자 블록이 끊기거나 장애물로 가로막혀 있으면 정상적인 보행이 어렵다. 또한, 보행

보조 스틱은 장애물의 존재를 즉각적으로 파악하기 어렵고, 안내견 역시 모든 장애물을 완벽히 인지하고 피하는 데 한계가 있어 위험한 상황이 발생할 가능성이 있다

셋째, 인도와 차도의 경계 인식 부족으로 인해 보행자 안전이 위협받는다. 도로 설계가 명확하지 않거나, 인도와 차도의 구분이 불분명한 경우 보행자가 차도로 진입하는 위험한 상황이 발생할 수 있다. 특히, 시각장애인을 위한 점자 블록이 제대로 설치되지 않았거나 부적절하게 배치된 경우, 인도와 차도의 경계를 인식하는 데 어려움이 발생할 수 있다. 점자 블록은 시각장애인이 보행 중 장애물을 피하고 안전하게 길을 찾을 수 있도록 안내하는 역할을 한다. 그러나, 일부 지역에서는 점자 블록이 끊기거나 차량 주차·시설물 설치 등으로 인해 가려지는 경우가 많다. 또한, 점자 블록이 차도와 보도의 경계를 명확히 표시해야 함에도 불구하고 설치 기준이 제대로 지켜지지 않아 시각장애인이 차도로 잘못 진입하는 위험한 상황이 발생한다.

따라서, 본 프로젝트는 시각 AI 모델을 이용한 시각 장애인을 위한 보행 보조 장치를 제안한다.

## 2. 프로젝트의 목표 및 내용

### 2.1) 프로젝트의 최종 목표

본 프로젝트의 목표는 두 가지 큰 목표를 내포한다.

하나. 사물의 충돌 여부에 대해서 판독할 수 있는 인공지능 시스템을 구축한다.

둘. 다중 사용자에게 어플리케이션을 통한 AI 시각 보조 서비스를 제공할 수 있는 SaaS 시스템 (Software as a Service)를 구축한다.

이를 위해 스마트폰, SaaS 서비스를 위한 시스템 그리고 AI 시스템을 사용하여 실시간 객체 탐지-추적 및 위험 평가 기능을 제공한다.

## 2.2) 프로젝트의 내용

### 2.2.1) AI 관련

본 시스템의 AI 설계는 두 단계로 구성되어 있다.

첫 번째 단계는 영상 스트림으로부터 실시간으로 보행 객체를 탐지하고, 관련 정보를 분석하는 역할을 수행한다. 이를 위해 YOLOv11 기반의 객체 탐지 모델을 사용하여 이동체와 고정체의 위치, 크기, 세부 정보 등을 추출하며, 이 데이터들은 각각의 Queue에 저장된다. 각 Queue에는 가장 최근에 추출된 데이터가 우선적으로 저장되어, 시간 순서에 따른 데이터를 유지할 수 있도록 한다.

두 번째 단계에서는, 해당 Queue로부터 최근 데이터를 분류기에 입력하여, 약 2~3초 후 발생할 수 있는 잠재적인 충돌 상황을 예측한다. 이 과정을 통해 시스템은 보행자와의 충돌 위험을 사전에 감지하고, 적절한 대응이 가능하도록 한다.

#### - 객체 탐지

본 객체 탐지 시스템은 YOLO v11의 객체 인식 및 AI 모델을 활용하여, 실시간으로 객체를 탐지하고, 탐지된 데이터를 기반으로 향후 충돌 가능성을 정량적으로 평가할 수 있도록 전처리하는 과정을 포함한다.

본 시스템은 객체와의 충돌 가능성을 예측하는 것이 핵심 기능이므로, 객체를 정확하게 인식하는 능력이 시스템의 전반적인 성능과 밀접한 관련이 있다. 이에 따라, 객체 인식 성능을 극대화하기 위해 YOLO 모델을 대규모 데이터셋을 기반으로 장시간 학습시킬 계획이다.

#### - 학습 데이터 구성

YOLO 모델의 학습을 위해 AI-Hub에서 제공하는 인도 보행 영상 데이터셋을 활용하였다. 이 데이터셋은 다음 두 가지 주요 구성으로 이루어진다.

- Bounding Box 데이터셋 (250GB)

객체의 위치 정보를 나타내는 바운딩 박스 좌표를 포함하며, YOLO 학습의 기본 입력 형식이다.

- Surface Masking 데이터셋 (50GB)

객체의 형태를 더 정밀하게 인식하기 위한 세그먼트(마스킹) 좌표가 포함되어 있으며, 추가적인 위험물 판단을 위해 사용된다.

대분류	장예물			
소분류	이동체		고정체	
라벨명 (뜻)	bicycle(자전거)	movable signage (이동판)	Barricade (배리어게이트)	Pole (대/기둥)
	bus(버스)	person(사람)	bench(벤치)	potted_plant(화분)
	Car (승용차)	Scooter (두 발로 밀수 있는 기구)	bollard(볼라드)	power_controller (전력제어함)
	Carrier (운수레)	stroller(유모차)	chair(1인 의자)	Stop (버스/대시 정류장)
	Cat (고양이)	Wheelchair (휠체어)	fire_hydrant (소화전)	table (탁자)
	dog(개)		Kiosk (ATM기기 등 독립된 단말기)	traffic_light (신호등)
	Motorcycle (오토바이)		parking_meter (주차요금정산기)	traffic_light_controller (신호등제어기)
			traffic_sign (교통표지판)	tree_trunk (가로수 기둥)

[표1 : Bounding Box 데이터셋]

대분류	노면					
소분류	Alley (사행, 차 모두 가능한 길)	Bike_lane (자전거 도로)	Braille_guide_blocks (점자블록)	Caution_zone (주의구역)	Roadway (차도)	Sidewalk (인도)
라벨명 (뜻)	Crosswalk (횡단보도)	(막걸리 얼음)	damaged (파손)	grating (그레이팅)	Normal (일반)	asphalt (아스팔트)
	Damaged (파손)		Normal (일반)	Manhole (맨홀)	Normal (일반)	blocks (보도블럭)
	Normal (일반)			repair_zone (보수구역)		Cement (시멘트)
	speed_dump (과속방지턱)			Stairs (계단)		Damaged (파손)
				tree_zone (가로수영역)		Other (기타)
						Soil/Stone (흙/돌)
						urethane (우레탄)

[표2 : Surface Masking 데이터셋]

## - 데이터 전처리 및 학습 방식

AI-Hub에서 제공되는 원본 데이터를 YOLO 포맷에 맞게 변환한 후, 전체 데이터를 무작위로 **Train:Validation = 8:2**의 비율로 분할하여 학습에 활용하였다. 이후, 사전 학습(pre-trained)된 YOLO v11 모델을 기반으로 추가 학습을 진행하였으며, 주요 학습 조건은 다음과 같다:

- 학습 Epoch: 20~40 Epoch
- 최적 이미지 크기(imgsz): 실험적으로 선택
- 사전 학습 모델: YOLO v11 공식 모델 중 선택
- **single\_cls = 1** 설정

(모든 객체를 하나의 클래스로 간주하여 학습하는 방식, 본 시스템에서는 객체 존재 유무만 판단하면 되므로 적용 가능)

아래 그림은 객체 인식을 수행한 모델의 결과 중 하나로, 보행자를 정확히 감지하고 추적하는 모습을 보여준다.

객체 탐지 모델이 사람이나 물체를 인식한 이후, 해당 객체와의 **충돌 가능성**을 정량적으로 평가하기 위해서는 먼저 **충돌 가능 영역(Collision Zone)**을 정의해야 한다. 사람의 보행 인지 방식을 모방하면, 보행자는 전방 시야 내에 특정 영역을 인지하며, 이 영역 내에 어떠한 고정체 또는 이동체가 존재할 경우 충돌 가능성이 있다고 판단한다. 본 시스템은 이러한 개념을 기반으로 충돌 가능 영역을 수학적으로 정의하고자



한다.

**VFoV**  
(수직시야각)  $= 2\tan^{-1}(\frac{S_v}{2F})$

**HFoV**  
(수평시야각)  $= 2\tan^{-1}(\frac{S_n}{2F})$

$S_v$  : 센서 수직 크기  
 $S_n$  : 센서 수평 크기  
 $F$  : 초점거리

EX) 갤럭시 S24 Ultra

**VFoV**

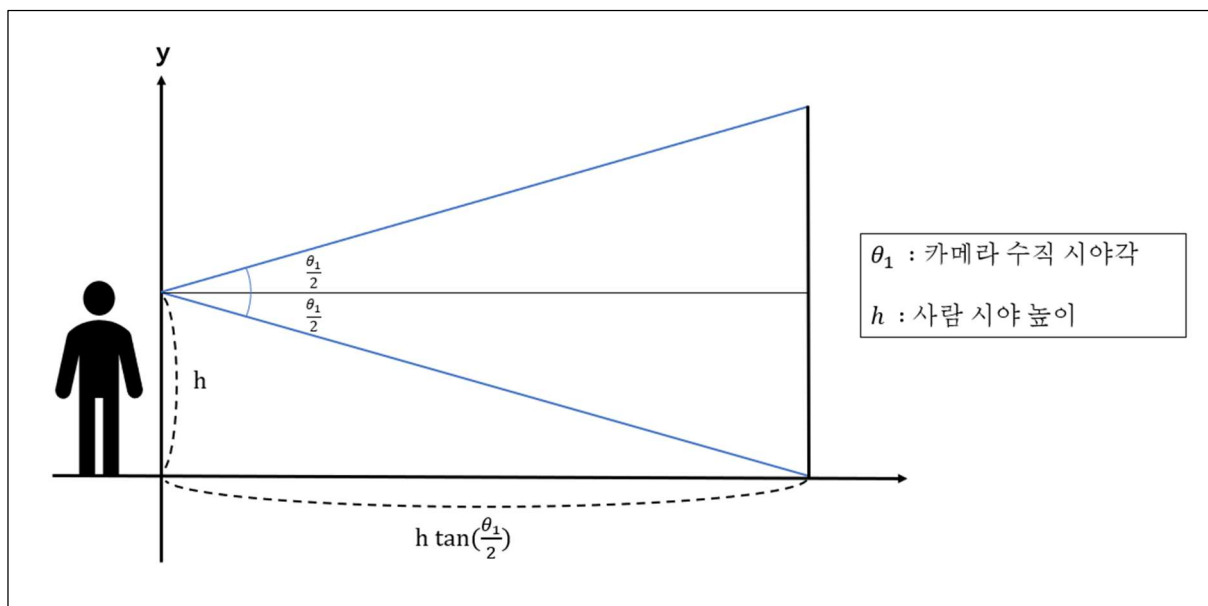
**HFoV**

$2\tan^{-1}(\frac{11.72}{2 \times 23}) = 28.59^\circ = \theta_1$  (수직)

$2\tan^{-1}(\frac{15.63}{2 \times 23}) = 37.53^\circ = \theta_2$  (수평)

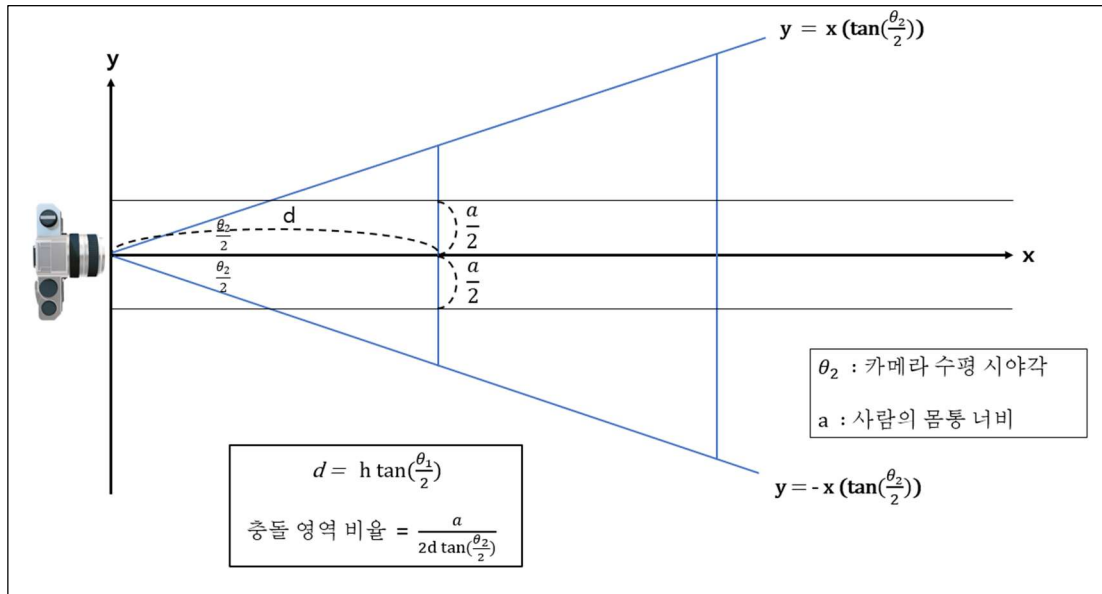
[수식 1] 센서에 따른 수직시야각과 수평 시야각

수직 시야각 (Vertical Field of View, VFoV) 및 수평 시야각 (Horizontal Field of View, HFoV) 을 계산하는 공식을 보여준다. 이 두 시야각은 카메라 센서의 물리적 크기와 초점 거리(Focal Length)를 바탕으로 위와 같이 계산할 수 있다.



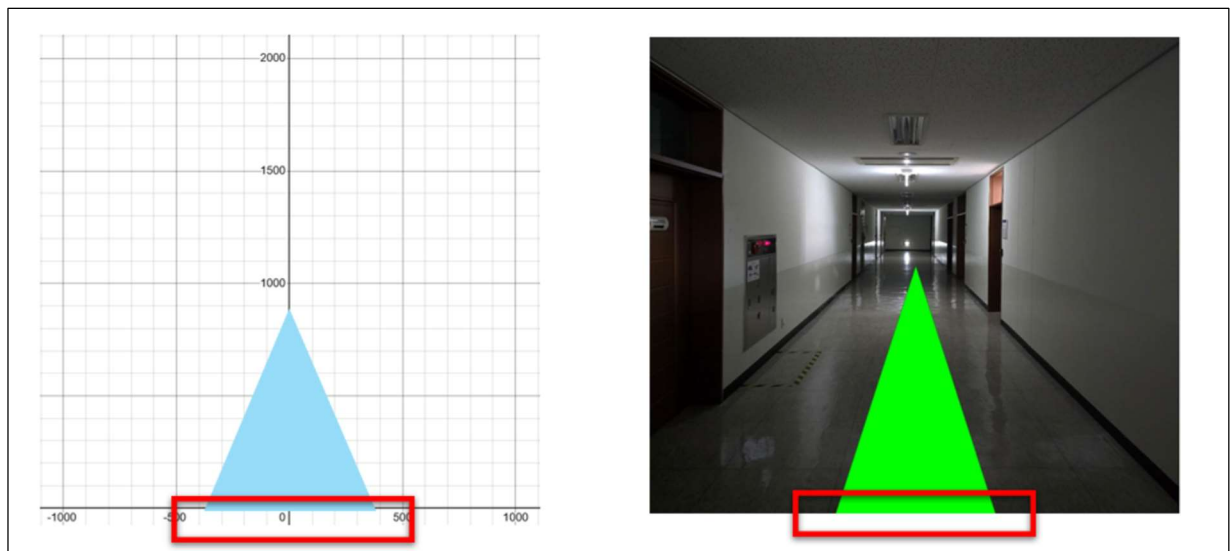
[그림 2] 수직 시야각 기반 거리 계산

[그림 2]은 사람의 눈높이(h)에서 바라본 카메라의 수직 시야각을 기반으로, 영상에서 가장 가까운 바닥 지점까지의 거리(d)를 계산하는 방법을 보여준다. 사람이 실제로 관측 가능한 가장 가까운 바닥지점까지의 거리를 측정 할 수 있으며, 이후 충돌 영역의 전방 경계를 설정하는 데 사용된다.



[그림 3] 수평 시야 각 기반 충돌 가능 폭 계산

[그림3] 은 카메라 수평 시야각과 앞서 구한 거리를 통해 충돌 가능 영역의 비율을 정량화 할 수 있다.



[그림4] 충돌 가능 영역의 실제 예시

[그림 4] 는 앞서 계산한 충돌 가능 영역 비율을 실제 이미지에 적용한 예시를 보여 준다.

정량적으로 구한 충돌 영역 비율을 기반으로 다음과 같은 방식으로 이미지 내 영역을 정의한다:

1. 가장 하단부에서 시작하여, 충돌 가능 비율만큼의 좌우 폭을 시야 내에서 차지하도록 설정한다.
2. 해당 폭을 기준으로, 이미지 중앙 높이 혹은 관심 깊이 위치까지 삼각형 형태로 연결함으로써 충돌 가능 영역(Collision Risk Zone)을 정의한다.
3. 이 영역은 사람이 인지하고 충돌을 피할 수 있는 실질적인 범위를 나타내며, 이후 객체와의 겹침 여부를 평가하는 기준이 된다.

영역 비율을 사진의 가장 아래 부분에서 표시한 이후, 사진의 중간 지점과 연결해주면, 사진과 같이 충돌 가능 영역을 정의할 수 있다.

위 영역을 통해서, 사진에서 인식된 여러 객체의 데이터를 다음과 같은 방식으로 처리할 수 있다.

1. 객체의 Bounding Box 와 충돌 가능 영역의 겹치는 정도

객체의 바운딩 박스가 충돌 가능 영역과 겹치는 픽셀 비율 (Intersection over Zone, IoZ)을 계산한다. 겹치는 면적이 클수록, 해당 객체가 실제로 충돌할 가능성이 높아진다.

2. 객체의 Bounding Box 좌표의 충돌 가중치

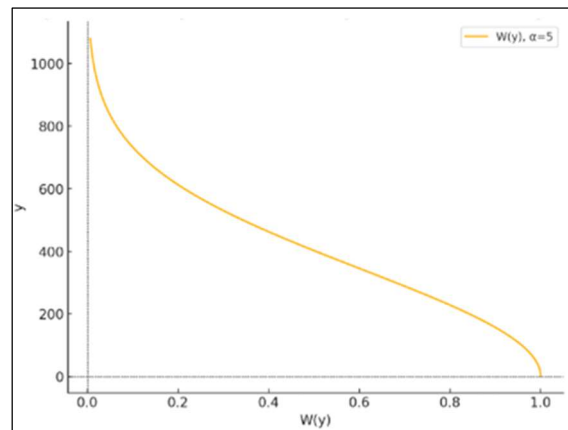
객체 인식에 사용되는 Bounding Box는 객체의 위치와 대략적인 크기를 파악하는 데 유용하지만, 이는 단순한 사각형 형태로 객체를 감싸기 때문에 실제 객체의 형상이나 바닥과의 접지 여부를 정확하게 반영하지 못한다. 특히 충돌 여부를 판단하기 위해서는 객체가 바닥과 얼마나 접촉하고 있는지, 혹은 충돌 가능 영역 내에 얼마나 깊게

포함되어 있는지 등의 정밀한 정보가 필요하다.

객체의 Bounding Box 좌표를 기반으로 보행자와의 충돌 가능성을 정량화하기 위해, 본 시스템은 두 가지 핵심 가중치 요소를 정의하고 이를 결합하는 방식으로 위험도를 평가한다. 이 가중치는 객체의 위치, 예상 경로, 시야의 중심성과 거리 등을 반영하여 충돌 가능성을 정밀하게 계산하도록 설계되었다.

#### - 현재 객체의 y 좌표에 대한 가중치

객체의 y 좌표는 이미지 상에서의 수직 위치를 의미하며, 일반적으로 y 값이 작을수록(즉 이미지 하단에 가까울수록) 보행자와의 실제 거리도 가까운 것으로 간주할 수 있다. 이를 바탕으로 본 시스템은 아래와 같은 지수 함수 기반의 거리 가중치 함수를 설계하였다



$$W_1(x) = e^{-\beta \frac{y}{1080}}$$

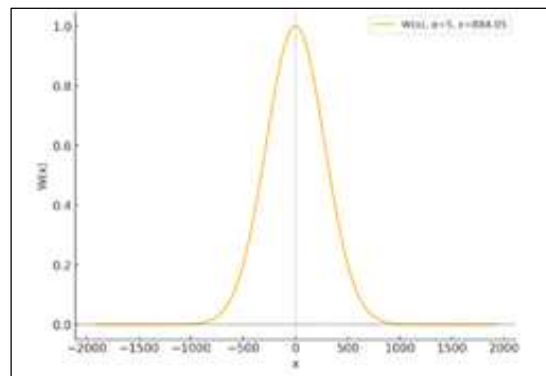
y 값은 현재 사진 객체의 y 좌표이며, 현재 y 값이 0픽셀에 가까울수록 객체와 가까

우므로, 높은 가중치가 설정된다. 즉, 객체가 가까울수록 위 가중치는 높게 부여한다.

#### - 예측한 객체의 x 축과의 교점에 대한 가중치

두 번째 요소는 객체가 보행자의 **예상 이동 경로**와 얼마나 일치하는지를 평가하는 가중치다. 만약, 객체의 예측한 x축의 값이 충돌 가능 영역에 들어오면, 이에 해당하는 충돌 영역의 예측 가중치를 곱해준다. 충돌 영역의 가중치 함수는 범위  $[-\epsilon, \epsilon]$ ,

$(\epsilon = \frac{h}{2d \tan(\theta/2)} \times 1920)$  에서 높은 값(1에 가까운 값)을 가지고 범위 외의 값은 낮은 가중치를 받는 함수를 설정하기 위해 지수 함수 기반의 함수를 설정한다. 아래는 지수 함수 가중치 함수와  $d = 2m$ ,  $\theta$ 는 57도,  $a$ 는 5값으로 했을 때의 가중치 함수의 그래프이다.



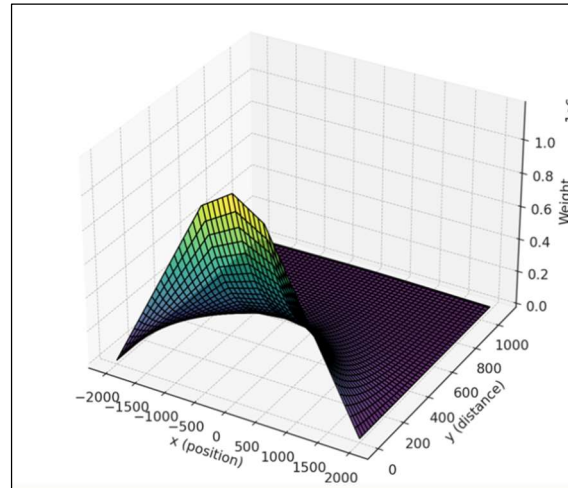
$$W_2(x') = e^{-a(\frac{x'}{\epsilon})^2} \quad (\epsilon = \frac{h}{2d \tan(\theta/2)} \times 1920)$$

위 가중치 함수는 보행자의 진행 방향을 기준으로 충돌 가능성을 평가하는 데 사용된다. 시야의 중앙부는 보행자의 실제 이동 경로와 밀접한 관계가 있으므로, 가중치 함수의 중앙 영역에 높은 가중치를 부여하고, 경계 영역( $x = \epsilon$ ) 부분에서는 가중치가 급격하게 감소하여 충돌 가능성이 낮음을 나타낸다.

최종적으로 위 두 가중치를 곱해 주고, 두 가중치의 곱이 일정 값 이상 크다면, 보행자에게 위험신호를 준다. 아래는 두 가중치를 곱한 공식과 3차원 그래프로 나타낸 그

럼이다.

$$W(x) = W_{1(y)} \cdot W_{2(x')} = e^{-\beta \frac{y}{1080}} \cdot e^{-\alpha \frac{x'}{\epsilon^2}}, \quad (\epsilon = \frac{h}{2d \tan(\theta/2)} \times 1920)$$



두 가중치를 곱한 최종 가중치는 객체와 보행자 간 충돌 가능성을 정량적으로 평가할 수 있는 아이디어이다. 첫 번째 가중치는 객체와 보행자 간 거리 기반 위험성을, 두 번째 가중치는 예측한 객체의 x축 교점을 통해 충돌 경로 내 중요도를 반영한다. 이때 최종 가중치가 임계값을 초과하면, 시스템은 충돌 가능성이 높다고 판단하여 진동과 음성 안내 등으로 긴급 경고를 발동한다. 이러한 방식은 실시간 데이터를 반영해 보행 환경의 동적 특성에 효과적으로 대응할 수 있으며, 보행 안전뿐 아니라 자율 주행, 스마트 시티 등 다양한 응용 분야로 확장이 가능하다.

### 3. 서피스 마스킹 데이터셋의 위협 객체의 충돌 가능영역 내 여부 파악

객체 인식에 사용되는 **Bounding Box**는 객체의 위치와 대략적인 크기를 파악하는 데 유용하지만, 이는 단순한 사각형 형태로 객체를 감싸기 때문에 실제 객체의 형상이나 바닥과의 접지 여부를 정확하게 반영하지 못한다. 특히 **충돌 여부를 판단하기 위해서**

는 객체가 바닥과 얼마나 접촉하고 있는지, 혹은 충돌 가능 영역 내에 얼마나 깊게 포함되어 있는지 등의 정밀한 정보가 필요하다.

이를 보완하기 위해, 본 시스템은 서피스 마스킹(Surface Masking) 데이터셋을 활용한다. 서피스 마스킹은 객체의 외곽선을 따라 픽셀 단위로 분할(Segmentation)된 정보를 제공하며, 각 객체의 정확한 모양, 위치, 바닥과의 접촉 면적을 파악할 수 있다.

서피스 마스킹 데이터는 각 객체의 세그먼트 마스크(픽셀 단위 영역)를 제공하므로, 충돌 가능 영역과의 정확한 겹침 영역을 계산할 수 있다. 위험 객체의 마스크 중 일부가 충돌 가능 영역 내에 존재할 경우, 그 영역이 위험 요소로 판단될 수 있는 근거가 된다.

$$\text{충돌 포함율} = \frac{\text{객체 마스크} \cap \text{충돌 영역}}{\text{객체 마스크 전체 면적}}$$

해당 모델을 통해 위험 객체의 충돌 포함율을 처리하여 넣을 수 있으며 해당 값을 통해 충돌 위험도를 측정할 수 있다.

위를 종합하면, 하나의 영상 프레임에서 추출되는 주요 데이터는 다음과 같다:

### 1. 객체의 Bounding Box 좌표

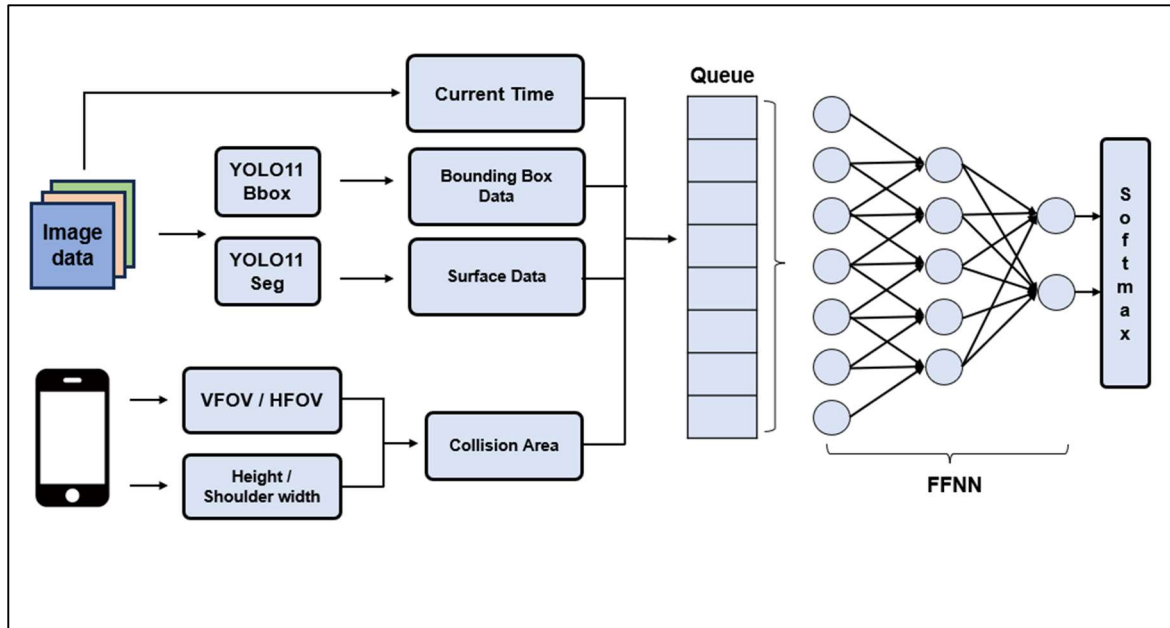
- ▶ 객체의 위치 및 크기 정보를 포함하는 직사각형 영역

### 2. Bounding Box와 충돌 영역의 겹침 비율

- ▶ 객체의 바운딩 박스가 충돌 위험 영역과 얼마나 겹치는지를 나타내는 정량적 수치

### 3. Surface Mask와 충돌 영역의 겹침 비율

- ▶ 객체의 실제 형태를 나타내는 마스크와 충돌 위험 영역의 중첩 정도



[그림5] AI 모델 전체적 구조

#### - 충돌 가능성 예측

객체 인식을 통해 추출된 객체의 여러 정보들은 시계열로 저장되어 시스템 내부의 Queue 구조에 저장된다. Queue는 일정 시간 동안의 객체 탐지 정보를 순차적으로 기록하는 역할을 하며, 실시간 충돌 예측의 핵심적인 입력으로 사용된다. Queue는 총 512개에서 최대 1024개까지의 데이터 항목을 보유할 수 있도록 구성되며, 시스템은 초당 약 20프레임(FPS = 20, 즉 0.05초 간격)으로 이미지를 처리한다. 한 프레임에는 평균적으로 약 5개의 객체가 탐지되므로, Queue에는 약 100프레임, 즉 약 5초간의 시계열 데이터가 지속적으로 유지된다.

이렇게 구성된 Queue는 각 시간 프레임에서 추출된 객체 정보와 충돌 영역 정보가 시간 순서대로 저장되어 있어, 충돌 가능성 예측에 적합한 형태의 입력을 제공한다. 예측 모델은 이 Queue 데이터를 기반으로 2~3초 후에 발생할 수 있는 충돌 여부를 판단하게 된다. 다시 말해, 현재부터 몇 초 뒤에 발생할 수 있는 위험 상황을 사전에 예측하고 사용자에게 경고를 제공하는 것이 본 시스템의 주요 목적이다.



예측을 수행하는 분류기(Classifier)는 다양한 AI 기법을 활용할 수 있으며, 특히 시계열 데이터를 다루기 위한 구조가 필요하다. 가장 단순한 구조는 **Feed-Forward Neural Network (FFNN)** 을 사용하는 것이지만, 이는 시계열 정보의 흐름을 반영하지 못하므로 정확도에 한계가 있다. 따라서 시계열 특성을 반영할 수 있는 **순환 신경망 계열(RNN-based)** 모델의 적용이 바람직하다. 대표적으로 **LSTM (Long Short-Term Memory)** 과 **GRU (Gated Recurrent Unit)** 모델이 있으며, 이들은 시간 축에서 데이터 간의 관계를 학습하여, 객체의 움직임과 충돌 위험의 변화를 보다 정교하게 예측할 수 있다. 또한, 트리 기반의 **XGBoost**와 같은 모델도 피쳐 기반으로 시계열을 압축하여 적용할 수 있어, 빠른 학습 속도와 높은 성능 측면에서 좋은 대안이 될 수 있다.

이러한 분류기를 학습시키기 위해서는, 실제 충돌 여부가 명확히 라벨링된 연속 영상 데이터가 반드시 필요하다. 그러나 현재 공개된 데이터셋 중에서는 이러한 형태의 시계열 충돌 라벨 데이터를 찾기 어려운 상황이므로, 본 AI 개발팀은 해당 데이터를 **직접 수집하고 라벨링**할 계획이다. 수집 방식은 주로 실내 평지 복도 환경에서 사람이 실제로 장애물이나 이동체와 **실제 충돌하는 상황을 촬영**하는 방식으로 진행된다. 충돌 시점과 종료 시점을 정확히 기록한 후, 전체 영상을 0.05초 단위로 분할하여 **충돌 발생 여부를 프레임 단위로 라벨링**하는 정교한 전처리 과정을 거친다.

이 과정을 통해 생성된 데이터셋은 약 **10GB 규모**로 구성될 예정이며, 시계열 분류기의 학습에 필요한 충분한 양의 데이터와 다양한 상황을 포함하도록 수집된다. 해당 데이터는 충돌 전후의 다양한 객체 행동 양상을 포함하고 있어, 모델이 시계열의 흐름 속에서 ‘충돌로 이어질 수 있는 패턴’을 학습할 수 있도록 설계된다.

또한, 확보된 원본 데이터에 대해 **데이터 증강(Data Augmentation)** 기법을 적용하여 학습 다양성을 확보하고 데이터의 양을 추가로 증가시킬 계획이다. 구체적으로는 **좌우 반전(Flip)**, 밝기 및 색상 조절(Color Jitter), 노이즈 삽입(Gaussian Noise), 작은

범위의 공간 이동(Shift/Translate) 등 시야와 객체의 변형에 영향을 주지 않는 범위 내에서의 증강 기법을 활용할 예정이다. 이러한 증강을 통해 원본 데이터의 분포를 확장하고, 모델이 다양한 환경에서도 일반화된 성능을 발휘할 수 있도록 학습 데이터의 양과 다양성을 동시에 확보할 계획이다.

궁극적으로, 본 시스템은 Queue에 저장된 시간 축 데이터를 기반으로 분류기를 통해 충돌 위험도를 사전에 예측하고, 실제 환경에서 사용자에게 2~3초 이내의 위험을 미리 알려주는 사전 경고 시스템으로 기능한다. 이 구조는 객체의 탐지 → 정량화된 충돌 영역 매핑 → 시계열 데이터 누적 → AI 기반 예측이라는 흐름으로 구성되며, 다양한 환경에서도 안정적으로 작동할 수 있도록 고안되었다.

## 2.2.2) SaaS 시스템 구현

### (요약)

- 클라우드 서버 연동
  - 스마트폰과 서버 간 데이터 처리 최적화로 고성능 연산 지원
- 다중 사용자 지원
  - 다양한 사용자에게 AI 기반 시각 보조 서비스를 제공하는 SaaS 형태 구축
- 유연한 접근성
  - 스마트폰 애플리케이션을 통해 서비스 가능

본 프로젝트는 시각 장애인을 위한 실시간 물체 감지·추적 및 위험 평가 시스템을 구축하고, 다중 사용자에게 모바일 어플리케이션을 통해 해당 AI 기반 시각 보조 서비스를 제공하는 것을 목표로 한다. 핵심 아이디어는 스마트폰 카메라를 통해 영상을 실시간 전송받아, 백엔드 AI 모델이 이를 분석하고 위험 요소를 감지하면 사용자에게 즉각 안내하는 것이다.

이를 위해 Kubernetes 클러스터 환경에서 GPU 노드를 활용한 AI 모델 추론을 수행하고, 외부 트래픽을 효율적으로 처리하기 위해 MetalLB와 Nginx Ingress를 도입하였다. 또한,

다중 사용자가 동시에 서비스를 이용할 때 발생할 수 있는 문제들을 해결하기 위해 회원 관리, 세션 관리, AI 모델 멀티 플렉싱 및 데이터 프라이버시(GDPR 준수) 등을 고려하였다.

이 문서는 전체 구현 시스템을 크게 (1) 클라우드 서버 연동, (2) 다중 사용자 지원, (3) 유연한 접근성의 세 가지 관점에서 상세히 설명한다.

### 2.2.3) SaaS 전체 시스템 개요

- Kubernetes 클러스터
  - test-001: Master 노드
  - test-002 ~ test-004: Worker 노드 (애플리케이션 및 인프라 구성 요소 배포)
  - GPU 노드: AI 연산 전담(딥러닝 모델 추론 가속)
- 외부 트래픽 처리
  - MetalLB: 로드 밸런싱 IP 할당 (예: 10.0.20.180)
  - Nginx Ingress: 외부로부터 들어오는 HTTP/HTTPS 요청을 라우팅하여 AI-Backend 및 AI-Prediction으로 연결
- AI-Backend & AI-Prediction
  - GPU 노드에서 실행되는 AI 모델(예: 물체 감지, 추적, 위험 평가)을 담당
  - 실시간 요청(이미지 데이터)을 받아 추론 수행 후 결과를 반환
- User & Application
  - 스마트폰 앱 혹은 웹 애플리케이션을 통해 사용자 영상/이미지를 업로드
  - 결과(충돌 위험 등)를 즉시 받아 음성, 진동 등의 방식으로 사용자에게 안내
- 네트워크 경로

- 외부 사용자 → DKU 게이트웨이(NAT) → Public IP([220.xxx.xxx.xxx](#)) → MetalLB IP(10.0.20.180) → Nginx Ingress → Kubernetes 내부 서비스

## 2.2.4) SaaS 구현 상세

### (요약)

#### 1. 클라우드 서버 연동

- Kubernetes 클러스터 구성 및 GPU 노드 설정
- MetalLB & Nginx Ingress를 통한 외부 트래픽 처리
- 서버 보안(SSL/TLS) 및 NAT/방화벽 설정

#### 2. 다중 사용자 지원

- 회원 관리(가입·로그인, JWT 인증, 권한(Role) 분리)
- AI 추론 요청 동시 처리(HPA, GPU 확장 등)
- WebSocket 기반 세션 관리 및 Rate Limiting
- GDPR 준수 및 데이터 프라이버시 보호

#### 3. 유연한 접근성

- 모바일 앱에서 실시간 카메라 이미지 전송(0.05초 간격)
- 음성 명령, 진동 피드백 등 시각 장애인 친화적 기능
- AI 추론 결과를 실시간으로 사용자에게 안내

### 2.2.4.1 클라우드 서버 연동

#### 1. Kubernetes 클러스터 및 GPU 노드 구성

- 클러스터 구성

- Master 노드(test-001)는 Kubernetes API 서버, 스케줄러, 컨트롤러 매니저 등을 실행하여 클러스터 전체를 관리한다.
- Worker 노드(test-002 ~ test-005)에는 실제 애플리케이션(예: Nginx, AI-Backend 컨테이너 등)이 배포된다.

- GPU 노드(10.0.20.161)는 딥러닝 모델 추론을 가속하기 위해 NVIDIA GPU(또는 AMD GPU)를 탑재한 서버로 구성한다.
- AI 모델 배포
  - GPU 노드에 AI 추론 컨테이너(Pytorch, TensorFlow 등)를 배포하여, 영상·이미지 분석을 전담한다.
  - 필요 시 모델 서빙 프레임워크(TensorFlow Serving, TorchServe 등)를 사용할 수 있다.
- MetalLB를 이용한 Load Balancing
  - 클러스터 내부에서 LoadBalancer 타입의 Service를 사용하면, MetalLB가 외부 IP(예: 10.0.20.180)를 할당하여 외부와의 통신을 가능하게 한다.
  - 이 외부 IP는 DKU 게이트웨이를 통해 Public IP와 매핑되어, 인터넷에서 접근할 수 있는 진입 지점이 된다.
- Nginx Ingress
  - 외부로부터 들어오는 HTTP/HTTPS 트래픽을 AI-Backend, AI-Prediction 등으로 라우팅한다.
  - WebSocket을 사용하는 경우, Nginx Ingress에서 upgrade 헤더와 관련 설정을 통해 WebSocket 연결을 지원한다.

## 2. 서버 보안 및 네트워크

- SSL/TLS 설정

- 외부에서 접속 시 HTTPS 프로토콜 사용을 권장하며, Nginx Ingress에 인증서(예: Let's Encrypt)를 적용한다.
  - WebSocket 연결 또한 SSL을 통해 암호화된 통신을 수행한다.
- 방화벽 및 NAT
    - 내부 인프라(10.0.20.0/24)와 외부 네트워크 사이에는 Router NAT와 DKU 게이트웨이가 위치한다.
    - Router NAT는 필요한 포트(예: 80, 443, 3000 등)만 오픈하여 외부 접근을 제한한다.

## 2.2.4.2 다중 사용자 지원

### 1. 회원 관리 (Authentication & Authorization)

- 회원 가입 & 인증
  - REST API 기반의 회원 가입 및 로그인 기능 제공.
  - JWT(Json Web Token)를 사용하여 인증 토큰을 발급하고, 이후 요청 시 헤더에 토큰을 포함해 인증한다.
- 접근 권한(Authorization)
  - 시각 장애인 사용자, 관리자 등 권한(Role)을 구분할 수 있다.
  - 백엔드에서 JWT를 검증해 요청이 합법적인지 확인하고, 적절한 리소스에만 접근을 허용한다.
- 데이터 프라이버시 & GDPR 준수

- 사용자 영상·이미지 등 민감 정보 처리 시, 사용자의 동의 및 개인정보 보호 규정 준수.
- 요청이 완료되면 이미지를 서버에 저장하지 않고 즉시 메모리 상에서 AI 모델에 전달 후 폐기한다.
- 필요 시, 개인정보보호 책임자(DPO)를 지정하거나 관련 정책(개인정보 파기 주기 등)을 문서화한다.

## 2. AI 서비스의 동시성 및 확장성

- 부하 관리

- 여러 사용자가 동시에 이미지 데이터를 전송할 경우, GPU 노드의 추론 부하가 급증할 수 있다.
- Kubernetes Horizontal Pod Autoscaler(HPA)를 사용해 AI-Backend 레플리카를 동적으로 증가시키고, GPU 리소스가 필요한 경우 Node 수를 추가하거나 멀티 GPU 서버로 확장한다.

- 멀티 플렉싱(Multiplexing) 전략

- WebSocket 연결마다 독립된 AI 모델 세션을 할당하거나, 공유 모델을 세션별 컨텍스트로 구분하는 방식 등 다양한 설계를 적용한다.
- 한 세션이 과도한 리소스를 사용하지 않도록 Rate Limiting(요청 제한) 정책을 설정한다.

- 세션 관리

- WebSocket 세션은 연결이 유지되는 동안 AI 모델의 상태를 보관하며, 연결 종료 시 자동 정리한다.
- 이를 통해 메모리 누수를 방지하고, 사용자별 독립성을 보장한다.

### 3. 사용자 단말 App 구현

- 실시간 이미지 전송
  - 스마트폰 카메라를 활용하여 0.05초 간격으로 이미지를 촬영한다.
  - WebSocket을 통해 서버와 지속적으로 연결하여, 암호화된 채널로 이미지를 전송한다.
- 음성 명령 및 진동 피드백
  - 시각 장애인 접근성을 강화하기 위해, 앱 내부에서 음성 명령을 인식하거나, 진동 등을 통한 안내를 제공한다.
  - 예: “촬영 시작” 음성 명령 시 자동으로 촬영 주기를 활성화하고, 위험 요소 감지 시 진동 알림 전송.
- 데이터 처리 흐름
  1. 촬영: 스마트폰 카메라에서 이미지를 캡처
  2. 전송: WebSocket SSL을 통해 AI 백엔드에 이미지 업로드
  3. 추론: GPU 노드에서 충돌 위험 평가
  4. 결과 반환: AI 백엔드가 추론 결과를 앱으로 전송
  5. 사용자 안내: 앱은 음성 안내 또는 진동으로 위험을 전달

## 2.2.5) 시스템 기능적 요구 사항

### 2.2.5.1 클라우드 서버(Kubernetes) 연동



## 1. 클러스터 구성

- (FR-001) Kubernetes Master 노드는 API 서버, 스케줄러, 컨트롤러 매니저 등을 구동하여 클러스터 전체를 관리해야 한다.
- (FR-002) Worker 노드들은 Master 노드와 통신하여, 애플리케이션(컨테이너)을 배포·운영해야 한다.
- (FR-003) GPU 노드는 딥러닝 모델을 구동하기 위한 NVIDIA GPU(또는 AMD GPU)를 탑재해야 하며, AI 추론용 컨테이너를 실행 가능해야 한다.

## 2. AI 모델 배포

- (FR-004) AI 모델(Pytorch, TensorFlow 등)을 위한 컨테이너 이미지를 생성하고, GPU 노드에서 해당 이미지를 실행해야 한다.

## 3. 로드 밸런싱 및 트래픽 처리

- (FR-006) MetalLB를 통해 LoadBalancer 타입의 Service를 생성할 시, 외부 IP가 자동 할당되어 외부 접근이 가능해야 한다.
- (FR-007) Nginx Ingress를 통해 HTTP/HTTPS 요청이 AI-Backend 및 AI-Prediction 서비스로 라우팅되어야 한다.
- (FR-008) WebSocket 연결을 지원하기 위해 Nginx Ingress에서 upgrade 및 connection 헤더를 처리해야 한다.

### 2.2.5.2 다중 사용자 지원

#### 1. 회원 관리(인증·인가)

- (FR-009) REST API를 통해 회원 가입 및 로그인 기능을 제공해야 한다.
- (FR-010) 인증 방식으로 JWT(Json Web Token)를 사용하여, 로그인 성공 시 JWT를 발급하고 이후 모든 요청은 토큰을 포함해야 한다.

- (FR-011) 관리자·일반 사용자 등 권한(Role)을 구분하여, 중요한 리소스에는 관리자만 접근 가능하도록 해야 한다.

## 2. AI 서비스 동시성

- (FR-012) 여러 사용자가 동시에 이미지를 전송할 경우에도 GPU 노드가 정상적으로 추론을 수행해야 한다.
- (FR-013) AI-Backend의 Pod 레플리카 수를 HPA(Horizontal Pod Autoscaler)로 조절하여 트래픽 증가 시 자동 확장해야 한다.
- (FR-014) GPU 리소스가 부족할 경우, 추가 GPU 노드를 쉽게 확장할 수 있도록 클러스터 구성이 유연해야 한다.

## 3. 세션 관리

- (FR-015) WebSocket 연결은 각 사용자 세션별로 독립적으로 유지되어야 한다.
- (FR-016) 세션이 종료되면 해당 세션과 연관된 AI 모델 상태(컨텍스트)는 자동으로 정리되어야 한다.
- (FR-017) Rate Limiting(요청 제한) 또는 QoS 정책을 통해 특정 세션이 과도한 리소스를 사용하지 않도록 제어해야 한다.
- (FR-018) 각 사용자마다 AI의 사용량은 산정되어야 하며, 이는 AI 세션 유지 시간으로 산정한다.

## 4. 데이터 프라이버시 & GDPR

- (FR-019) 사용자 이미지는 서버에 영구 저장하지 않고, AI 모델 추론 후 즉시 폐기해야 한다.
- (FR-020) 개인 영상·이미지 등 민감 데이터를 처리할 때, 사용자 동의 절차

차 및 개인정보 보호법(예: GDPR)을 준수해야 한다.

- (FR-021) 필요 시 개인정보 파기 정책을 문서화하고 준수해야 한다.

### 2.2.5.3 유연한 접근성

#### 1. 실시간 이미지 전송

- (FR-022) 스마트폰 앱은 0.05초 간격으로 카메라 이미지를 촬영하여 WebSocket을 통해 전송할 수 있어야 한다.
- (FR-023) 전송되는 이미지는 SSL을 통해 암호화되어야 하며, 전송 실패 시 자동 재시도 기능을 제공해야 한다.

#### 2. 음성 명령 및 진동 피드백

- (FR-024) 앱은 음성 명령(예: “촬영 시작”)을 인식하여 사진 촬영 주기를 활성화할 수 있어야 한다.
- (FR-025) 위험 요소가 감지되면 진동 알림, 음성 안내 등으로 사용자에게 즉각적인 피드백을 제공해야 한다.

#### 3. 데이터 처리 흐름

- (FR-026) 전송된 이미지는 AI 백엔드가 GPU 노드로 전달하여 물체 감지 및 위험 평가를 수행해야 한다.
- (FR-027) 추론 결과(충돌 위험 등)는 앱에 즉시 반환되어, 사용자에게 안내 메시지(음성/진동)로 표시되어야 한다.

### 2.2.6) 시스템 비기능적 요구사항 (Non-Functional Requirements)

#### 1. 성능 및 확장성

1. (NFR-001) GPU 노드는 초당 최소 X장의 이미지를 처리할 수 있어야 한다. (X는 실제 프로젝트 성능 목표에 따라 설정)

2. (NFR-002) HPA를 통해 AI-Backend Pod가 증가할 때, 1분 이내에 새 인스턴스가 준비 상태로 전환되어야 한다.
3. (NFR-003) 시스템은 피크 트래픽(예: 동시 사용자 1,000명) 상황에서도 1초 이하의 응답 시간을 유지하는 것을 목표로 한다.

## 2. 보안 및 안정성

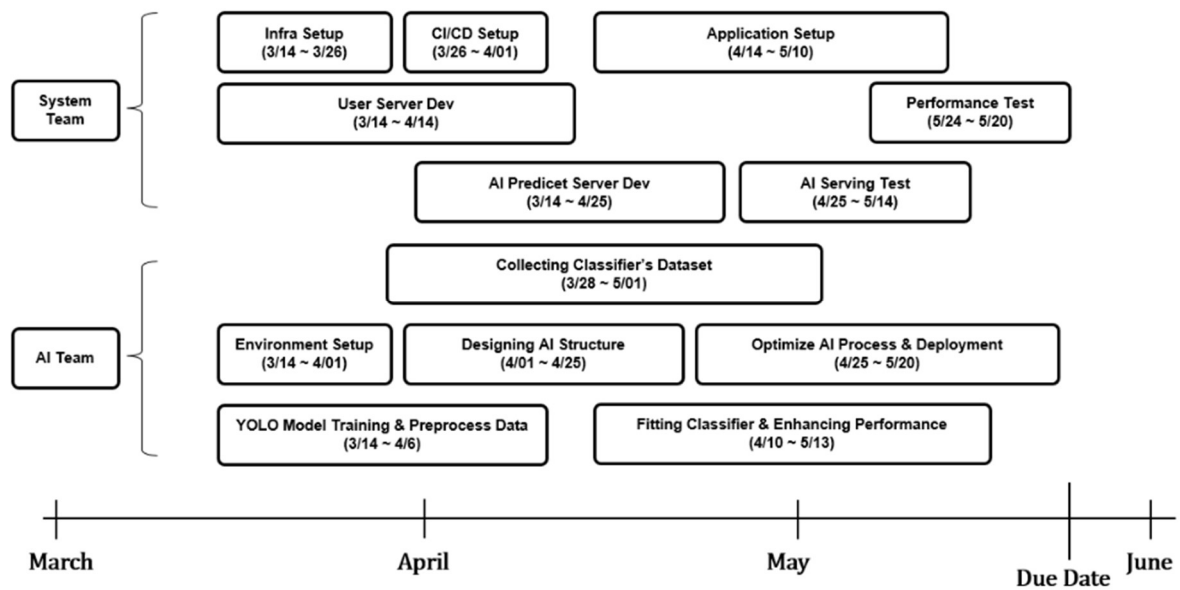
1. (NFR-004) 모든 외부 통신은 HTTPS/TLS를 사용하여 암호화되어야 한다.
2. (NFR-005) 서버 장애나 업데이트 발생 시에도, Kubernetes의 Rolling Update 기능을 통해 무중단 배포가 가능해야 한다.

## 3. 유지보수성 및 로깅

1. (NFR-006) 모든 주요 이벤트(사용자 로그인, AI 추론 요청, 에러 발생 등)에 대한 로그를 수집·모니터링(Prometheus, ELK Stack 등)할 수 있어야 한다.
2. (NFR-007) CI/CD 파이프라인(Jenkins, GitLab CI 등)을 구축하여, 코드 변경 시 자동으로 빌드·테스트·배포 프로세스를 수행해야 한다.
3. (NFR-008) AI 모델 버전 관리를 위해, 모델 업데이트 시 이전 버전과 호환성을 유지하거나 Blue-Green Deployment 방식을 적용할 수 있어야 한다.

## 3. 프로젝트의 추진전략.방법 및 추진체계

### 3.1) 프로젝트의 추진전략 방법



- 작업은 System 팀과 AI 팀으로 구분하여 진행한다.

## - 시스템 팀

- System 팀은 SaaS 시스템 구현을 위해 애자일 모델을 도입하여 개발할 것이며, 계획서에 첨부된 요구사항을 기반으로 하여 구체적인 요구사항 분석 명세서와 설계서를 작성한다.

- SaaS 시스템은 각각의 서비스가 분리된 마이크로서비스 아키텍처의 형태로 구현되며, 유연한 개발을 수행할 수 있도록 도모한다.

- 또한, 빠른 개발과 배포를 통해, AI 팀의 Prediction 모델과 협력할 수 있도록 CI/CI 파이프라인을 구축한다. CI/CD 파이프라인을 통해 구현한 시스템을 빠르게 배포하고 테스트할 수 있도록 인프라 환경을 구축한다.

- 인프라 환경을 Prometheus와 Grafana를 통해 모니터링하며, 시스템 전체의 작업 부하를 추적하며 성능 개선을 모색한다. .

## - AI 팀

-AI 팀은 충돌 예측을 위한 객체 인식 및 충돌 분류 모델을 개발하며, AI 개발 프로세스에는 애자일(Agile) 방식 일부를 도입하여 반복적인 실험과 피드백을 기반으로 모델을 개선할 예정이다.

- YOLO 기반 객체 탐지 모델은 사전학습(pretrained) 모델을 기반으로 하되, 실제 환경에 맞는 대용량 커스텀 데이터셋을 구성하여 재학습(fine-tuning)하며 최적화된 성능을 확보한다.

- 학습 데이터는 충돌 여부에 대한 정밀한 시계열 라벨링을 포함하여 AI팀이 직접 수집 및 전처리하며, 이를 기반으로 시계열 예측이 가능한 FFNN, LSTM, GRU, XGBoost 등의 분류기 모델을 실험적으로 비교·검증한다.

- 분류기의 성능은 정확도, 정밀도, F1-score 등의 지표로 평가되며, 실시간성 확보를 위해 전체 AI 연산을 100ms 이하로 유지하는 것을 목표로 한다.

- 학습된 모델은 System 팀과 협력하여 SaaS 시스템 내 서비스 형태로 탑재되며, 지속적인 모니터링 및 성능 개선 루프를 고려한 구조로 개발된다.

## 4. 프로젝트 성과의 활용 방안 및 기대 효과(사회적 가치)

### 4.1) 프로젝트 성과의 활용방안

본 프로젝트를 통해 개발된 AI 기반 충돌 예측 시스템은 시각장애인의 안전한 보행을 지원하는 데 중점적으로 활용될 수 있다. 특히 웨어러블 디바이스와 연동하여 실시간 객체 인식

및 충돌 위험 예측 기능을 제공함으로써, 실생활에서의 보행 안전성을 높이는 데 기여할 수 있다.

핵심 기술은 모바일 기반으로 구현되어, 스마트 글라스, AI 안내기, 진동 경고 장치 등 다양한 형태의 보조 장치에 적용 가능하며, 필요 시 소형화 및 경량화된 하드웨어와 결합하여 실사용에 적합한 형태로 제공될 수 있다.

또한, 해당 기술은 시각장애인 보행 지원뿐 아니라 **실내 자율주행 로봇**, **교통약자 보조 시스템**, **스마트시티 인프라** 등 다양한 분야에서도 확장적으로 활용될 수 있으며, 추후 산업용 또는 공공시설용 보행 보조 기술로도 응용 가능하다.

#### 4.2) 프로젝트 성과물을 통한 기대효과(사회적 가치)

본 시스템의 도입은 시각장애인의 **보행 안전성 향상**에 직접적으로 기여할 수 있으며, 기존 보조기기에 대한 의존도를 줄이고 **자율적인 이동 능력**을 강화하는 데 도움이 된다. 이를 통해 시각장애인의 이동에 대한 심리적 부담을 완화하고, 보다 적극적인 사회 활동 참여를 유도할 수 있다.

또한, 안정적인 이동 지원은 **이동권 보장**의 측면에서도 긍정적인 효과를 기대할 수 있으며, 궁극적으로는 개인의 삶의 질 향상에 기여할 수 있다.

더불어, 시각장애인 대상 보조 기술의 실질적 보급은 **장애인의 사회적 포용성과 독립성 강화**에 기여하고, 교통약자나 고령자 등 다른 취약계층을 위한 기술 발전에도 기반이 될 수 있다.

이러한 기술의 개발과 적용은 향후 **실생활 밀착형 AI 활용 사례**로 이어질 수 있으며, 공공 안전, 복지 기술, 사회적 약자 지원 등의 다양한 분야에서 실용적 가치를 제공할 수 있다.

## 5. 참고문헌(Reference)

[1] 권대근 외. “시각장애인을 위한 Seeing AI 응용서비스 연구.”

한국멀티미디어학회 논문지, 2022. <https://www.dbpia.co.kr/pdf/pdfAiChatView.do?nodeId=NODE11958395>

[2] 이상훈 외. “딥러닝 기반 시각 장애인을 위한 보행 보조 시스템.”

KIPS 2021 춘계학술대회 논문집.

<https://manuscriptlink-society-file.s3-ap-northeast->

[1.amazonaws.com/kips/conference/kips2021spring/presentation/KIPS\\_C2021A0080.pdf](https://manuscriptlink-society-file.s3-ap-northeast-1.amazonaws.com/kips/conference/kips2021spring/presentation/KIPS_C2021A0080.pdf)

[3] 신재훈 외. “시각장애인 이동지원 시스템을 위한 카메라 기반 실시간 장애물 검출 방법.”

한국정보기술학회 논문지, 2021. <https://www.dbpia.co.kr/pdf/pdfView.do?nodeId=NODE10569017>

[4] 김유진 외. “딥러닝 기반 스마트 글래스를 활용한 시각장애인 보행보조 시스템 개발.”

한국인터넷정보학회논문지, 2023. <https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE12019958>

[5] Yuan, Z., et al. “A Wearable Assistant System for Visually Impaired People.”

IEEE Sensors Journal, vol. 15, no. 12, pp. 6968–6976, Dec. 2015.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7259447>

[6] Kim, Minsoo, et al. “Real-Time Semantic Segmentation for Visually Impaired People Using Smart Glasses.”

IEEE Access, vol. 9, pp. 116308–116319, 2021. <https://ieeexplore.ieee.org/abstract/document/9451710>

[7] Choi, Jonghyun, et al. “Realtime Depth-aware Obstacle Avoidance for Visually Impaired via Glass-mounted Stereo Cameras.”

IEEE Transactions on Biomedical Circuits and Systems, 2024.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10302240>