



Tmap API × 온디바이스 시를 결합한
시각장애인 보행 보조 및 도시 환경 개선 솔루션

윤○○○ · 이지호 · 최○○○

2025.02

3팀(공부많이된다)

"Guiding the Way to Safety"

목차

01

프로젝트 개요

Overview & Background

02

팀 구성 및 역할

Team Roles & Contributions

03

수행 절차 및 방법

Methodology & Timeline

04

수행 경과

Results & Trouble Shooting

05

자체 평가

Self-Evaluation & Retrospective

기획 배경

Overview & Background



70%+

보행 약자

인도 위 무단 방치 장애물로
심각한 부상 위험 노출



40%

점자블록 단절

전국 점자블록 중 파손되거나
끊겨서 기능 상실한 구간 비율



N/A

방해물 감지 어려움

기존 서비스는 횡단보도 및
점자블록 위주의 경로 안내만 하여
도로 위 위험을 감지 어려움



"기존 보조 도구는 발밑만 봅니다. WalkMate는 전방을 봅니다."

Goal 11 지속가능한 도시와 공동체

Target 11.2

취약계층·장애인을 위한
안전하고 접근 가능한 교통체계 제공

→ WalkMate 실시간 장애물 탐지 및 보행 안전 혁신

Target 11.7

장애인을 위한
공공장소 및 녹지에 대한 보편적 접근 제공

→ 체스트 하네스형 장치로 공공장소 자유 이동 실현

Goal 10 불평등 완화

Target 10.2

장애와 무관하게
모든 사람의 사회·경제·정치적 포용 증진

→ 타인 도움 없이 스스로 길 찾기를 통한 자립 지원

 국내 시각장애인 약 25만 명 이동 자립 지원

 보행 관련 장애인 사고 30%+ 감소 기여 가능

"WalkMate는 기술로 이동권 장벽을 허뿁니다"

WalkMate 솔루션

Solution & Differentiation

WHO

시각장애인 및 보행 약자

WHY

인도 위 무단 장애물 점자블록 단절

WHAT

온디바이스 AI 탐지 + Tmap 내비게이션

WHERE

언제 어디서나 (체스트 마운트)

HOW

0.3초 실시간 인식
→ 음성 안내
→ DB 누적



System Architecture Flow



온디바이스 AI

서버 통신 지연 없는 **Zero Latency**
네트워크가 끊겨도 실시간 탐지 지속
YOLOv26n TFLite 최적화

초단거리 정밀 탐지

기존 내비게이션이 무시하는
10m 이내 미세 장애물 회피 가이드
핀홀 카메라 기반 거리 측정

실시간 위험 DB

탐지-저장-알림 **1초 미만** 루프
도시 안전 인프라 확장을 위한
PostGIS 기반 공간 데이터 측정



**스마트폰이 여러분을 위한 능동적인 가이드 역할을 한다고 상상해
보세요.**

팀 구성 및 역할

Team Roles & Contributions

최 멘토님 Support: DB 설계 인사이트 제공 (GPS 좌표 저장 방식)

최

팀장 · Frontend

React

Capacitor

Tmap API

Voice UI

- ✓ Capacitor+React 하이브리드 앱 구축
- ✓ Tmap API + 센서 퓨전(GPS/나침반)
- ✓ 100% Voice UI (No-Touch 인터페이스)
- ✓ 위험물 TTS 우선순위 Queue 시스템

"하드웨어와 소프트웨어의 경계를 무너뜨린 프론트엔드"

이지호

Backend · Cloud

FastAPI

AWS S3

Supabase

PostGIS

- ✓ FastAPI 무중단 API 아키텍처 설계
- ✓ AWS S3 + Supabase 이원화 파이프라인
- ✓ PostGIS 공간 데이터 RPC 필터링 로직
- ✓ WAL 기반 Zero-Refresh Admin 구현

"데이터의 수집부터 적재까지, 신뢰할 수 있는 백엔드 환경 구현"

최

AI · Algorithm

YOLOv26n

TFLite

Pinhole

Transfer Learning

- ✓ YOLOv26n 80→13 레이블 가지치기
- ✓ 볼라드·입간판·킵보드 추가 학습
- ✓ 핀홀 카메라 거리 계산 알고리즘 개발
- ✓ 3단계 위험도 분류기(Risk Level) 구현

"물리 공식으로 AI의 눈을 뜨게 한 알고리즘 엔지니어"

개발 타임라인

Project Timeline & Methodology

STEP 1

02.09 ~ 02.10



기획 & 페르소나

아이디어 확정

- 시각장애인 페르소나 설계
- 핵심 요구사항 정의
- 프로젝트 목표 설정

STEP 2

02.11 ~ 02.12



설계 & AI 학습

아키텍처 수립

- 시스템 아키텍처 설계
- YOLO 모델 학습 시작
- 데이터셋 라벨링

STEP 3

02.13 ~ 02.19



전략적 PIVOT

노선 변경 & 통합

- 점자블록 탐지 포기
- Tmap + 장애물 탐지 전환
- 파트너십 BM 수립

STEP 4

02.19 ~ 02.24



테스트 & 최적화

성능 안정화

- 30FPS 온디바이스 확보
- GPS 오차 보정 테스트
- 클라우드 부하 테스트

기술 스택

Comprehensive Tech Stack & Development Environment

</> Development

OS / Hardware

Galaxy Book2 MacBook Air M4 Android Device

Languages

Python 3.10.4 Java SDK 17 TypeScript / JS

IDE / Environment

Android Studio (Pixel9) VS Code Colab Antigravity

React

☰ Infrastructure & Backend

Framework / Server

FastAPI ngrok (Tunneling)

Database

Supabase (PostgreSQL)

Storage & Version Control

AWS S3 Git GitHub

🧠 AI & Data

LLM (GenAI)

Gemini ChatGPT Pro

Computer Vision

YOLOv26n Roboflow

Model Optimization

TFLite (On-Device)

👥 Collaboration & Design

Documentation

Notion Discord Chrome

Visualization

Napkin AI MS PowerPoint Genspark FFmpeg golpoi

Monitoring / Demo

QuickTime Player

AI 모델 커스터마이징

Custom AI Model & Algorithms

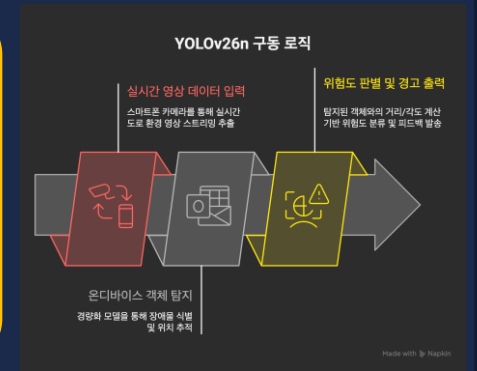
① 모델 개요

YOLOv26n (You Only Look Once) 핵심 원리

이미지 전체를 단 한 번만 훑어(Look Once)
객체의 위치(Bounding Box)와 종류(Class)를 동시에 예측

프로젝트 특화

모바일 환경에서의 실시간성(30 FPS)을 위해 레이어를 최적화(80 -> 13)하고,
보행 장애물 탐지에 특화된 전이 학습 적용



② 모델 선정

모델 선정 근거: 실시간성 및 효율성 극대화

온디바이스 최적화: 시각장애인 보행 안전을 위한 초저지연(Latency < 300ms) 환경 구축을 위해 가벼운 Nano(n) 계열 모델 선정.

성능 비교: 기존 YOLOv8/v10 대비 연산 효율이 개선된 YOLOv26n을 채택하여 모바일 기기 발열 최소화 및 배터리 효율성 확보.

정확도 균형: 소형 모델임에도 보행 장애물(키펠드, 볼라드 등) 탐지에 특화된 전이 학습(Transfer Learning)을 통해 실무 적용 가능한 수준의 mAP 달성.



AI 모델 커스터마이징

Custom AI Model & Algorithms

③ 모델 구조 분석

입력 해상도: 640 × 640 RGB 이미지 스트림 사용.

레이어 구성: 총 260개 레이어로 이루어진 심층 신경망 구조.

최적화(Optimization): TFLite Float 32-bit 변환을 통해 모델 크기를 5.4MB로 압축. 모바일 기기의 메모리 점유율을 낮추고 배터리 효율을 향상

④ 데이터셋 구조

기존 클래스(10종): COCO Dataset 기반 (사람, 자동차, 버스 등 보편적 객체)

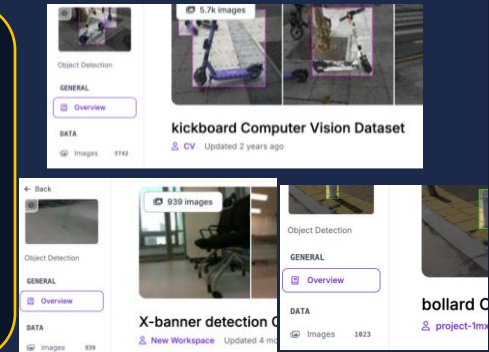
특화 클래스(3종): Roboflow 수집 데이터 (볼라드, 현수막, 등 보행로 특화)

Person (Class 0): 268,029건 (최다 분포)

Car (Class 2): 45,449건

특화 객체: Bollard(3,584건), Banner(2,388건), Kickboard(3,364건)

학습 방법: 전이 학습(Transfer Learning) 적용.



AI 모델 커스터마이징

Custom AI Model & Algorithms

✂ ① 커스텀 레이블



person car bus truck bicycle
motorcycle traffic light stop sign bench
dog

🌟 추가 학습 (Transfer Learning)

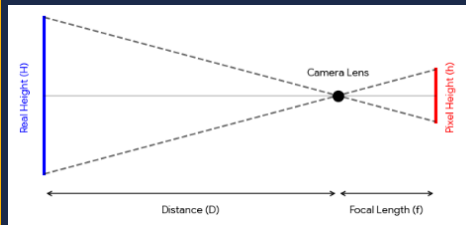
⚠ 블라드 🗨 입간판 ⚡ 키포드

📏 ② 거리 계산 알고리즘

$$\text{Distance (m)} = \frac{(\text{Real_H} \times \text{Focal_L})}{\text{BoundingBox_Pixel_H}}$$

$$\frac{\text{실제 높이}(H)}{\text{거리}(D)} = \frac{\text{픽셀 높이}(h)}{\text{초점 거리}(f)}$$

$$D = \frac{H \times f}{h}$$



🚫 ③ 3단계 위험도 분류

HIGH (3) 키포드 · 자전거 · 차량 · 버스 · 트럭
● 5m 이내 즉시 경고

MED (2) 사람 · 신호등 · 스탑 사인 · 개
● 3m 이내 경고

LOW (1) 블라드 · 벤치 · 표지판 · 나무
● 참고 정보 제공

💡 위험도별 TTS 우선순위 자동 조정

사용자 앱 UI/UX

User Application Features & Experience

100% Voice UI

"~로 안내해줘" > STT 인식 > Tmap 연동

화면 터치가 전혀 필요 없는 완전 음성 인터페이스 구현

No-Touch Interface

지능형 센서 퓨전

GPS Bearing (90%) + 나침반 LPF (10%)

이동 중엔 GPS 궤적 우선,
정지 시 나침반 보조로 체스트 마운트 흔들림 보정

방향 안정화 알고리즘

시계 방향 보정

경로 이탈 시 절대 각도가 아닌 상대적 방향 안내

"3시 방향으로 돌아주세요"

지도를 보지 않고도 직관적인 복귀 가능

직관적 UX

TTS 우선순위 큐 (Priority Queue)

P1: 장애물 경고(즉시) > P2: 길안내 (대기 후 재생)

오디오 충돌 방지를 위한 지연 재생 시스템 구축

S/W 아키텍처로 하드웨어 음성 채널 한계 극복

거리 기반 알림 필터

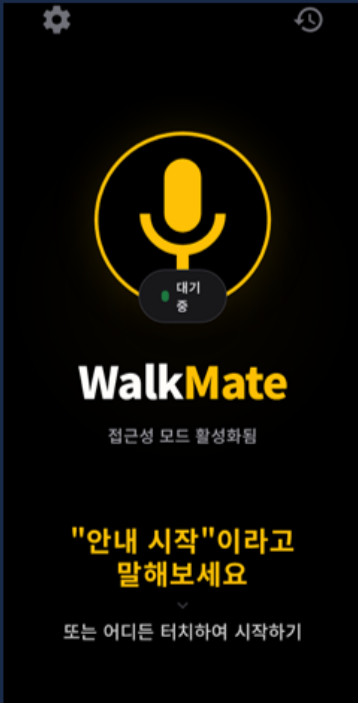
HIGH: 5m 이내 MEDIUM: 3m 이내

위험도에 따른 차등 알림으로 알림 피로도 최소화

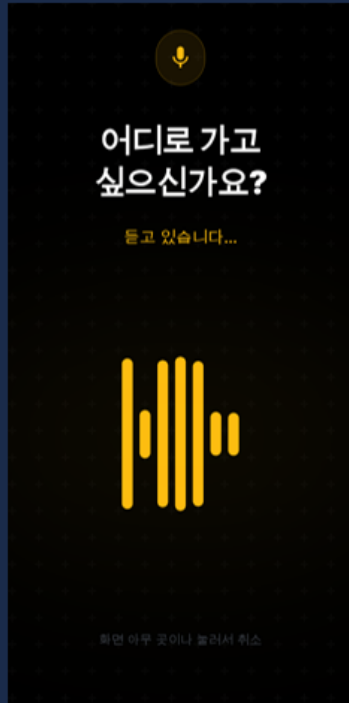
무의미한 잦은 경고음 제거 설계

사용자 앱 UI

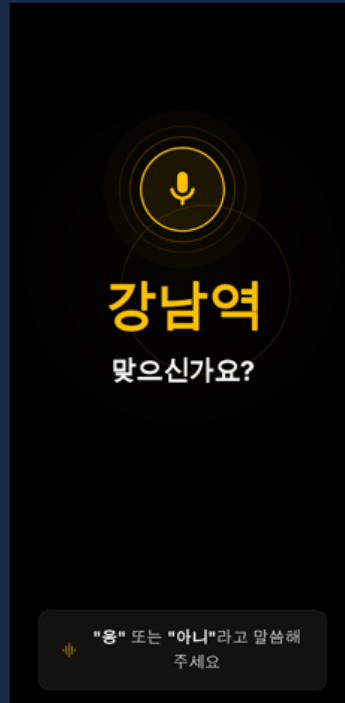
User Application Features & Experience



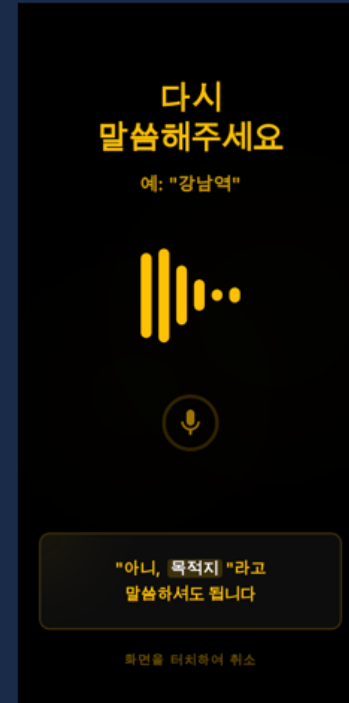
메인 화면



목적지
입력 화면



재확인 화면



목적지
재입력 화면



안내 중 화면

관리자 대시보드

Admin Dashboard & Monitoring System

실시간 히트맵 (Heatmap)

수집된 위험물 좌표를 바탕으로 **도심 내 위험 구역 밀집도**를 히트맵 클러스터로 시각화

앱-웹 간 지연 없는 실시간 데이터 반영

`</>react-leaflet`

통계 대시보드

위험물 발생 추이, 시간대별 빈도, 유형별 통계를 차트로 시각화하여 데이터 기반 의사결정 지원

행정기관용 리포트 자동 생성 데이터 제공

`Recharts`

썸네일 뷰어 & Action

현장 사진 실시간 로드 및 행정 조치 상태(Action) 관리

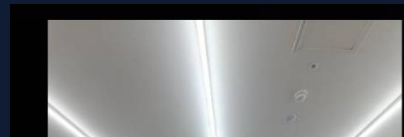
접수 → 처리중 → 완료 상태 추적 테이블

`aws` AWS S3 Integration

User Friendly

다크모드, 사이드바 collapse, filter, pagination, items per page, 정렬
데이터베이스 csv 파일로 내보내기, 체크박스, 처리 상태 변경 페이지 구축

관리자 UI



대시보드

위험 히트맵

마스터 DB

위험 지역 히트맵

접수된 장애물 데이터를 기반으로 실제 지도 위에 위험 밀집 구역을 시각화합니다.

지도 필터

위험 등급 (RISK LEVEL)

전체 보기

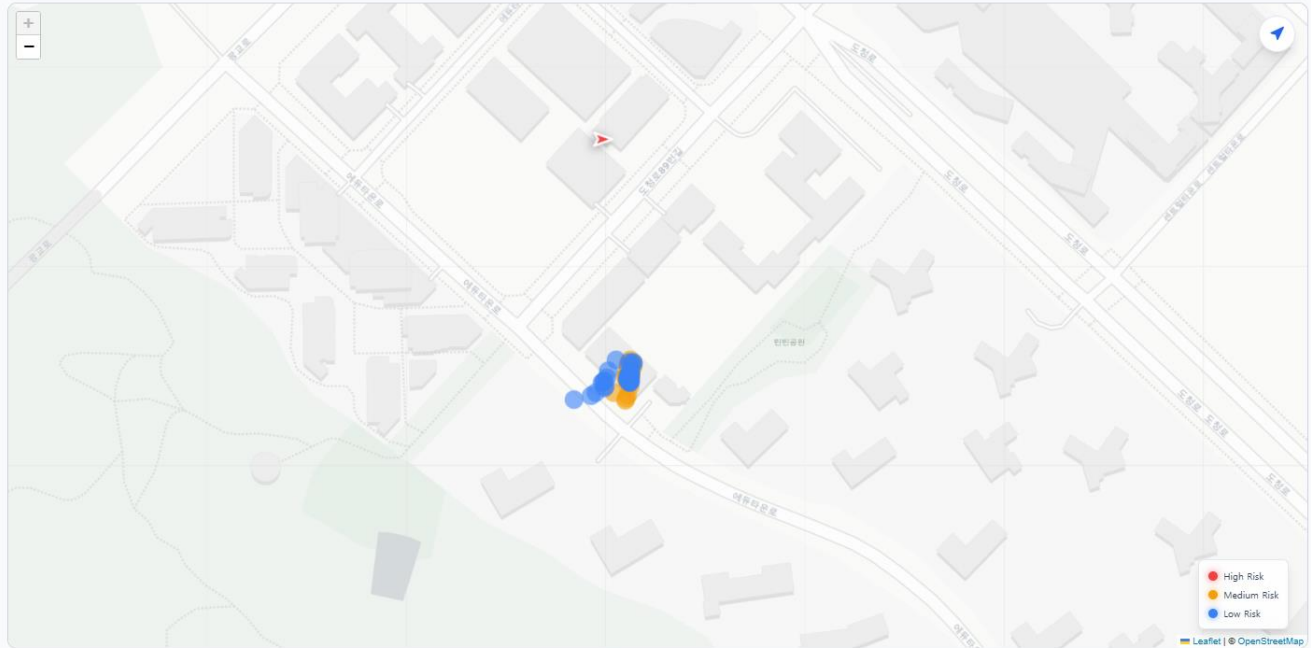
High Risk

Medium Risk

Low Risk

현재 표시된 데이터

160 건



로그아웃



Low

534f0dad-7d73-46f2-a572-894480b56f1b

경면 근접 문제 감지X

2026.02.24

오후 1:05:02

위도: 37.288735, 경도: 127.047427

New



Low

0e7ef5e1-8c2b-4ad7-9053-4f756189260b

경면 근접 문제 감지X

2026.02.24

오후 1:04:54

위도: 37.288752, 경도: 127.047432

New

백엔드 파이프라인

Backend Architecture & Data Pipeline



엔터프라이즈급 로깅 시스템

Enterprise-Grade Logging Architecture

🔗 상용 서비스 수준의 운영 안정성 및 가시성 확보

📄 데일리 로그 로테이션

날짜별 자동 파일 분리로 서버 용량 및 읽기 속도 최적화

```
app.log.2026-02-19
app.log.2026-02-20
app.log (현재)
```

매일 자정 자동 아카이빙, 메인 파일은 **현재 로그만** 기록

용량/속도 최적화

🌐 브라우저 실시간 모니터링

/logs **엔드포인트** 개설로 터미널 접속 없이 즉시 확인
 프론트엔드 개발자도 **백엔드 권한 없이** API 상태 체크 가능
 부서 간 양방향 디버깅 및 협업 속도 비약적 단축

협업 가속화

📈 트래픽 추적

Request Lifecycle 추적: 시작(▶) ~ 완료(◀)
 Device / API 경로 / HTTP 코드 / **Latency (ms)** 단위 기록
 병목 현상 파악 및 서버 성능 최적화 정량적 지표 확보

성능 최적화

The screenshot shows a file explorer on the left with a red box highlighting the 'logs' directory containing 'app.log' and three rotated log files: 'app.log.2026-02-19', 'app.log.2026-02-20', and 'app.log.2026-02-21'. To the right, a terminal window displays a log of HTTP requests with their lifecycle details:

Line	Timestamp	Level	Message	Latency	Device
24	2026-02-23 10:51:10,381	INFO	[요청 완료]	200 소요시간: 165.95ms	
25	2026-02-23 10:52:32,608	INFO	[요청 시작] POST /api/v1/navigation/path/		Device: WalkMate-App
26	2026-02-23 10:52:32,736	INFO	[요청 완료]	200 소요시간: 128.51ms	
27	2026-02-23 10:52:42,402	INFO	[요청 시작] POST /api/v1/reports/		Device: WalkMate-App
28	2026-02-23 10:52:43,559	INFO	[요청 완료]	200 소요시간: 1157.60ms	
29	2026-02-23 10:52:50,913	INFO	[요청 시작] POST /api/v1/reports/		Device: WalkMate-App
30	2026-02-23 10:52:51,337	INFO	[요청 완료]	200 소요시간: 423.73ms	
31	2026-02-23 10:52:59,425	INFO	[요청 시작] POST /api/v1/reports/		Device: WalkMate-App
32	2026-02-23 10:52:59,702	INFO	[요청 완료]	200 소요시간: 277.41ms	
33	2026-02-23 18:34:35,548	INFO	[요청 시작] GET /logs		Device: Mozilla/5.0 (Macintosh; Intel Mac OS X 1
34	2026-02-23 18:34:35,552	INFO	[요청 완료]	200 소요시간: 3.17ms	

시연 영상 및 성능

개요 팀 방법론 성과 **성과** 트러블슈팅 평가

A screenshot of a computer monitor displaying a terminal window and a web browser. The terminal window shows logs for an ngrok session, including session status, account information, and HTTP requests. The web browser shows the WalkMate Admin System dashboard, which includes a navigation menu, a header with user information, and a main content area with a table of tasks. The table has columns for checkboxes, task names, priority, ID, status, creation time, location, and actions.

<input type="checkbox"/>	방배빌	위험 등급	ID / 유형	발생일	발생 시간	위치	상태	자세히 보기
<input type="checkbox"/>		Low	55d91a59-8d2a-47e9-9564-deb63d68d510 person	2026.02.24	오후 4:52:03	위도: 37.288731, 경도: 127.047434	New	

시장 규모 분석

Market Size Analysis (TAM · SAM · SOM)

TAM
Total Addressable

약 12.8조 원 글로벌 AI 보조공학 + 스마트시티 안전 + MaaS 시장 합산

SAM
Serviceable Available

약 1.2조 원 국내 시각장애인 24.6만 + 고령 보행 약자 1,370만 + CSR/PM 견인 시장

SOM
Serviceable Obtainable

약 15.4억 원 2026 목표: PM 8사 + K-ESG 8사 + 지자체 10처 + 내비 3사

TAM 출처

Grand View Research (2025), MarketsandMarkets (2025), Precedence Research (2024)

SAM 출처

KOSIS 보건복지부 장애인현황 (2024), 국가데이터처 인구총조사 (2024)

정부 예산 출처

행정안전부 제1차 국가보행안전 기본계획 (2022~2026)

ESG 출처

한국경제인협회 & K-ESG얼라이언스 사회적 가치 보고서 (2025)

수익 구조 3대 축

Revenue Model: Three Pillars



B2B — PM 업체

견인료 방어 API

타겟

지쿠 빔 스윙 디어 8개사

월 150만 원

API 호출료 포함

ROI 포인트

킵보드 1대 견인비 약 4만 원
→ 월 38대 선제 수거 시 본전 달성



B2G — 지자체

스마트 안전 관리

타겟

서울/경기 도로교통과 NIA

단계별 요금제

Basic

월 50만 원
핵심 위험 구간 10곳 모니터링

Standard

월 150만 원
관내 전역 위험 지도 + PM 단속 연동

Premium

월 250만 원
관제 API 연동 + 민원 자동 접수



ESG — 대기업 후원

디지털 안내견 보급

타겟

삼성화재 현대차 SKT

파트너십 등급

Official

월 5천만 원
앱 내 로고 노출 + 연간 ESG 리포트

Premium

월 1억 원
케이스 브랜딩 + 보이스 광고

Visionary

월 1.5억 원+
독점 기술 협력 + 서비스 결합

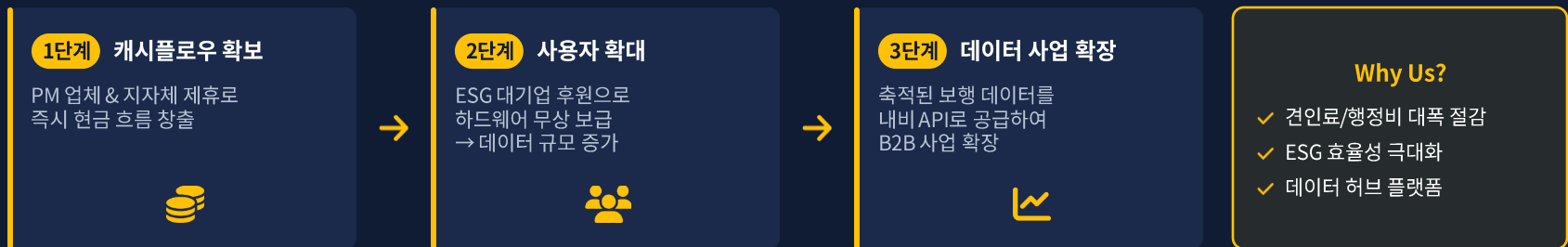
파트너십 로드맵 & SOM 달성

Partnership Roadmap & Revenue Scenario

파트너 그룹별 매출 구조



3단계 확장 로드맵



ROI 분석 & 흑자 전환

ROI Analysis & Break-Even Point Roadmap

흑자 전환 타임라인



1년차 적자 — 도입기

인프라 구축 및 AI 모델 고도화 집중 투자



2년차 확장 — 성장기

수도권 5개 구청 + 주요 PM 3사 제휴 완료 시 매출 가시화



3년차 안정기 — BEP 달성

런칭 18~24개월 내 흑자 전환 예상 (플랫폼 규모의 경제 달성)

이해관계자별 ROI



지자체 관점

75% 행정비 절감

별도 순찰 인력 운영비(연 약 1억 원) 대비 대폭 절감



PM 업체 관점

일 1.3대 견인 방어로 이익

일 평균 1.3대의 견인 리스크만 방어해도 투자 수익 발생



대기업 관점

ESG 효율성 극대화

안내견 1마리 육성비(1억 원) = 수만 명에게 즉시 서비스 제공



"사용자에게 과금하지 않습니다. 대신 그들이 만드는 안전 데이터의 가치로 수익을 창출합니다."

AI 트러블슈팅 (Part 1)

AI Model Troubleshooting: Detection Strategy & Environment

ISSUE 01

🔴 점자블록 탐지 실패

BEFORE — 문제 현상

normal 잘 잡음 → broken 적당히 잡음 → **아주 부서지면 못 잡음** → 학습 시 일반 바닥까지 오탐지 / 점자블록 없는 곳이 더 많음 / 지팡이로 이미 인지 중

↻ PIVOT — 장애물 탐지로 전환

AFTER — 해결 과정 및 결과

점자블록 포기 → **장애물 객체 탐지 모델** 개발 → 안정적 탐지 확보



ISSUE 02

🔴 로컬 환경 불일치

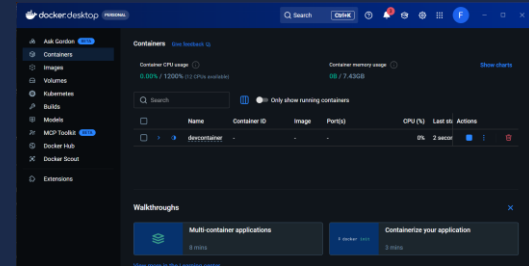
BEFORE — 문제 현상

로컬마다 다른 환경 → **매번 Pull 받아야 함** → 모델 엔지니어링 개발 지연

🐳 Docker 컨테이너 도입

AFTER — 해결 과정 및 결과

Docker 컨테이너로 환경 통일 → 일관성 확보 → 개발 효율 향상



AI 트러블슈팅 (Part 2)

AI Model Troubleshooting: Image Processing & Model Selection

ISSUE 03

🔄 Bounding Box 90도 회전 문제

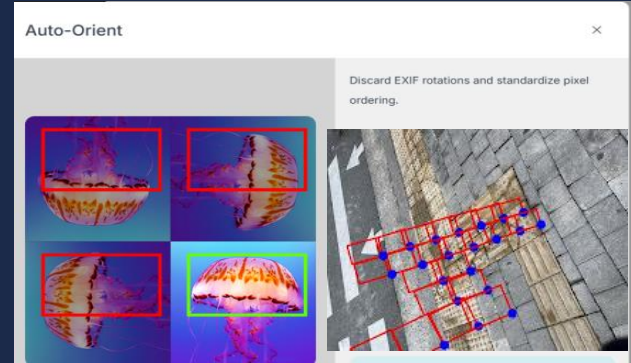
BEFORE — 문제 현상

세로 촬영 → 메타데이터 남음 → 추론은 가로사진 → jpg 합칠 때 **Bbox 좌표 그대로** + 사진만 세로 회전 → **Bbox가 90도 돌아간 것처럼 보임**

⚙️ Roboflow auto-orient 전처리

AFTER — 해결 과정 및 결과

Roboflow 다운로드 시 **auto-orient 전처리** 적용 → 메타데이터 자동 보정 → Bbox 정상 표시



ISSUE 04

🧠 모델 선택 문제

BEFORE — 문제 현상

yolo11n-obb.pt tflite 변환 미지원, **yolo11n-seg** 추론 시간 과다 예상 → 온디바이스 배포 불가

↔️ yolo26n.pt 모델로 변경

AFTER — 해결 과정 및 결과

yolo26n.pt 활용 → tflite 변환 지원 + 빠른 추론 속도 → 온디바이스 배포 성공

모델	파일 이름	작업	추론	검증	훈련	내보내기
YOLO11	yolo11n.pt yolo11s.pt yolo11e.pt yolo11l.pt yolo11x.pt	객체 탐지	✅	✅	✅	✅
YOLO11-seg	yolo11n-seg.pt yolo11s-seg.pt yolo11e-seg.pt yolo11l-seg.pt yolo11x-seg.pt	인스턴스 분할	✅	✅	✅	✅

모델	파일 이름	작업	추론	검증	훈련	내보내기
YOLO26	yolo26n.pt yolo26s.pt yolo26m.pt yolo26l.pt yolo26x.pt	객체 탐지	✅	✅	✅	✅
YOLO26-seg	yolo26n-seg.pt yolo26s-seg.pt yolo26m-seg.pt yolo26l-seg.pt yolo26x-seg.pt	인스턴스 분할	✅	✅	✅	✅

센서 & 오디오 트러블슈팅 (Part 1)

Sensor & Audio Troubleshooting: TTS & Navigation

ISSUE 01

🔊 TTS 음성 안내 중복 및 청각 피로도

BEFORE — 문제 현상

AI가 초당 수십 프레임 감지 → **TTS 겹침** → 주변 소리 못 들음

Throttling & Priority Queue

AFTER — 해결 과정 및 결과

동일 객체 3초 차단 + **거리 기반 우선순위** → 안내 피로도 최소화

```
// 2) 통합 루프: 3초마다 촬영 -> 추론 -> 전송
// 모든 엔트를 큐에 넣기 (기존처럼 말하는 중이라고 무시하지 않음)
const safeSpeak = (text: string, isObstacle: boolean = false) => {
  if (!text) return;

  // 똑같은 멘트가 큐에 중복으로 수십 개 쌓이는 것만 방지 (★ 차단 로직)
  if (ttsQueue.current.some(m => m.text === text && m.isObstacle === isObstacle)) return;

  ttsQueue.current.push({ text, isObstacle });
  processTtsQueue();
};
```

ISSUE 02

📍 저속 보행 시 내비게이션 화살표 떨림

BEFORE — 문제 현상

저속/정지 시 GPS로 방향 못 잡음 → **화살표 지그재그 회전**

🔗 Sensor Fusion & LPF

AFTER — 해결 과정 및 결과

자기계 센서 + GPS 합성 + **LPF(alpha=0.15)** → 부드러운 방향 전환

```
// ★ [핵심] 3. 센서 퓨전 (Sensor Fusion) - GPS 기반 가속도 보정
// 제스처 히네스는 플랫폼이 심해 자이로만으로 방향이 심하게 틀어집니다.
// 따라서 사용자가 걷고 있을 때(최근 4초 이내 1.5m 이상 이동)는, 이동하는 방향(gps 궤적)을 시선 방향으로 크게 신뢰합니다.
const now = Date.now();
if (targetGpsHeading.current !== null && (now - gpsActiveTime.current) < 4000) {
  let diffGps = targetGpsHeading.current - currentHeading;
  diffGps = ((diffGps + 540) % 360) - 180;
  // GPS 궤적으로 90% 확 끌고 옵니다. (자석 매국 무시)
  currentHeading = currentHeading + (diffGps * 0.9);
  currentHeading = (currentHeading + 360) % 360;
}

// 4. 지역 통과 필터(LPF) 적용 및 360도 경계선 차단거리 보간 로직
if (lastSmoothedHeading === null) {
  lastSmoothedHeading = currentHeading; // 첫 값은 그대로 적용
} else {
  let diff = currentHeading - lastSmoothedHeading;
  diff = ((diff + 540) % 360) - 180;

  // 부드럽게 새 각도 반영 (기존 각도에 차이값의 일정 비율만 더함)
  // 걷고 있을 때 더 빠르게(0.35) 반응하고, 서 있을 때 더 부드럽게(0.15) 처리
  const dynamicAlpha = (now - gpsActiveTime.current) < 4000 ? 0.35 : 0.15;
  lastSmoothedHeading = lastSmoothedHeading + dynamicAlpha * diff;
}
```

센서 & 오디오 트러블슈팅 (Part 2)

Sensor & Audio Troubleshooting: Navigation & Accessibility

ISSUE 03

📍 체크포인트 이탈 및 스킵 자동 처리

BEFORE — 문제 현상

경로 이탈 후 다음 지점 도달 → 이전 지점 복귀 무한 루프

🔍 미래 지점 역순 탐색 (Reverse Search)

AFTER — 해결 과정 및 결과

15m 이내 미래 지점 역순 탐색 → 건너뛴 지점 자동 완료 → 최적 안내

```
// 사용자 체크포인트를 건너뛰었을(bypass/skip) 가능성을 고려하여,
// 현재 위치에서 반경 15m 이내에 들어온 가장 먼(미래의) 체크포인트를 찾습니다.
let reachedIndex = -1;

// ★ 역순 탐색 (미래 지점부터 현재 지점 방향으로 거꾸로 검사)
for (let i = routeData.length - 1; i >= nextIndex; i--) {
  const dist = getDistance(curLat, curLng, routeData[i].latitude, routeData[i].longitude);

  if (dist < 15) { // 스킵 판정을 위해 기존 10m에서 15m로 약간 완화
    reachedIndex = i; // 가장 먼 미래의 지점을 현재 도달한 지점으로 판정
    break;

    // ★ 현재 단계 완료 처리 (스킵된 단계를 포함하여 한 번에 인덱스 점프 업데이트)
    lastGuideIndex.current = reachedIndex;
  }
}
```

ISSUE 04

🦿 제로 배리어(Zero-Barrier) UX

BEFORE — 문제 현상

스크린 리더 실행 번거로움 + 오토치 종료 + 음성 인식 실패

👉 홈 더블 클릭 + 3회 터치 + 분할 버튼

AFTER — 해결 과정 및 결과

홈 버튼 2회 → 실행 / 3회 터치 → 종료 / 화면 상단='응', 하단='아니오'

```
// -----
/* 화면 상단 절반 (음성 '예' 와 동일 효과): 클릭 시 '확인(onConfirm)' 실행 */
<button
  className="flex-1 w-full outline-none focus:bg-primary/5 active:bg-primary/10
  transition-colors"
  onClick={() => { stopListening(); onConfirm(); }}
  aria-label="Confirm Destination" // 스크린 리더용 라벨
></button>

/* 화면 하단 절반 (음성 '아니오' 와 동일 효과): 클릭 시 '취소(onDeny)' 실행 */
<button
  className="flex-1 w-full outline-none focus:bg-red-500/5 active:bg-red-500/10
  transition-colors"
  onClick={() => { stopListening(); onDeny(); }}
  aria-label="Deny Destination" // 스크린 리더용 라벨
></button>
```

DB 트러블슈팅 (Part 1)

Database & Backend Troubleshooting: Logging & Error Handling

ISSUE 01

Ngrok 환경 로그 확인의 어려움

BEFORE — 문제 현상

ngrok 테스트 중 팀원이 실시간 로그를 보지 못해 디버깅 지연

ID	생성 시간	방법	x	y	w	h	카메라/장치	이벤트
44b31c0f444c0b449868179534e	13시 20분 19초	mouse	0.6461	0.8356	0.1025	0.1321	3.7m (C)	
8793304a2be47951c1a042840025450	13시 20분 16초	방탄 군갑 투척 공격X	0.0000	0.0000	0.0000	0.0000	-(C)	

Log Sharing Endpoint 구현

AFTER — 해결 과정 및 결과

FastAPI 로그 엔드포인트 추가 → 웹에서 실시간 로그 확인 가능 → 디버깅 속도 개선

```

2026-02-23 10:05:41,100 - INFO - [요청 시작] GET /mirror | Device: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7;
2026-02-23 10:05:41,104 - INFO - [요청 완료] 200 | 소요시간: 3.36ms
2026-02-23 10:05:41,289 - INFO - [요청 시작] GET /api/v1/reports/ | Device: Mozilla/5.0 (Macintosh; Intel Mac OS
Safari/537.36
2026-02-23 10:05:42,895 - INFO - [요청 완료] 200 | 소요시간: 1687.48ms
2026-02-23 10:06:02,123 - INFO - [요청 시작] GET /logs | Device: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
2026-02-23 10:06:02,125 - INFO - [요청 완료] 200 | 소요시간: 1.88ms
2026-02-23 10:19:24,484 - INFO - [요청 시작] POST /api/v1/navigation/path/ | Device: WalkMate-App
2026-02-23 10:19:24,584 - INFO - [요청 완료] 200 | 소요시간: 198.31ms
2026-02-23 10:19:33,539 - INFO - [요청 시작] POST /api/v1/reports/ | Device: WalkMate-App
    
```

Commits: 433cf50, 4b88602, 34e6f65

ISSUE 02

예외 처리 누락으로 인한 서버 크래시

BEFORE — 문제 현상

예외값 발생 시 서버 크래시 → 프론트 에러 유발

Error Handling & Soft Delete

AFTER — 해결 과정 및 결과

Exception Handling + Soft Delete 적용 → 서버 안정화 및 통신 신뢰성 향상

```

# 기존 reports.py 내부 코드 예시
from fastapi import FastAPI

app = FastAPI()

@app.post("/api/v1/reports/")
def create_report():
    # ...
    # 예외 처리 누락
    raise ValueError("Invalid report data")
    # ...

```

```

2026-02-19 17:26:20,223 - INFO - [요청 시작] POST /api/v1/reports/ | Device: Wa
2026-02-19 17:26:20,666 - INFO - [요청 완료] 200 | 소요시간: 442.91ms
2026-02-19 17:26:23,295 - INFO - [요청 시작] POST /api/v1/reports/ | Device: Wa
2026-02-19 17:26:23,757 - INFO - [요청 완료] 200 | 소요시간: 461.79ms
2026-02-19 17:26:26,215 - INFO - [요청 시작] POST /api/v1/reports/ | Device: Wa
2026-02-19 17:26:26,522 - INFO - [요청 완료] 200 | 소요시간: 306.81ms
2026-02-19 17:26:29,190 - INFO - [요청 시작] POST /api/v1/reports/ | Device: Wa
2026-02-19 17:26:29,586 - INFO - [요청 완료] 200 | 소요시간: 395.96ms
2026-02-19 17:26:32,265 - INFO - [요청 시작] POST /api/v1/reports/ | Device: Wa
2026-02-19 17:26:32,679 - INFO - [요청 완료] 200 | 소요시간: 414.10ms
2026-02-19 17:26:35,073 - INFO - [요청 시작] POST /api/v1/reports/ | Device: Wa
2026-02-19 17:26:35,425 - INFO - [요청 완료] 200 | 소요시간: 352.19ms

```

```

content={
    "detail": "Internal Server Error",
    "error": str(exc) # 개발 단계 공유를 위해 에러 메시지를 함께 전달
}
    
```

Commit: f6e18ba (에러 및 예외처리)

자체 평가

Self-Evaluation & Retrospective

8.5 / 10

★★★★★★☆☆☆

Tmap + 온디바이스 AI + 앱 완전 통합
데이터베이스 구축, 음성 UI 등 실무 활용성 높음

향후 개선 방향

🔊 골전도 이어폰을 활용한 방향 안내
음성 안내 및 골전도 이어폰을 활용한 진동을 통한 방향 안내 기능 구현

🗣️ 단방향 TTS → 양방향 대화형 안내
VAD + LLM 연동으로 자연스러운 보행 가이드 구현

목표 달성 현황

		항목	현재	목표	측정 방법
김○○	Frontend · 팀장	실시간 음성 안내/음성 인식 (STT / TTS)	95% (소음 심한 경우 작동 X)	정상 작동	위험 감지 신호 수신 후 1초 이내에 스마트폰 스피커를 통해 경고 음성 송출 및 목적지 음성 입력 가능 여부
		네비게이션	수행 완료	정상 작동	경로 설정 및 경로 수정 등 작동 여부
		관리자 UI 구현	구현 완료	구현	스마트폰에서 촬영된 사진 데이터가 관리자 페이지에 목록 형태로 보이는지 여부
이지호	Backend · Cloud	위험 데이터 자동 저장	100%	성공률 100%	위험 탐지 시 사진(S3)과 위치(DB)가 누락 없이 적재되는지 확인
		관리자 지도 시각화	수행 완료	수행 완료	수집된 위도/경도 데이터가 웹 대시보드 마커로 실시간 표시
		API 응답 안정성	100%	99% 이상	앱에서 서버로 데이터 전송 시 HTTP 201 응답 성공 비율
최○○	AI · Algorithm	방해물객체 탐지모델 개발	정상작동	정상작동	mAP50, mAP50-95
		방해물객체 위치 측정	수행완료	수행완료	실제 거리와 추정치 비교

66

보행자가 걷는 것만으로도 도시는 더 안전해집니다

PROJECT



walkmate

Guiding the Way to Safety

Team 공부많이된다

