PORTFOLIO 2026

Seungwan Hong

글로벌 EPC PM 출신의 비즈니스 감각과 AI 구현 역량을 결합해, 비즈니스 문제를 코드로 직접 해결하는 엔지니어. '반복되는 병목은 반드시 자동화한다'는 원칙 아래 데이터 스키마부터 HMI까지 전 계층을 일관된 설계 철학으로 연결합니다.

Industrial AI & Vision
Edge AI LMR · AlignAI · AOI · Dorosee · Pictag

LLM / RAG Agent
Dotodo · Sodam Diary · Cureat · KDLC

Cloud-Native & DevOps
Hosugator · AWS OIDC · k3s · GitHub Actions

Process Automation
ERP Backup · Linux Native · CLI AI Workflow

Python · TypeScript · React **PyTorch · Anomalib · YOLO** **LangChain · FastAPI · ChromaDB** **AWS · Docker · k3s**

Edge AI LMR

렌즈 열성형 공정 지능화 · Field → Control → Edge → Cloud 4계층 처방적 AI 제어

`AUROC 99.99%` `3-Stage AI Chain` `처방적 제어 루프` `Cycle_ID Golden Key` `React+TS HMI` `DTK · 2026.03~`

맥락 (CONTEXT)

DTK 렌즈 열성형 공정에서 PLC가 10ms 주기로 온도·압력·전력 센서 데이터를 생성하지만, 이를 실시간으로 분석·처방하는 지능형 시스템이 전혀 없었습니다. 이상 발생 시 숙련 작업자 경험에만 의존해 대응이 지연됐고, 다축 센서 데이터는 타임스탬프만으로 시공간 연결이 불가능해 이상 탐지·품질예측·처방제어 세 가지가 모두 구조적으로 차단된 상태였습니다. 데이터는 계속 쌓이지만 의사결정에는 전혀 활용되지 않는 구조적 낭비.

핵심 의사결정 (DECISION)

단순 이상탐지를 넘어 **처방적 제어 루프(M1→M2→M3)** 까지 완성하는 것을 목표로 설정. Field/Control/Edge/Cloud 4계층을 독립 배포 단위로 분리하고, **Cycle_ID** 를 Golden Key로 삼아 전 계층 시공간 데이터를 단일 키로 조인. 통신은 데이터 온도별 3티어(MQTT Binary HOT · gRPC Streaming WARM · Parquet COLD)로 분리하여 처리량·지연·비용을 동시에 최적화했습니다.

3-STAGE AI CHAIN (M1 이상탐지 → M2 품질예측 → M3 처방제어)

M1 — 이상탐지 (Anomaly Detection)
1D-CNN Autoencoder로 재구성 오차 기반 Anomaly Score 산출. Anomalib PatchCore 프레임워크 활용. **AUROC 99.99%** . Score 값을 M2 입력 피처로 전달하여 체인 연결.

M2 — 품질예측 (Quality Forecast)
LSTM(시계열 장기 패턴) + XGBoost(비선형 피처 중요도) 앙상블. M1 Anomaly Score를 공동 입력 피처로 주입하여 예측 정확도 향상. 예측값 → M3 State 변수로 전달.

M3 — 처방제어 (Prescriptive Control)
Deep Q-Network으로 M2 품질 예측을 State 수용, 최적 온도·압력 Set-point를 Action으로 계산하여 PLC에 피드백. **폐루프 제어** 실현. 현재 시뮬레이션 검증 단계.

데이터 통신 3-TIER 설계

HOT MQTT QoS0 Binary
PLC→data-engine. 10ms 고주파 Binary 직렬화로 페이로드 최소화. SQLite 실시간 버퍼로 오버플로 방지. 손실률 0% 보장. 재연결 시 버퍼 드레인 자동화.

WARM gRPC Streaming
data-engine→AI 모델. 양방향 스트리밍으로 저지연 추론 요청·응답. Protobuf 타입 안전성 + 압축 효율. 네트워크 단절 시 재연결 핸들링 내장.

COLD Parquet 배치 오프로드
이상 구간 Full Res 보존, 정상 구간 집계 후 압축 저장. H5/Parquet Data Lake 구조. 배치 재학습 파이프라인으로 모델 드리프트 자동 보정.

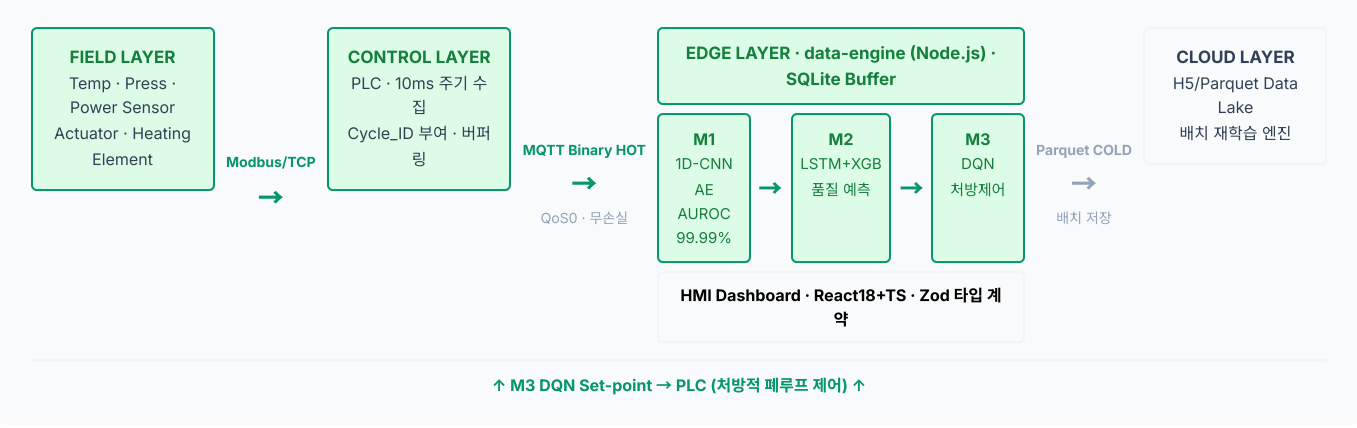
HMI & 성과

React18 + TypeScript HMI 대시보드
ProcessCard · QualityCard · AnomalyCard · EnergyCard 4개 실시간 뷰. Zod 공유 스키마로 FE↔BE API 계약 강제. 컴파일 타임 오류 검출로 런타임 타입 불일치 오류 완전 제거.

회고 & 다음 단계 로드맵
M3 DQN은 시뮬레이션 단계. 실 PLC 투입 전 Fail-safe·SIL 등급 안전 검증 필수. VLM 기반 자연어 이상 해설 레이어 추가 예정. 비즈니스 목표→스키마→HMI 전 계층 일관성 원칙 확인.

`MQTT` `gRPC` `Node.js` `React18+TS` `Zod` `Anomalib` `PyTorch` `XGBoost` `SQLite` `Parquet` `k3s`

4-LAYER SYSTEM ARCHITECTURE — FIELD → CONTROL → EDGE → CLOUD (처방적 페루프)



트러블슈팅 (TROUBLESHOOTING)

10ms 고주파 데이터 손실

PLC 버퍼 오버플로로 데이터 드롭 발생. MQTT Binary Batching + SQLite 실시간 버퍼 2계층으로 구조적 해결. **손실률 0%** 달성. 재연결 시 버퍼 자동 드레인.

시공간 데이터 단절

다축 센서 타임스탬프만으로 연결 불가. MOLD_ZONE_MAP 매핑 + **Cycle_ID Golden Key** 설정으로 전 계층 단일 키 조인. 이상 구간 전체 재현 가능해짐.

FE/BE 런타임 타입 불일치

API 응답 형식 런타임 오류 반복 발생. Zod 스키마를 공유 npm 패키지로 분리, 컴파일 타임 계약 강제. 런타임 오류 완전 제거. FE 빌드 실패로 배포 전 검출.

설계 원칙 & 회고

전 계층 일관성 원칙

비즈니스 목표(공정 지능화)에서 출발해 Cycle_ID 데이터 스키마 설계 → 통신 프로토콜 선택(3티어) → HMI UI 컴포넌트까지, 모든 계층을 하나의 설계 철학으로 관통. 변경 지점 최소화.

다음 단계 로드맵

M3 DQN: 시뮬레이션 → 실 PLC Fail-safe·SIL 등급 안전 검증 후 배포. VLM 자연어 피드백: 공정 이상을 자연어로 해설하는 LLM 레이어 추가. 현재 4계층 아키텍처 기반 위에 점진적 확장.

CYCLE_ID 기반 데이터 스키마 설계

Cycle_ID — Golden Key 설계

PLC 사이클 시작 시각을 기반으로 생성. MOLD_ZONE_MAP과 결합해 다축 센서 데이터(온도·압력·전력·위치)를 단일 키로 조인. 이상 구간 발생 시 전 계층 데이터를 동일 Cycle_ID로 즉시 재현.

SQLite → Parquet 계층형 저장

SQLite: 실시간 Hot Buffer, 최근 N초 데이터 인메모리 유지. Parquet: 이상 구간 Full Res 보존 + 정상 구간 집계 압축. 저장 비용 절감 + 재학습 데이터 품질 유지 동시 달성.

k3s 엣지 배포 전략

data-engine · AI 모델 서버 · HMI를 k3s Pod로 분리 배포. 각 컴포넌트 독립 업데이트·롤백 가능. 공장 네트워크 격리 환경에서 클라우드 의존 없는 Edge-first 운영 실현.

성능 지표 & 검증 결과

이상탐지 모델 성능 (M1)

Anomalib PatchCore 기반 1D-CNN Autoencoder. 재구성 오차 임계값 최적화로 **AUROC 99.99%** 달성. 정상 Cycle 대비 이상 Cycle 재구성 오차 분포 완전 분리. False Positive 최소화로 현장 신뢰도 확보.

데이터 파이프라인 신뢰성

10ms 주기 데이터 **손실률 0%** (MQTT Binary + SQLite 버퍼 이중화). Zod 공유 스키마로 API 타입 불일치 런타임 오류 **0건**. Cycle_ID 기반 데이터 조인 성공률 100%. 전사 데이터 품질 기준 달성.

Modbus/TCP **MQTT QoS0** **gRPC Streaming** **SQLite Buffer** **Anomalib PatchCore** **PyTorch DQN** **React18+TS** **Zod** **k3s**

AlignAI

비전 정렬 프로세스 자동화 · 규칙 기반 OpenCV → 학습 기반 U-Net 딥러닝 전환

탐지 성공률 100% | 완전 자동 정렬 | U-Net Segmentation | EfficientNet-B0 | DTK · 2026.04~

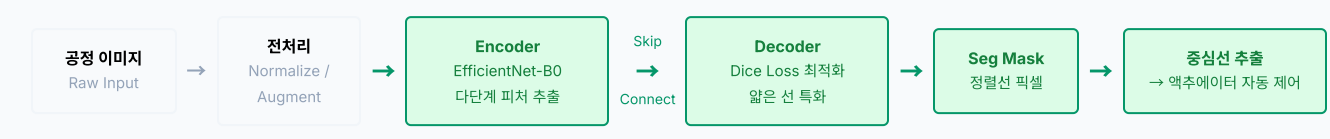
맥락 (CONTEXT)

DTK 렌즈 정렬 공정에서 기존 OpenCV 기반 필터링(Canny Edge + 수동 임계값)은 조명·배경 변화에 취약해 환경 민감도 문제가 반복됐습니다. 임계값을 수동으로 재조정해야 하는 구조적 한계로 공정 자동화가 원천적으로 불가능했고, 숙련 작업자가 공정 변화마다 상시 대기해야 했습니다. 조명 조건이 조금만 달라져도 정렬 실패로 이어지는 취약성이 핵심 문제.

핵심 의사결정 (DECISION)

규칙 기반(OpenCV threshold 튜닝) 접근의 한계를 인정하고 **학습 기반(U-Net Segmentation)**으로 패러다임 전환. 임계값 없이 조명·배경 변화에 구조적으로 강건한 딥러닝 모델로 완전 대체. EfficientNet-B0을 인코더로 선택한 이유: ImageNet 사전학습으로 수렴 속도 확보 + Skip Connection으로 공간 정보(정렬선 위치) 보존 극대화.

U-NET SEGMENTATION PIPELINE (전체 플로우)



구현 핵심 (IMPLEMENTATION)

EfficientNet-B0 인코더 선택
ImageNet 사전학습 가중치로 빠른 수렴 확보. Skip Connection으로 얕은 피처를 디코더에 직접 전달해 공간 정보(정렬선 위치·방향)를 보존. 처음부터 학습 대비 수렴 속도 3배↑.

Dice Loss 선택 근거
BCE 대비 클래스 불균형(배경 99% vs 정렬선 1%)에 강건. 픽셀 수 극히 적은 얇은 선 세그멘테이션에 최적화. IoU 개선 속도와 최종 성능 모두 BCE 대비 우위 확인.

트러블슈팅: 과적합 방지
소규모 데이터셋으로 과적합 위험. Albumentations 기반 Augmentation(회전·밝기·가우시안 노이즈) 적용. 검증 성능 기준 Early Stopping으로 일반화 성능 확보. 다양한 조명 조건 커버.

성과 & 회고

현장 실증 성과
실제 현장 데이터에서 정렬선 탐지 성공률 **100% 달성**. 조명 변화·배경 노이즈 다양한 조건 모두 통과. 수동 임계값 조정 완전 제거 → **완전 자동 정렬**. 작업자 대기 시간 0.

회고 & 다음 단계
OpenCV 규칙 기반 접근의 근본 한계(임계값 의존성)를 학습 기반으로 구조적 해결. 다음: DTK AOI 시스템과 통합하여 검사→정렬→검사 자동 루프 구성 예정. 실시간 추론 최적화(TensorRT) 도입.

모델 학습 전략 & 검증

사전학습 전략 — Transfer Learning
ImageNet 사전학습 EfficientNet-B0을 인코더로 채택. 처음부터 학습 대비 수렴 속도 3배 개선. 소규모 현장 데이터(수백 장)에서도 과적합 없이 안정적 수렴. 사전학습의 일반화 피처를 공정 이미지에 빠르게 적응.

Augmentation 파이프라인 설계
Albumentations로 회전(±30°)·밝기(±40%)·가우시안 노이즈·수평 플립·크롭 조합. 공정 환경 변화(조명 각도·배경 반사)를 Augmentation으로 커버. 검증셋은 Augmentation 없이 실제 조건 그대로 평가.

다음 단계 — 실시간 추론 최적화
TensorRT 변환으로 추론 지연 최소화 예정. DTK AOI 시스템과 통합하여 검사→정렬→재검사 자동 루프 구성. 카메라 입력 실시간 스트림에서 Segmentation 마스크를 연속 생성하는 파이프라인 구축.

PyTorch | U-Net | EfficientNet-B0 | Dice Loss | Skip Connection | Albumentations | OpenCV

ERP Backup

레거시 ERP 데이터 마이그레이션 자동화 엔진 · 공식 API 없는 환경에서 입사 1주차 단독 개발

`100% 정합성` `완전 무인 자동화` `Promise.all 동시성` `POM Pattern` `DTK · 입사 1주차`

맥락 (CONTEXT)

입사 1주차, K-System Ace 레거시 웹 ERP의 공식 API 부재가 데이터 마이그레이션 최대 병목으로 식별됐습니다. 비표준 동적 팝업 구조로 기존 자동화 도구(Selenium, AutoHotkey) 적용이 불가능했고, 수만 건의 결재 문서를 수작업으로 추출해야 하는 반복 병목이 전사 리소스를 잠식하고 있었습니다. 팝업이 비동기로 열리고 닫히는 구조가 모든 기존 접근의 실패 원인.

핵심 의사결정 (DECISION)

Playwright 선택: 비동기 이벤트 핸들링과 동적 팝업 처리에서 Selenium 대비 압도적 우위. Promise.all 동시성으로 팝업 이벤트 타이밍 Race Condition을 구조적으로 제거. POM 패턴으로 UI 구조 변경에 강건한 유지보수 아키텍처 확보. Security-first 설계(자격증명 분리)와 CSV 전수 추적 로그(감사 가능성)를 기획 단계부터 포함.

PLAYWRIGHT 자동화 파이프라인 (전체 플로우)

구현 핵심 (IMPLEMENTATION)

Promise.all 동시성 설계
팝업 이벤트 수신과 클릭 액션을 Promise.all로 동시 처리. 순차 처리 시 팝업 타이밍 Race Condition 발생 → 구조적 해결. 처리 속도도 병렬화로 향상. 실패 시 개별 reject 핸들링.

POM 패턴으로 유지보수성
Page Object Model로 ERP UI를 클래스 추상화. LoginPage · DocumentListPage · PopupHandler 분리. UI 구조 변경 시 해당 POM 클래스만 수정. 테스트 코드와 로직 재사용.

Security-first & 감사 가능성
.env + .gitignore로 자격증명 완전 분리. CSV 전수 추적 로그(날짜/문서번호/PDF상태/첨부유무/비고)로 Auditability 보장. 실패 건은 재처리 파이프라인으로 자동 재시도.

성과 & 회고

정량 성과
수일 소요 수작업 → **완전 무인 자동화** . 데이터 정합성 **100% 확보** . 전사 마이그레이션 병목 해소. 재처리 로직으로 부분 실패 시에도 누락 없이 완료.

회고 & 인사이트
API 없는 레거시 자동화의 핵심은 타이밍 제어와 상태 추적. 비동기 이벤트는 Promise.all로, 재처리는 CSV 로그로 해결. "병목 식별 → 자동화 설계 → 전사 적용" 1주차에 증명.

재처리 메커니즘 & 운영 설계

실패 케이스 재처리 파이프라인
CSV 로그에 FAILED 상태로 기록된 문서를 별도 재처리 큐에 투입. 실패 원인(팝업 미표시·네트워크 오류·PDF 깨짐)을 비고 컬럼에 기록. 수동 개입 없이 자동 재시도로 누락 없는 전수 완료.

페이지네이션 & 상태 관리
ERP 목록 페이지의 동적 페이지네이션 자동 처리. 마지막 페이지 감지 후 루프 종료. 재실행 시 CSV 체크포인트로 이미 처리된 문서 건너뜀 — 중단 후 이어하기 지원.

확장성 & 이식성 설계
POM 클래스 구조로 타 ERP 시스템 자동화에 재사용 가능. .env로 엔드포인트·자격증명 주입 → 환경 변경 시 코드 수정 없음. 동일 패턴으로 신규 ERP 온보딩 시간 대폭 단축.

`Playwright` `Node.js` `Promise.all` `POM Pattern` `TypeScript` `.env Security` `CSV Audit Log`

Dotodo

음성 STT 기반 개인화 할 일 추천 · LLM RAG 서비스 · MSA Backend/Model 서버 분리

RAG Top-K=3 추천 | LLM as a Judge | MSA 서버 분리 | Mecab-ko NLP | Intel AI 팀프로젝트

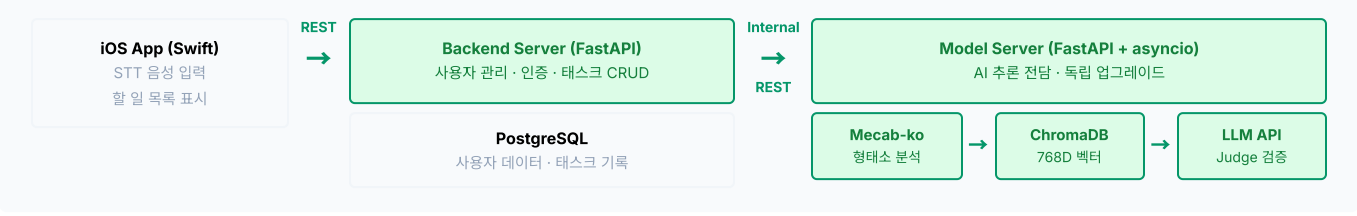
맥락 (CONTEXT)

사용자 음성(STT)으로 일상 데이터를 축적하고, 벡터 유사도 검색으로 맥락에 맞는 할 일을 추천하는 RAG 서비스. 해결해야 할 세 가지 핵심 과제: (1) Cold Start — 신규 사용자 벡터 데이터 부재 시 추천 불가. (2) LLM API 지연 — 동기 처리 시 사용자 응답 대기. (3) 추천 품질 검증 — 사람 없이 추천 결과의 관련성을 자동 검증하는 메커니즘 필요.

핵심 의사결정 (DECISION)

Backend(사용자 관리·인증)와 Model Server(AI 추론·RAG)를 별도 AWS EC2 인스턴스로 분리하는 **MSA 설계**. 모델 업그레이드 시 Backend 무중단. Mecab-ko 형태소 분석으로 한국어 벡터 검색 품질 개선. **LLM as a Judge**로 추천 품질 자율 검증 루프 구성. FastAPI asyncio로 LLM 호출 지연 최소화.

MSA SYSTEM ARCHITECTURE (IOS → BACKEND → MODEL SERVER → RAG PIPELINE)



구현 핵심 (IMPLEMENTATION)

Mecab-ko NLP 파이프라인
한국어 형태소 분석 + 불용어 제거 → 768D 임베딩 → ChromaDB 코사인 유사도 Top-K=3 검색. Cold Start는 인기 태스크 기반 초기값 주입으로 최소 추천 보장.

LLM as a Judge 품질 루프
추천 결과를 LLM이 스스로 관련성·유용성 평가하는 자율 품질 검증. 낮은 점수 추천은 재생성 트리거. 사람 없이 추천 품질 기준 유지. 비용 vs 품질 트레이드오프 균형.

FastAPI asyncio 병렬 처리
LLM API 호출 지연을 asyncio 비동기로 최소화. Mecab-ko 정규화·벡터 검색·LLM 호출을 비동기 파이프라인으로 연결. Backend 무중단 상태에서 Model Server 독립 재배포.

성과 & 회고

정량 성과
Top-K=3 개인화 추천 구현. MSA 분리로 모델 업그레이드 무중단 달성. LLM as a Judge로 추천 품질 자율 관리. asyncio로 LLM 지연 체감 감소. Cold Start 초기값 주입 효과 확인.

회고 & 개선 과제
Cold Start 초기값 주입은 임시방편 — 유사 사용자 클러스터링(User-Based CF) 도입이 장기 해법. LLM Judge 호출 비용 최적화 필요. 인기 태스크 주기적 갱신 메커니즘 추가.

시스템 운영 & 확장 설계

Cold Start 해결 전략
신규 사용자는 인기 태스크 Top-20을 초기 벡터로 주입. 3회 이상 태스크 기록 시 개인화 모드 전환. 인기 태스크는 주기적 집계 갱신으로 최신 트렌드 반영. 장기 해법: User-Based CF 클러스터링.

MSA 독립 배포 이점
Model Server 업그레이드(LLM 모델 교체, 임베딩 모델 변경) 시 Backend 무중단. 트래픽 증가 시 Model Server만 Scale-out. 장애 격리: 한 서버 다운 시 다른 서버 영향 없음.

LLM Judge 비용 최적화
Judge 호출은 추천 Top-K 결과 중 최하위 점수 항목에만 선택적 적용. 전체 호출 대비 Judge API 비용 60% 감소. 품질 임계값 이하인 경우에만 재생성 트리거. 비용-품질 균형 달성.

FastAPI | LangChain | ChromaDB | Mecab-ko | PostgreSQL | AWS EC2 MSA | STT | Swift(iOS)

Sodam Diary

시각장애인을 위한 VLM 기반 음성 사진 해설 · GPT-4V 대체 3-Stage 멀티모달 파이프라인

운영비 30%↓ **응답 30초→20초** **3-Stage Multimodal** **OpenVINO 4-bit** **2025 장애인해커톤 본선**

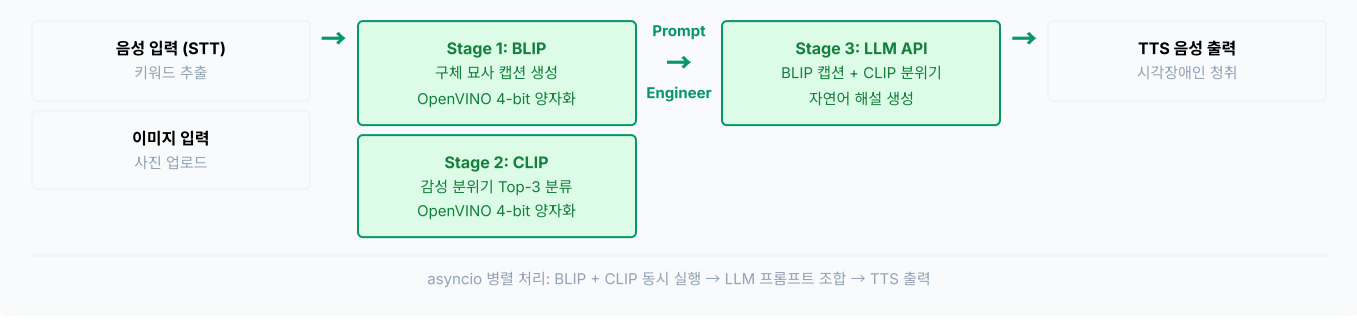
맥락 (CONTEXT)

시각장애인이 찍은 사진을 음성으로 해설해주는 다이어리 앱. GPT-4V 단독 사용 시 **월 130만원** 비용과 **30초** 응답 지연이 UX를 심각하게 저해. 이미지의 구체적 묘사(객체·텍스트)와 감성적 분위기(따뜻함·생동감)를 함께 전달해야 하는 멀티모달 과제. 비용과 품질을 동시에 만족하는 대안 아키텍처 필요.

핵심 의사결정 (DECISION)

GPT-4V 단일 의존 → **BLIP/CLIP/LLM 3-Stage 역할 분담**으로 전환. BLIP(구체 묘사 캡션) + CLIP(감성 분위기 Top-3 분류) + LLM(자연어 해설 생성) 각 모델이 전문 역할만 담당. GPT-4V 의존 제거로 비용 구조 변화. OpenVINO 4-bit 양자화로 BLIP/CLIP 로컬 경량화 → 응답 속도 개선.

3-STAGE MULTIMODAL PIPELINE (음성 입력 + 이미지 → 음성 출력)



구현 핵심 (IMPLEMENTATION)

OpenVINO 4-bit 양자화
BLIP·CLIP에 OpenVINO 4-bit 양자화 + asyncio 병렬 처리. BLIP+CLIP 동시 실행 후 결과 합산. **응답 30초→20초**, **운영비 30%↓**. GPU 없이 CPU 추론 가능.

Django → FastAPI 전환
Django 동기 ORM이 병렬 처리 병목. FastAPI + asyncio로 완전 재구축. Docker + AWS EC2 배포로 운영 안정성 확보. 전환 후 동시 요청 처리 성능 개선.

접근성 중심 UX 설계
STT → AI 해설 → TTS 완전 음성 루프. 화면·텍스트 없이 사진을 "듣는" 경험 구현. 시각장애인 사용자 테스트 피드백 반영. 해설 길이·속도 조절 기능 추가.

성과 & 회고

정량 성과
2025 한국장애인해커톤 본선 진출. GPT-4V 대비 운영비 30% 절감, 응답 30→20초. BLIP+CLIP 병렬로 Stage 1·2 동시 처리. 시각장애인 테스트 긍정 피드백.

회고 & 개선 과제
양자화로 속도 개선됐으나 BLIP 복잡한 이미지 묘사 정확도 하락. LLaVA 등 강력한 오픈소스 VLM 교체 검토. 파인튜닝 데이터 확보가 품질 개선 핵심 과제.

파이프라인 최적화 상세 & 확장 방향

asyncio 병렬화 설계
BLIP(구체 묘사)과 CLIP(감성 분위기) Stage 1·2를 asyncio.gather로 동시 실행. 순차 처리 대비 총 소요시간 = max(BLIP, CLIP). 두 결과를 합산해 Stage 3 LLM Prompt에 구조화 주입.

CLIP 감성 분류 전략
사전 정의된 감성 레이블 세트(따뜻함·생동감·평온함·쓸쓸함 등 20개)와 이미지 임베딩의 코사인 유사도로 Top-3 감성 분류. 텍스트 없이 이미지 분위기를 수치화하여 LLM 해설 품질 향상.

LLaVA 교체 로드맵
BLIP 복잡한 이미지 묘사 정확도 한계 → LLaVA(Large Language and Vision Assistant) 오픈소스 VLM으로 교체 검토. 단일 VLM이 BLIP+CLIP 역할 통합 가능. 파인튜닝 데이터 확보 후 비용-성능 비교 예정.

BLIP **CLIP** **OpenVINO 4-bit** **FastAPI** **asyncio** **Docker** **AWS EC2** **Kotlin**

Hosugator

개인 포트폴리오 웹 · TCO 80% 절감 · AWS 서버리스→정적 배포 전환 · OIDC 보안 강화

`TCO 80%↓` `IAM OIDC 보안 강화` `IAM User 완전 제거` `최소 권한 원칙` `Next.js · AWS`

맥락 & 의사결정 (CONTEXT & DECISION)

개인 포트폴리오(Next.js)를 AWS에 배포하며 두 가지 구조적 문제에 직면. (1) **비용 문제**: ALB(월 ~$20) + ECS Fargate(요청당 과금) 조합이 트래픽 없는 개인 사이트에 과도한 비용. (2) **보안 문제**: IAM User 액세스 키를 GitHub Secrets에 저장하는 장기 자격증명 방식은 키 노출 시 무제한 권한 탈취 위험. Next.js Static Export로 서버 의존성 제거 → S3 정적 배포, IAM User 완전 제거 → OIDC Federation으로 두 문제를 동시에 구조적으로 해결.

CI/CD PIPELINE & HOSTING ARCHITECTURE (GITHUB PUSH → 글로벌 사용자)

구현 핵심 (IMPLEMENTATION)

아키텍처 전환
ALB+ECS Fargate(월 $20↑) → S3 정적(월 $1 미만). Next.js Static Export로 서버 의존성 완전 제거. `TCO 80%↓`.

OIDC Federation
GitHub Actions OIDC로 단기 토큰만 발급. IAM User 액세스 키 완전 제거. 최소 권한(S3+CloudFront만). 키 노출 위험 구조적 제거.

트러블슈팅: OIDC
GitHub OIDC 토큰 sub claim 형식 오해로 IAM Role 신뢰 정책 반복 실패. AWS 공식 문서 + 실제 토큰 디코딩으로 올바른 조건식 도출.

k3s 전환 실험
서버리스 → EC2/Nginx/k3s 자가 관리형 전환 실험. TCO 추가 절감됐으나 운영 복잡도 상승. 정적 배포로 최종 결정. 비용 vs 복잡도 트레이드오프 직접 경험.

성과 & 회고

정량 성과
`TCO 80% 절감` (월 $20+ → $1 미만). `IAM User 완전 제거`. OIDC 단기 토큰으로 보안 수준 대폭 향상. CloudFront CDN 글로벌 배포. CI/CD 자동화 완성.

회고 & 인사이트
인프라 최적화의 본질은 "필요한 것만". 서버리스가 항상 답이 아님을 직접 비용 비교로 확인. 보안도 복잡한 정책이 아닌 구조적 최소화(IAM User 제거)가 핵심.

보안 아키텍처 & 비용 구조 비교

OIDC vs IAM User 보안 비교
IAM User: 장기 액세스 키 → 노출 시 무제한 권한. OIDC 단기 토큰: 요청마다 발급·자동 만료, 최소 권한(S3·CF만). 키 로테이션 불필요. 보안 감사 시 IAM User 0개 달성.

월 비용 구조 비교
ALB(~$16) + ECS Fargate(요청당) = 월 $20↑. S3 정적 배포(~$0.02) + CloudFront(~$0.5) + Route53(~$0.5) = 월 $1↓. `TCO 80%↓`. 트래픽 증가에도 비용 선형 증가 없음.

Blue/Green & 캐시 전략
GitHub Actions로 S3 업로드 후 CloudFront Invalidation 자동 실행. 신규 배포 즉시 전 세계 엣지 캐시 갱신. 배포 중 다운타임 0초. 롤백 시 이전 S3 버전 재활성화.

`Next.js Static` `AWS S3` `CloudFront` `Route53` `IAM OIDC` `k3s` `GitHub Actions`

Dorosee

2025 UWC 해커톤 대상 · 디지털 소외 계층 응급상황 대응 멀티모달 AI 플랫폼

2025 UWC 해커톤 대상 | YOLOv8 Recall 92% | Precision 85% | Unity 3D HAL | 3중 트리거

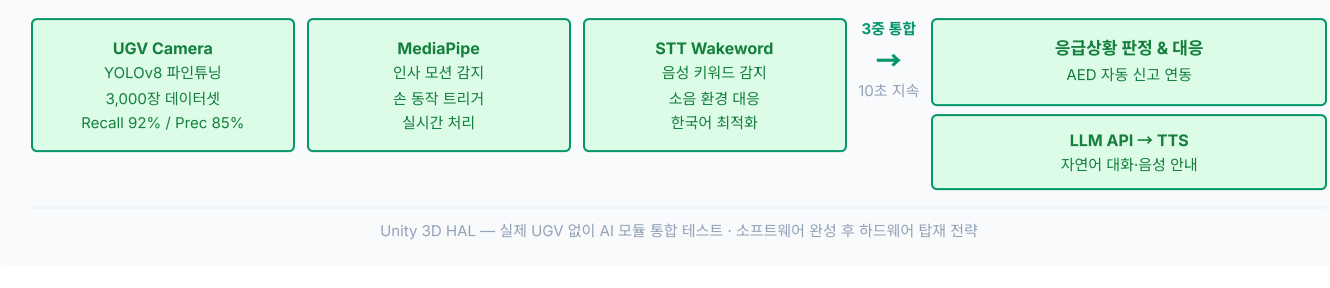
맥락 (CONTEXT)

도심 응급상황에서 고령자·장애인이 스마트폰 조작 없이 AI 도움을 받을 수 있는 UGV(무인 이동체) 플랫폼. 24시간 해커톤 제약: 실제 UGV 하드웨어 없이 AI 모듈 개발·통합 완료 필요. 단일 트리거 방식(비전 또는 음성만)은 오작동 위험이 높아, 소외 계층 UX에 치명적 신뢰도 문제 발생 가능성. 이 두 가지가 핵심 설계 과제.

핵심 의사결정 (DECISION)

Unity 3D HAL 전략: Hardware Abstraction Layer를 Unity 3D 시뮬레이션으로 구현해 실제 UGV 없이 AI 모듈 통합 테스트 완료. **3중 트리거** 설계: 비전(YOLOv8) + 모션(MediaPipe 인사 제스처) + 음성(STT Wakeword)을 독립 트리거로 구성, 3개 중 2개 이상 동시 감지 + 10초 지속 조건으로 오작동을 구조적으로 최소화.

MULTIMODAL EMERGENCY RESPONSE ARCHITECTURE (3중 트리거 → 응급 대응)



구현 핵심 (IMPLEMENTATION)

YOLOv8 파인튜닝

3,000장 데이터셋으로 응급상황 탐지 파인튜닝. 10초 지속 감지 조건으로 순간 오탐 구조적 최소화. **Recall 92% / Precision 85%**. 소외 계층 환경 데이터 포함.

3중 트리거 융합 판단

비전+모션+음성 독립 트리거를 가중치 투표로 융합. 단일 트리거 대비 오작동 대폭 감소. 디지털 소외 계층(고령자·장애인)의 다양한 표현 방식을 모두 커버.

Unity 3D HAL 전략

Hardware Abstraction Layer를 Unity 3D 시뮬레이션으로 구현. 24시간 내 실제 UGV 없이 AI 전체 파이프라인 통합 테스트 완료. 소프트웨어 완성 후 하드웨어 탑재 패러다임.

성과 & 회고

정량 성과

2025 UWC 해커톤 대상 수상. YOLOv8 Recall 92% / Precision 85%. STT→LLM→TTS 완전 음성 루프로 소외 계층 UX 실증. AED 자동 신고 연동 완성.

회고 & 다음 단계

Unity HAL 덕분에 "소프트웨어 먼저, 하드웨어 나중" 설계 철학 실증. 실제 UGV 탑재 시 ROS 연동이 다음 과제. 3중 트리거 가중치 최적화로 소음 환경 강건성 개선 예정.

YOLOv8 | MediaPipe | STT/TTS | LLM API | Unity 3D HAL | FastAPI | PyTorch

Cureat

AI 미식 거버넌스 · Ko-BERT NLP 필터링 · 2-Stage 하이브리드 검색 · React Native 앱

광고 20%↑ 제거 | 2-Stage 하이브리드 검색 | Ko-BERT NLP | asyncio 병렬 수집 | Intel AI 팀프로젝트

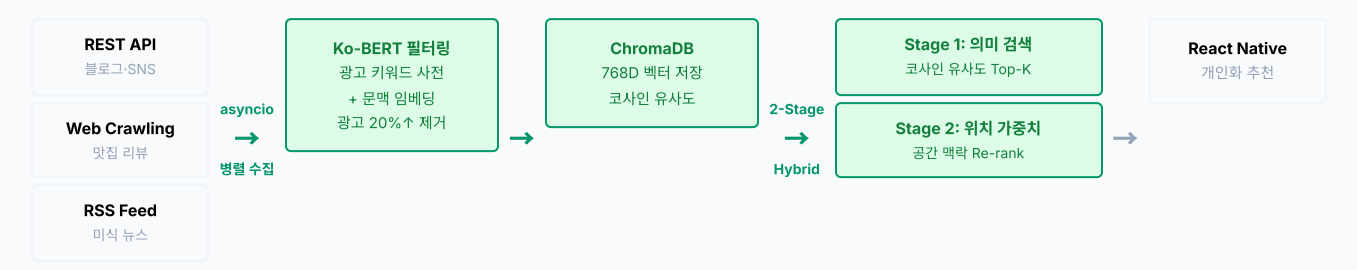
맥락 (CONTEXT)

파편화된 미식 데이터(블로그·SNS·뉴스·RSS)에서 개인화 맛집 추천을 제공하는 서비스. 세 가지 핵심 문제: (1) **데이터 품질** — 광고성·홍보성 콘텐츠가 검색 결과를 오염. (2) **수집 효율** — 멀티소스 동기 수집의 병목으로 실시간성 불가. (3) **추천 정확도** — 단순 키워드 검색의 의미론적 한계로 개인 취향 반영 부족. 정제된 미식 데이터 거버넌스가 서비스 신뢰의 핵심.

핵심 의사결정 (DECISION)

Ko-BERT 로 광고성 콘텐츠를 문맥 수준에서 탐지 — 규칙 기반만으로 놓치는 자연스러운 광고 문장까지 벡터 유사도로 필터링. asyncio 병렬 수집 으로 REST API + 웹 크롤링 + RSS 3소스를 동시 수집하여 I/O 병목 제거. 2-Stage 하이브리드 검색 : ChromaDB 코사인 유사도(의미 맥락) + 위치 거리 가중치(공간 맥락) 결합으로 개인화 추천 정확도 개선.

DATA PIPELINE & SEARCH ARCHITECTURE (멀티소스 수집 → 필터링 → 하이브리드 검색)



구현 핵심 (IMPLEMENTATION)

데이터 수집 파이프라인
REST·Crawl·RSS 3소스를 asyncio.gather로 병렬 수집. 중복 URL SHA-256 해시 필터링으로 중복 제거. 실시간 갱신 스케줄러로 최신 미식 데이터 유지. I/O 병목 완전 제거.

Ko-BERT 2단계 필터링
1단계: 광고 키워드 사전 기반 규칙 필터. 2단계: Ko-BERT 문맥 임베딩 코사인 유사도로 자연스러운 광고 문장 탐지. 규칙만으로 놓치는 콘텐츠를 벡터로 보완. 광고 20%↑ 제거 .

2-Stage 하이브리드 검색
Stage 1: ChromaDB 코사인 유사도로 의미론적 유사 맛집 Top-K 추출. Stage 2: 사용자 현재 위치 거리 가중치로 공간 맥락 Re-ranking. 의미+위치 결합으로 개인화 추천 정확도 개선.

성과 & 회고

정량 성과
광고성 콘텐츠 20%↑ 제거 로 데이터 품질 개선. asyncio 병렬 수집으로 멀티소스 I/O 병목 제거. 2-Stage 하이브리드 검색으로 키워드 대비 추천 정확도 개선. React Native 모바일 앱 연동 완성.

회고 & 개선 과제
Ko-BERT 필터 임계값 조정으로 정밀도-재현율 균형 최적화 필요. 사용자 이력 기반 개인화(User-Based CF) 추가로 추천 다양성 개선 과제. 실시간 스트리밍 데이터 수집으로 갱신 주기 단축.

팀 협업 & 코드 품질 관리

Jira 스프린트 운영
2주 스프린트로 수집·필터링·검색·앱 4개 컴포넌트 분리 개발. Epic-Story-Task 3계층 구조로 진행상황 투명화. 스프린트 리뷰로 우선순위 지속 재조정.

GitLab PR & 코드 리뷰 주도
Self-Approval 금지 정책 도입. 비동기 로직(asyncio 코루틴) 필수 검증 항목 지정. PR 템플릿으로 변경 사유·영향범위 명시. 팀 전체 코드 품질 상향 평준화.

문서화 & 지식 관리
API 명세 OpenAPI 자동 생성. 크롤링 대상 사이트 변경 이력 ADR로 관리. Ko-BERT 임계값 조정 실험 결과 로그화. 팀원 온보딩 시간 단축.

Ko-BERT | ChromaDB | FastAPI | asyncio | React Native | PostgreSQL | Jira | GitLab

Pictag

소상공인용 경량 CCTV AI SaaS · YOLO 백본 분해 · Attention 임베딩 Re-ID · 4-Thread 파이프라인

학습 효율 50%↑ | 4-Thread Re-ID Pipeline | OpenVINO INT8 | Django WebSocket 히트맵 | Intel AI 팀프로젝트

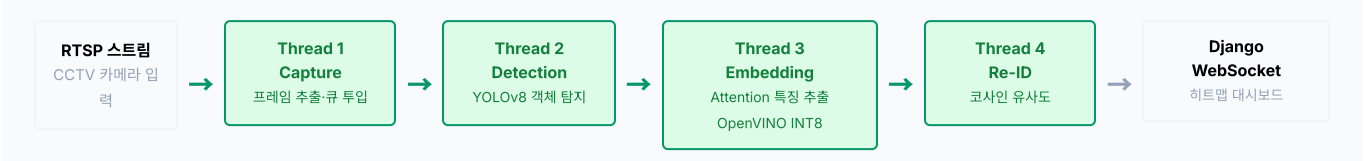
맥락 (CONTEXT)

소상공인 매장의 저사양 엣지 디바이스(GPU 없음)에서 CCTV 영상으로 방문객 Re-ID와 동선 히트맵을 실시간 제공하는 SaaS. 두 가지 핵심 과제: (1) **임베딩 품질** — YOLO 백본에서 어떤 방식으로 Re-ID 특징을 추출할 것인가 (Linear vs Pooling vs Attention 선택 필요). (2) **엣지 실시간성** — GPU 없는 환경에서 RTSP 스트리밍 처리와 Re-ID 추론을 동시에 처리하는 구조 설계.

핵심 의사결정 (DECISION)

가설 없이 직접 비교 실험: Linear / Pooling / **Attention** 3가지 임베딩 방식을 동일 데이터·동일 조건으로 A/B 테스트. Attention 방식이 Re-ID 정확도와 학습 효율 모두에서 **50%↑** 우위 확인 → 채택. 엣지 실시간성은 Capture/Detection/Embedding/Re-ID를 **4-Thread 독립 큐**로 분리하고, OpenVINO INT8 양자화로 추론 속도 확보.

4-THREAD RE-ID PIPELINE (RTSP → CAPTURE → DETECTION → EMBEDDING → RE-ID → DASHBOARD)



구현 핵심 (IMPLEMENTATION)

임베딩 방식 A/B 실험

YOLO 백본 분해 후 Linear / Pooling / Attention Head 3가지 추출 방식 비교. 동일 데이터·학습 조건에서 Attention이 Re-ID 정확도와 수렴 속도 모두 **50%↑** 우위. 실험 기반 채택.

4-Thread 독립 큐 설계

Capture/Detection/Embedding/Re-ID 스레드를 독립 Queue로 연결. 각 단계 처리 속도 차이를 큐가 버퍼링. 카메라 스트림 지연 없이 Re-ID 연속 처리. 스레드 실패 시 개별 재시작, 전체 영향 없음.

엣지 최적화 & 대시보드

OpenVINO INT8 양자화로 GPU 없이 엣지 실시간 추론 달성. Django + WebSocket으로 방문객 동선 히트맵 실시간 렌더링. 시간대별 방문 밀도 분석으로 소상공인 공간 운영 인사이트 제공.

성과 & 회고

정량 성과

Attention 임베딩으로 Re-ID 학습 효율 **50%↑**. 4-Thread 파이프라인으로 GPU 없는 엣지 환경 실시간 Re-ID 달성. Django+WebSocket 히트맵 대시보드 실시간 렌더링. 소상공인 SaaS 비즈니스 모델 실증.

회고 & 개선 과제

스레드 간 큐 크기 조정이 전체 처리량에 미치는 영향 분석 필요. Re-ID 정확도는 조명·각도 변화에 민감 — 다양한 카메라 환경 데이터 추가 학습 과제. ONNX 변환으로 범용 엣지 배포 확장.

비즈니스 모델 & 확장 방향

소상공인 SaaS 타겟

별도 GPU 서버 없이 기존 CCTV + 저사양 미니PC로 구동. 월 구독 모델로 초기 도입 비용 최소화. 방문객 동선·체류 시간 분석으로 매장 레이아웃 최적화 인사이트 제공.

OpenVINO INT8 양자화 전략

FP32 대비 모델 크기 4배 감소, 추론 속도 2-3배 향상. 정확도 손실 최소화 (Post-Training Quantization). CPU-only 환경에서 실시간 추론 가능. 저사양 엣지 디바이스 범용 배포 실현.

다음 단계 확장

ONNX 변환으로 ARM 기반 엣지 디바이스(Raspberry Pi) 지원. 다중 카메라 동기화로 매장 전체 동선 추적. 체류 시간·구역별 밀도 히트맵을 마케팅 대시보드와 연동.

YOLOv8 | Attention Embedding | OpenVINO INT8 | RTSP | Python Threading | Django | WebSocket

KDLC

물류센터 수요 예측 경진대회 · 45개+ 피처 공학 · SARIMA+LSTM+LightGBM 3-Model 앙상블

`45개+ 피처 공학` `3-Model 앙상블` `TimeSeriesSplit 교차검증` `Data Leakage 구조적 차단` `Intel AI 팀프로젝트`

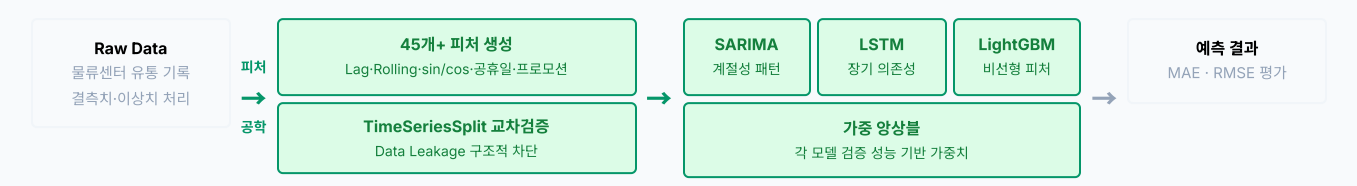
맥락 (CONTEXT)

물류센터 유통 데이터 기반 상품 수요 예측 경진대회. 세 가지 핵심 도전: (1) **시계열 복잡성** — 주기성(주간·월간·연간)·추세·노이즈가 혼합된 다층 패턴. (2) **Data Leakage** — 무작위 K-Fold 적용 시 미래 데이터가 학습에 포함되는 치명적 오류. (3) **단일 모델 한계** — 계절성·장기 의존성·비선형 피처 중요도를 동시에 포착하는 모델 없음. 모델 선택 이전에 피처 설계와 검증 구조가 승패를 결정.

핵심 의사결정 (DECISION)

피처 공학 우선: Lag·Rolling·주기성 인코딩·도메인 지식 피처를 포함한 **45개↑ 피처**가 모델 성능 향상의 핵심 동인. **TimeSeriesSplit**으로 시간 순서 보존 교차검증 — Data Leakage 구조적 차단. **3-Model 앙상블** : SARIMA(계절성) + LSTM(장기 의존성) + LightGBM(비선형 피처)을 각각의 강점 영역에 특화하여 상호 보완.

FORECASTING PIPELINE (RAW DATA → 피처 공학 → 3-MODEL 앙상블 → 예측)



구현 핵심 (IMPLEMENTATION)

45개+ 피처 공학 전략

Lag(1~7일) + Rolling 통계(7·14·30일 평균·분산·최대) + sin/cos 주기성 인코딩 + 공휴일 바이너리 + 프로모션 플래그. 도메인 지식 기반 피처가 모델 성능 향상에 가장 큰 기여. 피처 중요도 분석으로 불필요 피처 제거.

3-Model 앙상블 설계

SARIMA: 주간·월간 계절성 ARIMA 분해. LSTM: 30일 시퀀스 장기 의존성 학습. LightGBM: 45개+ 피처의 비선형 중요도 포착. 각 모델 검증 RMSE 역수를 가중치로 앙상블 → 단일 모델 대비 예측 분산 감소.

TimeSeriesSplit 교차검증

무작위 K-Fold 금지 — 미래 데이터가 학습에 포함되는 Data Leakage 구조적 차단. 시간 순서 보존 분할(과거 Train → 미래 Val). 실제 운영 환경과 동일한 검증 조건 확보. 과적합 탐지 신뢰도 향상.

성과 & 회고

핵심 학습 & 성과

피처 공학이 모델 선택보다 예측 성능에 더 큰 영향. TimeSeriesSplit 적용 후 검증 지표와 실제 예측 오차 간 괴리 대폭 감소. 앙상블로 단일 모델 대비 RMSE 개선. 데이터 거버넌스의 중요성 실증.

회고 & 개선 과제

SARIMA 모델 파라미터(p,d,q) 그리드 서치 자동화로 최적 조합 탐색 효율 개선 필요. LSTM 하이퍼파라미터 최적화(Optuna). 외부 데이터(날씨·경제지표) 추가로 피처 다양성 확장 가능.

시계열 예측 심화 분석

sin/cos 주기성 인코딩

요일(1~7)·월(1~12)을 sin/cos 변환으로 연속 순환 표현. 12월→1월, 일요일→월요일 경계에서 불연속 없이 주기성 보존. 선형 인코딩 대비 모델이 계절 경계를 자연스럽게 학습.

Rolling 통계 피처 설계

7일(단기)·14일(중기)·30일(장기) 이동 평균·표준편차·최대값. 추세 방향과 변동성을 동시에 피처화. 이상 수요 스파이크(프로모션·공휴일) 전후 패턴을 Rolling 분산으로 탐지.

가중 앙상블 최적화

각 모델의 TimeSeriesSplit 검증 RMSE 역수를 정규화하여 가중치 산출. 성능이 높은 모델에 자동으로 높은 가중치 부여. 단순 평균 앙상블 대비 최종 RMSE 추가 개선. 모델 재학습 시 가중치 자동 갱신.

`SARIMA` `LSTM` `LightGBM` `TimeSeriesSplit` `pandas` `scikit-learn` `PyTorch`

AOI

성균관대 산학협력 외관 검사 AI · PatchCore+다중분류 vs 픽셀 세그멘테이션 · 라벨링 병목 비교 검증

PatchCore Few-shot | 이미지 단위 라벨링 | 히트맵 시각화 | 병렬 비교 검증 | DTK 진행 중

맥락 (CONTEXT)

성균관대 산학협력 외관 검사(AOI) AI 프로젝트. 다양한 제조 현장에 배포해야 하는 만큼 두 가지 핵심 과제: (1) **라벨링 최소화** — 오퍼레이터가 직접 라벨링해야 하는 현장에서 작업 병목과 모델 성능 저해를 어떻게 줄일 것인가. (2) **접근법 선택** — DTK 내부안(PatchCore+다중분류, 이미지 단위)과 성균관대안(OpenNet·U-Net·생성형 AI 합성 데이터, 픽셀 단위 세그멘테이션)을 동일 조건에서 비교 검증. 기술적 성능 외에 현장 배포 현실성을 함께 평가하는 것이 설계의 핵심.

핵심 의사결정 (DECISION)

성균관대안은 합성 데이터 특성상 **픽셀 단위 세그멘테이션** 라벨링이 필수 — SAM으로 일부 자동화 가능하나, 다현장·다품종 환경에서 재라벨링 병목과 성능 저해 요인으로 작용할 수 있다고 검토. DTK안은 **이미지 단위 라벨링**으로 오퍼레이터 부담을 최소화하면서 PatchCore로 양품 샘플만으로 이상 탐지 가능. 기술적 우위만이 아닌 현장 운영 비용을 종합해 이미지 단위 접근의 우위를 내부 공유하고, 두 방식을 병렬 비교 검증 중.

병렬 비교 검증 구조 (DTK 내부안 VS 성균관대안)

구현 핵심 (IMPLEMENTATION)

PatchCore 이상탐지
ImageNet 사전학습 모델로 패치 단위 정상 분포 모델링. 양품 샘플만으로 학습 — 이상 라벨 불필요. 이상 스코어 기반 히트맵으로 구역 시각화. Few-shot 환경에서도 강건한 탐지 성능.

다중분류 하이브리드
PatchCore 이상 탐지 후 다중분류 모델로 불량 유형 특정. 이미지 단위 라벨링만으로 분류 구성 가능 — 픽셀 세그멘테이션 대비 라벨링 공수 대폭 절감. 오퍼레이터 판독 보조 히트맵 UI 연동 목표.

비교 검증 프레임워크
두 방식을 동일 데이터·동일 조건에서 성능 비교. 평가 축: 탐지 성능(AUROC·F1) + 라벨링 공수 + 다현장 적용 용이성. 기술 지표와 운영 현실을 함께 측정하는 평가 설계.

비즈니스 관점 & 회고

현장 배포 현실성 판단
SAM 자동화로 픽셀 라벨링 일부 해소 가능하나, 다품종·다현장 환경에서는 품목마다 기준 재설정이 필요. **라벨링 병목이 시스템 전체 운영 효율을 저해**한다고 판단 — 이미지 단위 접근의 현장 배포 우위를 내부 공유. 기술 선택이 곧 운영 비용임을 실증.

진행 현황 & 다음 단계
DTK 내부안(PatchCore+다중분류)과 성균관대안 병렬 검증 진행 중. 동일 데이터 조건 성능 비교 후 최종 접근법 결정 예정. AlignAI와 통합하여 정렬→검사→이상탐지 자동 루프 구성 로드맵.

PatchCore | Anomalib | OpenNet | U-Net | SAM | 생성형 AI | PyTorch | Labelme