

AI 도로영상 분석 플랫폼 기획서

25.07.02(Wed.)

The KY 영상분석팀

Table of Contents

I. 프로젝트 개요

II. 주요 서비스 기능

III. 업무 **FLOW**
(서비스 시나리오)

IV. 시스템 구성도

V. 개발환경 및 도구

VI. 핵심 화면 흐름

VII. 개발 일정

VIII. 기대효과 및 향후계획

IX. **Q&A**

프로젝트 개요

□ 프로젝트 배경

- 도로 사고영상 분석 수요 증가
- 수작업 분석의 한계
- '한문철의 블랙박스 리뷰'와 같이, 디지털 플랫폼에서 교통사고 관련 정보를 공유하고, 온라인 상담을 제공하는 서비스 필요.

□ 프로젝트 목적

- AI로 사고 객체 검출 및 검출률 시각화
- 사용자 편의 제공 및 판단 보조

KY 영상분석팀

□ 목표기간 : 4주('25.06.02~'25.07.04)

□ 팀원(3명)

- 이경윤(프론트엔드) : Notion관리, 프론트엔드 개발
- 최소희(백엔드) : FLASK API·DB 설계·AI 모델 통합
- 고수혁(팀장, AI모델,기타) : PPT· 회의록· 테스트 시나리오

□ 솔루션명 : **The KY AI 영상 분석 Platform**

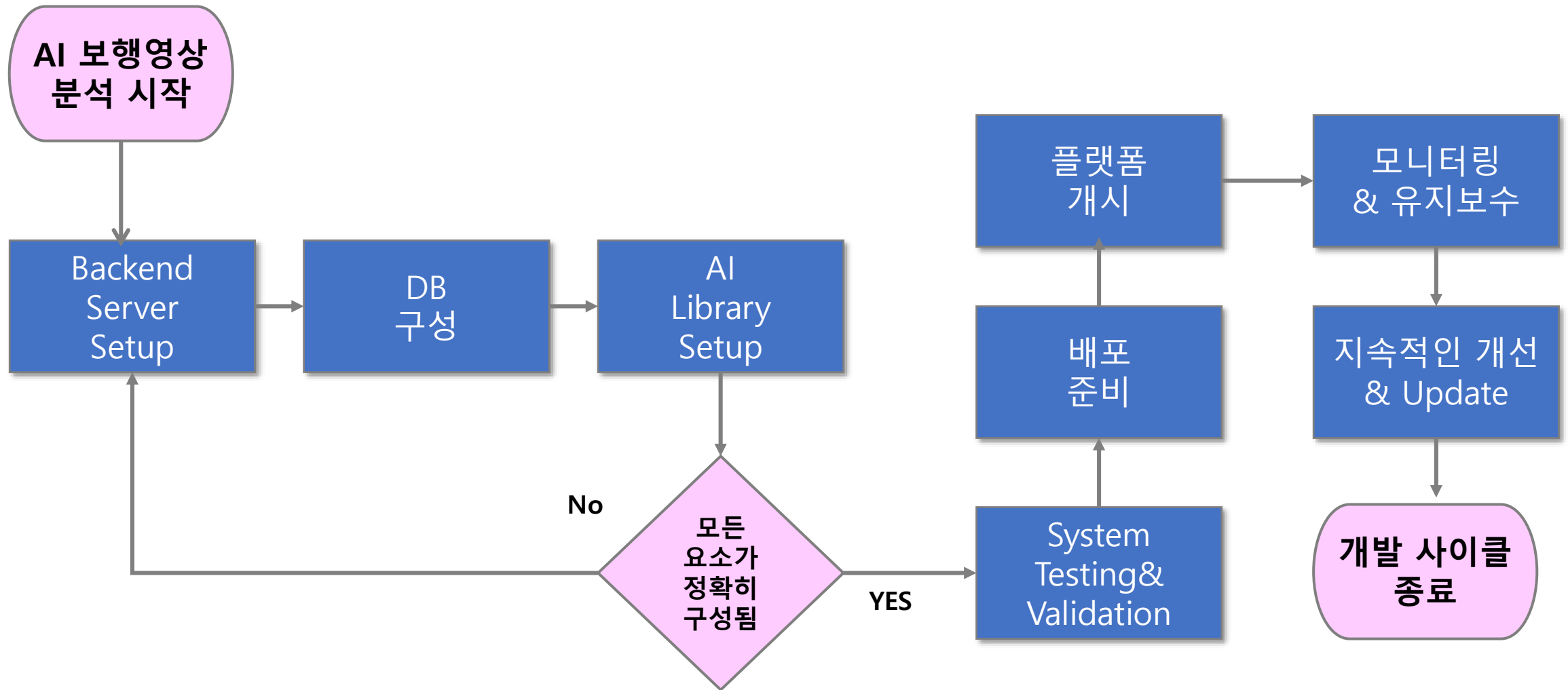
□ 개발 목표

1. 최소 구현: 교통 영상 객체 검출 (ex;보행자·차·자전거·차량)
2. 차후 구현 : 사고 상황분석 및 과실비율 산정
3. 배포 : 웹 서비스 형태, 배포시 도메인 등록 + 차후 클라우드 서버

주요 서비스 기능

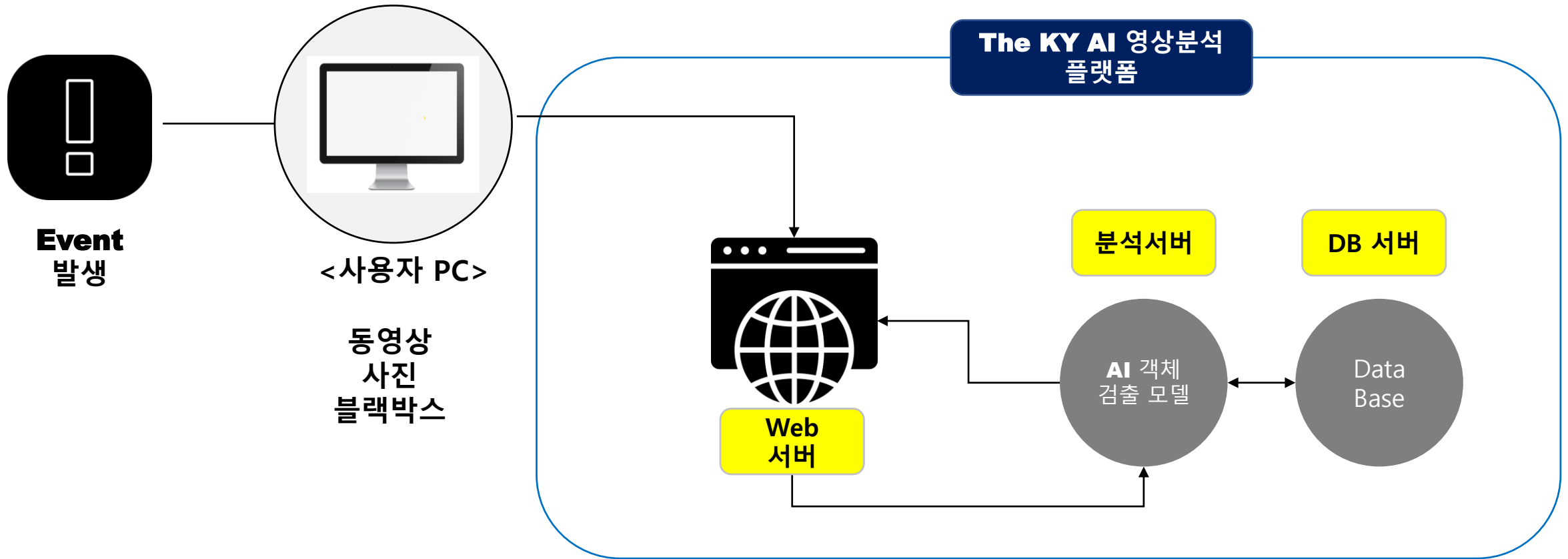
1. 회원가입 / 로그인 기능
2. 객체 검출시 AI 엔진 활용
3. AI 객체 검출
4. 검출 결과 DB 저장 및 다운로드 API
5. (확장) 사고 분석 · 과실 비율 리포트

개발 프로세스 (Development Process)



업무 FLOW (서비스 SCENARIO)

<영상 촬영현장>



업무 FLOW (서비스 SCENARIO)

사용자 영상수집

- 드론/카메라/차량 블랙박스
에서 교통영상 수집

영상 업로드

- AI 영상분석 플랫폼에
영상 업로드

AI 영상 분석

- 객체탐지(차량, 보행자, 신호등 etc.)
- 검출률 제공
- 교통량 측정
- YOLO8로 객체검출 및 검출률
제공

결과 전송

- User 정보와 함께
DB에 분석 결과 저장
- 사용자 조회시 제공

활용 시나리오



블랙박스 분석 지원

- 교통사고 발생시, 블랙박스를 바탕으로 사고경위 및 원인파악에 대한 정보 제공



교통사고 분석 Report

- 교통사고 발생시 귀책&손해사정의 근거자료로 활용
 - ✓ 귀책 판단) 책임, 과실, 고의 여부
 - ✓ 손해사정) 화재, 해상, 책임보험



차량사고 사전예방

- 사고 원인&사고 빈번 발생 구간에 대한 교통정보 제공으로 교통사고 사전예방 지원
 - ✓ 신호등 최적화를 통한 교통흐름 개선
 - ✓ 구급차/소방차 경로 최적화 지원
 - ✓ 운전자 교육에 활용



보행자 안전

- 스쿨존 보행자 감지 및 차량 속도 제어
- 횡단보도 보행자 우선 신호 운영

사용자 인터페이스

사용자가 영상을 시스템에 업로드하면, 특정시점의 분석결과가 조회 가능하도록 인터페이스를 그림과 같이 제공 예정.

영상 업로드

영상 분석 시스템

업로드 결과 확인 로그아웃

로그인 성공!

tester님 환영합니다

파일 선택:

새 파일 업로드

파일 선택

선택된 파일 없음

분석 시작

분석결과

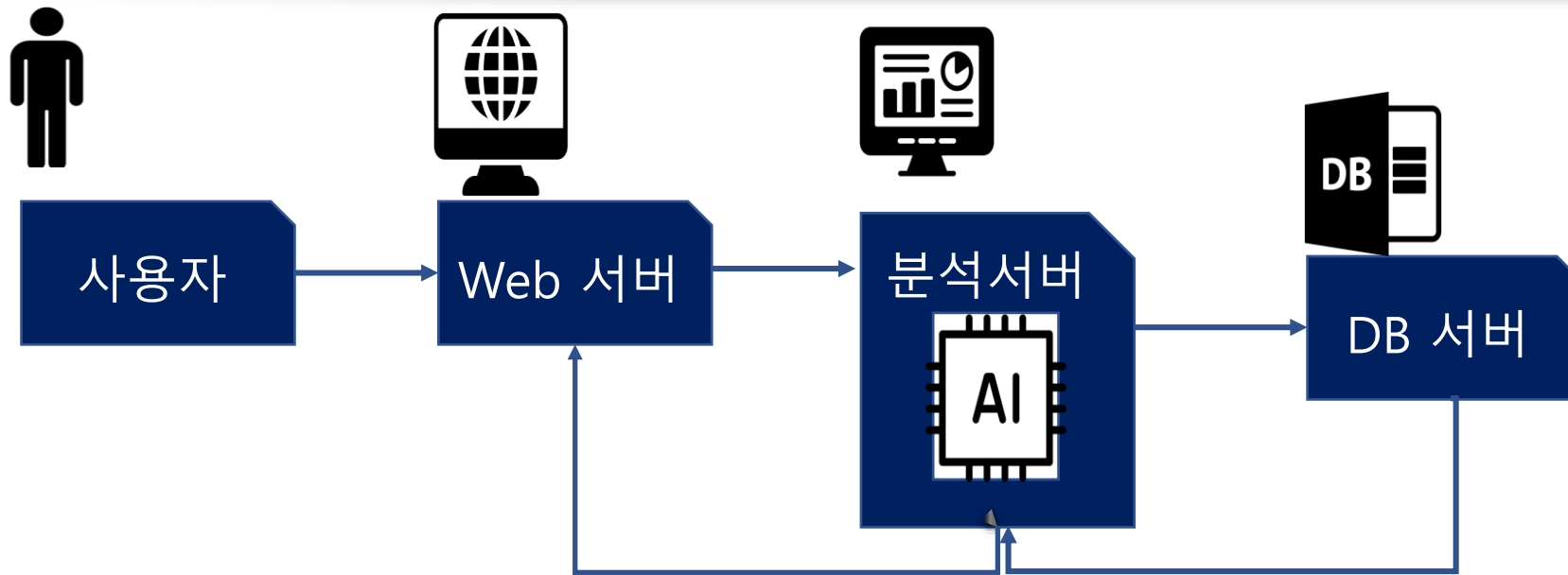
영상 분석 시스템

업로드 결과 확인 로그아웃

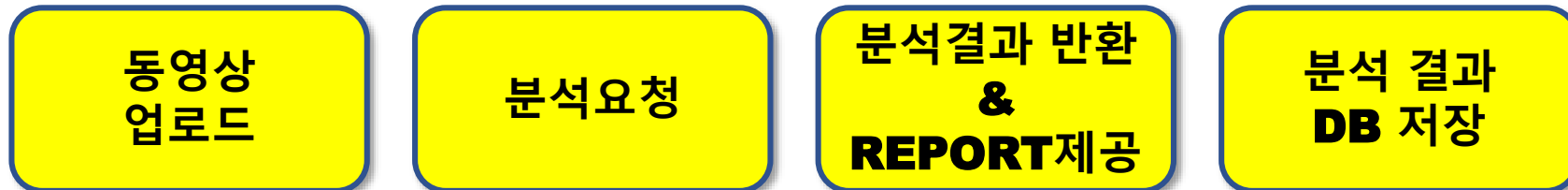
tester2님의 분석 결과 이력

파일명	타입	상태	다운로드
seoul.mp4	-	진행 중	대기중
215258_medium.mp4	video	완료	다운로드
highway-7213206_1280.jpg	image	완료	다운로드

시스템 아키텍처 & 인증



<데이터 흐름도>



● WEB 서버

- Jinja2 템플릿 사용 : Flask 내장 템플릿 엔진
- Jinja2 : 서버에서 데이터를 주입해 최종 HTML을 만들어 내는 역할

● 세션 기반 인증 활용

- 세션 기반 인증을 위해 Session과 Cookie사용
- 장점 1) 서버에 저장하기 때문에 편하고 효율적
- 장점 2) 구현이 명확하며 실제 서버에서 로그인 상태를 확인하기 유용

● 데이터 흐름

1. 사용자 → 영상 업로드
2. AI 서버로 전송 → AI 학습 모델로 추론
3. 검출결과 저장 → 사용자 웹 UI 조회/다운로드

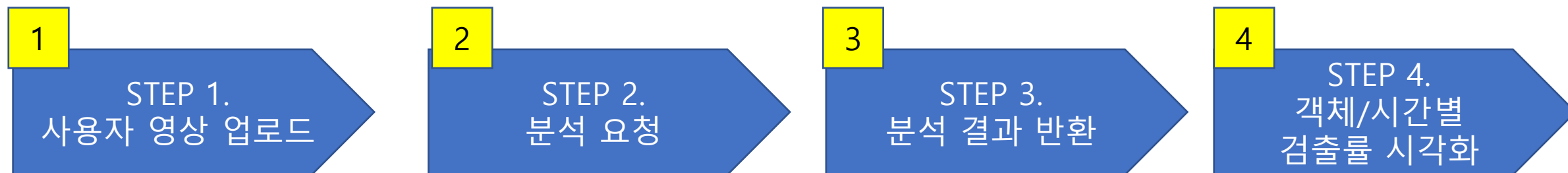
개발환경 및 도구

분 류	도구/라이브러리	역할 및 이유
서버 언어	Python 3.13	AI·API·스크립트 통합개발
AI 모델	YOLOv8	빠른 추론·COCO 사전학습 활용
DB	SQLite	SQL 친숙도·로컬테스트 용이
백엔드 프레임워크	FLASK	경량 REST API·JWT 인증 지원
프론트 엔드	HTML/CSS/JS, Axios	로그인·업로드·결과조회 화면 개발
인증·보안	세션기반 / Flask	
배포 패키징	AWS S3 버킷에 런칭	<ul style="list-style-type: none"> • 정적 웹 배포) HTMLS/CSS/JS 파일을 S3에 업로드 하여 웹으로 배포 • 이미지/동영상 저장소) 사용자 업로드 파일이나 분석된 AI 결과 저장에 사용 • AI 결과 공유) YOLO 결과 이미지나 리포트 파일을 S3에 저장해 외부에 공유 가능 • 백업 저장소) 로그/데이터베이스 백업 등을 주기적으로 저장 • 클라우드 Lambda 함수 연동) S3에 업로드될 때 자동으로 처리하는 workflow 가능
버전 관리	Git + Git Hub	브랜치별 협업·PR 리뷰
프로젝트 관리	Notion	일정·태스크·회의록·문서 관리 총괄

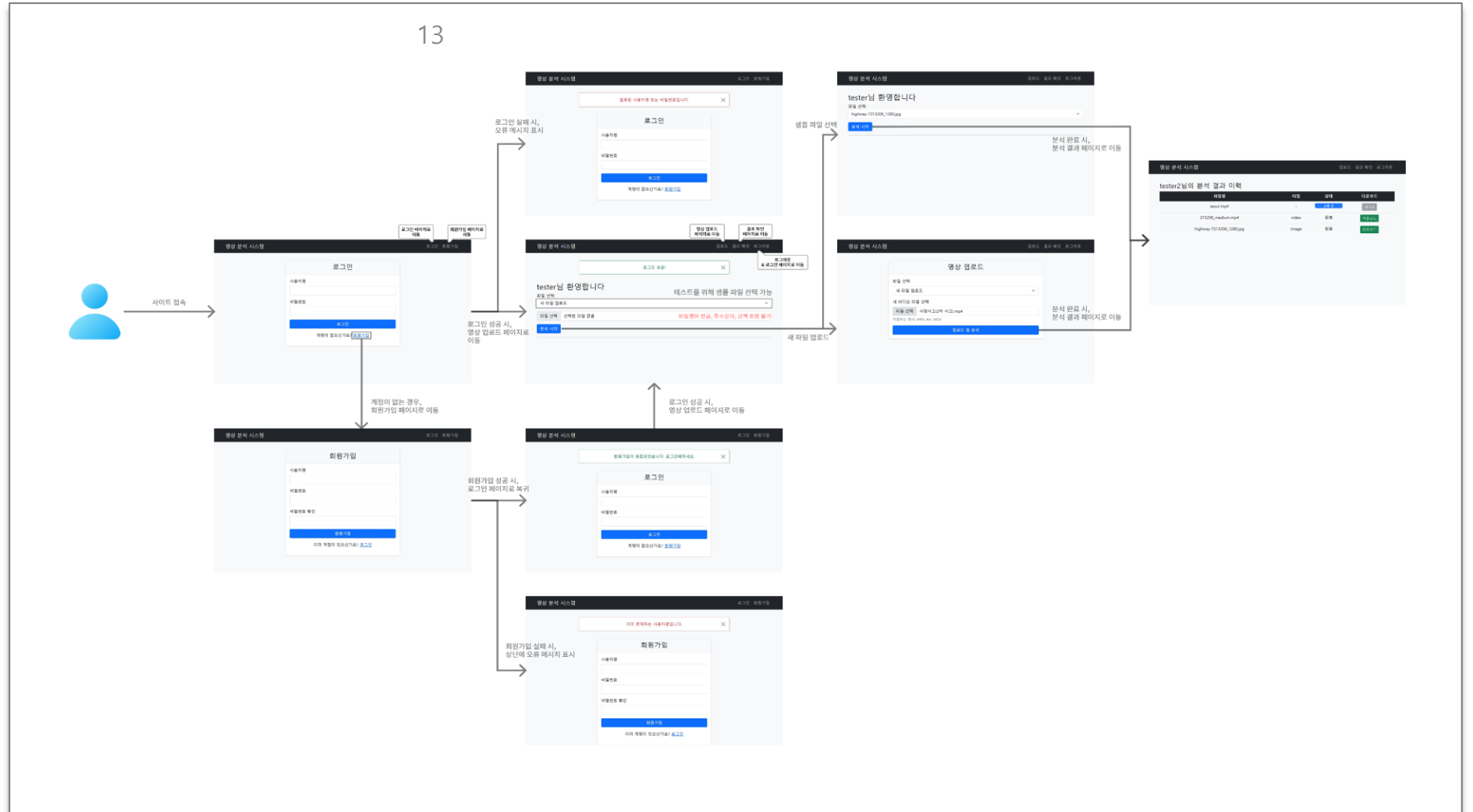
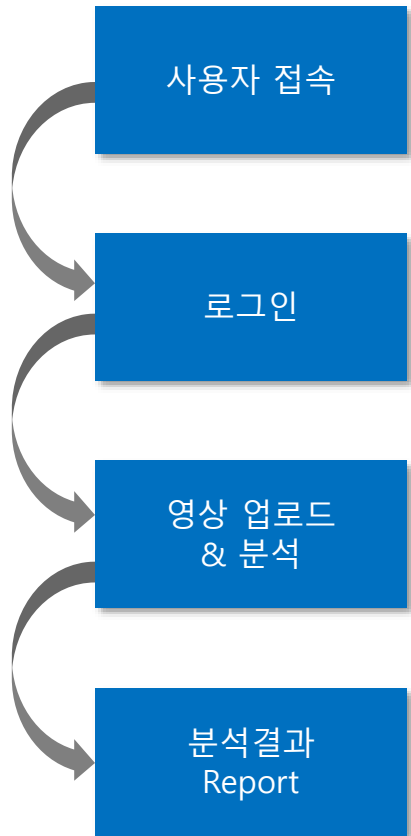
AI 모델 관련 작업

- 초기 단계 : YOLOv8-n(nano)
 - CPU 모드만으로도 충분한 실시간 처리 성능 확보
- 차후 고도화·클라우드 배포 : YOLOv8-s(small)
 - GPU 서버 환경에서 속도·정확도 모두 안정적
 - API 변경없이 모델 크기만 바꿔 재학습/배포 가능
- ▶ YOLOv8 선택 이유:
 - 다양한 컴퓨터 비전 작업 지원
 - 객체감지, 인스턴스 분할, 포즈 추정, 이미지 분류 등
 - Anchor Free Detection 지원
 - 직관적인 API제공

핵심 화면 흐름



핵심화면 흐름 > 사용자 접속



개발 일정

TASK		6월				7월				
업무영역	세부과제	W1	W2	W3	W4	W1	W2	W3	W4	담당자
전반	• 프로젝트 기획 및 역할분담		'25.06.02~							ALL
BackEnd	• 요구사항 확정&ERD 설계→DB설계									최소희, 이경윤
전반	• 개발환경 셋업(venv,SQLite,Git)									ALL
전반	• Notion보드, API 명세서, 템플릿 구성									ALL
FrontEnd, BackEnd	• 회원가입/로그인 • 세션 기반 인증 API 구현									최소희, 이경윤
DB	• SQLite 내장 DB로 구축									이경윤
FrontEnd	• 로그인·영상 업로드 화면 기본 레이아웃 구현									이경윤
BackEnd, AI 모델	• YOLOv8 모델 연동 및 분석서버 엔드포인트 구현									최소희, 고수혁
BackEnd	• 영상 → 프레임 → 객체검출 파이프라인 검출									최소희
FrontEnd	• 결과 조회·다운로드 기능 완성									이경윤, 고수혁
전반	• 통합 테스트 및 버그 수정 • 최종 문서화(배포 가이드·API 명세) 및 demo 준비					~'25.07.04				고수혁

API 명세서

현재 **GIT HUB** 상에 **API** 명세서를 작성 & 관리하고 있음.

API 명세서

인증 관련

POST /register

- 설명: 회원가입
- 요청: username, password, confirm_password (form)
- 응답: 성공/실패 메시지, 리다이렉트
- 인증: 불필요

POST /login

- 설명: 로그인
- 요청: username, password (form)
- 응답: 성공/실패 메시지, 세션 저장, 리다이렉트
- 인증: 불필요

GET /logout

- 설명: 로그아웃

개발코드 및 결과화면

개발코드

```
requirements.txt
1 flask
2 ultralytics
3 flask-sqlalchemy
4 werkzeug
5 flask_session
6 #flask-MigrateVenv
7 flask-migrate

app.py
...
return self.wsgi_app(environ, start_response)
...
File "C:\Users\USER\250625\flask-yolo8\flask-yolo8\venv\Lib
\site-packages\flask\app.py", line 1514, in wsgi_app
response = self.handle_exception(e)
File "C:\Users\USER\250625\flask-yolo8\flask-yolo8\venv\Lib
\site-packages\flask\app.py", line 1511, in wsgi_app
response = self.full_dispatch_request()
```

결과화면

영상 분석 시스템 업로드 결과 확인 로그아웃

고수혁님의 분석 결과 이력

파일명	타입	상태	다운로드
highway-7213206_1280.jpg	image	완료	다운로드
215258_medium.mp4	video	완료	다운로드
10881-226635366_medium.mp4	video	완료	다운로드
night-7530755_1280.jpg	image	완료	다운로드
highway-7213206_1280.jpg	image	완료	다운로드

개발코드 및 결과화면

다운로드

영상 분석 시스템 업로드 결과 확인 로그아웃

고수혁님의 분석 결과 이력

파일명	타입	상태	다운로드
highway-7213206_1280.jpg	image	완료	다운로드
highway-7213206_1280.jpg	image	완료	다운로드
215258_medium.mp4	video	완료	다운로드
10881-226635366_medium.mp4	video	완료	다운로드
night-7530755_1280.jpg	image	완료	다운로드
highway-7213206_1280.jpg	image	완료	다운로드

분석화면
(img)



개발코드 및 결과화면

다운로드

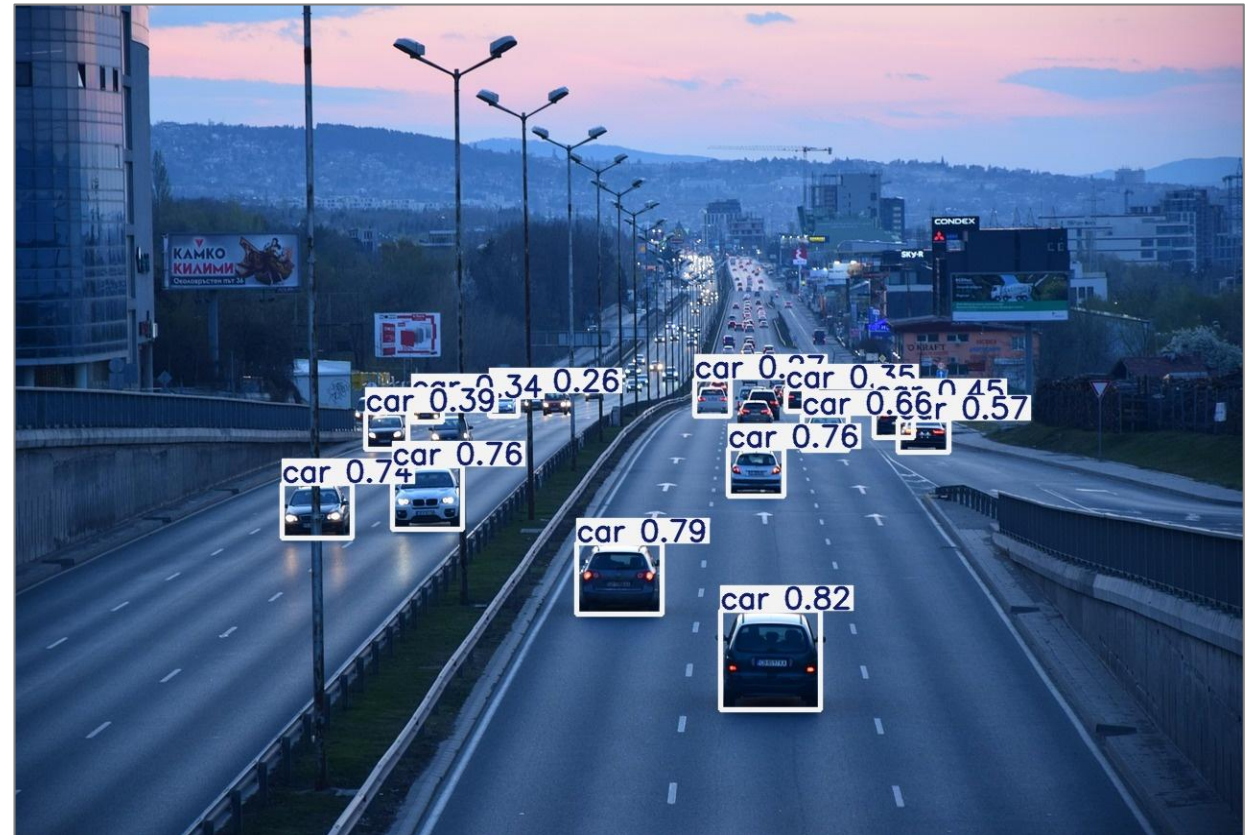
영상 분석 시스템

업로드 결과 확인 로그아웃

고수혁님의 분석 결과 이력

파일명	타입	상태	다운로드
highway-7213206_1280.jpg	image	완료	다운로드
highway-7213206_1280.jpg	image	완료	다운로드
215258_medium.mp4	video	완료	다운로드
10881-226635366_medium.mp4	video	완료	다운로드
night-7530755_1280.jpg	image	완료	다운로드
highway-7213206_1280.jpg	image	완료	다운로드

분석화면
(.mp4)



기대효과 및 향후 계획

기대효과

- 교통사고 영상 분석 자동화
- 인터넷을 통한 사고영상 업로드&Report 조회 가능
- 교통 사고 경위 정량화 및 객관화

고도화 방향

- 귀책사유 추정 로직 탑재 가능
- AI Library 도입을 통한 분석 모델 업그레이드
- (COCO아닌) 데이터 전처리/수집부터 시작하여 데이터 분석 수행
- 사용자의 모바일 기기를 통한 서비스 활용 지원

참고자료(Appendix)

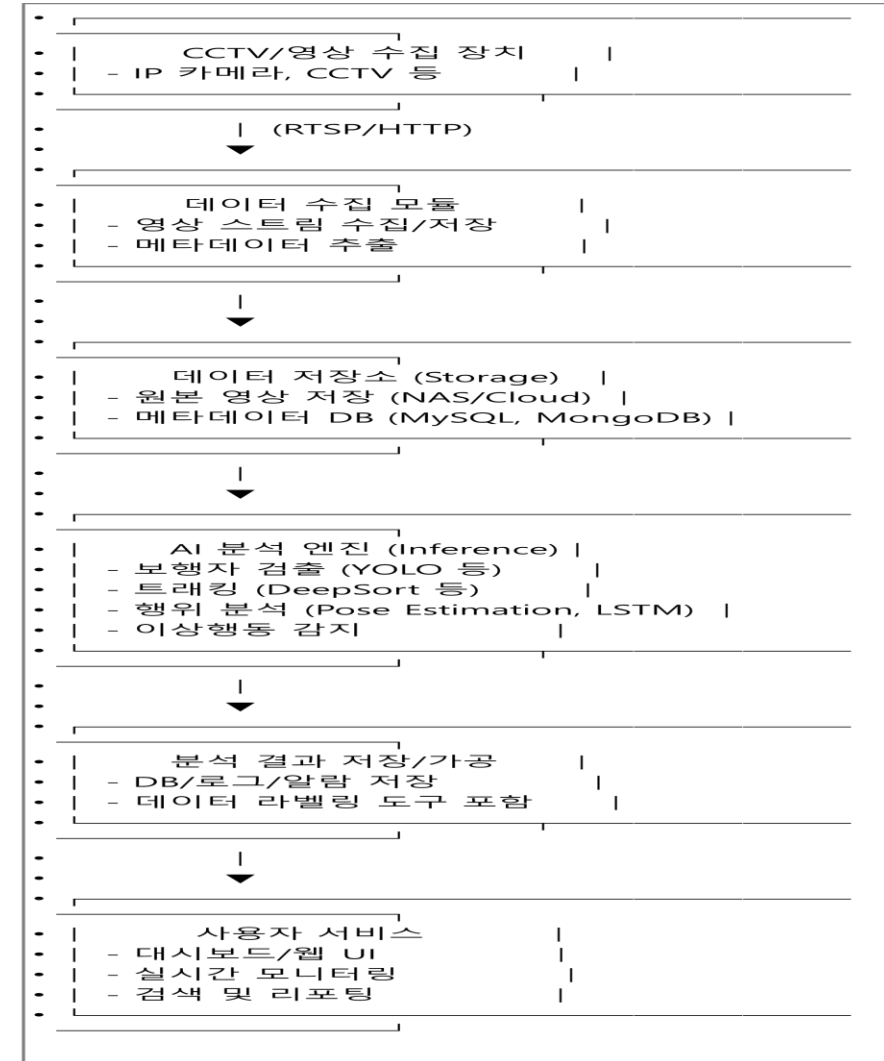
- Yolo 버전 시리즈에 대한 기입 & 작동원리
- API 명세서
- Sequence Diagram
- AI 허브 > 교통사고 영상 분석에서 사용했던 알고리즘
&AI 모델 deep 리서치 필요
(ex; DetectoRS,VTN)

참고자료(Appendix)

<AI 영상분석플랫폼 업무 FLOW>

※ 추가 가능 기능 :

- **AI 학습/재학습 시스템**: 수집된 데이터를 통해 모델을 지속적으로 개선할 수 있는 파이프라인 (AutoML 또는 주기적 Fine-tuning).
- **권한 및 보안 관리**: 사용자 접근 제어, 데이터 암호화, 로그 감사 기능.



참고자료(Appendix)

구성 요소	기술 예시
수집	RTSP, FFmpeg, Kafka
저장	AWS S3, NAS, PostgreSQL, MongoDB
AI 모델	YOLOv5, DeepSort, OpenPose, Detectron2
분석/처리	Python, PyTorch, TensorFlow, ONNX
실시간 처리	Kafka, Redis, Flask/FastAPI
프론트엔드	React, Vue.js, D3.js (시각화)
백엔드/API	Django, FastAPI
배포/운영	Docker, Kubernetes, Nginx, Grafana

참조자료(Reference)

- AI-HUB> 교통사고 영상 데이터
✓ URL) [AI-Hub](#)
- YOLOv8) [You Only Look Once – 나무위키](#)
- [세션 인증 방식 VS Token 인증방식\(인증과 인가\)](#)



*****END OF DOCUMENT*****