Strictly Confidential

AI 교통영상 분석 플랫폼 개발
최종 발표

25.07.04(Fri.)

The KY 영상분석팀

Table of Contents

01 프로젝트 개요

02 프로젝트 팀 구성 및 역할

03 프로젝트 수행 절차

04 프로젝트 수행 결과

05 자체 평가 의견

프로젝트 개요

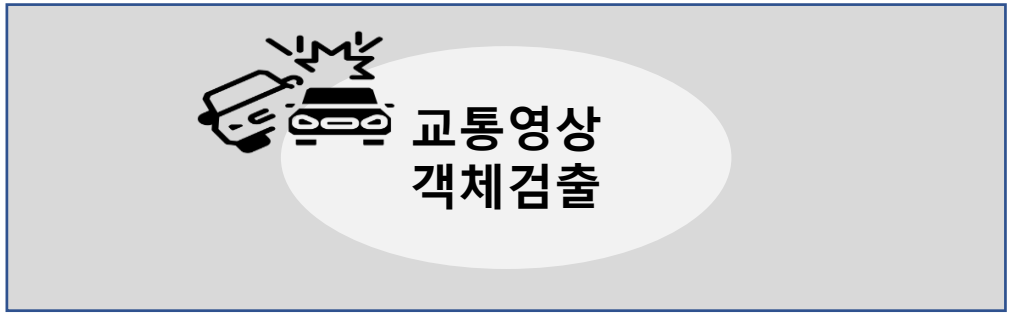
❒ 프로젝트 배경

- 교통사고 영상분석의 수요 증가
- 영상분석시 수작업 분석의 한계

❒ 프로젝트 목적

- AI 모델에 기반한 플랫폼(솔루션)개발을 통해 교통사고 객체검출 및 검출률 시각화
- 사용자에게 교통영상 분석을 통한 편의제공 및 사고발생시 판단 보조의 자료 제공

< AS-IS>



교통사고 분석 SOLUTION

< TO-BE>

프로젝트 개요

❒ 기간 및 개발목표

- 기간 : **'25.06.02**(Mon.)~**'25.07.04**(Fri.) **(4주)**
- 팀명 : The KY 영상 분석팀
- 솔루션명 : **The KY AI 영상 분석 Platform**
- 개발 목표
  - ✓ 최소 구현: 교통 영상 객체 검출
    (ex; 보행자·차·자전거·차량)
  - ✓ 차후 구현: 사고 상황 분석 및 과실 비율 산정
  - ✓ 배포: 웹 서비스
    (Flask 기반 REST API + HTML/JS 프론트)

프로젝트 개요

1 프로젝트 주제

AI 기반 교통영상 분석 플랫폼 개발

2 개발 내용

- **사용자가 등록한 교통영상에 대한 객체검출 및 검출률을 시각화하는 플랫폼 개발**
- **사용자 UI 개발과 영상분석을 위한 서버개발 및 모델 활용**
  - **Web 서버 구축)** 사용자의 교통영상 업로드 및 분석결과 조회
  - **분석 서버 구축)** 객체검출을 위한 서버
  - **DB 서버 구축)** 상황 로그를 남기기 위한 서버

3 개발 환경

- Frontend) HTML/CSS/JS 등
- Backend) Python, FLASK
- AI 라이브러리) YOLOv8
- IDE) Pycharm
- DB)SQLite
- 개발코드) GitHUB
- 일정/문서관리)Notion

4 주요기능

- **회원가입/로그인 기능**
- **AI 엔진을 활용한 객체 검출**
- **검출 결과 DB 저장 및 다운로드 API 제공**
- **(확장) 사고 분석·과실 비율 리포트 제공**

5 활용방안 및 기대 효과

- 교통사고 상황분석의 판단 자료
- 교통사고 귀책 여부/손해사정의 근거자료
- '한문철의 블랙박스 리뷰'와 같이 디지털 플랫폼에서 교통사고 정보를 공유하고, 온라인 상담을 제공하는 서비스로도 활용 가능

프로젝트 팀 구성 및 역할

교육생	역할	분야	담당 업무	
이경윤	팀원	프런트 엔드	NOTION 관리	FrontEnd 개발
최소희	팀원	백엔드	FLASK API, DB 설계	AI 모델통합
고수혁	팀장	개발테스트/업무 보고/자료조사/ 팀 관리	개발테스트, 자료작성	프로젝트/팀원관리, 업무보고
김민정	멘토	**PM** 및 코칭	프로젝트 예시 소개, 코칭 및 Feedback	

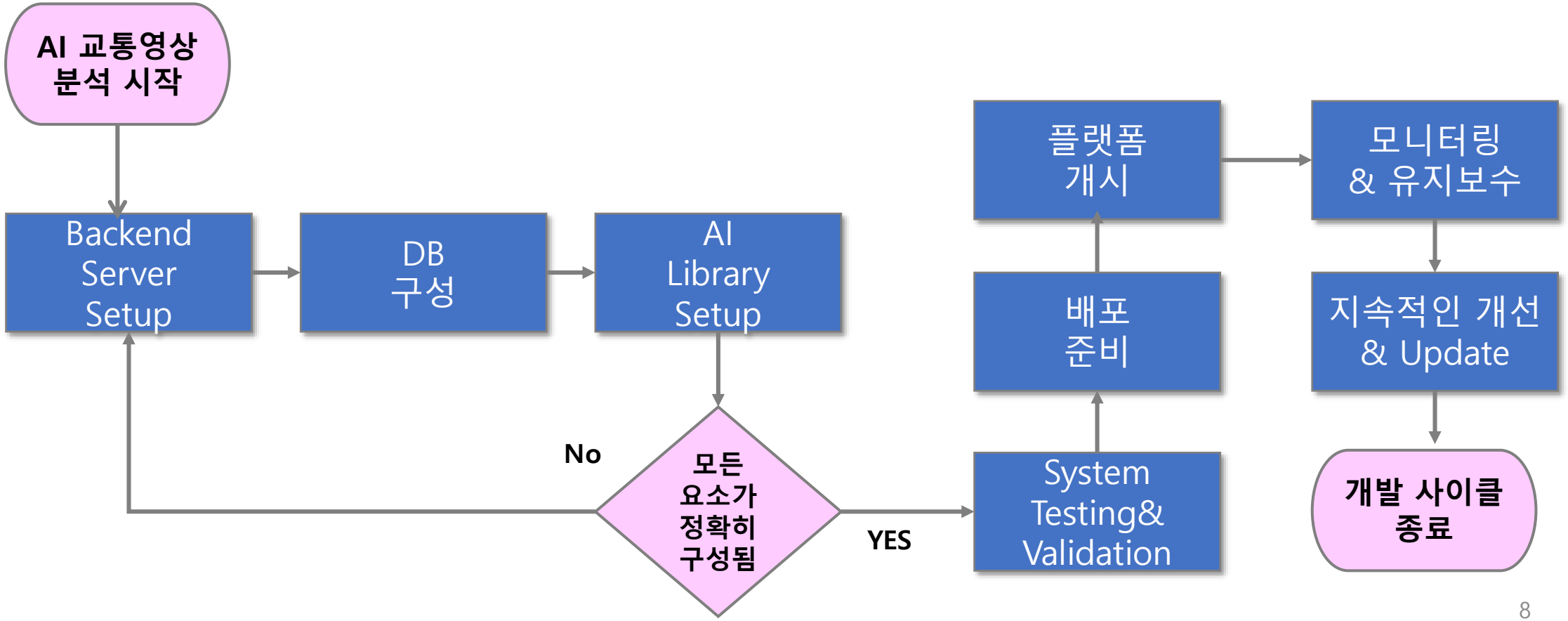
프로젝트 수행절차

일정& 수행과제

		TASK		6월					7월		
구분	업무영역	세부과제	W1	W2	W3	W4	W1	W2	W3	W4	
기획	전반	• 프로젝트 기획 및 역할분담	06.02~								
설계	BackEnd	• 요구사항 확정&ERD 설계→DB설계									
개발	전반	• 개발환경 셋업(venv,SQLite,Git)									
	전반	• Notion보드, API 명세서, 템플릿 구성									
	FrontEnd, BackEnd	• 회원가입/로그인 • 세션 기반 인증 API 구현									
	DB	• SQLite 내장 DB로 구축									
	FrontEnd	• 로그인·영상 업로드 화면 기본 레이아웃 구현									
	BackEnd, AI 모델	• YOLOv8 모델 연동 및 분석서버 엔드포인트 구현									
	BackEnd	• 영상 → 프레임 → 객체검출 파이프라인 검출									
	FrontEnd	• 결과 조회·다운로드 기능 완성									
테스트&배포	전반	• 통합 테스트 및 버그 수정 • 최종 문서화(배포 가이드· API 명세) 및 demo 준비					~07.04				

프로젝트 수행결과

개발프로세스



프로젝트 수행결과

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	• **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
버전 관리	Git + Git Hub	브랜치별 협업· PR 리뷰
프로젝트 관리	Notion	일정·태스크·회의록·문서 관리 총괄

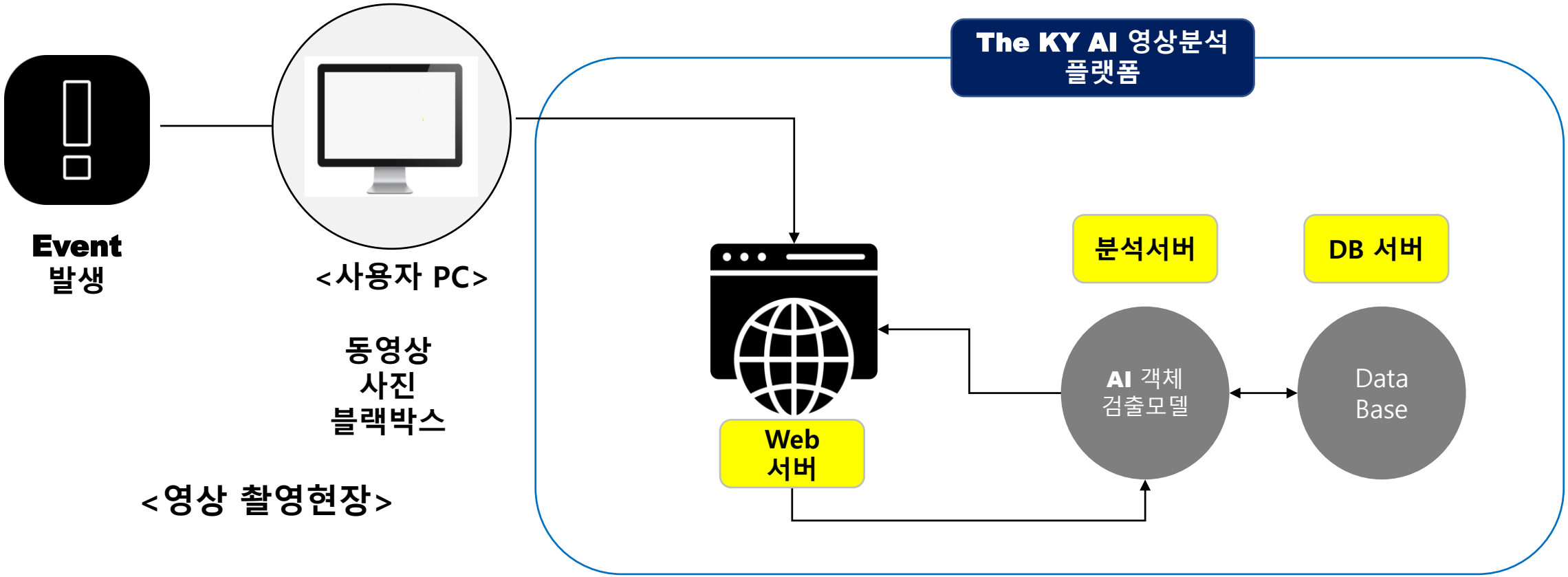
AI 모델 관련 후속 작업

- 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

프로젝트 수행결과

서비스 FLOW

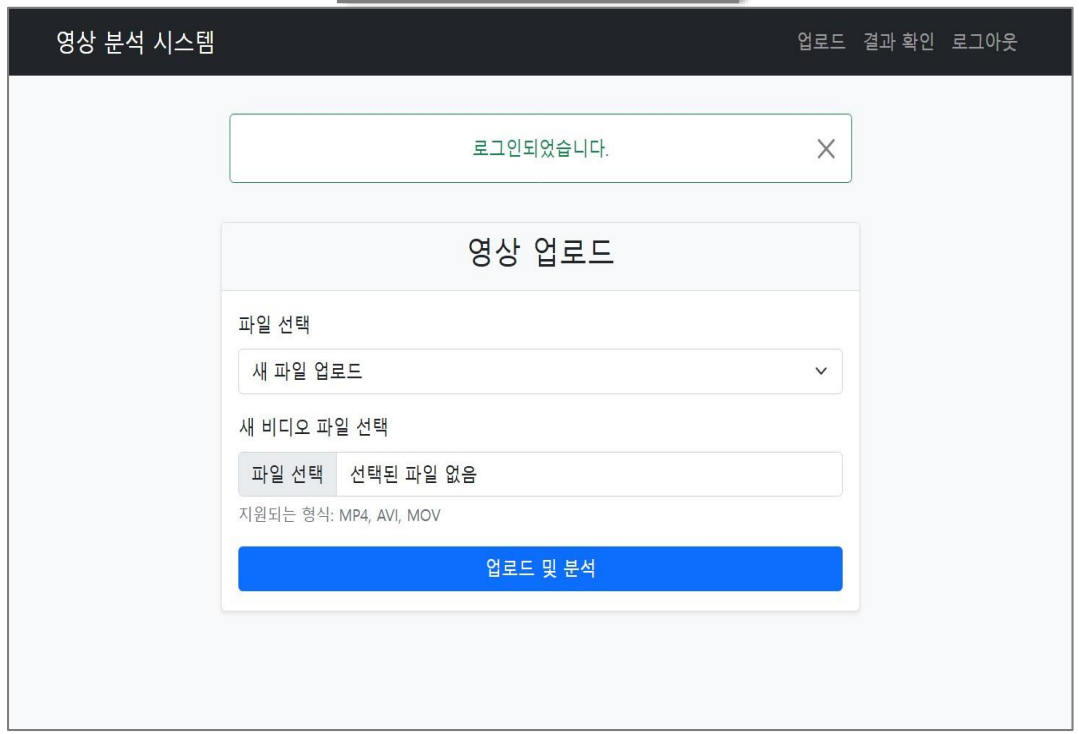


프로젝트 수행결과

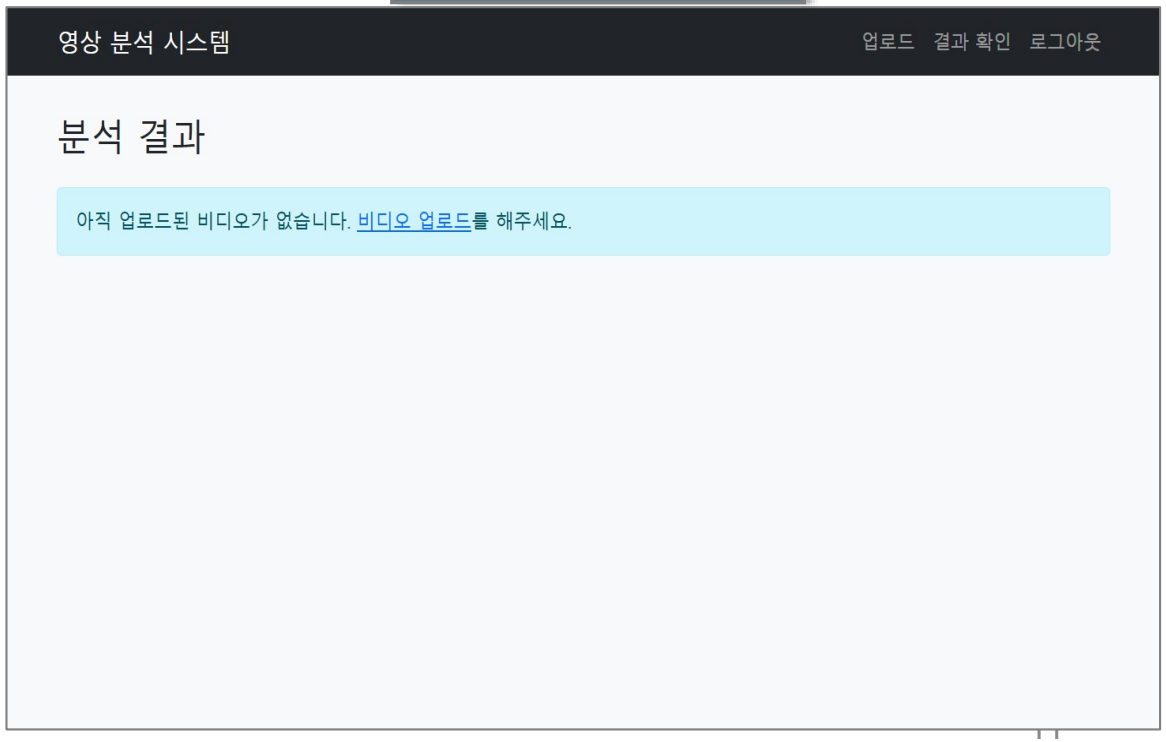
Web 서버

영상업로드/ 분석결과 반환의 사용자 I/F 제공

영상 업로드



분석결과



프로젝트 수행결과

분석서버

Flask 서버와 Yolov8의 분석모델 결합으로,교통영상의 AI 모델에 기반한 객체탐지 가능

분석모델(YOLOV8)

- 2023년 1월 10일, YOLOv8 Ultralytics에 의해 출시,
  컴퓨터 비전의 특정 작업에 특화된 다양한 모델 제공
  ※ YOLO) Ultralytics에서 개발한 최신버전의 객체탐지(object detection) 모델

주요 기능	세부내역
고급 백본 및 넥 아키텍처	YOLOv8 최첨단 백본 및 아키텍처를 채택하여 특징추출 및 객체감지 성능이 향상
앵커 프리 스플릿 Ultralytics 헤드	앵커 기반 접근 방식에 비해 더 나은 정확도와 효율적인 탐지 프로세스에 기여
최적화된 정확도-속도 트레이드오프	정확도와 속도 간의 최적의 균형을 유지하는 데 중점을 두어 다양한 애플리케이션 영역의 실시간 물체 감지 작업에 적합
다양한 사전 학습 모델	다양한 사전 학습 모델을 제공하므로 특정 사용 사례에 적합한 모델을 쉽게 찾을 수 있음.

객체 감지

인스턴스 분할

포즈& 키포인트감지

방향성 객체 감지

분류

▷작동모드와 호환

모델	파일 이름	작업	추론	유효성 검사	교육	내보내기
YOLOv8	yolov8n.pt yolov8s.pt yolov8m.pt yolov8l.pt yolov8x.pt	탐지	✅	✅	✅	✅
YOLOv8-cls	yolov8n-cls.pt yolov8s-cls.pt yolov8m-cls.pt yolov8l-cls.pt yolov8x-cls.pt	분류	✅	✅	✅	✅
YOLOv8-obb	yolov8n-obb.pt yolov8s-obb.pt yolov8m-obb.pt yolov8l-obb.pt yolov8x-obb.pt	방향 탐지	✅	✅	✅	✅
YOLOv8-pose	yolov8n-pose.pt yolov8s-pose.pt yolov8m-pose.pt yolov8l-pose.pt yolov8x-pose.pt yolov8x-pose-p6.pt	포즈/키포인트	✅	✅	✅	✅
YOLOv8-seg	yolov8n-seg.pt yolov8s-seg.pt yolov8m-seg.pt yolov8l-seg.pt yolov8x-seg.pt	인스턴스 세분화	✅	✅	✅	✅

프로젝트 수행결과

분석서버

분석모델(YOLOV8)

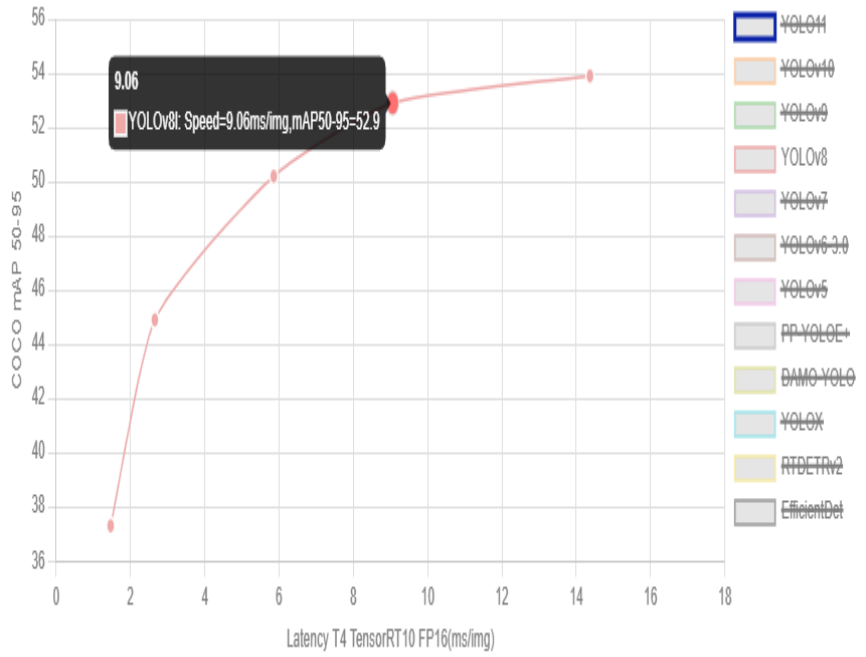
▷ COCO 데이터셋에서 학습된 YOLO8 성능

성능

탐지(COCO) | 감지(Open Images V7) | 세분화(COCO) | 분류(이미지넷) | 포즈 (COCO) | OBB(DOTAv1)

사전 학습된 80개의 클래스를 포함하여 COCO에서 학습된 이러한 모델의 사용 예제는 탐지 문서를 참조하세요.

모델	크기 (픽셀)	mAPval 50-95	속도 CPU ONNX (ms)	속도 A100 TensorRT (ms)	매개변수 (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8



프로젝트 수행결과

분석서버

FLASK

- Python 기반의 경량 웹 프레임워크(※ WAS 역할)
- 웹서버를 구축하거나 Restful API 만들 때 사용
- 프런트 엔드(HTML)와 백엔드(비즈니스 로직)를 연결하는데 사용

<Flask 애플리케이션의 기능>

YOLOV8로 학습된 모델 사용을 사용하여

✓ 이미지 또는 비디오 데이터 처리하기

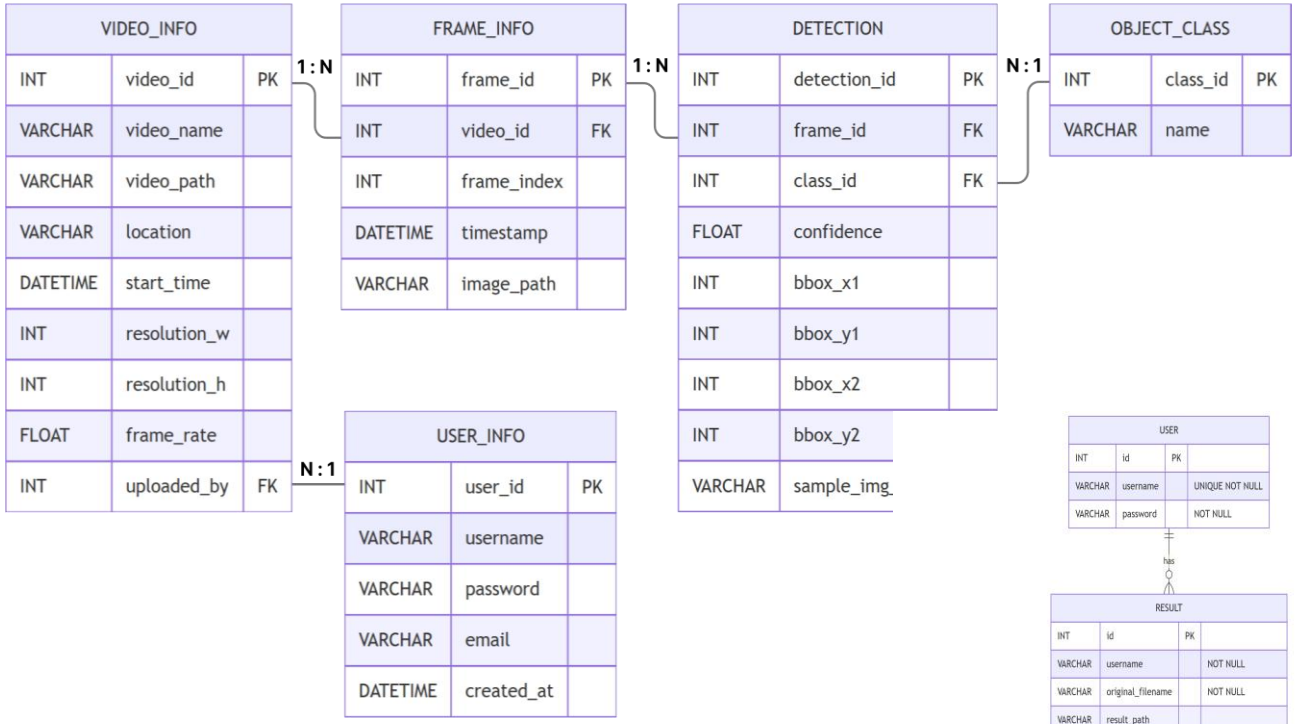
✓ 처리된 결과를 웹 페이지에 출력하기

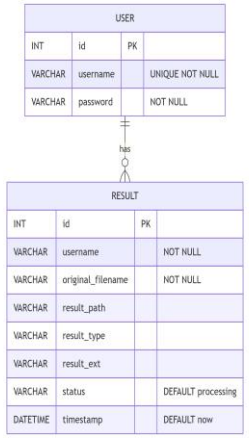
- **결론)** FLASK는 웹 요청을 처리하고, Yolov8은 이미지/영상에서 객체를 탐지하는 역할을 하여, 둘을 함께 사용하면 AI 기반 웹 애플리케이션을 만들 수 있음.(ex;교통영상분석 플랫폼, 사람/동물 탐지기 등)

프로젝트 수행결과

DB서버 >ERD

Q)USER와 RESULT 테이블의 삽입 위치?





1.USER_INFO → VIDEO_INFO (1:N)

- USER_INFO : 로그인 가능한 사용자 정보 저장
- 각 사용자가 업로드한 영상은 VIDEO_INFO.uploaded_by 외래키로 참조
→ 하나의 사용자(user_id)가 여러 영상을 가질 수 있음

2.VIDEO_INFO → FRAME_INFO (1:N) 9

- VIDEO_INFO : 분석할 영상 한 건당 기본 정보(영상 ID, 이름, 경로, 해상도, 프레임레이트 등) 저장
- 영상 하나에 속하는 개별 프레임들은 FRAME_INFO 에 여러 건으로 저장
→ 하나의 영상(video_id)이 여러 프레임(frame_id)을 가짐

3.FRAME_INFO → DETECTION (1:N)

- FRAME_INFO : 각 프레임의 순번(frame_index), 타임스탬프, (논의 필요)프레임 이미지 경로 등 저장
하나의 프레임에서 검출된 객체들이 DETECTION 에 각각 레코드로 저장
하나의 프레임(frame_id)이 여러 검출(detection_id)을 가짐

ION → OBJECT_CLASS (N:1)

ETECTION : 각 검출의 바운딩 박스 좌표와 신뢰도, (논의 필요)샘플 이지 경로 등 저장
출 정보 각각은 반드시 하나의 객체 종류(OBJECT_CLASS)에 속해야
→ 검출은 객체 클래스로 모여드는 N : 1 관계
반대로 OBJECT_CLASS 테이블에는 "person", "car" 같은 객체 종류만 유하게 관리 → 하나의 클래스가 여러 검출에 참조될 수 있음

프로젝트 수행결과

서비스 사용 Process

1

사용자 영상수집

- 드론/카메라/차량 블랙박스에서 교통영상 수집

2

영상 업로드

- AI 영상분석 플랫폼에 영상 업로드

3

AI 영상 분석

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

4

결과 전송

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

프로젝트 수행결과

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

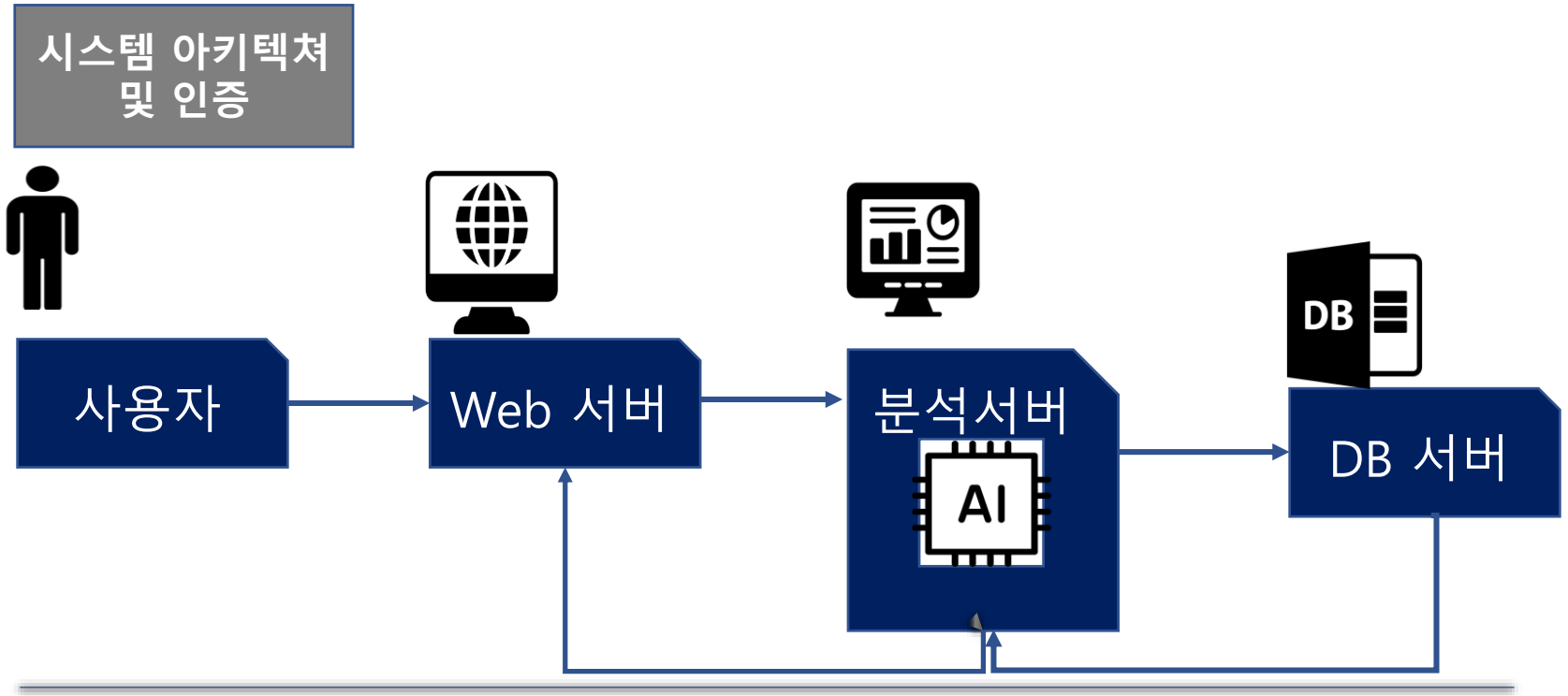
차량사고 사전예방

- 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

프로젝트 수행결과



<데이터 흐름도>

동영상 업로드

분석요청

분석결과 반환 & REPORT제공

분석 결과 DB 저장

- **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

프로젝트 수행결과

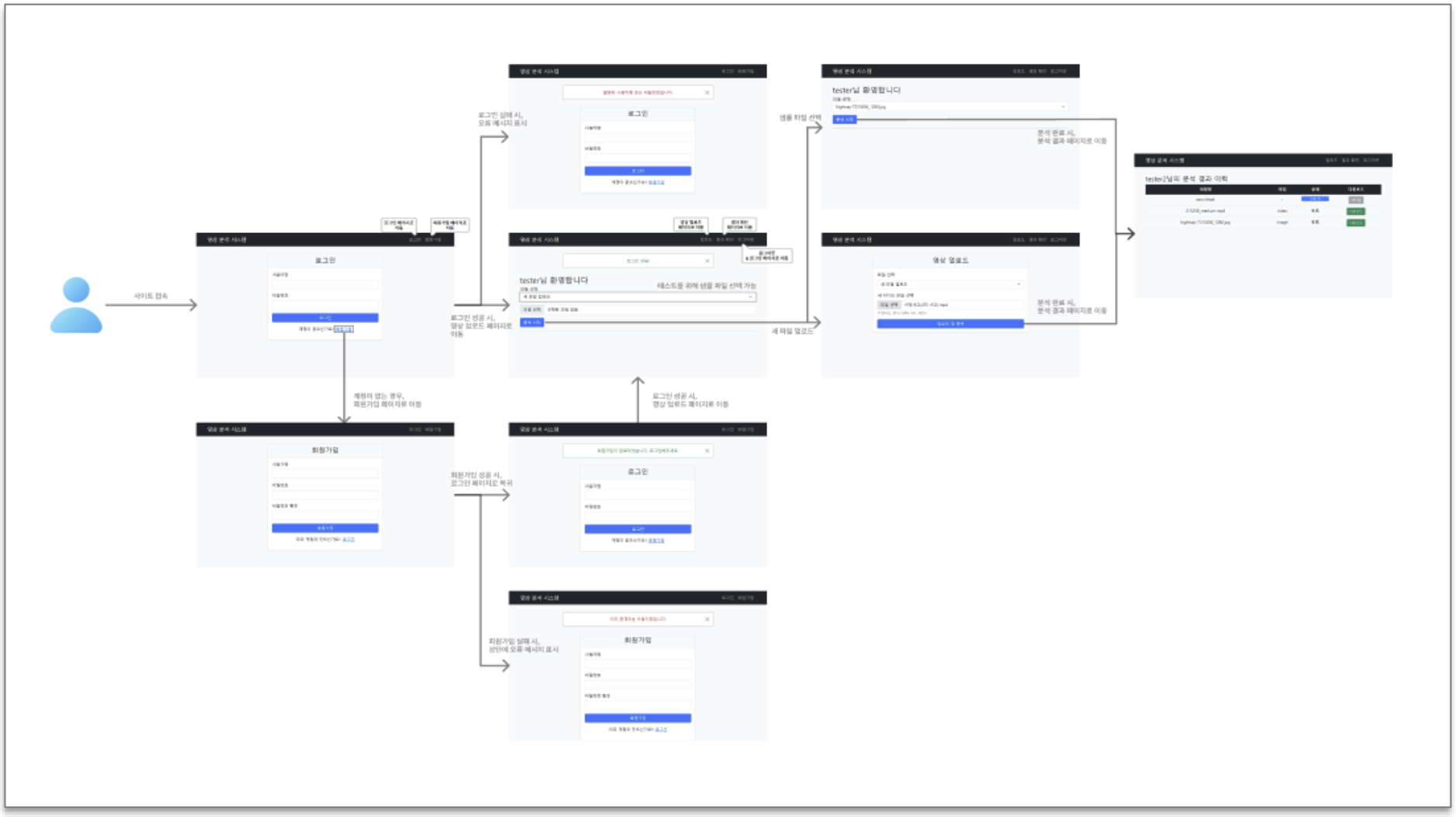
핵심 UI 화면흐름

사용자 접속

영상 업로드 및 분석

분석결과 반환

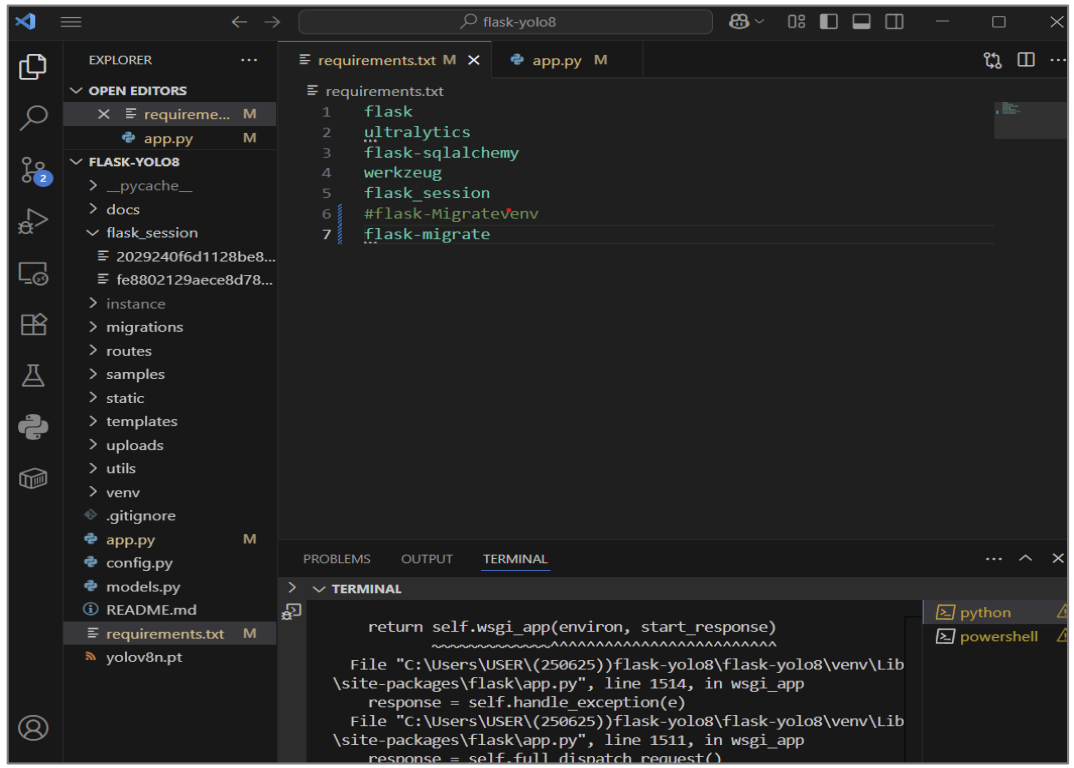
분석결과 Report



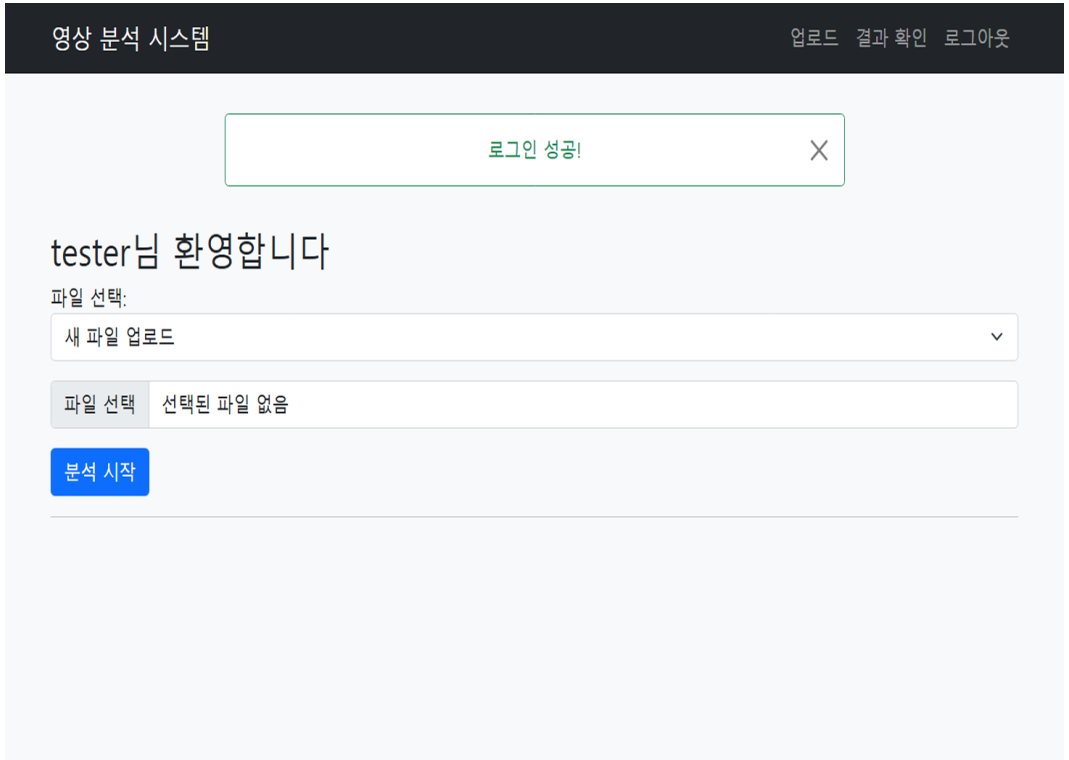
프로젝트 수행결과

개발 테스트

개발코드



테스트 환경



프로젝트 수행결과

개발 테스트

다운로드

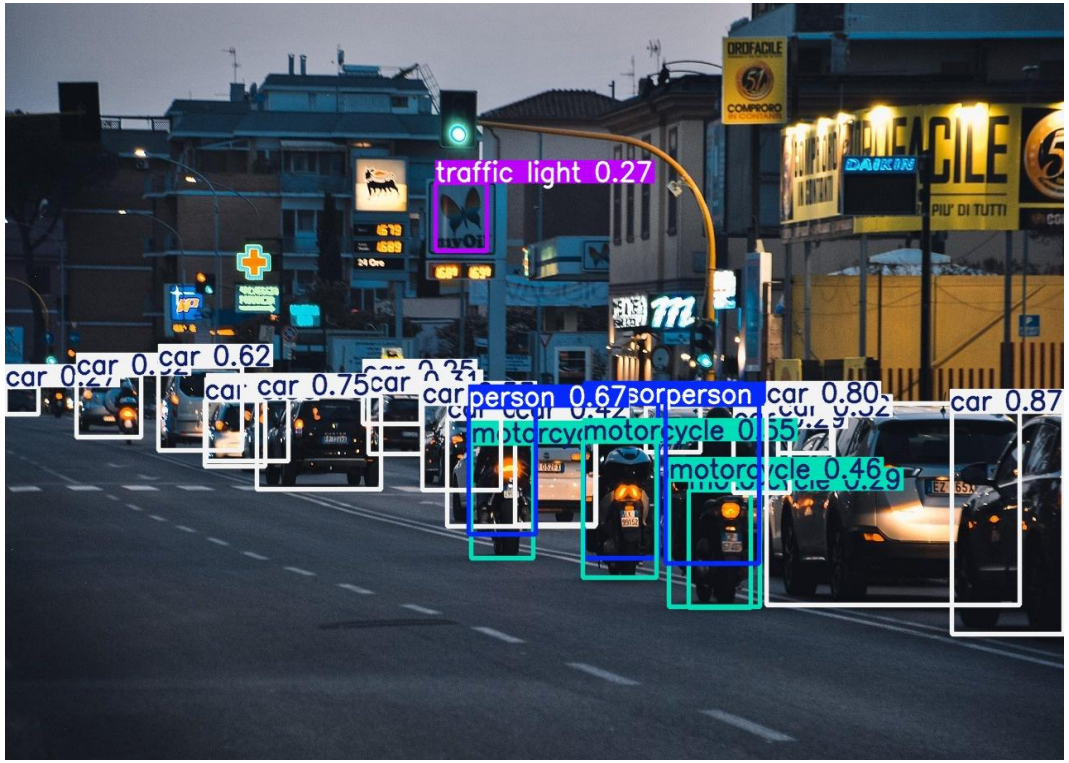
분석화면
(이미지)

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



프로젝트 수행결과

개발 테스트

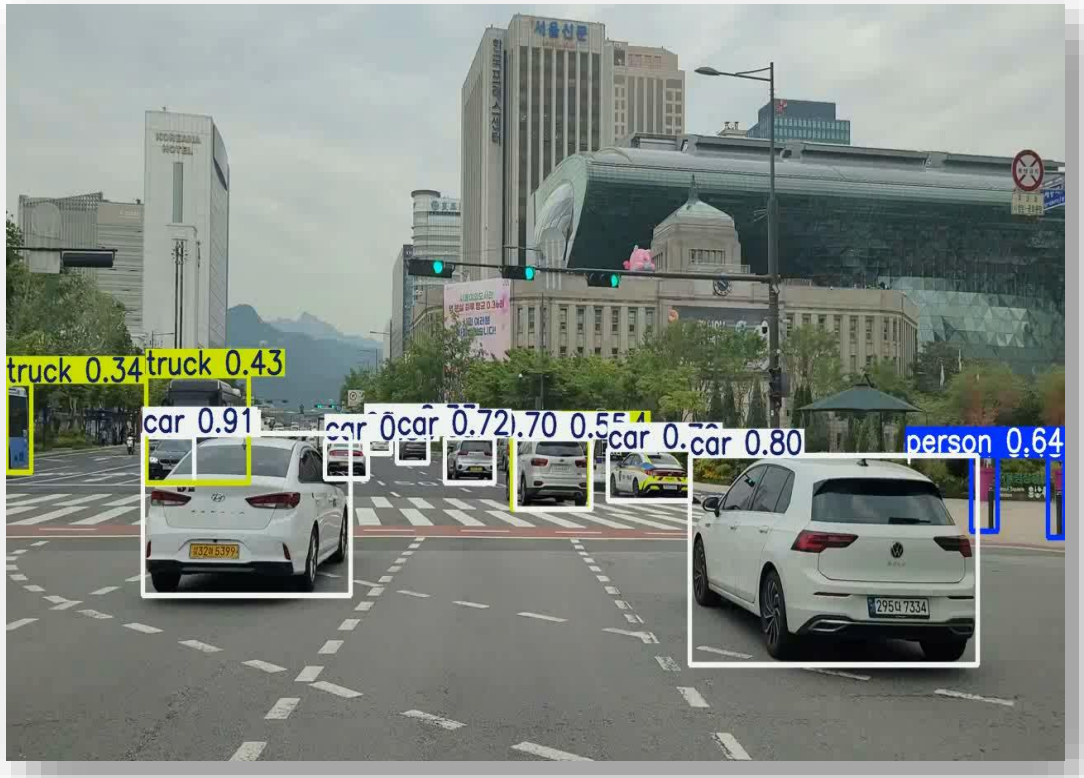
다운로드

분석화면
(동영상)

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

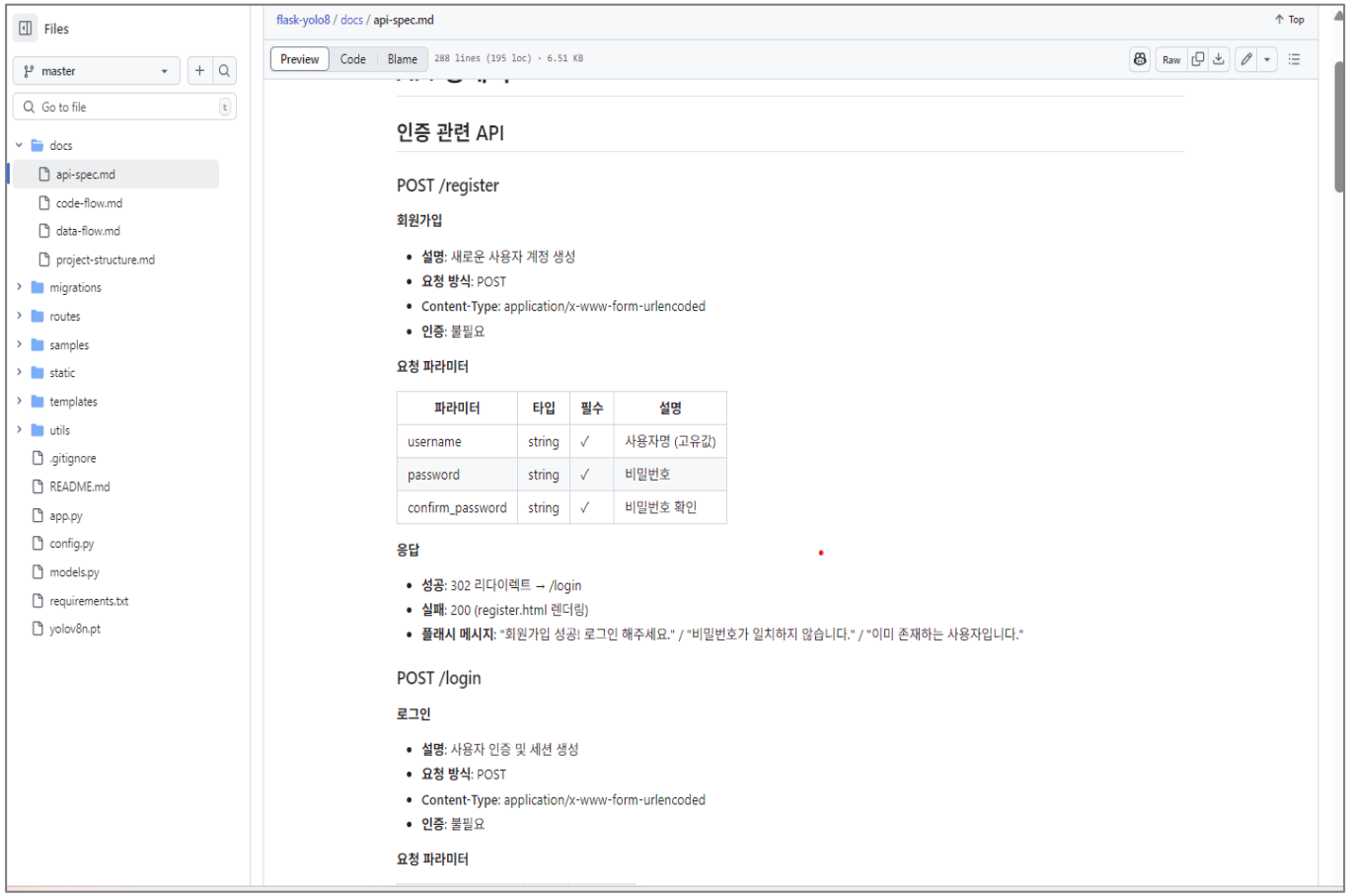


프로젝트 수행결과

현재 **GIT HUB** 상에 **API** 명세서등의 문서를 작성 **&** 관리하고 있음.

API 명세서

- **api-spec.md)** API 명세서
- **code-flow.md)**코드 흐름 설명
- **data-flow.md)**데이터 흐름 다이어그램
- **project-structure.md)**프로젝트 구조 및 파일 구조



기대효과 및 향후계획

기대효과

- 교통사고 영상 분석 자동화
- 인터넷을 통한 사고영상 업로드&Report 조회 가능

고도화 방향

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

자체평가 의견

최소희

- 현재 세션 기반이지만 규모가 커진다면 토큰 구현해볼 수 있음
- 대용량 미디어 파일 저장과 제공을 위해서 AWS S3 사용해볼 수 있음
- 추후 DetectoRS와 VTN 사용해서 사고 분석 프로그램으로 확장해볼 수 있음

이경윤

- 처리 중 상태 표시, 로그인 피드백, 파일명 필터링 등 작은 기능들이 실제 사용자 경험에 큰 영향을 준다는 것을 경험함
- Markdown 기반 기술 문서를 정리하면서, 코드만으로는 부족했던 팀원 간 이해를 돕는 소통 도구로서의 문서화의 중요성을 체감함
- UI 완성도나 분석 결과의 시각화 등 디자인적 디테일을 더 개선하지 못해 아쉬움

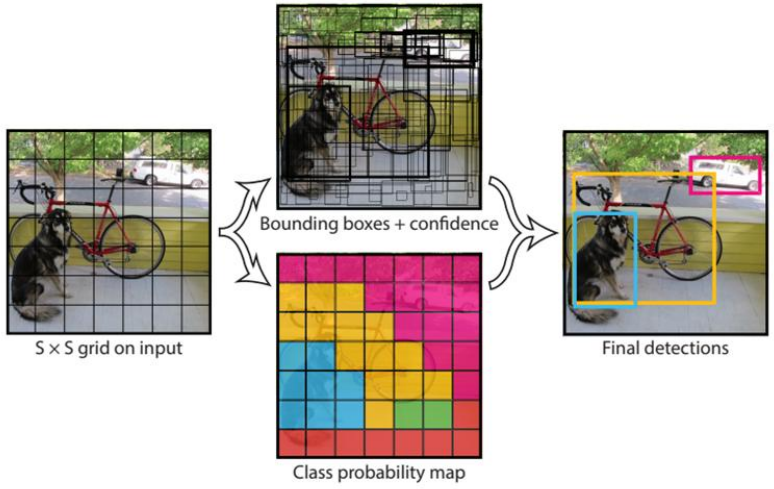
고수혁

- VSCode상의 개발코드, Flask, Yolo, COCO dataset 등 전문적인 객체검출 기술을 맛보고 배울 수 있어서 유익한 시간이었음.
- 팀 프로젝트를 수행함에 있어, 팀원간 커뮤니케이션/협업/산출물 완성이라는 기본적인 자세가 더 필요하다고 느꼈음.
- 스스로 개발연습 및 인공지능에 대하여 자기계발을 지속해야 할 필요성을 많이 느낌.

참고자료(Appendix)

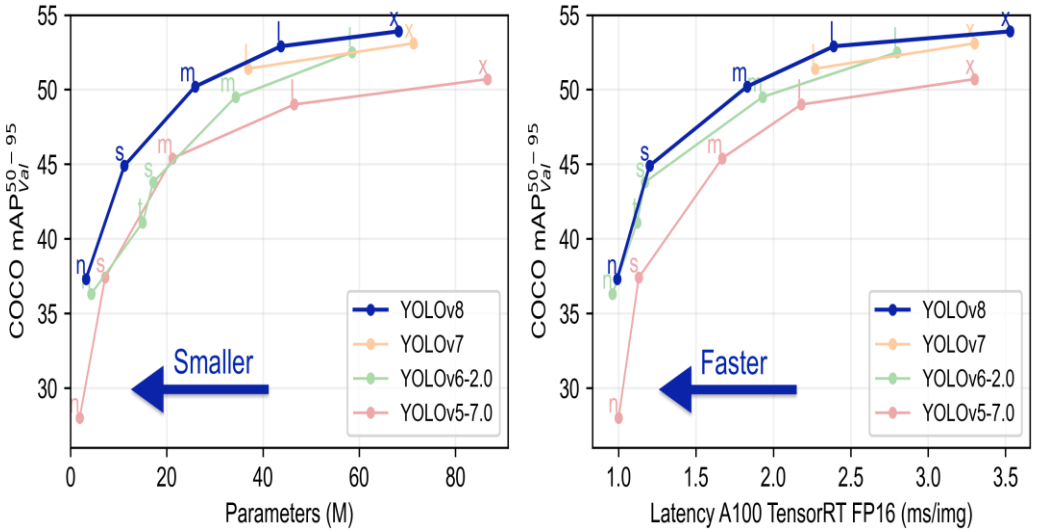
<주요 YOLO 버전과 COCO dataset>

버전	세부내역
YOLOV1	• 2016년에 발표된 최초 버전으로, 실시간 객체 검출을 위한 딥러닝 기반의 네트워크 • 객체감지를 위한 단일회귀 문제로 정의
YOLOV5	• 2020년 6월에 발표된 버전으로 YOLOv4와 비교하여 객체검출 정확도에서 10% 이상 향상되었으며, 더 빠른 속도와 더 작은 모델 크기를 가짐 • Pytorch 기반으로 훈련이 쉽고 코드가 간단
YOLOV8	• 2023년 1월 발표된 버전으로, 개체 감지, 인스턴스 세분화, 포즈추정 및 이미지 분류 모델을 train하기 위한 통합 프레임워크로 구축됨 • 배포 유연성이 뛰어남.



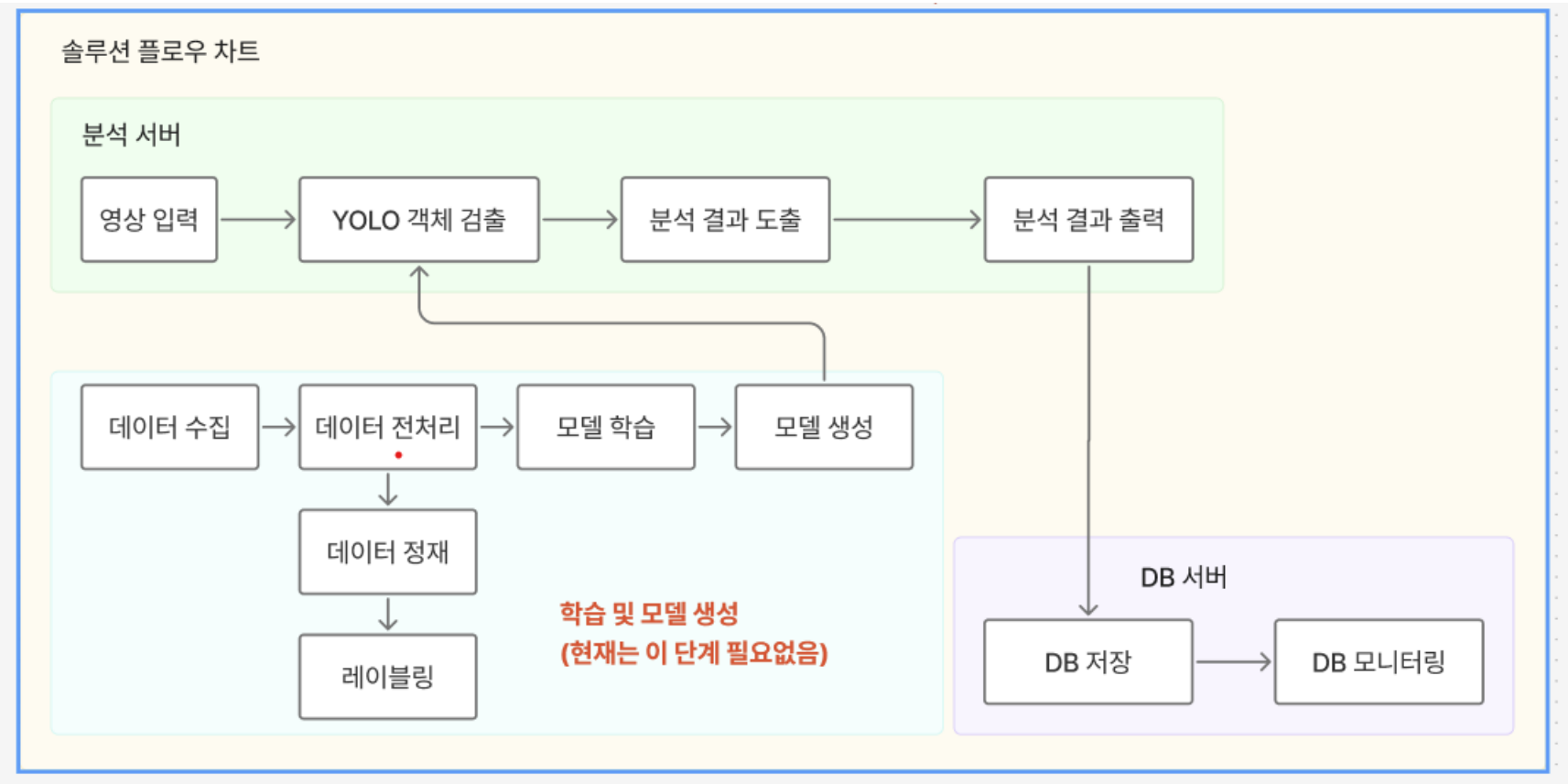
COCO

- 객체감지, 세분화, 캡션에 사용되는 대규모 데이터 셋트
- 33만개의 이미지 포함, 20만개의 이미지에는 객체감지, 분할 및 캡션작업을 위한 주석 포함
- 객체 감지를 위한 평균 평균 정밀도 (mAP), 세분화 작업을 위한 평균 평균 정확도 (mAR) 등 표준화된 평가 지표를 제공하므로 모델 성능을 비교하는 데 적합



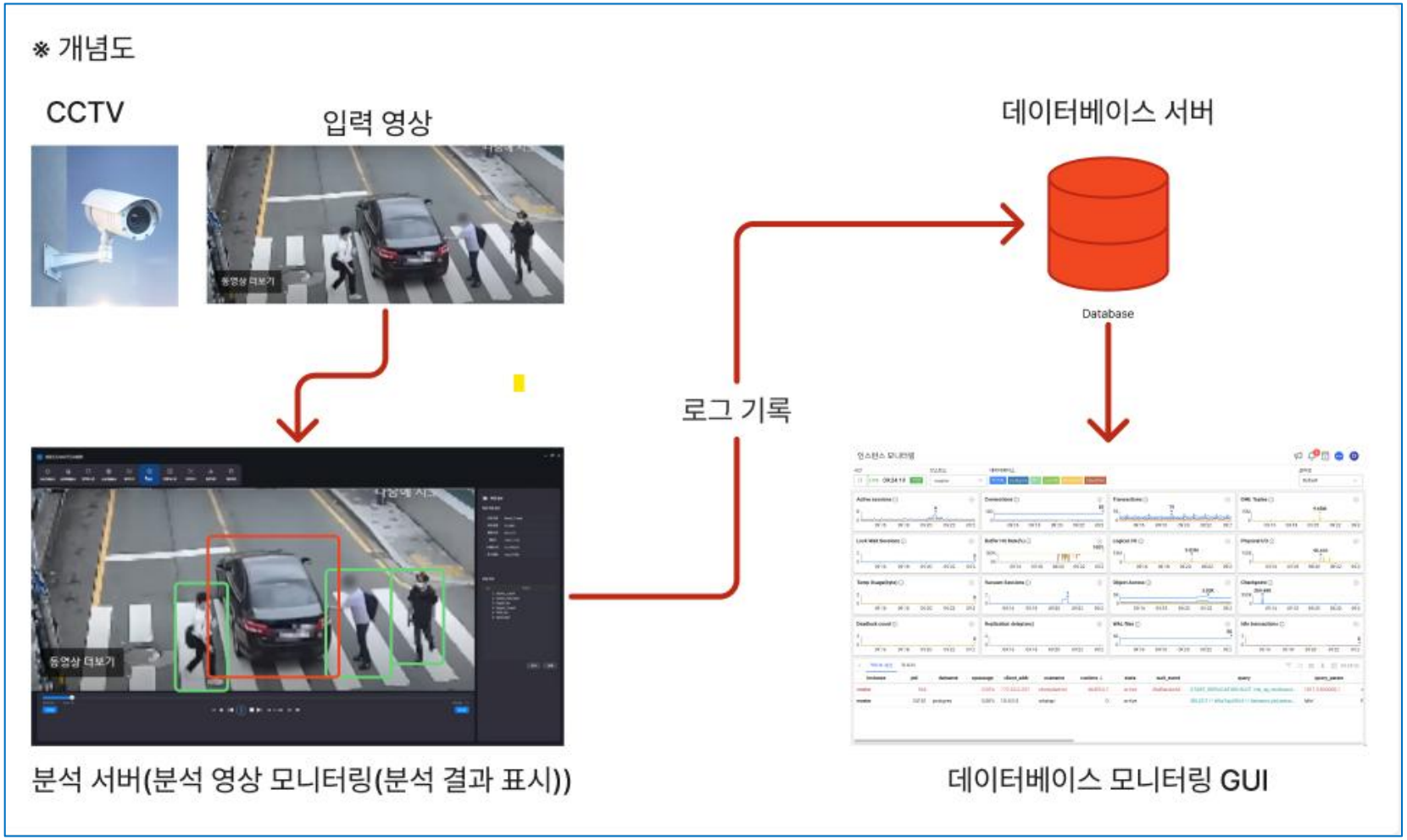
참고자료(Appendix)

<도로 교통영상 분석 솔루션 Flowchart>



참고자료(Appendix)

<도로 교통영상 분석 솔루션 개념도>



참조자료(Reference)

- Notion) https://www.notion.so/yunslog/
- Github) https://github.com/Clean314/flask-yolo8
- AI-HUB> 교통사고 영상 데이터
  ✓ URL) AI-Hub
- Yolov8) You Only Look Once – 나무위키
- 세션 인증 방식 VS Token 인증방식(인증과 인가)
- Yolo) https://velog.io/@qtly_u/n4ptcz54
- COCO)https://docs.ultralytics.com/ko/datasets/detect/coco/#what-are-the-key-features-of-the-coco-dataset

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