AI와 함께 만든 기능, 직접 확인한 작동 범위

01 출발점과 나의 역할

프로그래밍 정식 교육 경험이 부족한 상태에서 팀 프로젝트에 참여했습니다. 터미널 사용과 수업 예제 연결부터 AI의 도움을 받았고, 작동하는 결과물을 먼저 만든 뒤 화면과 검증 기록을 보며 수정했습니다.

담당: 화면·영상 업로드·차량 감지 사진 비교와 저장, 검증 요청, 배포 준비 및 발표자료 지원.
팀 공동 기여: 문답·백엔드, 사고 기준·법규·판례 자료, 별도 AI 검색 프로토타입.

02 AI 활용과 피드백의 반복

01 구체적인 요청
기존 코드를 유지하며 담당 기능 구현 요청. 결과를 사진 중심으로 바꾸고 원본 비교·확대 기능 추가.

02 AI의 구현 지원
Codex가 코드 작성과 자동 검사 수행. YOLO로 영상 프레임의 차량을 감지하고 사진 생성.

03 결과 확인·수정
화면을 보고 질문과 안내 문구 수정 요청. 실제 영상 처리 시간, 재시작과 백업 복원 확인 요청.

03 확인한 기능과 배운 점

영상 업로드 → 첫 차량 감지 사진 생성 → 원본 비교·확대 → 사진 저장
“차량을 찾은 장면”과 “충돌 순간 판정”은 다릅니다. 현재 구현에 맞게 화면과 발표 문구를 고쳤습니다. AI가 생성한 코드의 전체 원리를 독립적으로 설명할 수 있는 수준을 달성했다고 주장하지 않습니다.

검증 기록과 다음 학습 계획

01 측정 조건을 함께 남긴 결과

2026년 10월 23일, macOS·Python 3.12·CPU 환경에서 기록한 로컬 시험입니다. 첫 감지 장면에서 처리를 종료한 결과이며, 전체 영상 분석 속도나 사고 판정 정확도가 아닙니다.

확인 항목	결과와 해석
첫 차량 감지 최초 요청	1차 3.168초 / 2차 1.886초 모델 최초 로딩을 포함한 각 요청
시험 영상	약 26.33초·790프레임 수업 영상 첫 프레임에서 차량을 감지하고 종료
재시작·백업 복원	별도 시험 폴더로 복원 후 자료 비교 2차 시험의 영상·사진 등 16개 파일 동일

02 구현한 범위와 남은 과제

현재: 문답과 영상 기능 연결, 차량 감지 사진 비교·확대·저장, 로컬 오류·성능·자료 보존 검증.
미구현·미검증: 충돌 순간 자동 판정, 영상 기반 과실 계산, 공개 클라우드 운영과 장애 복구, 실제 사용자 효과. AI 검색 연구는 현재 웹앱과 별도입니다.

03 다른 학습자도 따라 할 수 있는 과정

담당 기능과 완료 기준 정하기 → 기존 코드·예제로 작은 기능 실행 → 화면에서 구체적인 수정 의견 전달 → 잘못된 입력·저장·재시작 확인 → 시험 조건과 한계 기록.
저의 학습 성과는 AI에 구현을 요청하고 확인할 기준을 구체화한 경험입니다. 코드 작성과 자동 검사는 AI의 도움을 받았습니다.

다음 계획: 사고 경험자 50명 인터뷰를 목표로 어려운 단계를 확인하고 화면을 개선할 예정입니다. 인터뷰 완료 실적이나 검증된 만족도·시간 절감 수치는 없습니다.

기록·출처
팀 개발 기록: docs/DEPLOYMENT_BENCHMARK.md (2026-10-23).
측정 원시 기록: docs/benchmarks/mac_run1.json, mac_run2.json.
YOLO 공식 문서: https://docs.ultralytics.com/
기준 자료 참고: https://accident.knia.or.kr/
이 문서는 제출 원고의 보충 요약입니다. 제3자 사고 영상·사진, 팀 발표자료 원본, 타요 캐릭터, 대화 캡처 및 연락처는 포함하지 않았습니다.