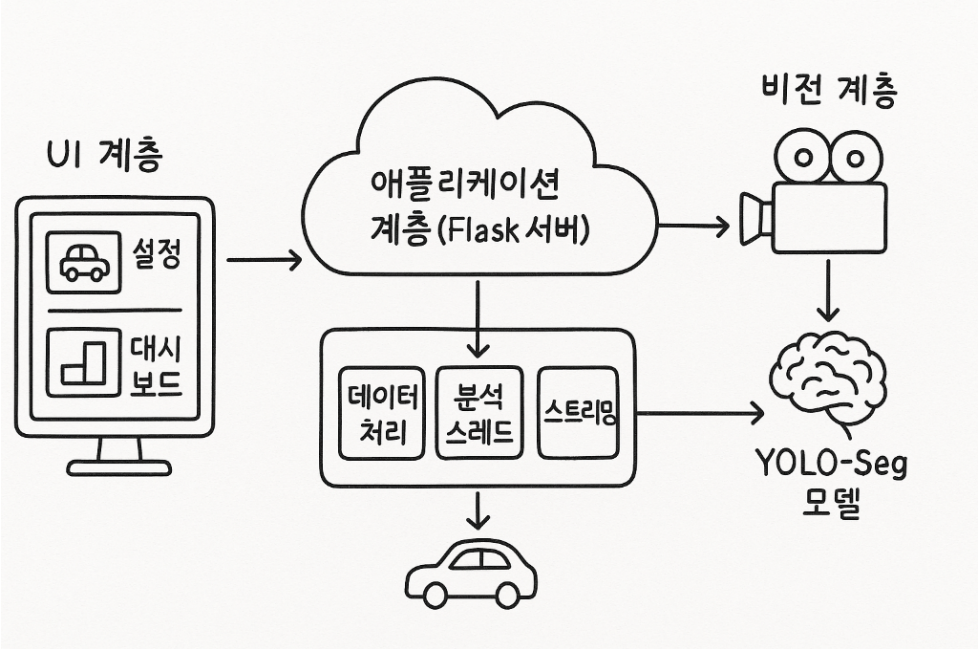
프로젝트 기술서

팀 명
가 나 다 라

프로젝트명	스마트 주차장 관리 시스템 (Smart Parking Management System)
수행기간	2025.09.18. ~ 2025.09.24.
담당역할	**최지은 (백엔드 및 시스템 통합)** • 역할: Flask 서버(app.py)를 개발하여 UI와 YOLO 로직을 연결하는 백엔드 시스템을 구축합니다. • 주요 임무: 비디오와 주차 공간 정보 업로드 처리, 백그라운드 스레드(영상 분석, 예약 관리) 실행, 실시간 비디오 스트리밍 API 개발, 최종 데이터베이스(예정) 연동. **이하은 (프론트엔드 개발)** • 역할: 사용자가 직접 조작하는 UI(index.html, dashboard.html)를 개발합니다. • 주요 임무: 비디오 업로드 및 캔버스 드로잉 기능 구현, 실시간 현황을 보여주는 대시보드 레이아웃 설계, 서버 API 호출을 통한 데이터 동기화, 사용자 친화적인 디자인 적용. **정은진 (컴퓨터 비전 및 모델 최적화)** • 역할: YOLO 모델을 시스템에 통합하고, 주차 상태를 판단하는 핵심 로직을 담당합니다. • 주요 임무: YOLO-Seg 모델 로드 및 최적화, 차량 마스크 추출 및 필터링, 마스크 간의 IoU(겹침 비율) 계산 로직 구현, 시스템의 정확성과 안정성 향상.
수행목표	**1. 주요 목표:** 컴퓨터 비전 기술을 활용하여 실시간으로 주차 공간의 점유 현황을 정확하게 파악하고, 사용자에게 시각적으로 제공하는 웹 기반 솔루션을 개발합니다. **2. 세부 목표:** 사용자가 직접 주차 공간을 정의할 수 있는 직관적인 UI를 제공합니다. YOLO 세그멘테이션 모델을 활용하여 차량을 픽셀 단위로 정밀하게 감지합니다. 감지된 차량 마스크와 주차 공간 마스크의 겹침을 분석하여 점유 상태를 판단합니다. 백엔드와 프론트엔드가 실시간으로 데이터를 주고받는 안정적인 시스템을 구축합니다.
사용 기술	백엔드: Python 3.10+, Flask (웹 프레임워크), OpenCV (비디오 처리), threading (백그라운드 작업), Werkzeug (파일 업로드).

	컴퓨터 비전: Ultralytics YOLO (세그멘테이션 모델), NumPy (수치 연산). 프론트엔드: HTML5, CSS3, JavaScript (드래그 앤 드롭, API 호출), Tailwind CSS (스타일링).
세부수행내용	
구 성 도	
상세 내용	1. 데이터 흐름 및 시스템 구조 • index.html에서 사용자가 비디오와 주차 공간 좌표를 POST 요청으로 서버에 전송합니다. • app.py는 이 데이터를 받아 백그라운드 스레드를 시작합니다. • analyze_video 스레드는 비디오 프레임마다 YOLO-Seg 모델을 실행하여 차량을 감지하고, PARKING_SPOTS의 상태를 갱신합니다. • dashboard.html은 API 엔드포인트(/yolo_feed, /video_feed, /parking_status)를 통해 실시간 데이터를 받아와 두 개의 동영상 화면과 주차 현황 정보를 표시합니다. 2. 비전 로직 • 객체 감지: YOLO 모델은 입력 프레임에서 차량(car, bus, truck)을 인식하고, 각 차량에 대한 정확한 픽셀 마스크를 반환합니다. • 마스크 기반 점유 판단: • create_mask_from_coords(): 사용자가 그린 주차 공간의 사각형 좌표를 픽셀 마스크로 변환합니다. • calculate_iou_from_masks(): YOLO가 감지한 차량 마스크와 주차 공간 마스크의 **겹치는 픽셀 비율(IoU)**을 계산합니다.

	• 이 IoU 값이 미리 설정된 임계값(IOU_THRESHOLD)보다 높으면 해당 주차 공간을 점유된 것으로 최종 판단합니다. 3. 안정성 및 최적화 • 다중 스레드: 영상 분석, 예약 모니터링, 웹 스트리밍을 각각 독립적인 스레드로 분리하여 시스템이 동시에 여러 작업을 처리하도록 합니다. • 데이터 공유: latest_frame 및 latest_yolo_frame과 같은 공유 변수와 threading.Lock을 사용하여 스레드 간에 안전하게 데이터를 주고받습니다. • 점유 지연: 차량이 잠시 감지되지 않더라도 즉시 상태를 바꾸지 않고, 일정 시간(OCCUPIED_RELEASE_DELAY) 동안 점유 상태를 유지시켜 시스템의 안정성을 높입니다.
참조	**Github : 프로젝트 깃허브** **PPT 및 결과물 :** **https://docs.google.com/presentation/d/11PgjsBpgpI2kGNIIB0MSXQI-zZ8Acih5GfWfCS4hQgI/edit?usp=sharing**