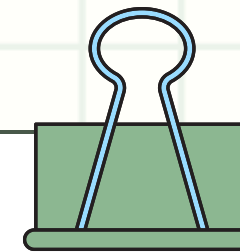
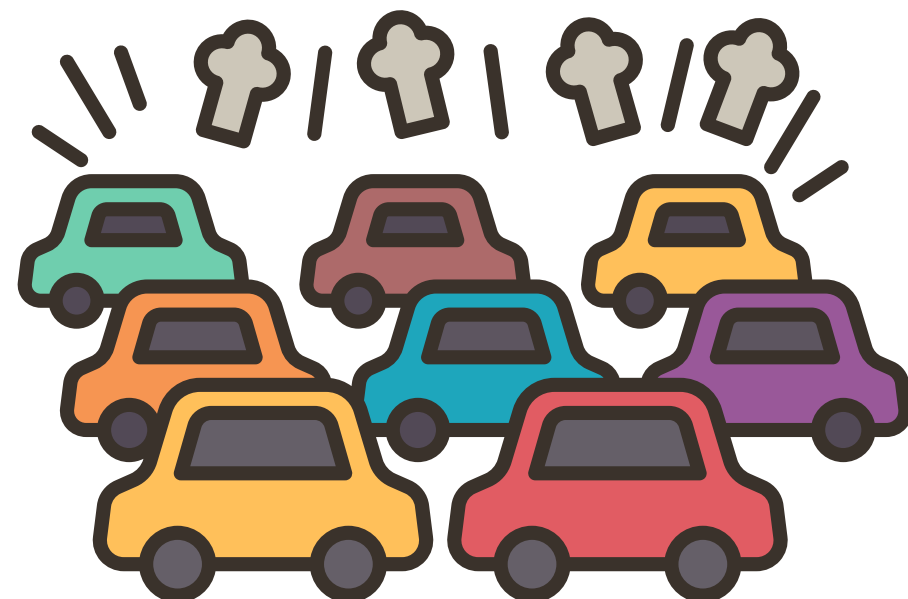


SW파일럿 Team Project 기획발표

AI 기반의 유동적인 신호등 시스템 구축
(AI BASED FLEXIBLE TRAFFIC SYSTEM)



목차



01

프로젝트 개요

02

프로젝트 추진방안

03

프로젝트 요구사항

04

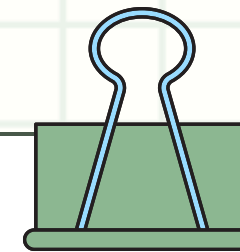
진행 계획

05

팀원 소개

06

기대효과



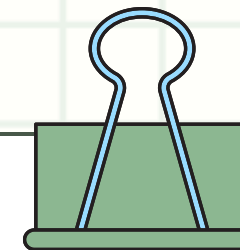
01 프로젝트 개요



도시 집중 현상



심각해지는 교통체증



01 프로젝트 개요

기존의 신호체계

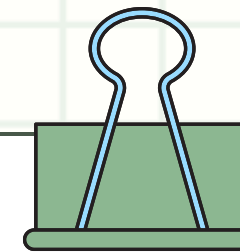
고정식(Fixed) 신호 방식	감응식(Actuated) 신호 방식
일정한 시간 주기로 고정	차량 감지 여부에 따라 실시간 변동
사전에 수집된 평균 교통량 데이터 기반	차량 검지기(루프 감지기)로 실시간 교통량 파악
<ul style="list-style-type: none">- 단순 구조- 유지비용 저렴	<ul style="list-style-type: none">- 실시간 대응 가능- 불필요한 대기 시간 최소화
<ul style="list-style-type: none">- 교통량 변화 반영 불가- 비효율적 대기 발생	<ul style="list-style-type: none">- 설치비용 높음- 검지기 오작동 시 마비 가능



자료 : <http://ops.fhwa.dot.gov>

루프 디텍터를 이용한 차량 감지

NEWSis.()



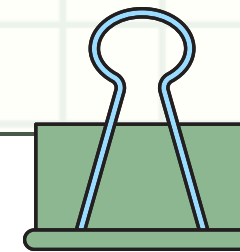
01 프로젝트 개요

그래서 생각했습니다.

- 교차로 상황을 실시간 인식하면서,
- 필요할 때만 신호하여 대기 최소화하고,
- 설치해야 할 인프라 부담을 낮추면서
- 인간의 관리 없이 자율적으로 제어되는

새로운 교통 시스템 솔루션을 만들 수 있을까?





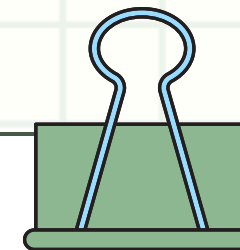
01 프로젝트 개요

AI 기반의 유동적인 신호등 시스템








프로젝트 목적

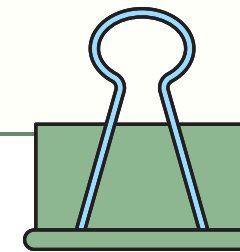
- AI가 사진에서 실시간 교통량을 인식
- 자체 알고리즘을 통해 신호 주기를 조정
- 지능형 자동 제어 시스템
 - 시뮬레이터 버전(MVP)을 구현
- 기존 방식의 한계를 극복
- 교통 문제를 해소하는 데 기여



01 프로젝트 개요

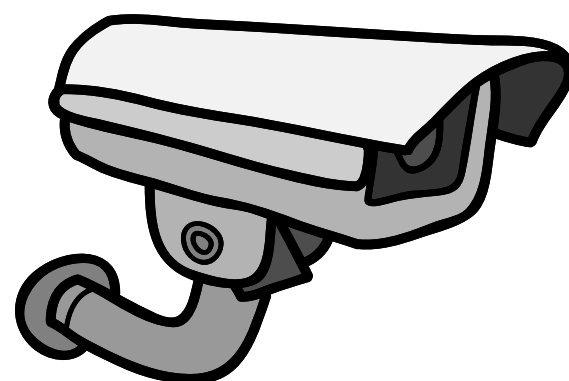
프로젝트 목표

	차량 인식	객체 탐지 모델로 차량 수 실시간 집계
	교통량 분석	동-서/남-북 방향별 차량 수를 슬라이딩 윈도우 방식으로 분석
	신호 조정 알고리즘	차량 수 비율 기반으로 신호 주기 자동 계산
	시뮬레이터 UI	PyQt로 신호등 애니메이션 + 교차로 UI 구현
	결과 시각화	실시간 대시보드 + 로그 저장 기능 제공



02 프로젝트 추진 방향

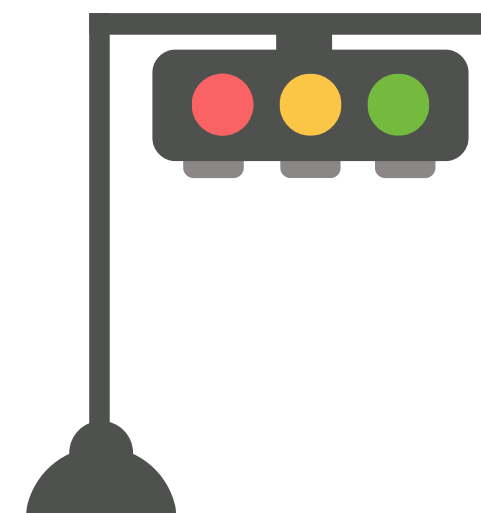
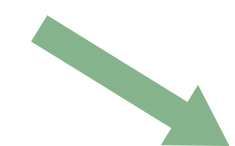
프로젝트 구조도



차량 감지 시스템
(CCTV 영상 분석)



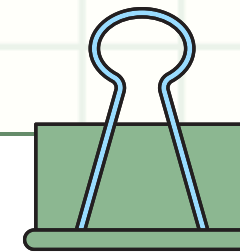
교통량 데이터 집계 및 해석



신호등 제어 시스템

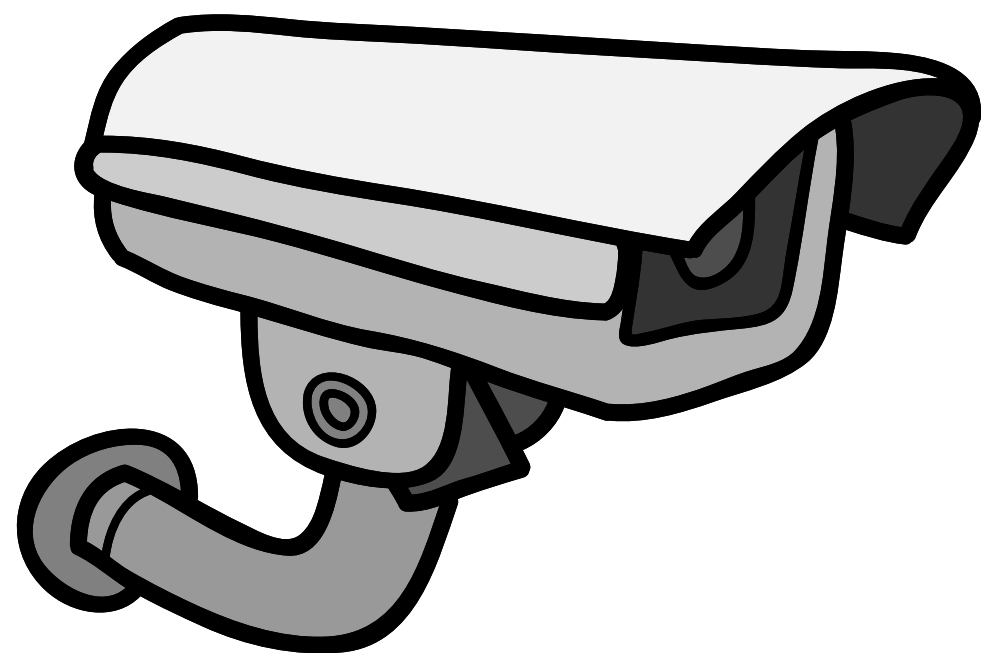


시각화 및 로그 시스템



02 프로젝트 추진 방향

모듈별 설명



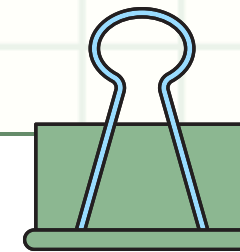
차량 감지 시스템
(CCTV 영상 분석)

기술 선택

- OpenCV + 딥러닝 기반 객체 탐지 모델
- YOLOv5 or YOLOv8 사용 예정

구현 요소

- CCTV 영상을 실시간으로 받아오는 스트림 처리
- 매 프레임에서 자동차 객체 탐지
- 관심영역(ROI) 내에 객체 수 계산
- 차량이 위치한 도로 방향 정보 tagging



02 프로젝트 추진 방향

모듈별 설명



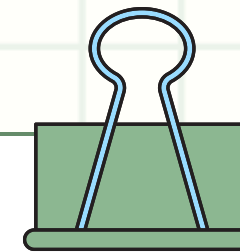
교통량 데이터 집계 및 해석

기술선택

- Pandas 라이브러리 활용

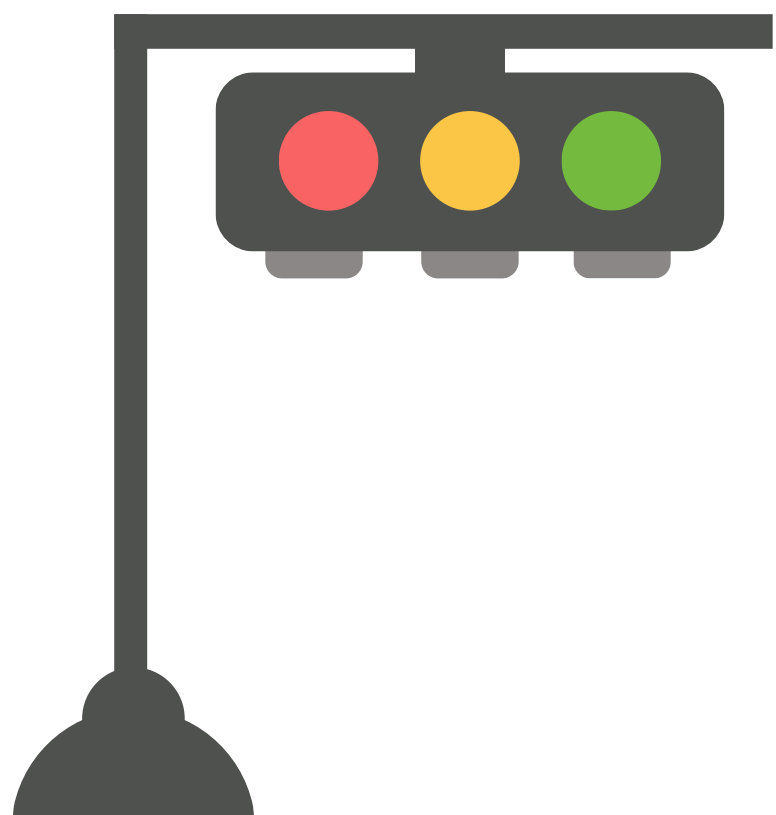
구현 요소

- 일정 시간 단위 (예 : 10초)로 방향별 차량 수 누적
- 테이블 데이터 형태로 데이터 전처리
- 신호 주기 조정 로직을 위한 입력값 구성
- 대시보드 시각화를 위한 입력값 구성



02 프로젝트 추진 방향

모듈별 설명



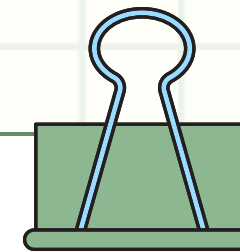
신호등 제어 시스템

기술선택

- PyQt 라이브러리 활용
- 신호 조정 로직
 - Rule based vs Reinforcement Learning

구현 요소

- PyQt 기반 GUI로 신호등 시뮬레이터 구축
- 신호 조정 로직 구현



02 프로젝트 추진 방향

모듈별 설명



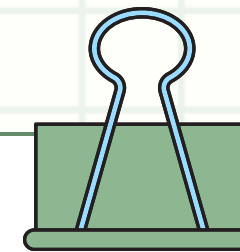
시각화 및 로그 시스템

기술선택

- PyQt 라이브러리 활용
- Matplotlib 라이브러리 활용

구현 요소

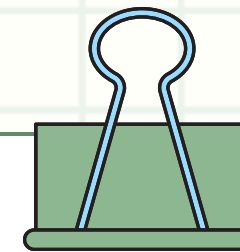
- 트래픽 대시보드 구현
- 각 방향 교통량 실시간 출력
- 현재 신호 주기 표시 및 조정 시점 출력
- 로그 저장 (csv형태)



03 프로젝트 요구사항

기능 요구사항

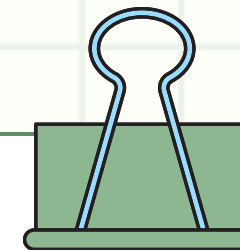
1. 차량 객체는 YOLO 기반 탐지기로 자동 인식되어야 하며, 방향(동-서/남-북)에 따라 분류되어야 한다
2. 집계된 차량 수 데이터를 바탕으로 신호 주기를 동적으로 조정해야 한다
3. 신호 조정 알고리즘은 최소 및 최대 시간 범위 내에서 동작해야 한다
4. PyQt 기반 UI에서 현재 신호 상태(녹색/적색/남은 시간 등)를 애니메이션 형태로 표시해야 한다
5. 대시보드에는 방향별 차량 수, 현재 신호 시간, 전체 사이클 로그가 표시되어야 한다
6. 모든 신호 사이클은 CSV 파일로 로그가 저장되어야 한다.
7. 사용자는 프로그램 실행 후 영상 파일 또는 실시간 스트림을 선택하여 테스트할 수 있어야 한다
8. 전체 시스템은 Presenter 구조를 통해 Model과 View를 독립적으로 관리해야 한다



03 프로젝트 요구사항

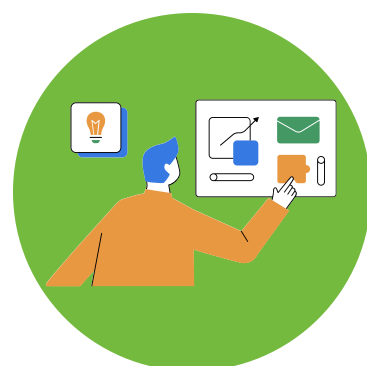
비기능 요구사항

1. 프로그램은 PyQt5 기반으로 구성되며 외부 서버에 의존하지 않아야 한다 (완전한 로컬 실행 가능)
2. 시스템은 예외 상황(영상 없음, 모델 로드 실패 등)에 대한 처리 로직을 포함해야 한다
3. 코드 스타일은 PEP8을 따르며, 각 클래스 및 함수에는 docstring이 포함되어야 한다
4. 최종 보고서 및 발표 자료는 문서 템플릿에 맞춰 제출되어야 한다



04 진행 계획

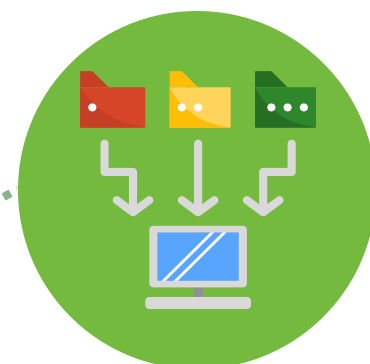
주차별 계획



Week 1

요구사항 분석 및 설계

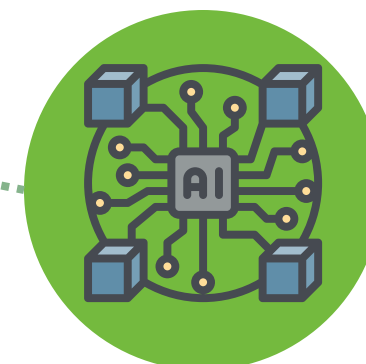
- Task 1 - 전체 요구사항 정리
- Task 2 - 기술 스택 확정
- Task 3 - 세부 역할 분담
- Task 4 - 진행 일정 세분화
- Task 4 - 시스템 아키텍처 작성



Week 2

CCTV 수집 환경 구축 AI 프로토타입 설계

- Task 1 - CCTV 스트림 수집 환경 구축
- Task 2 - 차량 검출 모델 조사
- Task 3 - 검출 모델 파인튜닝
- Task 4 - 검출 모델 평가 리포트

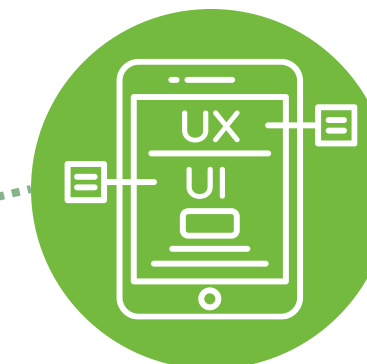


Week 3

교통량 집계

신호제어 알고리즘 구현

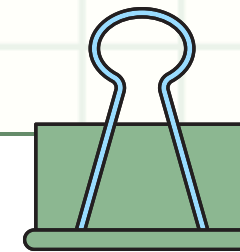
- Task 1 - 교통량 통계 로직 개발
- Task 2 - 신호주기 조정 알고리즘 설계
- Task 3 - 신호주기 조정 알고리즘 개발
- Task 4 - 알고리즘 실용성 테스트
- Task 5 - 기존 방식 vs 알고리즘 비교 분석



Week 4

대시보드 UI·통합 테스트

- Task 1 - 실시간 교통량 현황 시각화
- Task 2 - 실시간 신호주기 시각화
- Task 3 - 테스트 케이스 작성·수행
- Task 4 - 버그 리포트 및 수정

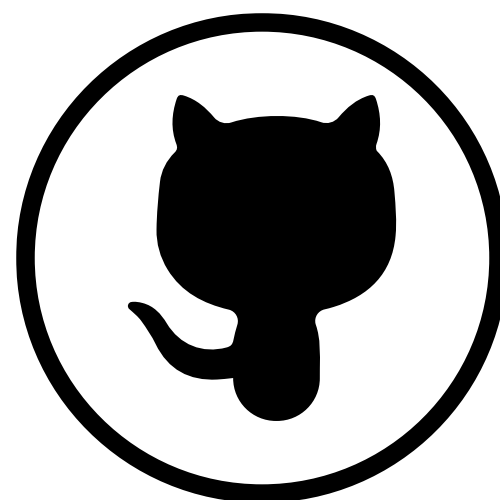


04 진행 계획

팀원간의 회의 일정 /소통 방법



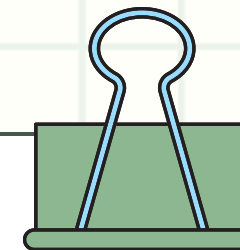
Slack을 통해
팀원 간 의사소통
및 회의 일정 조정



각 task가 끝난 후
해당 내용을 githubPR로 공유



비대면 회의 시
Discord로 화면공유/회의 진행



05 팀원 소개

팀장



황문규

객체 탐지 모델 결정

CCTV 스트림 처리

모델 연결



권재영

GUI 시뮬레이터 개발

UI 테스트

도로 내 시설 시각화



송채현

모델-View 연결

상태 흐름 관리

통합 컨트롤러



신정현

신호 조정 로직 설계

신호 조정 로직 개발

테스트 케이스 작성

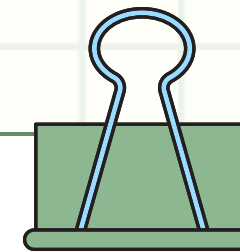


조은일

대시보드 구현

실시간 데이터 표시

로그 기능 구현



06 기대효과

교통 흐름 개선

평균 통행 시간 감소 기대
불필요한 대기 최소화
교차리 처리 효율 향상



사고 예방

급정거·충돌 위험 감소
보행자 보호 강화
꼬리물기 현상 방지



체감 비용 절감

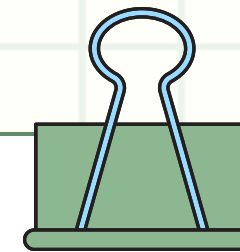
출퇴근 시간 단축
연료비 절감
차량 비용 · 물류비 · 인건비 절감



탄소 감축 및 에너지 절약

신호 대기 불필요한 가속 · 감속 감소
연료 소비 절감
온실가스 및 배기가스 감소





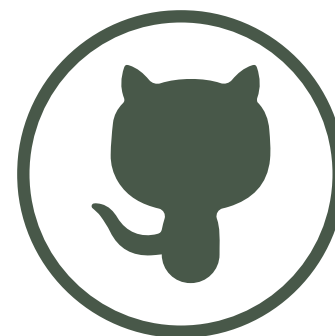
질의 응답 및 연락처

자유롭게 질문해주세요.



이메일

moongye2202@knu.ac.kr



프로젝트 GitHub

<https://github.com/moon9H/AdaptiveTraffic>