

승강기 대기시간 단축 프로젝트

홍찬호

- [프로젝트 기본 정보]
- 프로젝트명: 승강기 대기시간 단축 프로젝트
 - 개발 기간: 2025.08.13 ~ 2025.08.21
 - GitHub 링크: https://github.com/JoshuaHong0215/miniproject/tree/main/01_project/src/final

- [문제 정의 및 목표]
- 문제: 회사, 아파트 등 이용빈도수가 많은 건물의 승강기의 대기시간으로 인하여 많은 시간이 지체됨
 - 목표: 승강기는 4초동안 문을 열고 7초 동안 대기하고 4초동안 문을 닫는 로직을 수행하고 이패턴은 건물의 층수와, 규모, 용도에 따라 달라지나 보편적인 패턴인 4/7/4로직에서 다음의 조건을 추가하여 어느정도의 효율이 있는지를 본다
 - 조건: 내부에서 외부로 이동한 객체를 기준으로 3초동안 객체가 감지가 되지 않고, 다른 층에 호출이 되어있는 상태라면 7초의 대기시간을 3초로 단축시킴
 - 이유: 급한 불일이나 업무가 생길때 많은 이들이 공감 할 만한 일상속의 소소하지만 확실한 문제라고 인식하였음

- [기술 스택]
- 언어: Python
 - 라이브러리: OpenCV, Tkinter, YOLO
 - 개발 도구: VS Code
 - AI: ChatGPT (GPT-5)

[성과 측정결과]

실험시간 : 09:00~19:00

총 정차 횟수: 2640회

측정 항목	개선 전	개선 후	결과
시간단축	100%	92.52%	7.48% 개선율
소요시간	1291.1분	1194.6분	96.5분 단축

- [핵심 기능]
1. Tkinter 카메라 영상 출력
 - OpenCV프레임을 PIL.Image → ImageTK.PhotoImage로 변환해서 canvas.creat_image()로 띄움
 2. 객체 감지 이벤트
 - ROI를 사용하여 객체를 감지하고 가운데 선을 중심으로IN, OUT으로 공간을 분리하여 IN → OUT순으로 이동한 객체가 기준이 되어 더 이상 OUT공간에 객체가 인식되지 않고 다른층에서 호출이 되어있으면 이벤트 발생
 3. 엘리베이터 호출 및 동작 시뮬레이션
 - Tkinter버튼으로 층을 호출하면 시각적으로 이동 경로 표시
 - Tkinter Canvas로 배치하고 after메소드와 move()메소드를 통해 애니메이션 구현

[문제 해결 사례]

문제1. 카메라 영상과 Tkinter GUI 동시 실행 문제 → 화면이 멈춤

- cv2.imshow와 Tkinter의 이벤트 루프가 충돌

해결방법 : Tkinter Canvas에 OpenCV프레임을 직접 띄우는 방식 사용

습득 : GUI와 영상처리 루프의 동기화 과정을 습득

문제2. 내려갈때 중간 층 패스

- 호출 큐 정렬 로직이 높은 층 우선 → 내림차순으로만 작동하여 이동 중간에 만나는 층을 별도로 체크하지않음

해결방법: Tkinter의 일정시간이 지난뒤 특정 함수를 실행하도록 하는 after() 반복실행과 호출 큐 검사를 사용해서 한 층씩 이동할 때마다 호출된 층이 있으면 즉시 멈추도록 바꿨다

습득 : after()메소드의 존재를 알게 되어 추후 다른 프로젝트에도 유용되어질거같다

- [학습 성과]
- Tkinter Canvas 영상 출력 (FPS 개선)
 - ROI 객체 감지 (좌표 디버깅)
 - 타이머 조건 단축 (if 충돌 해결)

[코드 품질 및 협업]

- 총 56회 커밋 (평균 5.8회/일)