K-Digital Training

01 프로젝트 개요

1 프로젝트 주제 및 선정 배경, 기획의도

주차장에 존재하는 주차공간의 수를 파악하기 위해 프로젝트 제작

필요성

1. **주차효율성 향상**
2. **사용자 편의성**
3. **경제적효과 상향(AI활용)**

2 프로젝트 내용

주차 공간의 이용 상황을 한눈에 파악할 수 있도록 AI가 차량을 자동으로 인식해 **슬롯별 점유 상태를 계산**하고, 그 결과를 **전체 주차장 현황**과 함께 색상 박스로 직관적으로 표현하는 **실시간 모니터링 시스템**을 구현

3 활용 장비 및 재료

VSCode
Python
miniconda
Google Colab
roboflow
React
Yolo 8
FastAPI

4 프로젝트 구조

1. React(프론트)
사용자 영상 업로드
2. FastAPI(백엔드)
-업로드된영상 current.mp4로 저장 (업로드될때마다 덮어쓰기)
-yolo로 차량 감지
-슬롯 플리곤과 차량 중심점 비교
-> 점유상태 판단
3. 스트리밍 및 처리 최적화
/stream에서 속도 조절
(프레임 마다 처리 x 프레임 5번 당 무거운 yolo 작업을 1번 하는 식)
4.분석결과관리
최신 차량/주차 슬롯 상태를 전역 변수에 지속 업데이트
5 .React(프론트)
최신 분석결과 받아 실시간 시각화

5 활용방안 및 기대 효과

사용자에게 현재 남은 주차 대수와 혼잡도 등을 실시간으로 안내한다.

실시간 정보 제공으로 빈자리 탐색 시간을 줄여, 이용 편의성과 서비스 만족도를 높일 수 있다.

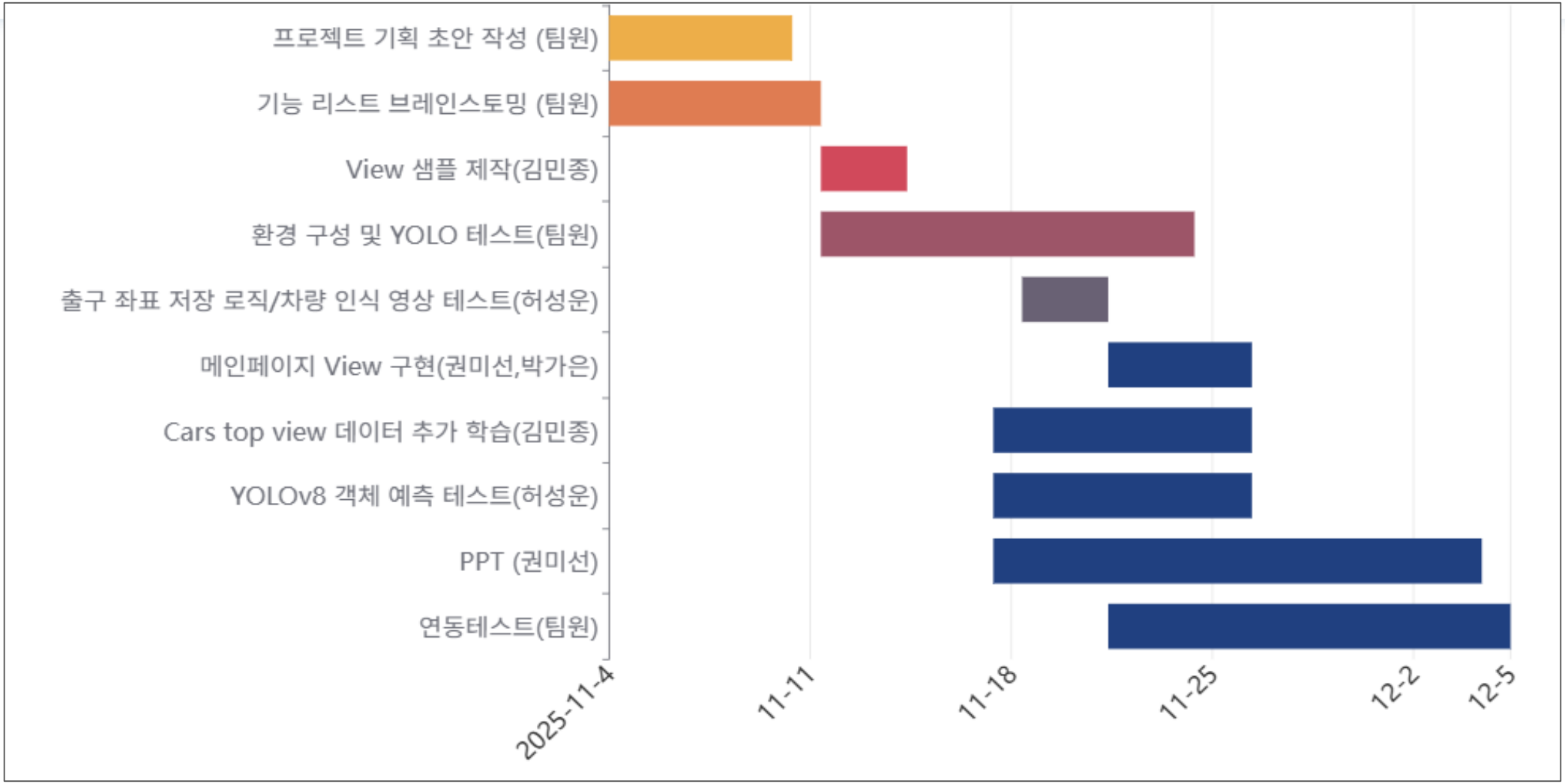
주차 공간 활용도를 극대화하고 관리 효율 및 운영의 경제성을 기대할 수 있다.

03 프로젝트 수행 절차 및 방법

구분	기간	활동	비고
사전 기획	11/4(화) ~ 11/11(화)	프로젝트 기획 및 주제 선정 기획안 작성	아이디어 선정
요구사항 분석	11/10(월) ~ 11/11(화)	필요 데이터 및 수집 절차 정의 플로우 차트 작성	벤치마킹 및 흐름도 정의
기능 구현	11/11(화) ~ 11/24(월)	코드 구현	기능별 구현 및 주간 회의
연동 테스트	11/24(월) ~ 12/4(금)	코드 테스트	최적화, 오류 수정
총 개발기간	11/4(월) ~ 12/5(금)(총 4주)		

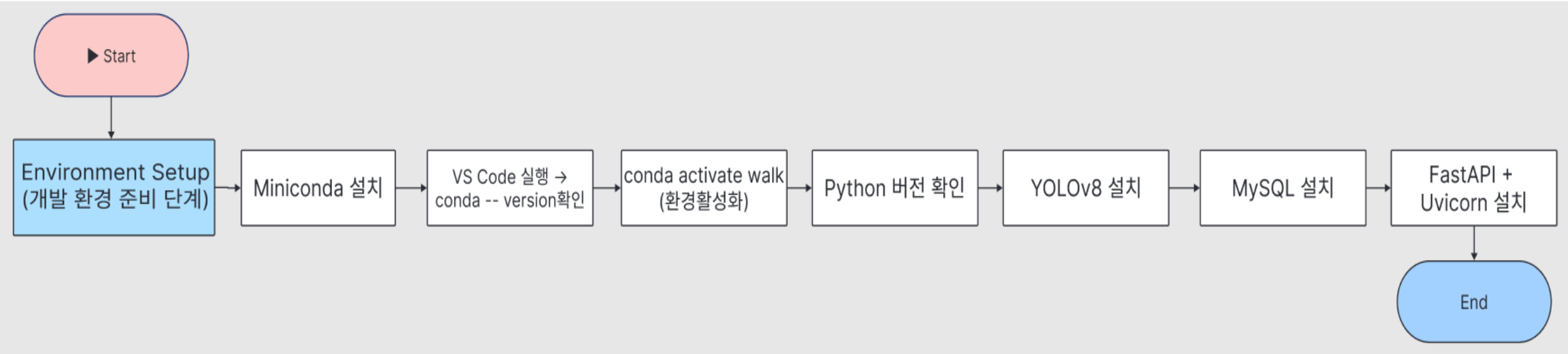
03 프로젝트 수행 절차 및 방법

개발 로드맵 (간트차트)



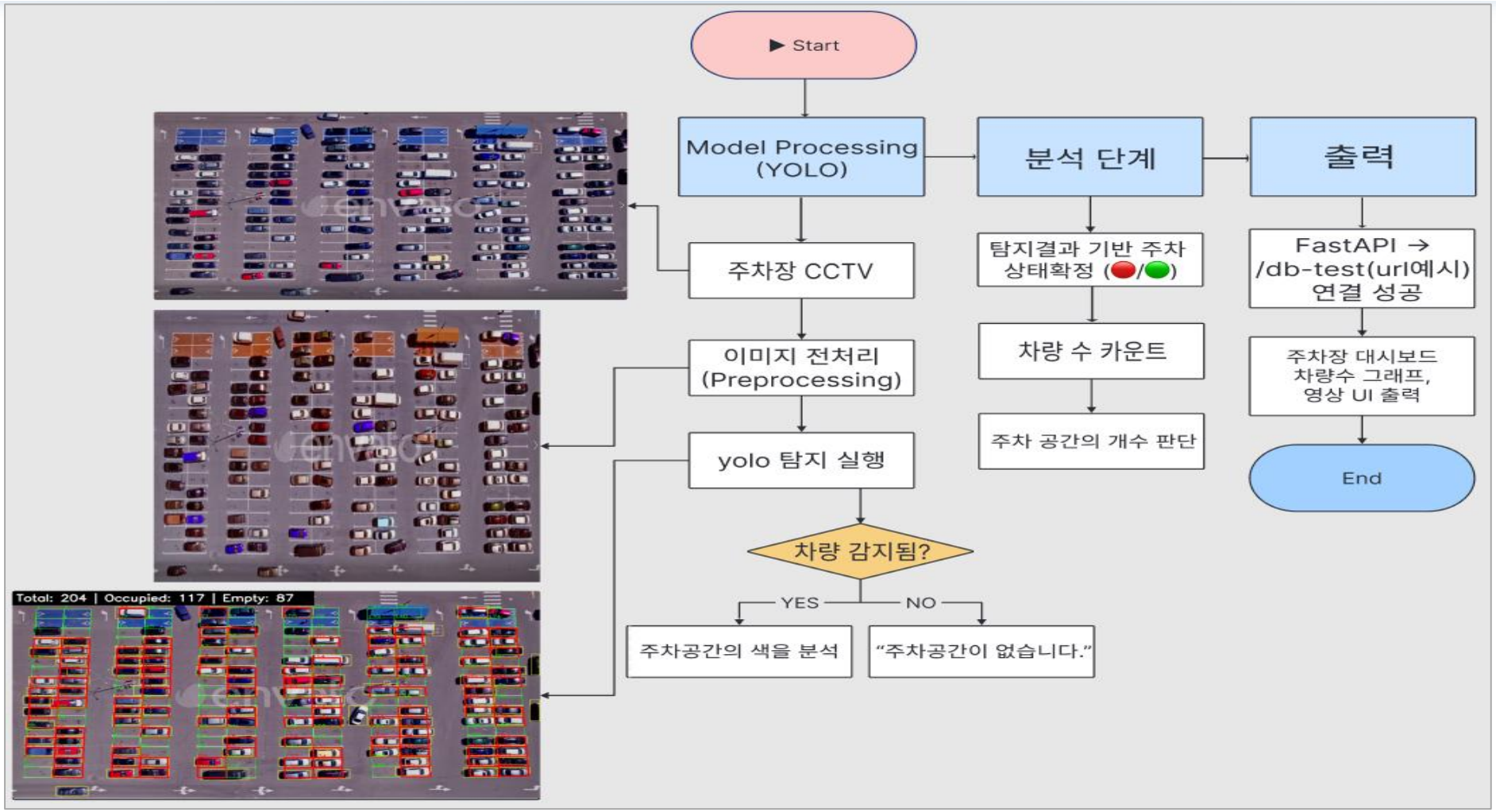
04 프로젝트 수행 경과

개발환경 플로우차트(Flowchart)



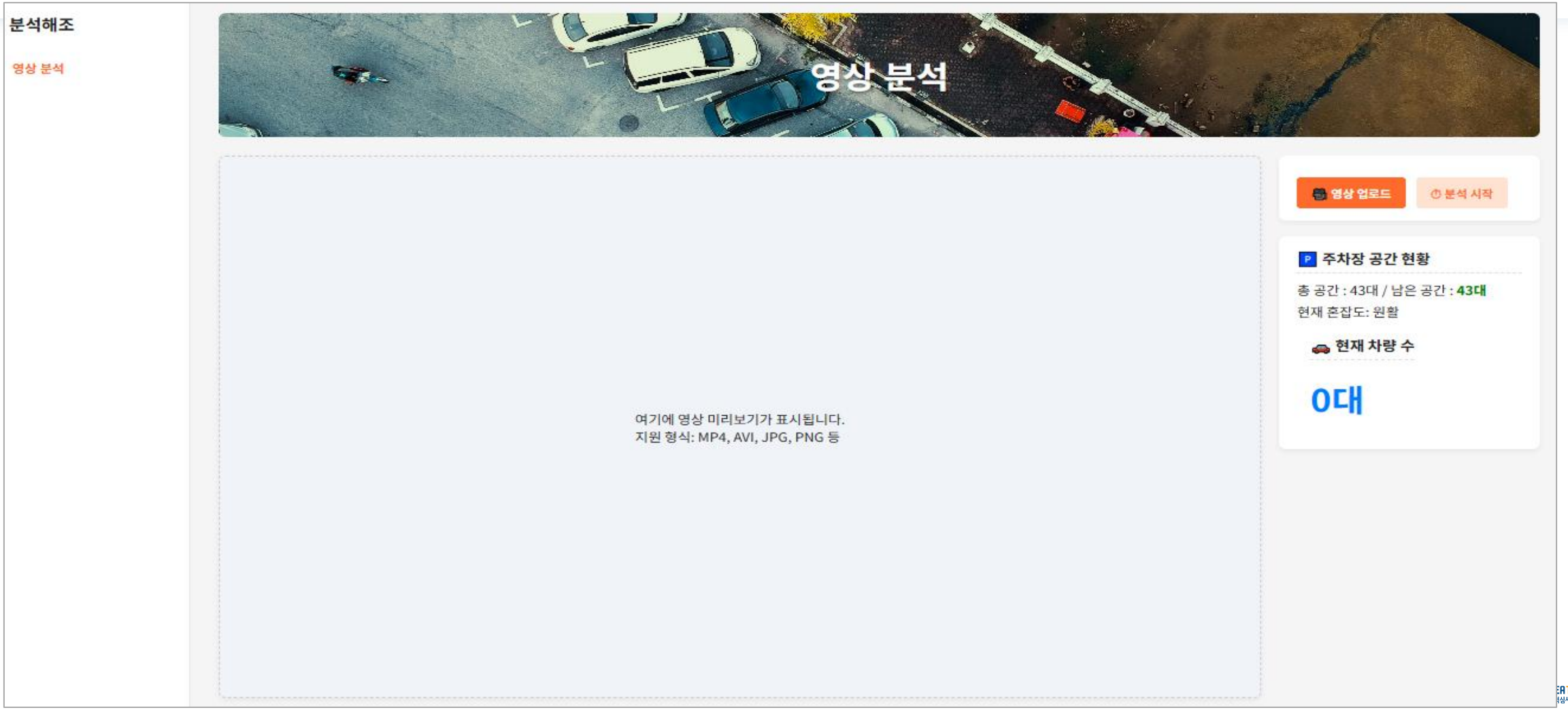
프로젝트 수행 경과

시스템 플로우차트(Flowchart)



프로젝트 수행 경과

주차장 분석 서비스 메인 화면



04 K-Digital Training 프로젝트 수행 경과

- **실시간 주차장 영상 분석 (** 영상 업로드 → 차량 감지 → 슬롯 상세 현황 / **점유** (빨간색) 및 **빈자리표시**(초록색) **)**

04 프로젝트 수행 경과

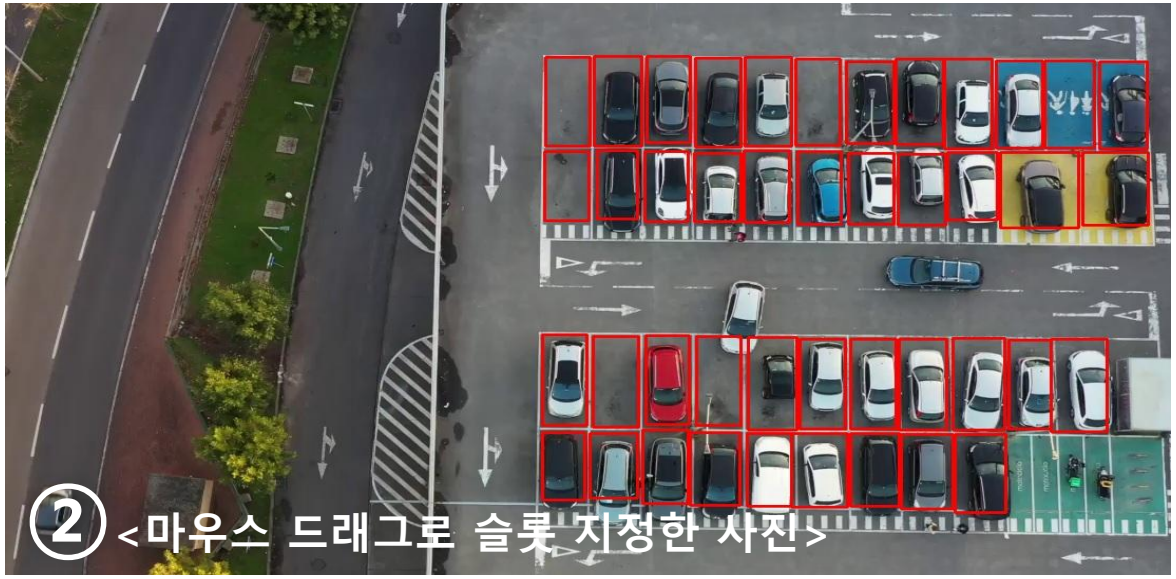
① <차량 인식 사진>

1.슬롯 좌표 파일 생성
주차장 각 자리의 4개 모서리 좌표를
마우스로 드래그하여 수작업으로 지정(slots.csv 파일로 저장)

2. 슬롯 좌표는 고정
주차장 자리 구조는 변하지 않으므로, 좌표는 고정된 값으로 사용

3. 슬롯 정보와 차량 분석 영상 결합(①,②)
CSV 파일에서 슬롯 좌표를 읽어와 차량 감지 코드와 결합
차량 중심점이 슬롯 폴리곤 안에 있는지 확인

4. 점유 판단
차량 중심점이 슬롯 폴리곤 내부에 있으면 점유
점유 → 빨간색, 빈자리→ 초록색

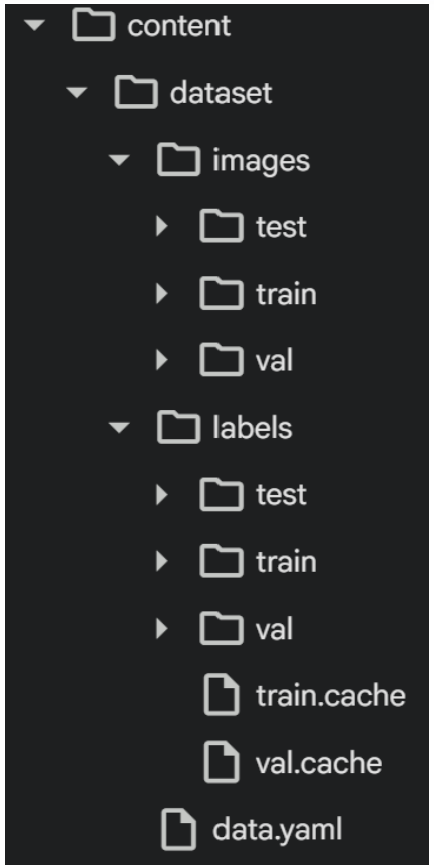


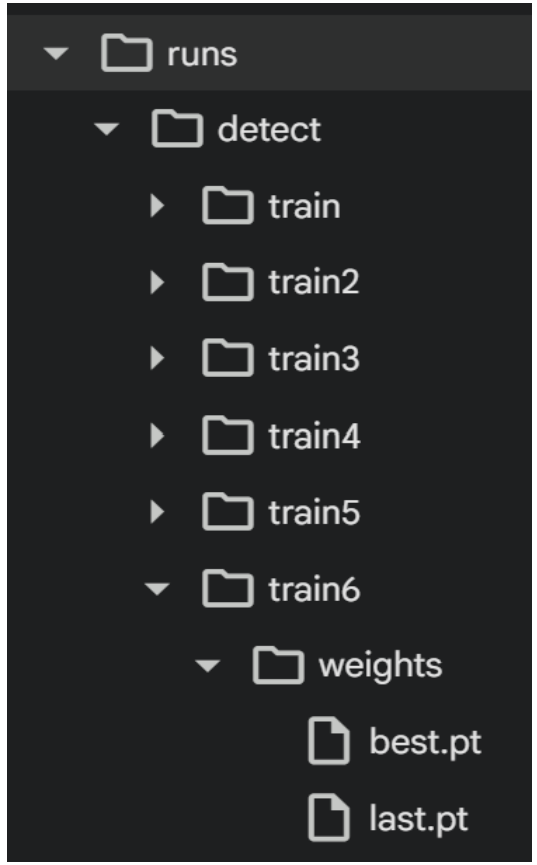
② <마우스 드래그로 슬롯 지정한 사진>

③

프로젝트 수행 경과

YOLOv8 데이터셋 구성 및 학습 과정 결과





<주차장 영상 데이터셋 구성> < YOLOv8 새 데이터 학습(Google Colab) > < 학습 결과(best.pt 파일 생성) >

프로젝트 수행 경과

Roboflow를 통한 라벨링 학습 화면

지나가거나 주차한 각 차량마다 보라색 바운딩 박스 씌워 객체 위치와 크기 지정

학습한 클래스(차량) 표시

04 프로젝트 수행 경과

YOLOv8 데이터셋 구성 및 학습 과정 결과

1. 첫 학습에 쓰인 설정값 (총 120장 사용)

```
model = YOLO("yolov8s.pt")

results = model.train(
    data="/content/dataset/data.yaml",
    imgsz=640,
    epochs=150,
    batch=16,           # GPU VRAM 보고 자동 조절
    patience=25,        # Early Stop 여유롭게
    lr0=0.001,          # 초기 학습률
    optimizer="AdamW", # 탑뷰 데이터에 성능 안정적
    mosaic=1.0,         # 데이터 증강 강화 (탑뷰에 유리)
    augment=True
)
```

2. 추가학습에 쓰인 설정값

```
from ultralytics import YOLO

model = YOLO("/content/drive/MyDrive/Yolo_train/best.pt")   # 기존 학습 모델 경로


model.train(
    data="/content/drive/MyDrive/Yolo_train/data.yaml",     # 새 + 기존 dataset 경로
    epochs=50,                                  # 필요에 따라 변경 가능
    imgsz=640,
    batch=4,                                   # GPU 여유 있으면 32까지 가능
    lr0=0.0005,                                 # 일반 학습보다 낮게 (Fine-Tuning 기본 전략)
    optimizer="AdamW",
    patience=15,
    amp=True,
    device=0,
    cache=False,
    pretrained=True                             # 기존 weight 사용
)
```

샘플 데이터 추가해서 총 400개 사용

04 프로젝트 수행 경과

- **YOLOv8 학습 기반 예측작업(** VSCode **)**

```python
from ultralytics import YOLO
import cv2
import os
from tkinter import Tk
from tkinter.filedialog import askopenfilename
```

```python
# 1 영상 선택
Tk().withdraw()
video_path = askopenfilename(
    title="영상 파일 선택",
    filetypes=[("Video files", "*.mp4;*.avi;*.mov;*.mkv"), ("All files", "*.*")]
)
if not video_path:
    print("❌ 파일을 선택해야 합니다.")
    exit()

print(f"🎞️ 선택된 파일: {video_path}")
```

```python
# 2 YOLO 모델 로드
model_path = "model/best.pt"
if not os.path.exists(model_path):
    raise FileNotFoundError(f"🚨 모델 파일이 없습니다: {model_path}")

print(f"📦 모델 로드 중: {model_path}")
model = YOLO(model_path)

tracker_path = r"C:\Users\ggp03\miniconda3\envs\walk\Lib\site-packages\ultralytics\cfg\trackers\bytetrack.yaml"
```

```python
# 3 Tracking 모드 적용 (깜빡임 해결)
results = model.track(
    source=video_path,
    conf=0.15,       # 신뢰도 기준 설정
    iou=0.50,        # 박스 겹침 기준
    imgsz=1280,      # 고해상도 분석
    project="runs/detect",
    name="predict",
    exist_ok=True,
    stream=True,
    tracker=tracker_path,  # 안정적 추적
    persist=True                # ID 유지-> 깜빡인 최소화
```

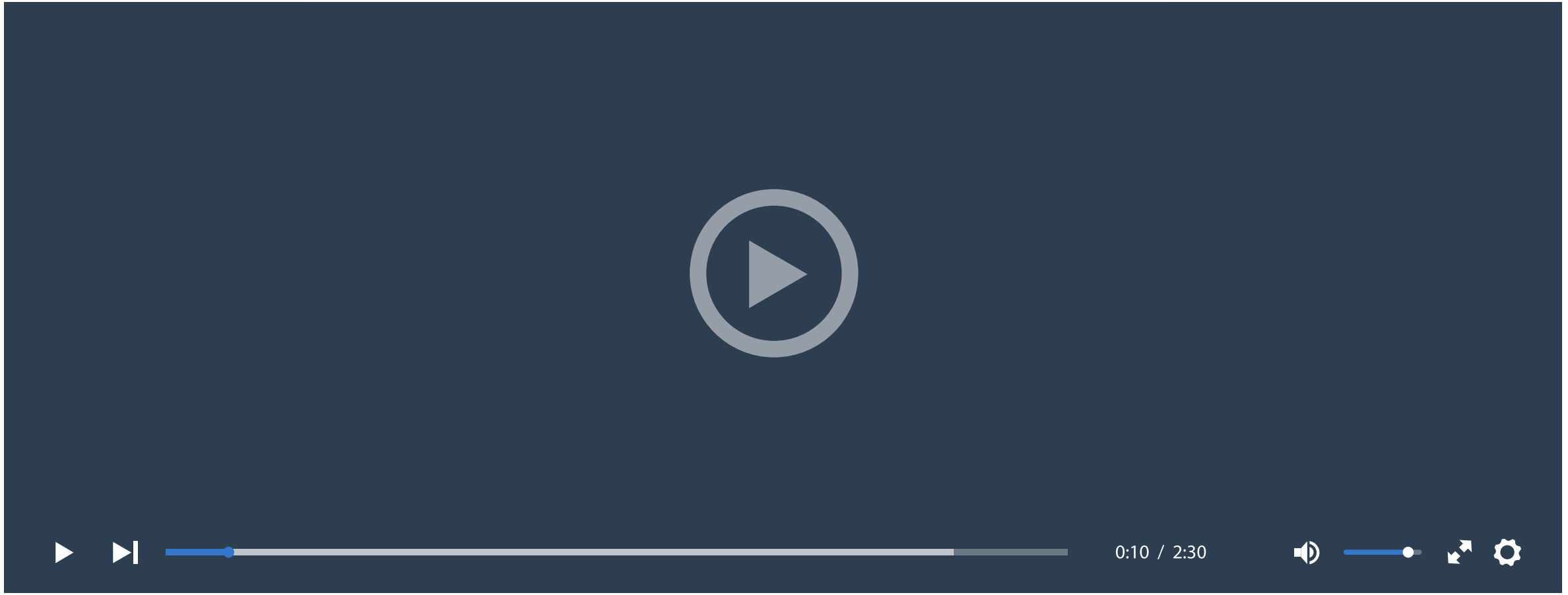
```python
# 4 후처리: 작은 박스 제거 + ID 라벨 표시
output_video_path = f"videos/optimized_{os.path.basename(video_path)}"

cap = cv2.VideoCapture(video_path)
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = cap.get(cv2.CAP_PROP_FPS)
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(output_video_path, fourcc, fps, (width, height))

# 작은 박스 기준 완화 (영상 크기 0.66% 이상)
min_size = max(width, height) // 200
```

```python
# 5 Windows 자동 실행
if os.name == "nt":
    os.startfile(output_video_path)
```

04 프로젝트 수행 경과



* 용량제한: 팀별 5-10분 내(100MB 이하), 기능별 소개 음성 포함, 별도 파일형태로 제출 가능

자체 평가 의견

평가 의견과 느낀 점

10점 만점에 7점

실시간 분석, 웹 출력, 추가 학습 등 핵심 기능은 잘 구현해 기본 완성도는 높았습니다. 다만, 인식 안정성, 데이터 부족, DB 미활용 등 보완할 점이 남아 있어 7점을 부여했습니다.

- Redis로 실시간 데이터를 저장 조회하는 구조를 구축해야 한다.
- 긴 영상 기반으로 슬롯 상태를 지속적으로 기록할 기능이 필요하다.
- 다양한 환경 데이터를 추가 수집해 모델을 더 잘 학습시켜야 한다.
- 차량 외 다른 객체도 인식할 수 있도록 모델을 확장해야 한다.

잘한 부분

- AI로 주차 가능 여부를 즉시 확인할 수 있는 시스템을 만들었다.
- 실제 주차장 데이터를 추가 학습에 활용해 모델 정확도를 높였다.
- 분석 영상을 웹에 실시간으로 표시하는 기능을 구현했다.

아쉬운 점

- 환경 변화에 따라 인식이 불안정했다.
- 영상 데이터가 부족해 학습 다양성이 낮았다.
- DB를 활용한 저장•분석 기능을 구현하지 못했다.

05 자체 평가 의견

평가 의견과 느낀 점

백엔드 : 모델 테스트를 위해
터미널에서 가상환경을 켜고 패키지를 설치하고
영상을 실행시키는 과정자체가 새로운 경험이였습니다
프론트 : 기존의 HTML과는 다르게 react는 코드를 수정하면
화면이 즉시 반영되는 점이 가장 인상적이였고
하나의 컴포넌트안에서 함께 관리할수있어 개발흐름이 훨씬 빠르고 직관적이라 개발이 훨씬 효율적이라는 것을 느꼈습니다

실시간으로 영상 분석하기 위해
리엑트를 통해 view 생성하여 차량을 감지하면 주차공간에 몇 개 남았는지 몇 개 들어가는지에 대해 한눈에 볼 수 있어서 좋았습니다 평소에 써보지 못한 다양한 환경 정비 재료로 주차공간의 수를 색상 박스로 구분할 수 있다는 점에서 흥미로웠고 신기했습니다

YOLOv8 기반 주차장 AI 영상 분석에서 자동차 탑뷰 인식률 향상을 위해 직접 데이터셋을 구축하고 Fine-Tuning을 진행했습니다. 공개 데이터만으로는 실제 환경 요소로 인해 인식률이 낮아, 주차장 영상 프레임을 수집·라벨링하여 맞춤형 데이터셋을 제작했습니다. 이후 Colab GPU 환경에서 하이퍼파라미터를 조정하며 재학습한 결과 정확도가 크게 향상됐고, 이를 통해 실제 환경에 맞는 데이터 구축의 중요성을 깨달았습니다.

슬롯 좌표를 지정할 때, 영상 원본의 파일을 이용하여
좌표를 지정하지 않으면 안된다는 점과 좌표를 지정하는데
다양한 방법이 존재하는 점, 점유율을 계산할 때에도
최적의 방법에 대해 고민을 가지고 적용해보는 시간이 되었습니다.
또 주차 슬롯 점유는 프레임마다
전부 무거운 yolo 분석을 할 필요가 없듯이(속도가 느려짐)
제작하려는 프로그램의 용도, 의도를 파악하고
분석 코드를 기획하는 것이 중요한 것 같습니다.

감사합니다

발표와 관련해 궁금한 사항이 있다면 질문해주세요.