



A.I. PADS

군중 밀집에 따른 사고를 예방하는 A.I. 서비스
A.I. Prevents Accidents Due to crowding Service

지하철역 혼잡도 분석 시스템
Subway Station Congestion Analysis System

미래산업과학고등학교 업서요 팀



목차



1. 팀 소개



2. 프로젝트 소개



3. 프로젝트 제작 과정

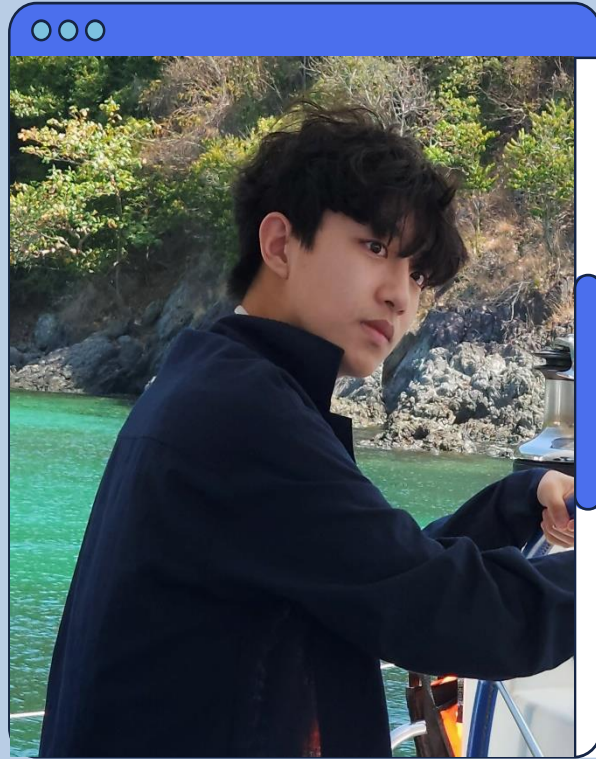


4. 프로젝트 결과 및 해석

팀원 소개



박진우 (팀장)



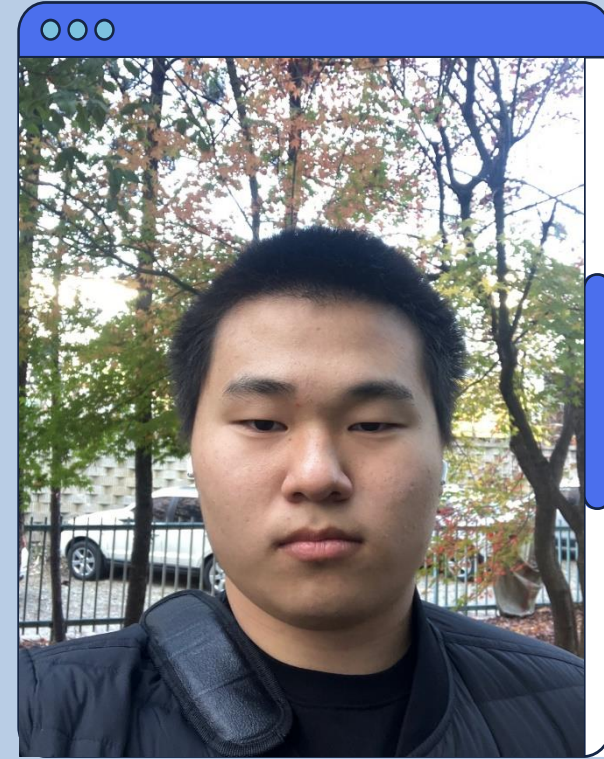
조운혁 (HW / SW)



김강현 (AI / SW)



이승우 (SW)



이동현 (SW)



해당 사진들은 “미래산업과학고등학교” 및 “메이커창작과” 와 **무관함**을 알려드립니다.



LG CNS AI 지니어스 아카데미 참여동기

AI 미션 클리어 활동을 진행하면서
작년 “이태원 참사”와 같은 압박 사고를
인공지능을 통해 예방하고자 참여하게 되었습니다.

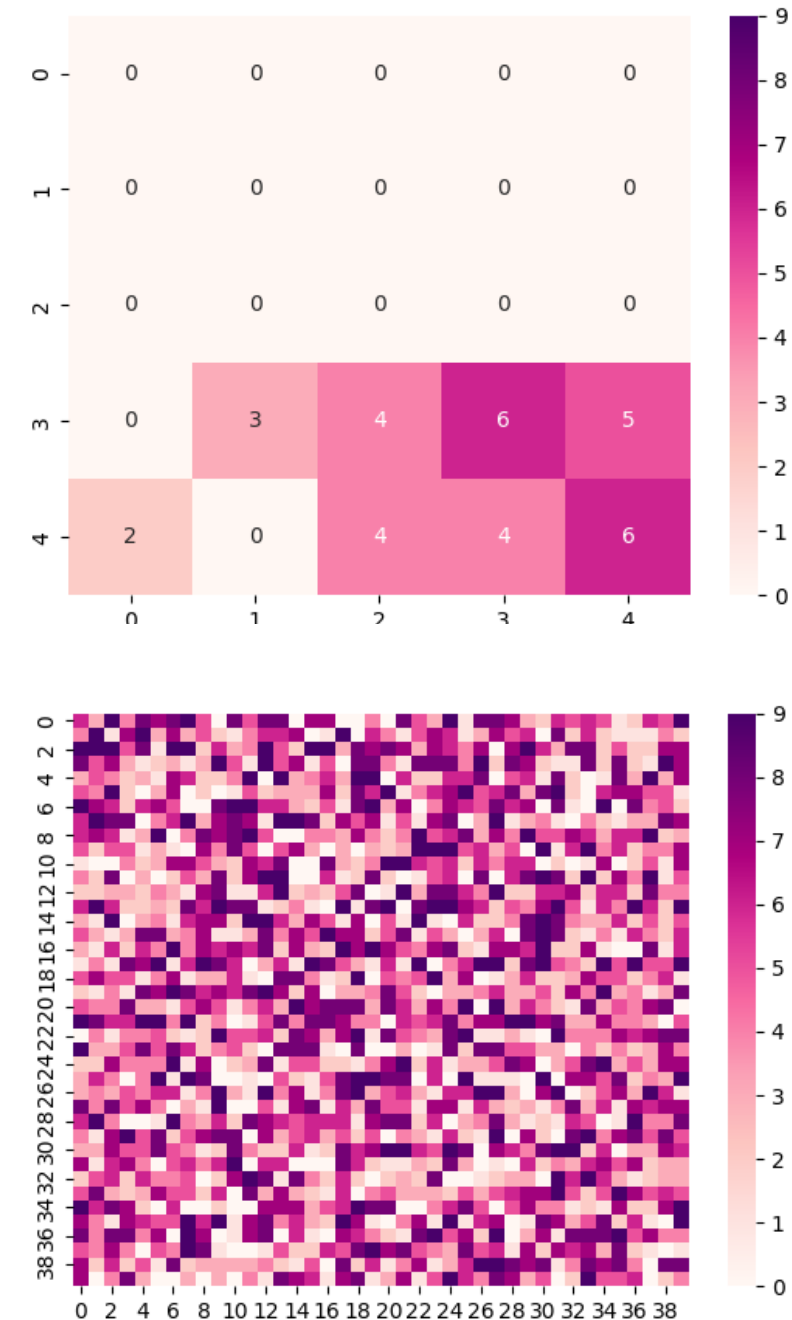
1. 프로젝트 소개



A.I. PADS

군중 밀집에 따른 사고를 예방하는 A.I. 서비스

A.I. Prevents Accidents Due to crowding Service



WHY ? 혼잡도?



김포 골드라인 실신 사고



2022 이태원 참사



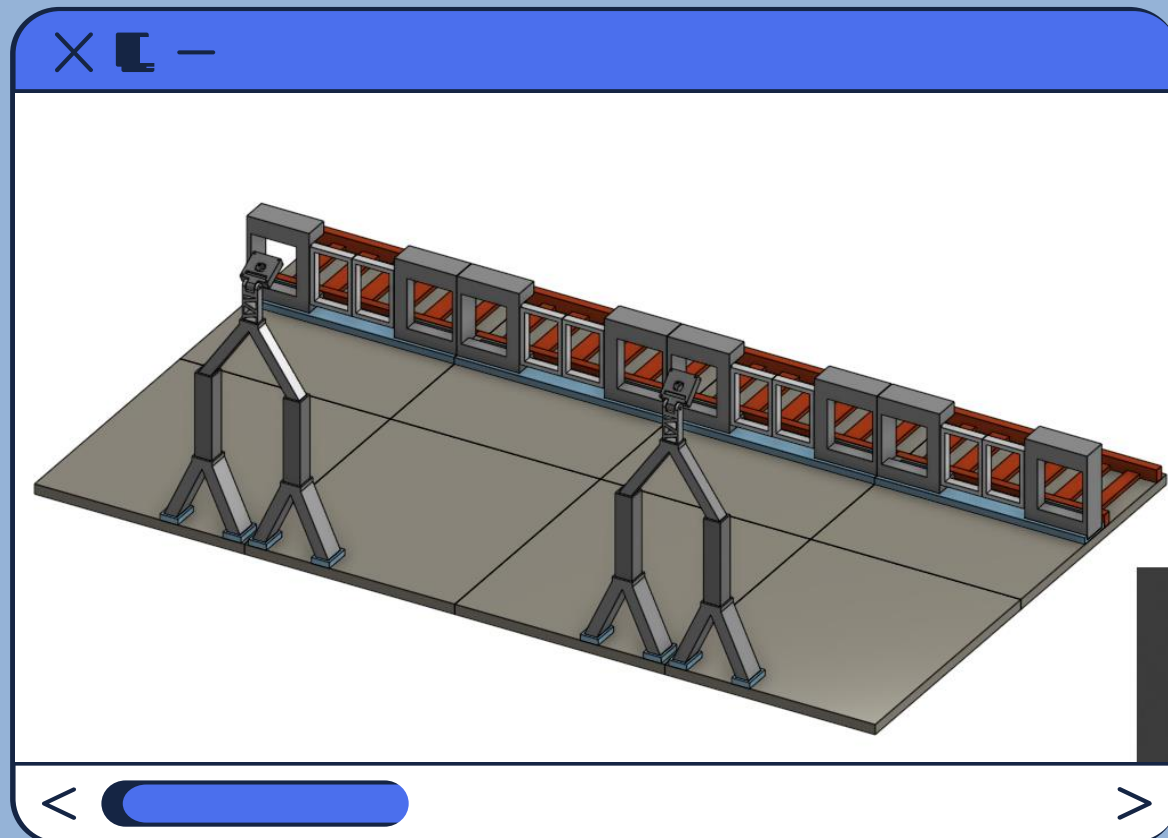
해외 축구장 압사 사고



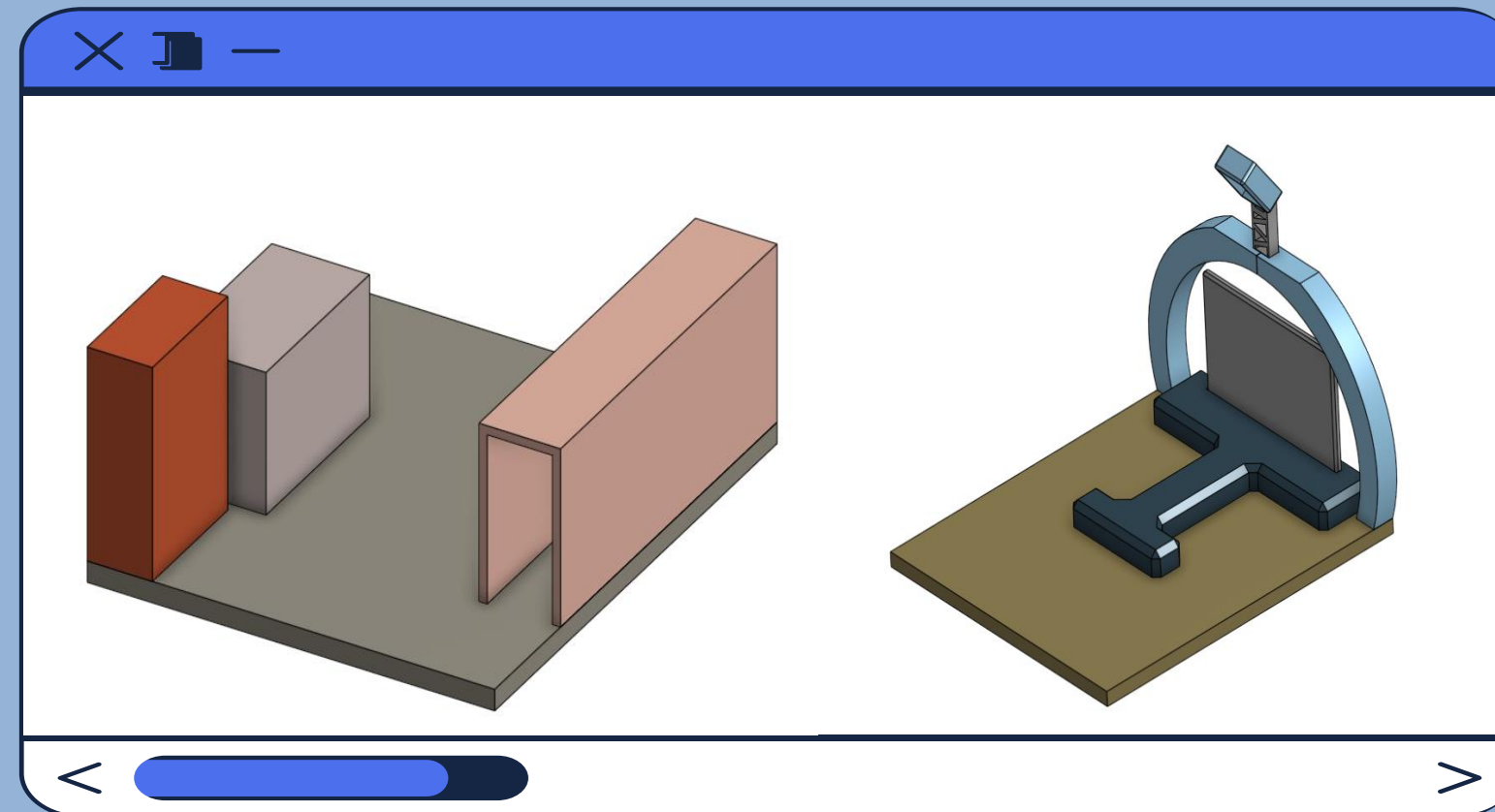
HW 설계 및 제작



3D프린팅 및 제작



지하철역 및 기타 장소
(콘서트 장 , 골목 등) 모델링



SW / AI 개발 계획

1

사람 인식 AI 모델 제작
RoboFlow에서 라벨링 및 데이터 학습

2

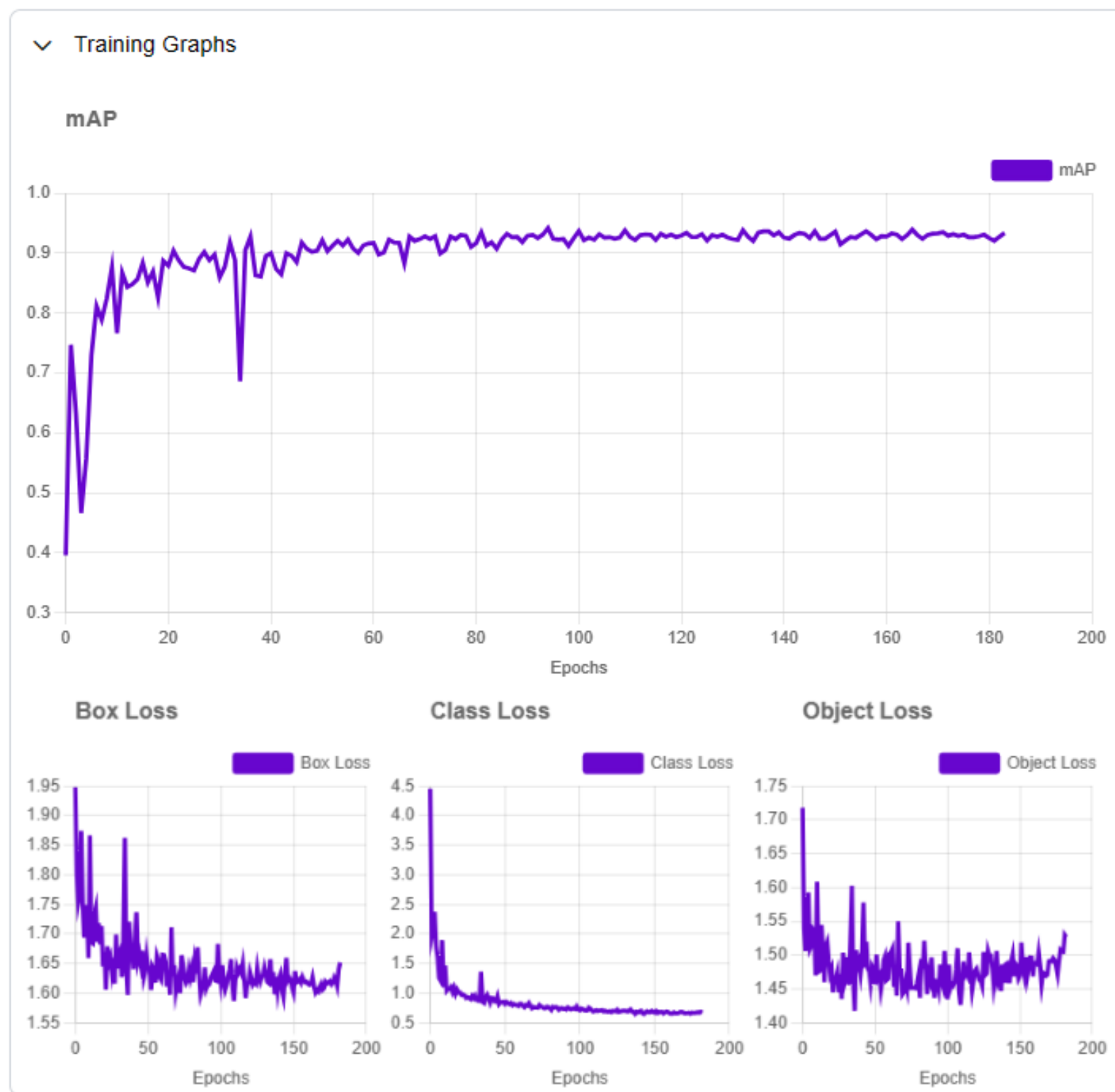
혼잡도 수치화
실시간 이미지를 통한 혼잡도 수치화(백분율)

3

혼잡도 수치를 이용한 정보 시각화
HeatMap을 통하여 혼잡도를 시각화

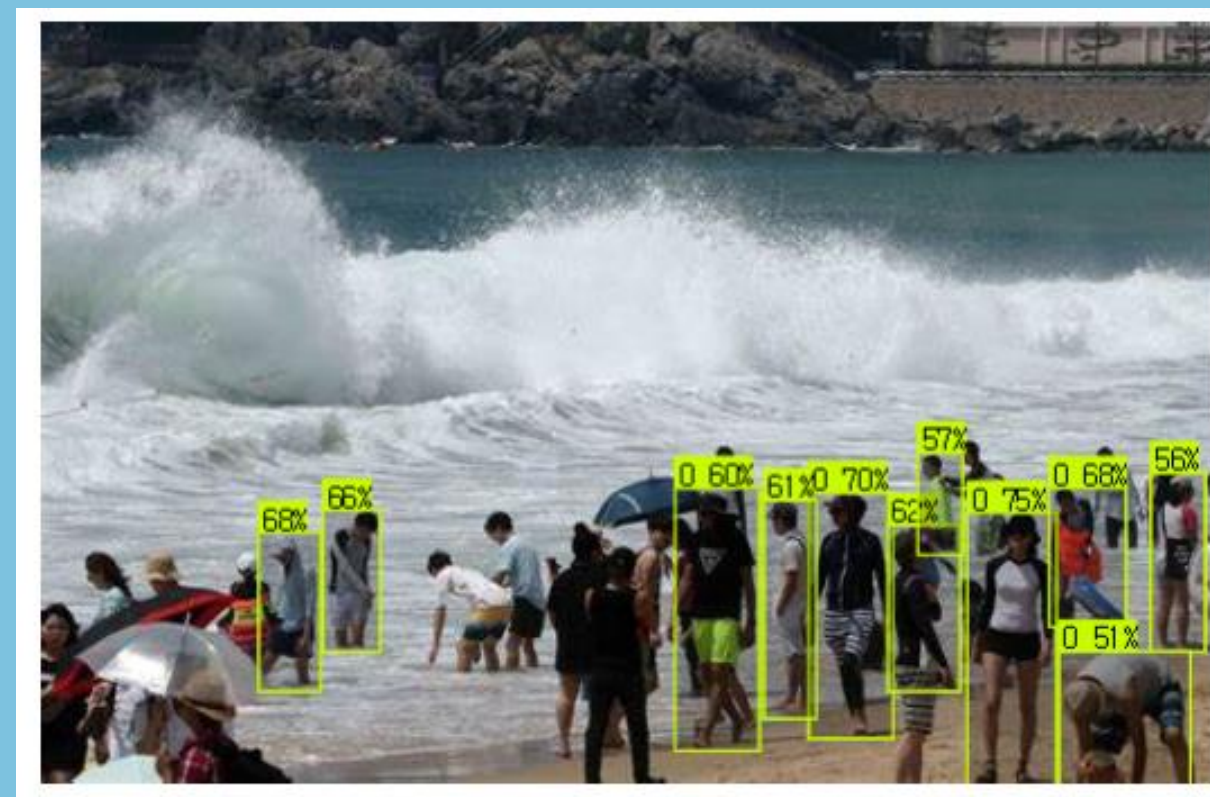
4

실시간으로 정보를 보여주는 웹사이트
Flask 를 사용한 사용한 웹서버 및 웹 구축



사람 인식 AI 모델 제작

RoboFlow를 사용하여 Image Classification model 제작



약 2000장 정도의 데이터 셋 제작 및 학습

혼잡도 수치화



혼잡도 측정 방법:

이미지 전체의 픽셀 수 대비 사람이 차지하는 픽셀 수의 비율을 계산하여 혼잡도를 구함

픽셀을 대신한 좌표 값을 이용한 계산:

사진의 각 좌표 값마다 사람의 영역인지 빈 공간인지 구분하여 사람의 영역을 계산하였다.

$$\text{혼잡도}(\%) = (\text{감지된 사람 수} / \text{전체 픽셀 수}) * 100$$

```
data.decode()=="recog":
    print('>> Received from ' + addr[0], ':', addr[1], data.decode())
    current_time = datetime.now()
    rf = Roboflow(api_key="TiNMvvhxIoHsoajs9cDg4")
    project = rf.workspace().project("lg-cns")
    model = project.version(2).model
    predict_json=model.predict("static/assets/img/saved_picture.jpg", confidence=40,
overlap=30).json()
    print(predict_json)

    with open("prediction.json", "w") as json_file:
        json.dump(predict_json, json_file)
    print("Prediction saved to prediction.json")
    try:
        with open(file_path, 'r') as json_file:
            data = json.load(json_file)
            print('JSON 데이터:', data)
    except FileNotFoundError:
        print(f'파일을 찾을 수 없습니다: {file_path}')
    except json.JSONDecodeError as e:
        print(f'JSON 파일 파싱 오류: {e}')
    except Exception as e:
        print(f'오류 발생: {e}')

    image_path = 'static/assets/img/saved_picture.jpg'

    desired_width = int(data.get('image').get('width'))
    desired_height = int(data.get('image').get('height'))
    print(desired_width,desired_height)

    image = cv2.imread(image_path)
    resized_image = cv2.resize(image, (desired_width, desired_height))
    predictions = data.get("predictions", [])
    results = []
    person_num=0

    for prediction in predictions:
        class_name = "person"
        x = prediction.get("x", 0)
        y = prediction.get("y", 0)
        width = prediction.get("width", 0)
        height = prediction.get("height", 0)
        person_num+=1
        # 결과 배열에 추가
        results.append([class_name, x, y, width, height])

    # 결과 출력
    for result in results:
        print(result)
    # 바운딩 박스 그리기
    for result in results:
        class_name, x, y, width, height = result
        x1, y1 = int(x - width / 2), int(y - height / 2) # 바운딩 박스 좌측 상단 꼭짓점 계산
        x2, y2 = int(x + width / 2), int(y + height / 2) # 바운딩 박스 우측 하단 꼭짓점 계산
        color = (0, 255, 0) # 바운딩 박스 색상 (여기서는 녹색)
        thickness = 2 # 바운딩 박스 두께

        cv2.rectangle(resized_image, (x1, y1), (x2, y2), color, thickness)

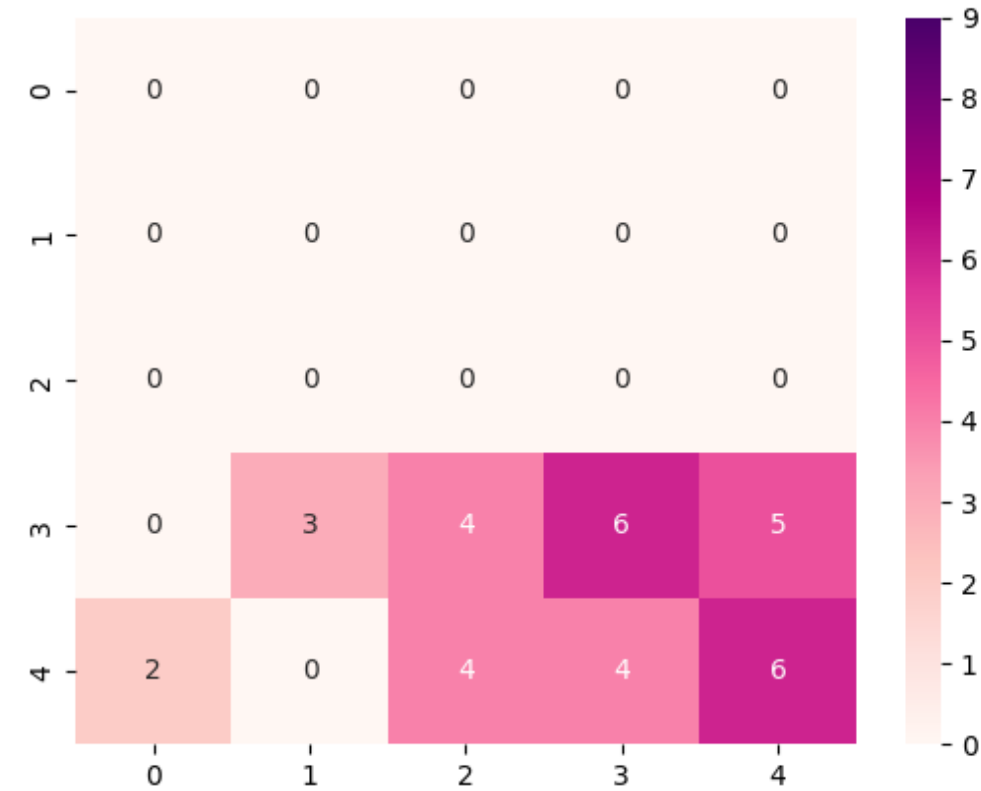
        text = f"{class_name}"
        cv2.putText(resized_image, text, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
color, thickness)

    output_image_path = 'static/assets/img/saved_picture.jpg'
    cv2.imwrite(output_image_path, resized_image)

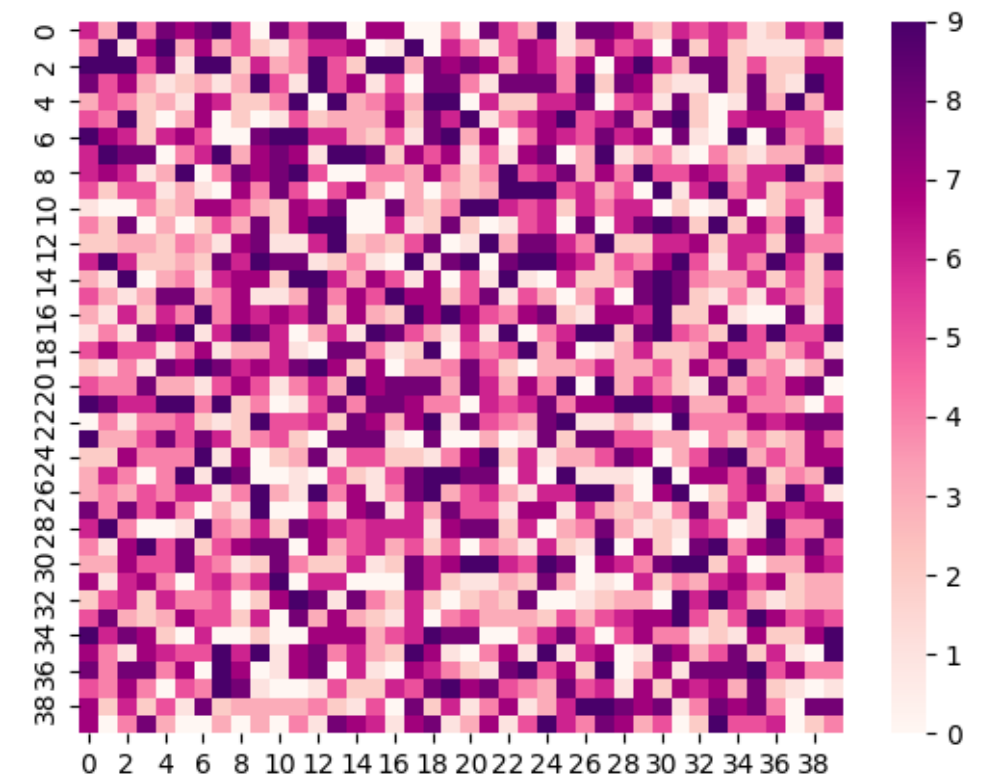
    Full_Size=desired_width*desired_height
    people_area=0
    for result in results:
        people_area += (int(result[3])*int(result[4]))
    Full_Density=(people_area/Full_Size)*100
    print(f'Full_Density%')
```

장소에 따른 히트맵

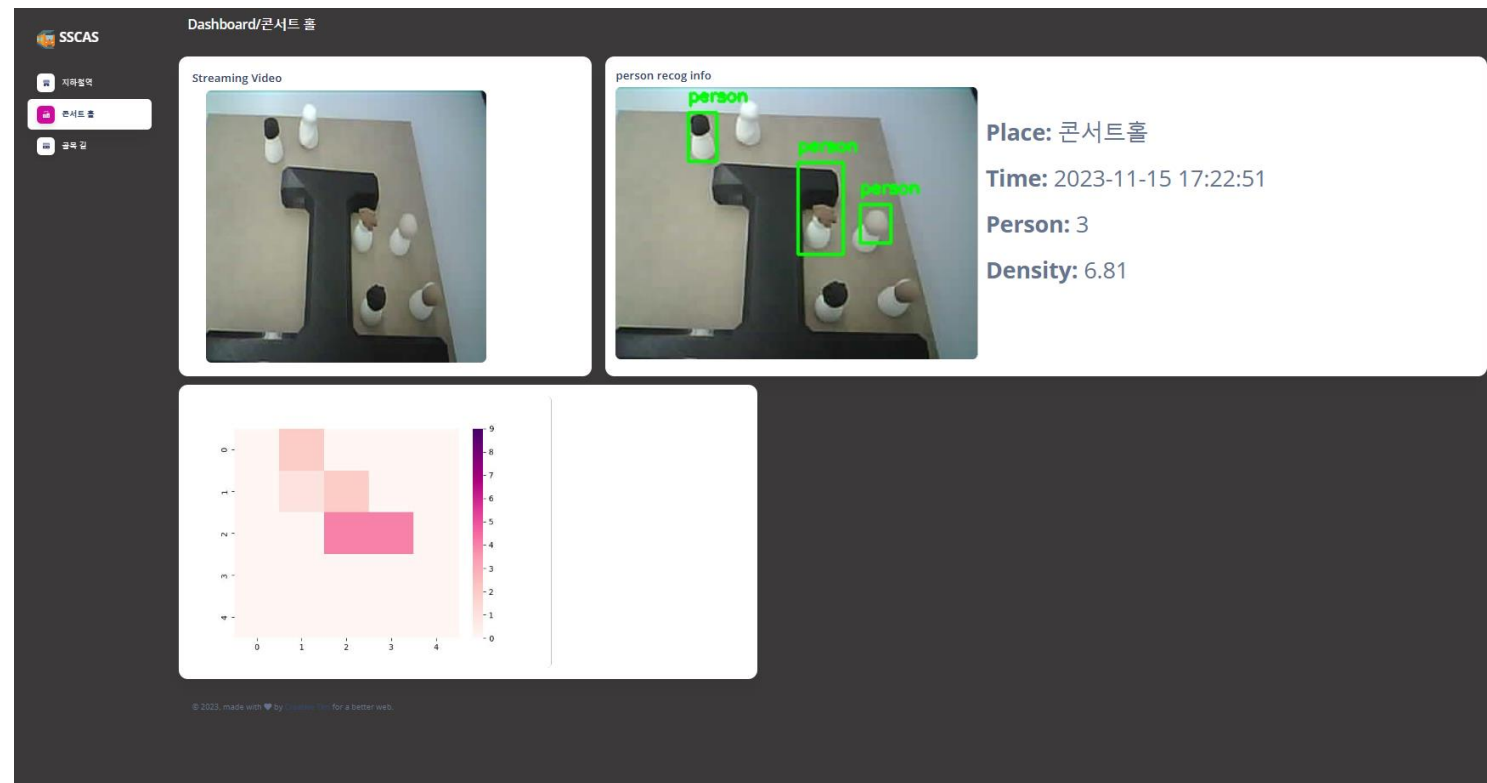
HeatMap: 히트맵은 색상 코딩 시스템을 사용해 다양한 값을 나타내는 데이터의 그래픽 표현이다.



사진(장소)을 가로, 세로로 5등분하여 각의 영역의 정확도를 구현
이렇게 정한 정확도를 범위에 따라 단계(1~9단계)를 나눠 Heat Map에
표시하였다.



AI-PADS WEB



WEB 사이트 제작

- ◆ Python의 Flask 프레임워크를 이용하여 웹서버 및 웹을 제작하였습니다.
- ◆ Dashboard 형태의 UI로 제작하여 각각의 장소를 선택하면 해당 장소의 혼잡도 및 상황을 모니터링 할 수 있도록 하였습니다.
- ◆ 실시간으로 계속 화면과 정보들이 업데이트되어 사용자에게 실시간으로 정보를 보여준다.

프로젝트 결과 (작동 영상)



프로젝트 결론 및 해석

객체 인식을 활용하여 혼잡도를 분석해 정보를 시각화 하여 웹 사이트에 나타내었다.

기대효과 :

- 장소에 도착하기 전 장소의 혼잡도를 알 수 있다.
- 축제나 행사를 진행 할 때 인원 배치를 보다 효율적으로 진행 할 수 있다.
- “이태원 참사”와 같은 압사 사고를 예방 할 수 있다.



감사합니다!

