



2. 학습 내용

<프론트엔드 파트>

웹사이트 구현을 위해 협업에서 사용되는 도구(React, Axios, WebSocket 등)의 전반적인 지식을 습득하고, 이를 활용한 웹사이트를 구현한다.

아래의 두가지 기능을 구현하였다.

1) 실시간 좌석 정보 페이지

해당 장소의 실시간 좌석 정보를 제공한다. WebSocket으로 실시간 좌석 정보를 받아온다.

2) 좌석 포화도 정보 페이지

해당 장소의 좌석 포화도 정보를 제공한다. Axios으로 좌석 포화도 정보를 받아온다.

<백엔드 파트>

프론트 팀과 디러닝 팀이 서로 데이터를 주고 받을 수 있도록 이어주는 역할을 하였으며, 그 과정에서 CentOS를 이용하여 서버를 운영하였다.

아래의 두가지 기능을 구현하였다.

1) API기능

디러닝 팀에서 분석한 데이터를 토대로 서버에 있는 데이터베이스에 저장해 프론트에서 요청을 할 시 데이터를 보내주는 역할을 하였다. 또한 디러닝 팀에서 분석에 필요한 데이터를 가지고 있다가 분석시 데이터를 넘겨주었다.

2) socket기능

모델팀에서 측정한 빈 좌석 유무 같은 경우에는 실시간으로 데이터를 넘겨주어야 하기 때문에 라즈베리파이에 있는 코드에서 spring으로 작성한 소켓에 클라이언트로 참여해 프론트와 통신을 할 수 있도록 하였다.

<디러닝 파트>

시스템에 필요한 기능을 디러닝을 활용하여 구현하는 것을 목표로 한다. 필요한 기능을 구현하는 것도 중요하지만, 디러닝에 대한 전반적인 이해와 해당 프로젝트에서 사용되는 YOLOv5와 LSTM의 이론에 관한 이해 또한 중요시하며 프로젝트를 진행하였다.

아래의 두가지 기능을 구현하였다.

1) 실시간 영상 객체 탐지로 포화도 분석

디러닝 팀은 라즈베리파이를 통해 얻은 데이터를 사용하여 유의미한 정보로 분석해내고 이를 백엔드 팀에 지속적으로 제공하였다. 라즈베리파이의 카메라에서 프레임을 읽어오고, 이 프레임을 실시간으로 분석(YOLOv5를 활용한 객체탐지)하여 좌석의 사용 가능 여부를 제공하였다. 해당 좌석의 사용 가능 여부를 일시, 테이블 위치, 사용 가능 여부 등과 함께 추출한다. 좌석의 사용 가능 여부는 인간 혹은 물체의 존재를 판단했으며, 이는 사람들이 좌석에 물체가 놓여있을 경우 해당 좌석을 사용하지 않는다는 점에서 유의미한 판단이라고 생각하였다.

2) LSTM 시계열 분석으로 금일 예측 포화도 제공

데이터베이스에서 누적된 데이터들을 사용하여 금일 시간대별 평균 공간 포화도 예측을 진행하였다. 이때, 포화도는 '불가용좌석 수/전체 좌석 수'로 정의하였다. LSTM 기반의 시계열 예측과 성능 강화를 진행하였다. 프론트엔드 측과 백엔드 측과의 지속적인 협의를 통해 유의미한 데이터들을 성공적으로 주고받을 수 있었다.

3. 회차별 학습보고

회차	학습 목표 및 활동	학습방법	학습시간
1	<p>프론트엔드 파트 Figma를 이용한 UI 설계 및 디자인</p> <p>백엔드 파트 Figma를 이용한 UI 설계 및 디자인 웹페이지 구성요소 회의</p> <p>딥러닝 파트 Figma를 이용한 UI 설계 및 디자인 수립</p>	<p>프론트엔드 파트 Figma 사용법 학습</p> <p>백엔드 파트 Figma 사용법 학습</p> <p>딥러닝 파트 Figma 사용법 학습</p>	<p>프론트엔드 파트 10시간</p> <p>백엔드 파트 8시간</p> <p>딥러닝 파트 8시간</p>
2	<p>프론트엔드 파트 프로젝트 생성 및 React 개념 학습</p> <p>백엔드 파트 MySQL 워크벤치를 활용하여 기본적인 문법 학습</p> <p>딥러닝 파트 Object detection의 역사 및 SSD 논문 학습</p>	<p>프론트엔드 파트 React 개념 학습</p> <p>백엔드 파트 MySQL기본 문법 학습</p> <p>딥러닝 파트 SSD 논문 정독</p>	<p>프론트엔드 파트 7시간</p> <p>백엔드 파트 10시간</p> <p>딥러닝 파트 8시간</p>
3	<p>프론트엔드 파트 반응형 화면 구현</p> <p>백엔드 파트: 데이터베이스 생성 및 설계</p> <p>딥러닝 파트 YOLOv1 논문 이해 및 YOLOv5 작동 방식 학습</p>	<p>프론트엔드 파트 react-responsive 학습</p> <p>백엔드 파트 데이터 종류 타입 회의, 테이블 구축</p> <p>딥러닝 파트 YOLO(v1) 논문 이해</p>	<p>프론트엔드 파트 10시간</p> <p>백엔드 파트 7시간</p> <p>딥러닝 파트 8시간</p>
4	<p>프론트엔드 파트 홈 화면 구현 및 화면 이동 구현</p> <p>백엔드 파트 데이터베이스 생성 및 설계</p>	<p>프론트엔드 파트 react-router-dom 학습</p> <p>백엔드 파트 검색을 통해 웹 흐름도 학습</p>	<p>프론트엔드 파트 10시간</p> <p>백엔드 파트 10시간</p>



	<p>딥러닝 파트 YOLOv5 설치 및 테스트</p>	<p>딥러닝 파트 YOLOv5 공식 깃허브 문서 이해</p>	<p>딥러닝 파트 8시간</p>
5	<p>프론트엔드 파트 장소 선택 화면 구현</p> <p>백엔드 파트 Spring 작동 방법에 대해 공부한 내용을 바탕으로 코드 작성</p> <p>딥러닝 파트 실시간 영상 객체 탐지 구현</p>	<p>프론트엔드 파트 overflow 속성 학습</p> <p>백엔드 파트 Spring 공식 문서 학습</p> <p>딥러닝 파트 베이스라인 코드 작성 및 test</p>	<p>프론트엔드 파트 8시간</p> <p>백엔드 파트 10시간</p> <p>딥러닝 파트 8시간</p>
6	<p>프론트엔드 파트 실시간 좌석 정보 화면 구현(1) 좌석 배치도 영역 구현</p> <p>백엔드 파트 스프링과 서버 MySQL 연결</p> <p>딥러닝 파트 라즈베리파이 카메라 모듈 연동 및 데이터베이스 연동법 학습</p>	<p>프론트엔드 파트 좌석 배치도 참고 및 scroll 속성 학습</p> <p>백엔드 파트 mysql 공식문서 및 오류명 검색을 통해 수정</p> <p>딥러닝 파트 카메라모듈 공식문서</p>	<p>프론트엔드 파트 6시간</p> <p>백엔드 파트 10시간</p> <p>딥러닝 파트 10시간</p>
7	<p>프론트엔드 파트 실시간 좌석 정보 화면 구현(2) 종류별 테이블 및 좌석 배치도 구현</p> <p>백엔드 파트 Socket 통신 공부</p> <p>딥러닝 파트 라즈베리파이 카메라 모듈 연동 및 데이터 분석용 임시 데이터셋 구축</p>	<p>프론트엔드 파트 함수 생성 및 useState 학습</p> <p>백엔드 파트 논문을 통해 웹소켓 작동방식 파악</p> <p>딥러닝 파트 pysql 공부 & 카메라 오류해결</p>	<p>프론트엔드 파트 7시간</p> <p>백엔드 파트 6시간</p> <p>딥러닝 파트 8시간</p>
8	<p>프론트엔드 파트 좌석 포화도 정보 화면 구현(1) 포화도 데이터 그래프로 표시</p>	<p>프론트엔드 파트 Recharts 학습</p>	<p>프론트엔드 파트 8시간</p>



	<p>백엔드 파트 파이썬과 스프링을 이용하여 Socket 통신 실습</p> <p>딥러닝 파트 시계열 데이터 분석 이론 학습 및 임시 데이터를 활용한 베이스라인 코드 작성</p>	<p>백엔드 파트 구글링을 통해 다른 사람의 코드를 분석하고 적용</p> <p>딥러닝 파트 stackoverflow로 오류 해결</p>	<p>백엔드 파트 10시</p> <p>딥러닝 파트 8시간</p>
9	<p>프론트엔드 파트 좌석 포화도 정보 화면 구현(2) 포화도 데이터 CustomTooltip으로 표시</p> <p>백엔드 파트 파이썬을 이용하여 서버 MySQL 연동 후 Socket 코드와 합치기</p> <p>딥러닝 파트 백엔드와의 협업 데이터 적재 및 전송 코드 지속 실행 설정</p>	<p>프론트엔드 파트 CustomTooltip 기능 학습</p> <p>백엔드 파트 MySQL과 socket통신을 동시에 실행시키기 위해 multiprocessing학습</p> <p>딥러닝 파트 백엔드측과 회의 및 코드 작성</p>	<p>프론트엔드 파트 6시간</p> <p>백엔드 파트 10시간</p> <p>딥러닝 파트 8시간</p>
10	<p>프론트엔드 파트 서버에 배포, 좌석 포화도 정보 및 실시간 좌석 정보 데이터 받아오기</p> <p>백엔드 파트 Spring으로 작성한 코드를 직접 서버에 배포</p> <p>딥러닝 파트 프론트엔드와의 협업 : 웹과의 연동, 테이블별 데이터 전송 테스트</p>	<p>프론트엔드 파트 git clone, Axios, WebSocket 학습</p> <p>백엔드 파트 검색을 통해 다른 사람들의 배포 방법을 보고 실제로 진행</p> <p>딥러닝 파트 프론트엔드 파트와 웹 연결 확인 및 코드 작성</p>	<p>프론트엔드 파트 10시간</p> <p>백엔드 파트 10시간</p> <p>딥러닝 파트 8시간</p>
11	<p>프론트엔드 파트 설명서 페이지 구현</p>	<p>프론트엔드 파트 설명서 예제 참고</p>	<p>프론트엔드 파트 6시간</p>



	<p>백엔드 파트 동아리방에 설치해서 직접 테스트</p> <p>딥러닝 파트 라즈베리파이 케이스 및 안내문 제작 및 테스트 수행</p>	<p>백엔드 파트 테스트를 위한 테스트 설치 및 데이터 적재 확인</p> <p>딥러닝 파트 2회에 걸친 미팅 및 동아리방 test</p>	<p>백엔드 파트 6시간</p> <p>딥러닝 파트 8시간</p>
12	<p>프론트엔드 파트 접속자 수 구하기</p> <p>백엔드 파트 학정에서 실제로 설치해서 사용 (예약종료, 자동실행 기능 추가)</p> <p>딥러닝 파트 학술정보원 라즈베리파이 설치 테스트 및 해당 공간에 맞는 분석 환경 설정</p>	<p>프론트엔드 파트 HITS 기능 학습</p> <p>백엔드 파트 학정에 설치하여 주기적으로 오류값이 없는지 확인</p> <p>딥러닝 파트 실 설치 및 분석 환경 세팅</p>	<p>프론트엔드 파트 6시간</p> <p>백엔드 파트 8시간</p> <p>딥러닝 파트 8시간</p>
13	<p>프론트엔드 파트 Beta 버전 표시</p> <p>백엔드 파트 유지보수 및 데이터 관리</p> <p>딥러닝 파트 학술정보원 라즈베리파이 설치, 누적 데이터 바탕 LSTM 성능 강화 및 결과 확인</p>	<p>프론트엔드 파트 Beta 버전의 의미 학습 및 구현</p> <p>백엔드 파트 지난 데이터 값들이 이상하진 않은지 측정은 잘 되고 있는지 확인</p> <p>딥러닝 파트 딥러닝 성능 강화 공부</p>	<p>프론트엔드 파트 6시간</p> <p>백엔드 파트 10시간</p> <p>딥러닝 파트 6시간</p>
14	<p>프로젝트 결과 정리 및 회고</p>	<p>프로젝트 결과 정리 및 회고</p>	<p>프론트엔드 파트 6시간</p> <p>백엔드 파트 6시간</p> <p>딥러닝 파트 6시간</p>



4. 팀원 간 역할 및 개별 성과

<서민정 팀장>

1) 역할

팀장, 딥러닝(객체 탐지, 시계열) 팀의 스터디 및 코드 작성, 하드웨어(라즈베리파이) 연결 및 유지보수 수행

2) 개별 성과

i. 컴퓨터 비전

이론 복습 및 스터디 (yolov5)

파이토치 프레임워크 실습

라즈베리파이에서 opencv 실습

ii. 시계열 분석

다양한 딥러닝 프레임워크 실습

전처리 과정 코드 작성

iii. 하드웨어

라즈베리파이 카메라 모듈 연결 및 실행

리눅스 명령어 실습

iv. 기타

프론트엔드 / 백엔드 팀원과 회의 및 통신

<김이나 팀원>

1) 역할

딥러닝(객체 탐지, 시계열) 팀의 스터디 및 코드 작성 , 하드웨어(라즈베리파이) 연결 및 유지보수 수행

2) 개별 성과

i. 컴퓨터 비전

이론 복습 및 스터디 (yolov5)

파이토치 프레임워크 실습

라즈베리파이에서 opencv 실습

ii. 시계열 분석

다양한 딥러닝 프레임워크 실습

전처리 과정 코드 작성

iii. 하드웨어

라즈베리파이 카메라 모듈 연결 및 실행

리눅스 명령어 실습

iv. 기타

프론트엔드 / 백엔드 팀원과 회의 및 통신



<이준민 팀원>

1) 역할

프론트 엔드, 반응형 웹사이트 구현

2) 개별 성과

i. 프로젝트 생성 및 배포

React 설치 및 프로젝트 생성

Git clone 배포

ii. 반응형 화면 구현 및 화면 이동 구현

react-responsive 설치 및 구현

react-router-dom 설치 및 구현

iii. 좌석 배치도 구현 및 실시간 좌석 정보 적용

WebSocket으로 실시간 정보 받아오기

iv. 좌석 포화도 그래프 구현 및 포화도 정보 적용

Axios으로 포화도 정보 받아오기

v. 설명서 페이지와 같은 부가적인 기능 구현

HITS으로 접속자 수 구하기

<안용식 팀원>

1) 역할

백엔드 (소켓, 데이터 베이스 ,서버관리) 코드 작성, 라즈베리파이 연결 및 유지보수

2) 개별 성과

i. API

Spring 스터디

서버 배포

데이터베이스와 통신과 API제공

ii. socket통신

실시간 데이터 통신

iii. Mysql

Mysql 명령어 공부

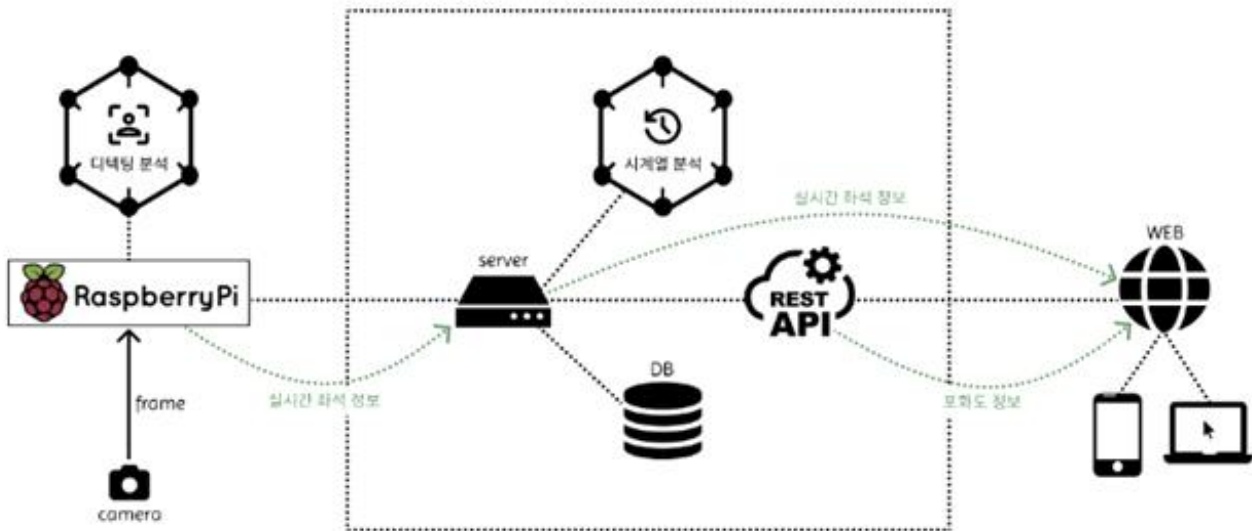
테이블 생성 및 데이터 관리

서버와 연결

5. 최종결과물 및 팀별 성과

총 14주차에 걸친 과제 수행을 통하여 프로젝트의 시스템 구축 및 배포를 완료하였다. 최종적으로 완성된 결과물은 아래와 같다.

< 시스템 구조도 >



* seatnullnull 시스템 구조도

< 하드웨어 >

포화도 분석을 위하여 라즈베리파이와 카메라의 구성을 사용하였다. 프레임을 사용한 공간 포화도 분석 과정은 해당 라즈베리파이에서 수행되며, 이 결과를 소켓 통신 및 데이터베이스 적재를 통해 웹에 표시한다. 이 하드웨어는 공간의 시야각이 확보된 곳에 설치해야 하며, 설치된 모습은 아래와 같다. 학술정보원과의 협의를 통하여 11월 25일 설치를 진행하였다.



* 학술정보원 하드웨어 설치 모습 및 안내문 부착 모습

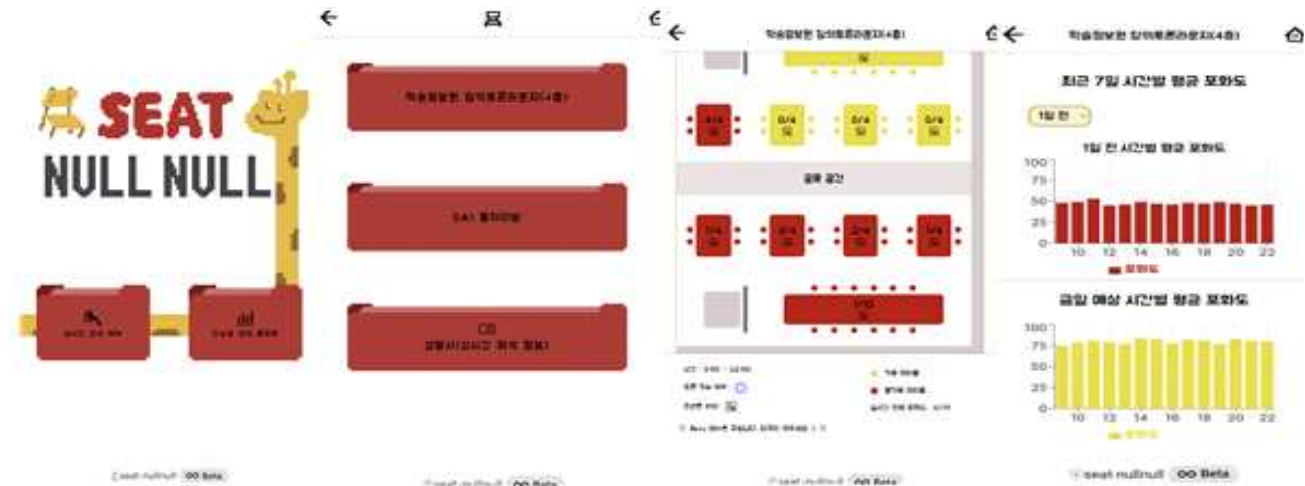
추가적으로 학생들에게 라즈베리파이와 카메라가 낯설지 않도록 친근하고 상세한 설명서를 작성하여 부착하였다. 이때, 안내문의 큐알 코드를 통해 홈페이지에 접속할 수 있도록 하였다.



< 소프트웨어 >

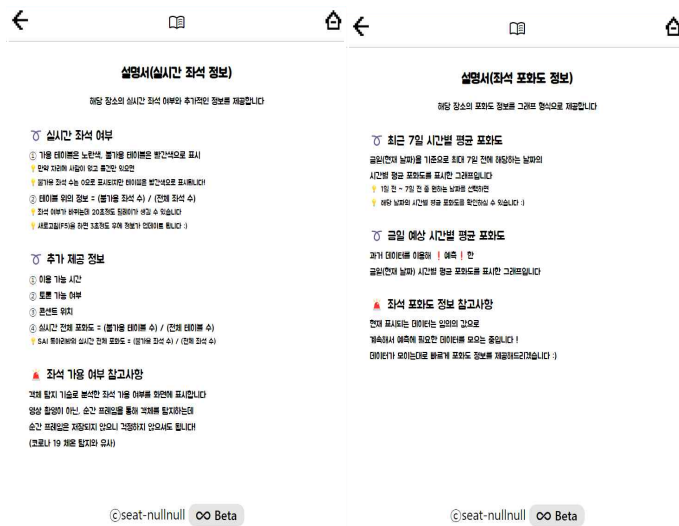
1) 웹페이지, 프론트엔드 파트 성과

학생들이 접속을 통하여 분석 결과를 확인할 수 있는 웹사이트를 설계하였다. 사이트 구성은 아래와 같다. 메인페이지에는 두가지 버튼이 존재한다. [버튼1 : 실시간 좌석 여부 / 버튼2 : 장소별 포화도 분석]
버튼1을 선택한 후 원하는 공간을 선택하면 해당 공간의 실시간 포화도 분석 결과를 확인할 수 있는 배치도를 확인할 수 있다. 버튼2를 선택한 후 원하는 공간을 선택하면 해당 공간의 과거 시간별 평균 포화도 및 금일 예측 시간별 평균 포화도를 확인할 수 있다.



* 웹사이트 UI

추가적으로, 학생들이 각각의 기능을 원활하게 사용할 수 있도록 설명서 페이지를 제작하였다.



* 학생들에게 웹페이지를 이용하기 위한 설명서

seatnullnull 링크 <http://seatnullnull.com:3000/>

2) 포화도 분석 / 포화도 예측, 딥러닝 파트 성과

프로젝트의 주요 기능인 포화도 분석의 경우, 라즈베리파이에서 읽어드린 프레임을 YOLOv5 모델을 활용한 코드를 백그라운드 실행을 통해 테이블별 인원수 및 물건 존재 여부를 분석한다. 이때, 공간에 맞는 테이블 바운딩을 통해 좌석별 분석을 시행한다. 또한 포화도 예측의 경우 데이터 하루의 적재가 마무리된 시점, 서버에서 코드를 실행하여 예측을 수행한다. 데이터베이스에서 최근 데이터를 불러드려 LSTM 모델의 입력으로 사용한 후, 해당 결과를 데이터베이스 예측 테이블에 저장하여 이를 다음날 페이지에 전송한다.



* 라즈베리파이의 테이블 바운딩 모습

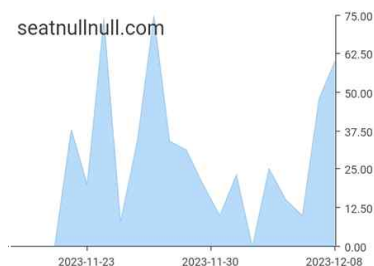
3) 데이터 누적 및 전송, 백엔드 파트 성과

포화도를 분석 결과를 학생들에게 실시간으로 전달하기 위하여 소켓을 사용하였으며, 포화도 예측을 위한 분석 포화도 데이터 누적 및 예측 포화도 저장을 위해 데이터베이스를 구축하여 데이터 누적 과정을 수행하였다.

Past_distribution		distributionAverage		Pre_distribution	
PK	데이터ID	PK	데이터ID	PK	데이터ID
	분포도 날짜 시간		분포도 날짜 시간 분		분포도 날짜 시간

* seatnullnull 데이터베이스 ERD

< 배포 후 학생 접속 수 >



*HITS seatnullnull 접속수 그래프

최종적으로 학술정보원이 열지 않는 일요일에는 접속자수가 낮으며, 전반적으로 20~70의 일 접속자수를 가지는 모습을 확인할 수 있었다.