**230204 멘토링 자료**

| 프로젝트 주제 | 개인형 이동장치 주차 안내 | | |
| --- | --- | --- | --- |
| 전체 기능 | **1. 개인형 이동장치 주차 점수**   * 사용자가 개인형 이동장치가 주차된 모습을 촬영한 사진과 사용자의 현재 위치와 기울기 센서로 주차 점수 계산   + 이미지 내 금지구역 마커의 유무 확인과 안내     - 금지구역 마커: 횡단보도, 자전거도로, 점자블럭, 보도블럭 중앙   + 현 위치 위도 경도 데이터를 통해 근처 금지구역 마커 유무 확인과 거리 파악 및 안내     - 금지구역 마커: 지하철역, 버스정거장, 택시승강장, 어린이보호구역(초등학교, 어린이집 등), 지상소화전   + 기울기 센서를 통해 현 이동장치의 위치 경사도를 파악   + 위 항목들의 주차 적합도를 계산한 뒤 점수 환산 및 적합도 안내   **2. 개인형 이동장치 주차가능구역 위치 안내**  **-** 현 위치의 위도 경도를 통하여 가장 가까운 주차장 위치 최단경로, 거리, 예상 소요 시간 안내  - 또는 금지 구역 마커가 포함 되어있지 않은 공간 위치, 경로 추천 | | |
| 기능별 진행 사항 | | **기능** | **주요 업무** | **세부 업무** | **상태** | | --- | --- | --- | --- | | 데이터베이스 구축 | 주차금지구역 및 주차가능구역 | 주차금지구역 및 주차가능구역 데이터 수집 | 진행 | | 회원 정보 |  | 예정 | | MariaDB 연동 | MariaDB 연동 | 완료 | | 회원가입/로그인 |  |  | 예정 | | 주차 점수 계산 | 이미지 분석 모델 | 이미지 데이터 수집 | 완료 | | 객체 인식 모델 생성 | 진행 | | 주차 이미지 판독 | 주차 이미지 서버 전송 | 완료 | | 이미지 객체 인식 | 완료 | | 이미지 판독 점수 환산 | 예정 | | 주차 위치 판독 | GPS 위치 정보 서버 전송 | 완료 | | 현 위치에서 주차가능구역 및 주차금지구역 사이의 거리 계산 | 완료 | | 위치 판독 점수 환산 | 예정 | | 주차 기울기 판독 | 기울기 센서 구현 | 예정 | | 기울기 센서 데이터 서버 전송 | 예정 | | 기울기 판독 점수 환산 | 예정 | | 결과 도출 | 주차 점수 계산 | 예정 | | 현 위치 인근 주차가능구역 및 주차금지구역 지도로 안내 | 완료 | | 주차가능구역 경로 안내 | 주차가능구역 경로 안내 | GPS 위치 정보 서버 전송 | 완료 | | 현 위치에서 주차가능구역 사이의 도보 경로 거리 계산 | 완료 | | 현 위치에서 도보 경로가 가장 가까운 주차가능구역 지도로 안내 | 완료 | | | |
| 객체 인식 모델 | **모델 학습 결과**   | **노면** | **킥보드** | | --- | --- | |  |  | | **클래스 별 정확도**   | **클래스** | **AP** | **TP** | **FP** | | --- | --- | --- | --- | | 보도블럭 | 63.27% | 430 | 319 | | 횡단보도 | 52.05% | 135 | 117 | | 점자블럭 | 70.98% | 306 | 129 | | 자전거도로 | 62.09% | 104 | 58 |   **학습 결과**   | **precision** | **recall** | **F1-score** | | --- | --- | --- | | 0.61 | 0.59 | 0.60 | | **클래스 별 정확도**   | **클래스** | **AP** | **TP** | **FP** | | --- | --- | --- | --- | | 킥보드 | 0.00% | 0 | 0 | | 바퀴 | 71.66% | 540 | 375 |   **학습 결과**   | **precision** | **recall** | **F1-score** | | --- | --- | --- | | 0.59 | 0.55 | 0.57 | | |  |  | | | |
| 데이터베이스 구조 |  | | |
| 시스템 구성도 |  | | |
| 피드백 | **-기능을 만든다기보다 보완에 신경 쓰고 피피티에 집중하는 것이 중요하다.**  **-모델은 왜 이걸 사용했는지, 모델 판별 정확도는 어떻게 되는지, 과정 등등 자세히 설명하면 좋을것 같음**  **-금지구역이 왜 선정되었는지 알림이 필요**  **-시연 영상은 현실성을 드러내기 위해 좌표랑 가까운(50m 100m)정도에서 찍을것**  <https://github.com/dobal-production/eksworkshop-custom/blob/main/020_prerequisites/06.install-helm.sh>  <https://github.com/dobal-production/eksworkshop-custom/tree/main/020_prerequisites>  thumbnail 이미지로 보여주기  **-아키텍처 구성도 보강** [**https://docs.aws.amazon.com/prescriptive-guidance/latest/containers-provision-eks-clusters-terraform/architecture.html**](https://docs.aws.amazon.com/prescriptive-guidance/latest/containers-provision-eks-clusters-terraform/architecture.html)  **시스템에는 kube 포함 x**  **서비스에는 kube 포함하여 진행 (deployment이런거 표현하지 말고고)**  **-이미지를 thumbnail화하여 용량을 줄여 response 시간을 줄여볼것**  **-CA HPA를 시연영상에 포함시킬것:**  **이용툴 : ops\_view**  [**https://github.com/dobal-production/eksworkshop-custom/tree/main/020\_prerequisites**](https://github.com/dobal-production/eksworkshop-custom/tree/main/020_prerequisites)  [**https://github.com/dobal-production/eksworkshop-custom/blob/main/020\_prerequisites/06.install-helm.sh**](https://github.com/dobal-production/eksworkshop-custom/blob/main/020_prerequisites/06.install-helm.sh)  **http://abfb5e29688474903be90ee45eda729b-1395073044.ap-northeast-2.elb.amazonaws.com/#**  **-aws 서비스 이용한것들 결과 보고에 잘 녹여내 볼것**  **- eksctl 사용 CloudFormation 사용 cm secret 사용 pvc 사용(관련 yaml 만들어서 첨부해볼것)**  **iam serviceaccount 사용**     * s3 public설정후 url로 진행하는 방법도 있고 빠르겠지만 시간이 없으니 개선방향으로 적을것 * 이미지에 정책을 기간제로 달거나 glacier로 보내는등 처리방법도 있음!   **-데이터 베이스 table 수정할것 ( 계정 관리)**  **score 테이블의 PK로 row id 대신에 날짜와 사용자 아이디로 concat key를 사용하는 것 추천**  **-점수 관리 시스템 구축 시작할것**  **-사진 보내는 기능 좀 더 간편화???**  **-점수 스코어링**  같은 dimension 상에서 존재하는 요소들을 묶어서(clustering) 군집으로 묶은 것에 대해 점수를 매기기  상관관계 파악을 통해 변수를 선정 → 그 변수로 그룹을 지정해서  개별 위치가 아닌, 지금 나와 있는 정보를 기반으로 군집 분석 → 군집 단위로 점수를 제공  - 주차가능구역  정의가 어려우면 러프하게 하고 나중에 수정하겠다…  혹은 현재 주차장이 부족하다. 증설이 필요하다 같은 느낌으로 사회/정책적으로 제언  **포트폴리오 예시**  <https://github.com/hellacles/BigData/blob/master/%EC%9C%B5%ED%95%A9_4%EC%A1%B0_%ED%97%AC%EB%9D%BC%ED%81%B4%EB%A0%88%EC%8A%A4_%EB%B0%B0%ED%8F%AC%EC%9A%A9.pdf> | | |
| 반영 방안 | 시급   1. UI  * 데이터베이스(score, user) * 로그인/회원가입 * score 계산 * iot 연결 * 이미지를 받아와서 yolo 서버로 post → temp 파일로 yolo 돌리고 → uri 변환(python base64) → 받아와서 render * 이미지 받아와서 파일 서버로 저장 → thumbnail로 저장 → yolo 서버 & html에 url로 받아오기 -이미지 처리??? * 예비 최종 발표 2/11일 발표 준비 앞당길 필요 있습니다.   후 순위   * post 이미지 thumbnail로   보고서 추가할 내용   * 기능 구성도 * 아키텍처 구성도 보강 * CA HPA 시연 영상 * 이미지 모델 설명 | | |

## 