

인하대학교

컴퓨터공학 종합설계

최종 보고서

[시각장애인을 위한 음성 안내 APP]

컴퓨터공학과

8조

김난영 김혜윤 이영주

목차

| | |
|--------------------|----|
| 1. 개요 | 3 |
| 1.1 주제 선정 배경 및 필요성 | 3 |
| 1.2 선행 연구 및 개발 현황 | 3 |
| 1.3 본 애플리케이션의 차별성 | 6 |
| 1.4 개발 목표 | 7 |
| 2. 프로젝트 구성 | 8 |
| 2.1 시스템 | 8 |
| 2.2 소프트웨어 | 9 |
| 2.3 데이터베이스 | 10 |
| 3. 프로젝트 개발 내용 | 13 |
| 3.1 개발환경 | 13 |
| 3.2 문제 및 해결방안 | 14 |
| 3.3 시험 시나리오 | 15 |
| 3.4 Prototype 구현 | 15 |
| 3.5 상세 설계 | 20 |
| 3.6 시험 / 테스트 결과 | 33 |
| 3.7 Coding & DEMO | 33 |
| 4. 프로젝트 결과 및 활용 | 33 |
| 4.1 프로젝트 활용 | 33 |
| 4.2 프로젝트 향후 계획 | 34 |
| 5. 일정 및 역할 분담 | 34 |
| 5.1 팀 역할 분담 | 35 |
| 5.2 개발 일정 | 35 |

1. 개요

본 장에서는 Google MAP API, T MAP API, Geo-fencing과 Real-Time Object Detection을 이용하여 시각장애인에게 안전한 길을 음성으로 안내하는 서비스에 대한 주제 선정 배경 및 필요성, 기존 연구 및 차별성, 개발 목표 그리고 본 문서의 개요를 소개한다.

1.1 주제 선정 배경 및 필요성

1.1.1 주제 선정 배경

올해 초부터 발생한 코로나19(Covid-19, 이하 코로나) 바이러스에 의해 비대면 회의 및 수업이 증가하고 있다. 실시간 화상 회의 플랫폼을 이용하여 온라인으로 수업을 듣던 중, 시각장애인들이 도우미 없이 수업을 듣는 방식에 의문이 들었다. 이를 시발점으로 하여 시각장애인들이 겪는 불편함에 대해 관심이 생겼다. 장애인들을 위한 보호 시스템이 부족한 현실에 도움이 되고자 프로젝트를 기획하게 되었다.

1.1.2 필요성



[Figure 1] 시각장애인의 스마트폰 보유율

조사한 자료에 따르면 시각장애인의 약 60%(2017년 기준)가 스마트폰을 사용하고 있다. 그러나 현재 국내의 시각장애인을 지원하기 위한 어플의 수가 적고, 그의 기능이 충분치 못하다. 기존의 어플을 보완하여 보다 유용한 기능을 제공한다면 시각장애인들이 어플을 이용함으로써 도움을 받을 수 있다.

현재 인하대학교 후문 신호등에는 음향 신호기가 부착되어있지 않다. 또한 보도와 차도의 구분이 모호하고 미흡한 점자블록으로 인해 보행 시에 어려움이 있다. 인하대

후문 주변의 안전한 경로를 제공하는 어플이 존재한다면 큰 도움이 될 것이라 생각하여 이 프로젝트를 기획하였다.

1.2 선행 연구 및 개발 현황

1.2.1 RightHear



[Figure 2] RightHear 로고

| 기능 | 단점 |
|---------------------------------|----------------|
| 사용자의 현재 위치를 알려준다. | 한국어를 지원하지 않는다. |
| 경로를 음성으로 안내한다. | |
| 추가 안내가 필요하다면 자원봉사자와 전화를 연결해 준다. | |
| 계단의 개수와 같은 현재 주변 상황을 음성으로 알려준다. | |

[표 1] RightHear 기능 및 단점

1.2.2 WayFindr



[Figure 3] wayfindr 로고

| 기능 | 단점 |
|---|-----------------------------|
| 근거리 무선통신장치 비콘(Beacon)과 블루투스를 이용하여 지하철역과 환승 통로를 음성으로 제공한다. | 영국 런던 내 지하철 역 내에서만 작동한다. |
| 이용자 위치를 파악하여 이동 경로를 음성으로 안내한다. | |
| 장애물, 계단의 개수, 복잡한 환승 경로 등을 제공한다. | |

[표 2] wayfindr 기능 및 단점

1.2.3 강남지팡이



[Figure 4] 강남지팡이 로고

| 기능 | 단점 |
|--|---|
| 사물인터넷(IoT) 기술을 활용한다. | 사용 가능 지역 제한 (대모산입구역과 하상장애인복지관 보행로, 하상장애인복지관, 양재천산책로(영동6교~대치교), 강남구청과 강남구청역 간 보행로, 강남구청 건물 내부 |
| 보행로에 설치한 비콘 센서와 반응하여 위치 및 주변 시설 정보(보행로 방향 및 장애물 등)를 음성서비스로 제공한다. | |

[표 3] 강남지팡이 기능 및 단점

1.2.4 커뮤니티 맵핑



[Figure 5] 커뮤니티 맵핑 로고

| 기능 | 단점 |
|--|--------------------|
| 비장애인들이 신호등을 앱으로 촬영하여 서버에 등록함으로써 도로 상황을 공유한다. | 정보 공유에 대한 참여율이 낮다. |
| 시각장애인은 공유된 정보를 바탕으로 음성 길안내 서비스를 실시간으로 받는다. | |
| GPS 정보를 기반으로 신호등과의 거리를 음성으로 안내해준다. | |
| 데이터를 공유한 사용자에게 보상을 지급한다. | |

[표 4] 커뮤니티 맵핑 기능 및 단점

1.3 본 애플리케이션의 차별성

1.3.1 터치 기능

시각장애인은 작은 스마트폰 화면에 컴포넌트들이 배치된 위치 및 형식을 알기 어렵다. 이러한 불편함을 해소하기 위해 컴포넌트를 한 번 터치하면 해당 컴포넌트의 내용이 음성으로 출력되며, 두 번 터치시 선택이 되도록 구현하였다. 이를 통해 VoiceOver 기능을 기본적으로 제공하지 않는 핸드폰에서도 보다 쉽게 애플리케이션을 사용할 수 있다.

1.3.2 서비스 지역

현재 인하대 후문에서는 앞서 소개한 국내 어플 서비스를 이용할 수 없다. 우리는 인하대 후문의 정확한 상황을 파악하여 주변 위험 요소 등 여러 정보를 제공할 예정이다.

1.3.3 경고 안내

보행 시 위험지역에 접근할 경우, 음성으로 알림을 주어 안전성을 제공한다. 인하대 후문 주변지역은 시각장애인의 보행을 돕는 점자블록이 충분히 설치되어 있지 않은 상태이다. 점자블록이 신호등 주변에만 설치되어 있는 상황을 보완하기 위해 경고 음성을 출력함으로써 보행 안전성을 높인다.

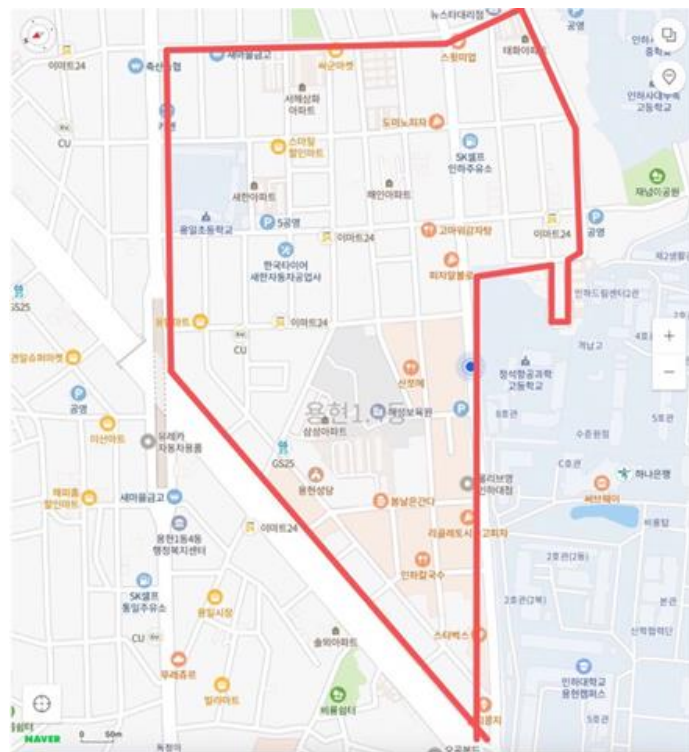
1.3.4 신호등 안내

신호등 색과 같은 신호등 정보를 실시간으로 음성 안내한다. 인하대 후문 주변 지역의 신호등에는 음향 신호기가 부착되어있지 않은 상태이다. 시각장애인이 횡단보도를 이용하여 횡단할 때 겪는 어려움을 보완하기 위해 실시간 객체 인식 모델을 기반으로 신호등 정보를 음성으로 안내한다.

1.4 개발 목표

본 프로젝트의 목표는 기존 ‘커뮤니티 맵핑’, ‘강남 지팡이’ 등 시각장애인을 위한 국내 어플리케이션에서 제공하지 않았던 신호등 정보와 위험 요소를 파악하여 인하대 후문 범위 내에서 안전한 길을 안내해주는 서비스를 제공하는 것이다.

소개에 앞서 먼저 우리가 지정한 인하대 후문 범위에 대해 설명하고자 한다.



[Figure 6] 인하대 후문 범위

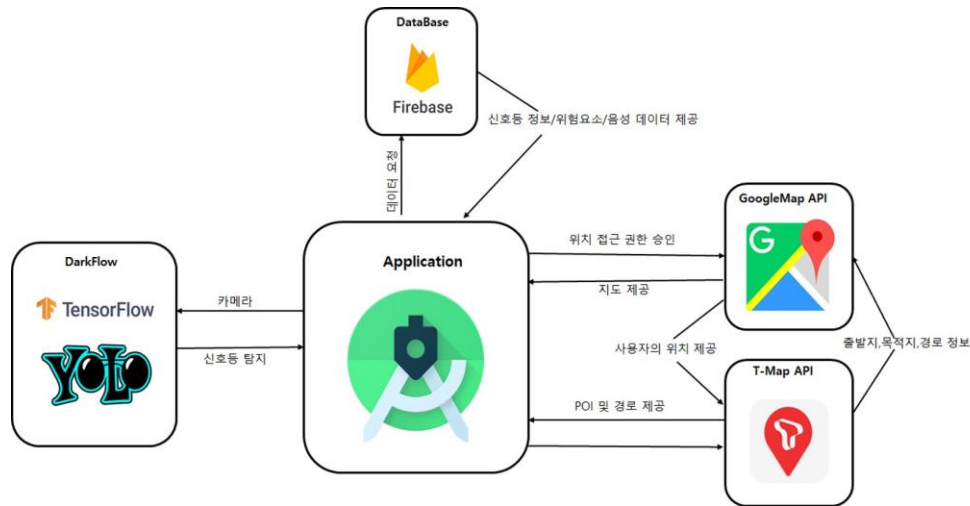
서비스 범위를 지정하기 위하여 부동산에 자문을 구하는 등 자료 조사를 통해 인하대 후문 앞 신호등을 기준으로 도보 10분 이내의 범위인 사진 상단에 위치한 용남치안센터 부터 하단에 위치한 굴다리까지로 지정하였다. 이는 위험 요소에 관한 정보를 제한된 인원 및 환경 속에서 관리하기 용이하게 하기 위해서이다.

본 애플리케이션은 다음과 같은 기능을 제공하고자 한다.

- 화면 구성을 파악하는데에 어려움이 있는 시각장애인에게 편의를 제공하기 위해 한번 터치 시 음성 출력, 두 번 터치 시 컴포넌트가 선택되도록 한다.
- STT API를 이용하여 사용자의 음성을 텍스트로 변환한다.
- TTS API를 이용하여 텍스트를 음성으로 변환하여 사용자에게 제공한다.
- Google MAP API를 이용하여 어플리케이션 내에 지도를 생성하고 GPS와 함께 사용자의 위치를 실시간으로 확인한다.
- Geofencing을 이용하여 신호등 및 위험요소에 사용자가 접근할 경우, 일정 범위 내에 들어가는지 확인한다.
- TMAP API를 이용하여 보행자의 도보 경로를 제공한다.
- DB에 보행자에게 안전하지 않을 수 있는 요소를 추가하여 경로 탐색 시 TTS API를 이용하여 위험요소가 가까이 있을 경우 음성으로 안내한다.
- Darkflow 모델을 이용하여 실시간으로 신호등을 탐지해 사용자에게 신호등 정보를 안내한다.

2. 프로젝트 구성

2.1 시스템 구성도

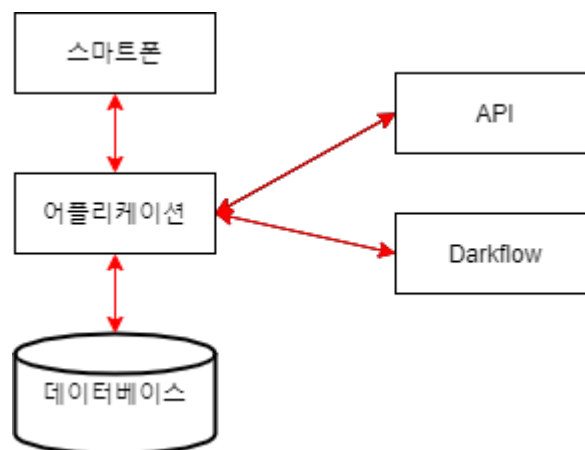


[Figure 7] 시스템 아키텍처

시스템 기능 설명

- 어플리케이션은 Android application을 통해 API를 요청 후 받아오고 경로 안내 서비스를 제공한다.
- 데이터베이스는 Firebase 서버를 이용하여 통신한다.
- 사용자가 입력한 목적지를 STT & TTS API로 가져오고 사용자의 위치를 Google MAP API로 언어와 출발지와 목적지를 설정한다.

2.2 소프트웨어



[Figure 8] 소프트웨어 구성도

본 프로젝트에서는 안드로이드 어플리케이션을 통해 API를 요청 및 받아오고, 사용자에게

경로 안내를 하며 Darkflow를 활용한 Object Detection을 통해 신호등 탐지 서비스를 제공한다. 본 절에서는 소프트웨어를 기술한다.

2.2.1 Android SDK

Android SDK를 이용하여 안드로이드 소프트웨어에서 제공하는 라이브러리들을 사용한다. 이는 View와 각종 Activity 및 Java 라이브러리를 포함한다.

2.2.2 Firebase 실시간 데이터베이스 SDK

안드로이드 어플리케이션에 필요한 데이터 저장하기 위한 데이터베이스로 사용된다. 신호등과 위험요소의 위치와 관련 음성 안내 텍스트를 저장하고, 사용자의 실시간 위치 및 경로 관련 정보를 저장하여 관리한다.

2.3 데이터베이스

2.3.1 데이터베이스 구조

| 신호등 관련 음성 안내 텍스트 TrafficLightText | | | | |
|-----------------------------------|-----|-------------|--------|--|
| 아이디 | id | primary key | int | |
| 메세지 | msg | NOT NULL | string | |

| 신호등 TrafficLight | | | | |
|------------------|-----------|-------------|--------|--|
| 아이디 | id | primary key | int | |
| 위도 | latitude | {33,43} | double | |
| 경도 | longitude | {124,132} | double | |
| 횡단보도 거리 | distance | {0-} | double | |
| 짝 | pair | unique | int | |

| 위험 요소 RiskFactor | | | | |
|------------------|-----------|-------------|--------|--|
| 아이디 | ID | primary key | int | |
| 위도 | latitude | {33,43} | double | |
| 경도 | longitude | {124,132} | double | |
| 조건 | option | 0/1 | bool | |

| 위험요소 접근 시 음성안내 텍스트 RiskFactorText | | | | |
|-----------------------------------|-----|-------------|--------|--|
| 아이디 | id | primary key | int | |
| 메세지 | msg | NOT NULL | string | |

[Figure 9] 데이터베이스 구조


2.3.2 데이터 베이스 테이블

| 신호등 TrafficLight | | | | |
|------------------|-----------|-------------|--------|--|
| 아이디 | id | primary key | int | |
| 위도 | latitude | {33,43} | double | |
| 경도 | longitude | {124,132} | double | |
| 횡단보도 거리 | distance | {0-} | double | |
| 짝 | pair | unique | int | |

[Figure 10] 신호등 정보 저장용 DB


인하대 후문 범위 내에 있는 신호등에 관한 정보를 저장한 데이터베이스이다. Primary

key인 신호등의 아이디(id)와 위도(latitude), 경도(longitude), 횡단보도 거리(distance), 신호등의 짝(pair)으로 구성되어 있다. GPS로 받은 사용자의 위치가 저장된 신호등의 일정범위 안에 포함되면 음성 안내가 출력된다. 짝을 지정하지 않았을 경우, 횡단을 했음에도 불구하고 또 다시 건너편 신호등 범위 안에 포함되어 안내가 중복되고 이는 경로 진행에 혼선을 유발한다. 이를 방지하기 위하여 신호등의 짝을 지정하였다.

| 잘못된 점자블록 | | WrongBrailleBlock | | |
|---|-----|-------------------|-------------|--------|
|  | 아이디 | id | primary key | int |
| | 위도 | latitude | {33,43} | double |
| | 경도 | longitude | {124,132} | double |


[Figure 11] 잘못된 점자블록 정보 저장용 DB

지정된 범위 내의 잘못 표기된 점자블록의 정보를 저장하는 데이터베이스이다. Primary key인 아이디(id), 점자블록의 위도(latitude)와 경도(longitude)로 구성되어 있다.

| 위험 요소 | | RiskFactor | | |
|---|-----|------------|-------------|--------|
|  | 아이디 | ID | primary key | int |
| | 위도 | latitude | {33,43} | double |
| | 경도 | longitude | {124,132} | double |
| | 조건 | option | 0/1 | bool |


[Figure 12] 위험 요소를 저장하는 DB

지정된 범위 내의 위험요소들의 정보를 저장하는 데이터베이스이다. 요소들의 위도(latitude), 경도(longitude), 조건(option) 그리고 primary key인 아이디(id)로 구성되어 있다. 여기서 조건은 경로 탐색시 무조건 배제 여부이다. 공사현장과 같이 경로상에 무조건적으로 포함하지 않을 곳은 0으로 지정하고 도착 지점이 포함될 수 있는 위험요소 같은 경우에는 1로 지정하였다.

| 신호등 관련 음성 안내 텍스트 | | TrafficLightText | | |
|---|-----|------------------|-------------|--------|
|  | 아이디 | id | primary key | int |
| | 메세지 | msg | NOT NULL | string |

[Figure 13] 신호등 관련된 텍스트를 저장하는 DB

신호등 탐지를 위해 진행과정 순서에 따라 사용자에게 안내할 텍스트를 저장한 데이터베이스 이다. 텍스트의 아이디(id), 메시지(msg) 그리고 primary key인 아이디(id)로 구성되어 있다.

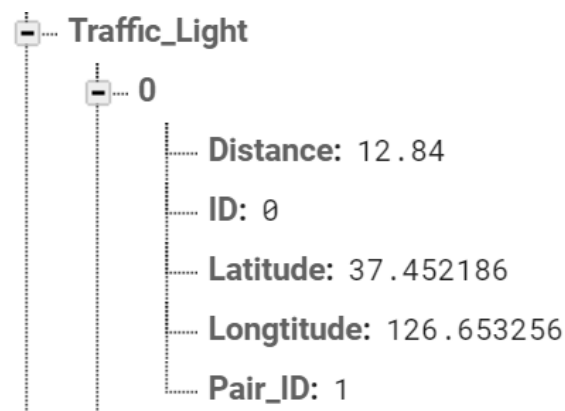
| 위험요소 접근 시 음성안내 텍스트 RiskFactorText | | | | |
|---|-----|-----|-------------|--------|
|  | 아이디 | id | primary key | int |
| | 메세지 | msg | NOT NULL | string |

[Figure 14] 위험 요소와 관련된 텍스트를 저장하는 DB

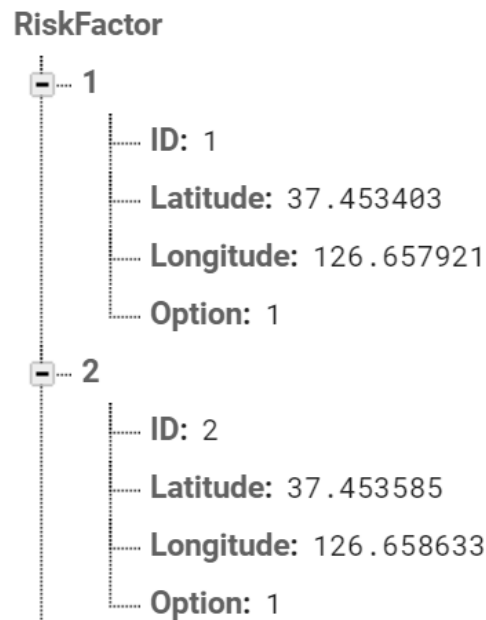
위험요소에 접근 시 음성으로 안내될 텍스트가 저장되어있는 데이터베이스이다. Primary key인 아이디(id)와 출력될 메시지(msg)로 구성되어있다.

2.3.3 Firebase 구조

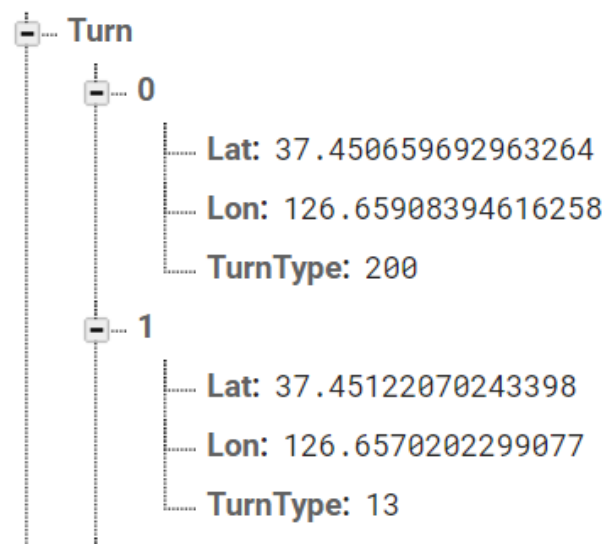
생성한 데이터베이스 테이블을 기반으로 Firebase에 데이터베이스를 추가하였다.



[Figure 15] Firebase Traffic_Light 데이터베이스



[Figure 16] RiskFactor 데이터베이스



[Figure 17] TurnType 데이터베이스

3. 프로젝트 개발 내용

3.1 개발 환경

| 구분 | | 항목 | 적용 내용 |
|----|------|----------------|-------------|
| | 개발도구 | Android Studio | Android IDE |
| | | JAVA | 애플리케이션 구현 |

| | | | |
|-----|-----------|----------------|------------------------------|
| S/W | 개발언어 | Python | 딥러닝 모델 구현 |
| | 라이브러리 | Darkflow | Annotation 데이터 이용하여 모델을 훈련 |
| | | Labellmg | 이미지 상의 객체의 위치와 클래스를 지정 |
| | 데이터베이스/서버 | Firebase | 위험지역, 신호등, 사용자 위치 및 경로 정보 저장 |
| | API | Google Map API | GPS 및 지도 제공 |
| | | T-MAP API | POI 정보 및 경로 제공 |
| | | STT/TTS API | 음성 인식 및 음성 출력 제공 |

[표 5] 개발 환경

3.2 문제 및 해결 방안

3.2.1 차량용 신호등과 보행자용 신호등 구분

| | |
|----|--|
| 문제 | 차량용 신호등과 보행자용 신호등이 카메라 화면에 같이 있을 경우, 보행자용 신호등만 인식하는가? |
| 해결 | 기존의 데이터셋을 사용하지 않고, 보행자용 신호등 이미지로만 이루어진 데이터셋을 새로 구축하여 이를 기반으로 모델을 훈련시킴으로써 해결했다. |

[표 6] 문제 및 해결 1

3.2.2 음성 인식을

| | |
|----|--|
| 문제 | 시끄러운 상황에서도 음성 인식이 잘 되는가? |
| 해결 | <p>일상 생활에서 발생하는 소음은 0dB ~ 60dB 이다. 사용자가 시끄러운 환경 속에서 음성 검색을 할 경우를 대비해 높은 데시벨의 상황에서 음성 인식을 해보았다.</p> <ul style="list-style-type: none"> - 오락실 : 최대 95dB - 카페 : 최대 79dB - 차도 옆 인도 : 최대 86dB <p>위와 같은 상황에서도 음성 인식이 잘 되는 것을 확인할 수 있었다.</p> |

[표 6] 문제 및 해결 2

3.2.3 보행자 기준 횡단보도 위치 구분

| | |
|----|---|
| 문제 | 사용자가 보행하는 방향에서 횡단보도 안내가 어느 방향인지 어떻게 판단하는가? |
| 해결 | <p>목적지까지의 경로를 탐색하는 과정에서 경로 내의 TurnType들을 저장한다. 그 중 TurnType 중 212는 좌측, 213은 우측에 횡단보도가 있음을 의미하는 것을 이용해 안내를 출력함으로 해결했다.</p> |

[표 7] 문제 및 해결 3

3.3 시험 시나리오

3.1.1 경로 상 위험지역이 포함된 경우

- 시나리오 : 명인짬닭을 출발지로 설정하고 미추홀구 노인 문화센터를 도착지로 지정한다.
- 예측 : 경로 내 포함된 2개의 위험지역에 접근했을 경우 경고 음성을 출력한다.
- 결과 : 위험지역으로 설정한 구역으로 진입할 시, 지정해놓은 경고 음성이 제대로 출력함을 확인했다.


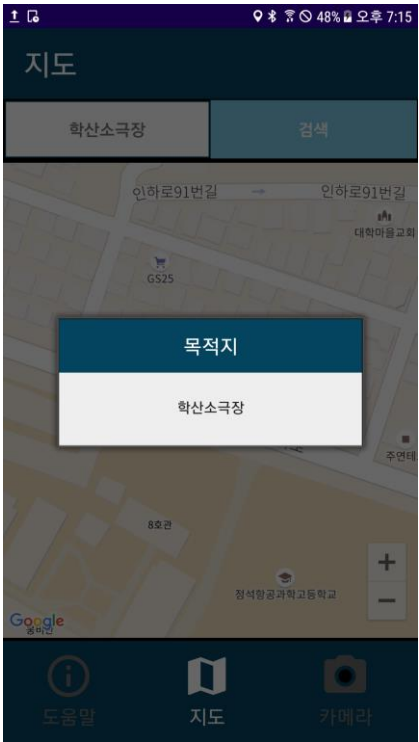
3.1.2 경로 상 신호등이 포함된 경우


- 시나리오 : 999마트를 출발지로 학산소극장을 도착지로 지정한다.
- 예측 : 경로 안내 시작 음성과 경로 내에 한 번의 우측에 위치한 신호등과 한 번의 좌회전, 도착 안내 음성을 순서대로 출력한다. 또한, 신호등을 디텍션하여 올바른 정보를 안내한다.
- 결과 : 경로 안내 시작 음성이 출력된 후, 우회전 알림 음성과 신호등 인식 음성이 순서대로 출력되었고, 좌회전 그리고 도착 알림 음성이 예상대로 출력되는 것을 확인했다.







3.4 Prototype 구현

본 프로젝트의 타겟 사용자는 시각장애인이므로 디자인보다는 실용성에 중점을 두고 단순한 구조로 화면을 설계하였다.

| 제목 | 이미지 | 기능 |
|-------|--|---------------------------------|
| 초기 화면 |  <p>The image shows the initial screen of the app. At the top, there is a status bar with a signal icon, 50% battery, and the time 7:02 PM. Below the status bar, the text 'FOR THE BLIND' is displayed. In the center, there is an illustration of a person walking a dog. Below the illustration, the text 'FORTHEBLIND' and '안전하게 길을 걷다' (Walk safely) are shown.</p> | 애플리케이션 실행 시 대기화면이다. |
| 팝업창 |  <p>The image shows a screenshot of the app with a map in the background. A tutorial pop-up window is displayed in the center. The pop-up window has a title '튜토리얼' (Tutorial) and contains 7 numbered steps. At the bottom of the screen, there are three icons: a person (labeled '도움말' - Help), a map (labeled '지도' - Map), and a camera (labeled '카메라' - Camera).</p> | 실행 시 첫 화면으로, 간단한 조작법에 대해 설명해준다. |

| | | |
|--------------|--|---|
| <p>지도</p> |  | <p>메인 화면으로 사용자가 목적지를 입력하면 경로를 표시하는 화면이다. 사용자의 위치를 GPS기반으로 실시간으로 보여준다.</p> |
| <p>음성 검색</p> |  | <p>사용자가 말한 목적지를 재확인 한 후 맞으면 경로 검색을 시작한다. 재확인 한 결과가 틀리면 사용자의 음성을 다시 입력받는다.</p> |

| | | |
|------------|--|--|
| <p>카메라</p> |  | <p>신호등 인식을 위한 카메라 화면이다.</p> |
| <p>도움말</p> |  | <p>도움말 화면이다. 튜토리얼과 기능안내 화면으로 갈 수 있다.</p> |

| | | |
|--------------|--|--|
| <p>튜토리얼</p> | <div data-bbox="336 208 746 954"> <h3>어플리케이션 조작법</h3> <p>1) 화면은 하단의 탭을 터치해서 전환할 수 있습니다.</p> <p>2) 하단 탭은 왼쪽부터 도움말, 지도, 카메라 순으로 위치해 있습니다.</p> <p>3) 한번 터치 시, 음성으로 설명이 출력됩니다. 두번 터치 시, 해당 요소가 선택이 됩니다.</p> <p>4) 초기 화면은 지도입니다. 안전한 경로를 안내하기 위한 화면입니다.</p> <p>5) 카메라 화면에서는 신호등의 색을 실시간으로 안내해줍니다.</p> <p>6) 도움말 화면에서는 해당 앱 사용 방법에 대해 자세히 알 수 있습니다.</p> <p>7) 팝업창을 닫기 위해서는 화면을 두 번 눌러주세요.</p> <div>  도움말  지도  카메라 </div> </div> | <p>튜토리얼 화면이다.</p> <p>팝업창과 동일하게 어플리케이션의 조작법을 간단히 설명해준다.</p> |
| <p>기능 안내</p> | <div data-bbox="336 999 746 1749"> <h3>기능 안내</h3> <div>지도</div> <div>카메라</div> <div>  도움말  지도  카메라 </div> </div> | <p>어플리케이션의 기능을 자세히 설명해준다. 선택 시 지도 혹은 카메라 설명 화면으로 넘어간다.</p> |

| | | |
|---------------|---|-------------------------------------|
| <p>지도 안내</p> | <div> <div>지도 안내</div> <div> <p>1) 안전한 경로를 안내해드립니다.</p> <p>2) 경로를 입력하기 위해서는 화면 상단에 있는 음성안내칸을 두 번 터치한 후 도착지점을 또박또박 천천히 말씀해주시기 바랍니다.</p> <p>3) 말씀하신 도착지가 맞는지 확인 음성이 나옵니다. 맞다면 두 번 터치해주시고, 재입력을 원하시면 아래에서 위로 화면을 쓸어 올려주세요.</p> <p>4) 안내 중 계단이나 잘못된 점자블록 등이 발견되면 경고 알림이 나오니 주의하시기 바랍니다.</p> <p>5) 신호등 위치에 근접하면 안내 음성이 나옵니다. 횡단보도를 건너다면 안내를 따라주시기 바랍니다.</p> </div> <div> <div>도움말</div> <div>지도</div> <div>카메라</div> </div> </div> | <p>지도에 대한 기능을 자세하게 설명해주는 화면이다.</p> |
| <p>카메라 안내</p> | <div> <div>카메라 안내</div> <div> <p>1) 신호등의 색과 남은 시간, 횡단 보도의 거리를 안내해드립니다.</p> <p>2) 신호등이 인식되면, 인식 완료 안내 후 정보가 안내됩니다.</p> <p>3) 시간이 안내된 후부터 카운트가 시작됩니다.</p> <p>4) 안내된 시간에 1~2초 정도의 오차가 있을 수 있으니 주의하시기 바랍니다.</p> </div> <div> <div>도움말</div> <div>지도</div> <div>카메라</div> </div> </div> | <p>카메라에 대한 기능을 자세하게 설명해주는 화면이다.</p> |

[표 8] ProtoType

3.5 상세 설계

3.5.1 APP 내 기능

본 어플리케이션에서는 시각장애인의 편의를 위해 한번 터치하면 해당 컴포넌트 내용이 음성으로 출력되고, 두번 터치하면 컴포넌트가 선택된다. 이 기능이 어플리케이션 전체에 적용되도록 구현하였다.

3.5.1.1 사용 방법 팝업

화면 구성을 파악하는 데에 어려움이 있는 시각장애인을 위하여 어플리케이션 시작 시 팝업창을 띄워 사용 방법을 안내한다. 한 번 터치시 안내사항이 음성으로 출력되고, 두 번 터치 시 팝업창이 닫히도록 구현하였다.

3.5.1.2 목적지 검색

목적지를 타이핑하여 검색하는 것이 어려운 시각장애인을 위해 목적지를 음성으로 검색하는 기능을 구현하였다. 음성으로 입력한 값을 텍스트로 변환하여 이를 다시 음성 변환을 통해 사용자에게 목적지가 맞는지 확인시킨다. 사용자가 음성으로 입력한 출력값과 원하는 목적지의 일치 여부에 따라 실행 결과가 달라진다.

- 일치할 경우 : 백 버튼을 터치하여 사용자 음성의 변환 결과 값이 목적지로 지정된다.
- 불일치할 경우 : 화면 두번 터치를 하여 이전화면으로 돌아간 뒤 목적지를 재검색한다.

3.5.1.3 안전한 경로 탐색

목적지가 입력되면 Google MAP API와 T MAP API를 이용하여 안전한 경로를 사용자에게 안내한다. 지정된 경로 상에 위험요소가 있는 경우, 사용자가 DB에 저장된 위험요소의 위치에 근접하면 경고 음성이 출력된다.

3.5.1.4 실시간 신호등 탐지

지정된 경로 상에 횡단하는 경우가 있을 경우, 사용자가 횡단하기 위해 DB에 저장된 신호등 위치에 근접하면 카메라로 전면을 비추라는 음성 안내가 출력된다. 카메라에 입력된 영상에 신호등의 정보를 검출하여 사용자에게 음성으로 안내함으로써 안전한 횡단을 돕는다.

3.5.2 Object Detection

Object Detection이란 이미지에서 특정 Object의 위치를 감지하여 해당 영역 안의 Object를 분류하는 작업이다.

3.5.2.1 darkflow 클론

<https://github.com/thtrieu/darkflow> 에서 darkflow repository를 클론한다.

3.5.2.2 동영상 촬영

구글에서 이미지를 크롤링 하는 것 대신 본 어플리케이션의 적용 범위가 ‘인하대학교 후문’ 이라는 것을 고려하여 직접 후문가의 신호등을 영상으로 촬영함으로써 훈련 데이터를 수집한다.

3.5.2.3 이미지 추출

촬영한 영상에서 이미지 파일을 추출함으로써 훈련을 위한 데이터 셋을 구축하였다.

```
kimnan-young@gimnan-yeong-ui-MacBook-Pro ~/NanyoungFiles
ffmpeg -i [입력동영상] -an -r [초당 출력할 프레임 수] -y -s [출력 해상도] [출력 이미지 파일 이름]%.jpg
```

[Figure 18] 이미지 추출 방법

```
kimnan-young@gimnan-yeong-ui-MacBook-Pro ~/NanyoungFiles ffmpeg
mpeg -i TLvideo_0.mp4 -an -r 5 -y -s 1080x1920 training_%.jpg
```

[Figure 19] 이미지 추출 예시

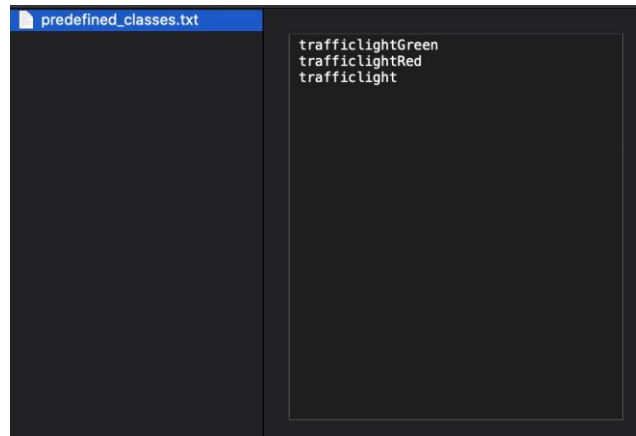
영상 스트림 프로그램인 ffmpeg을 활용하여 위와 같이 이미지를 추출하였으며, 아래의 명령어를 실행하면 영상이 저장된 경로에 이미지가 추출되어 저장된다.

| | |
|------------|--------------|
| -an | 오디오 무시 |
| -r 5 | 초당 5 프레임 추출 |
| -s | 결과 이미지 크기 지정 |
| -1080x1920 | 결과 이미지 크기 |

[표 9] 이미지 추출 명령어 요소 설명

3.5.2.4 predefined_classes.txt 수정

신호등 검출을 위해 총 세개의 클래스 라벨을 지정하였다.



[Figure 20] 수정된 predefined_classes.txt

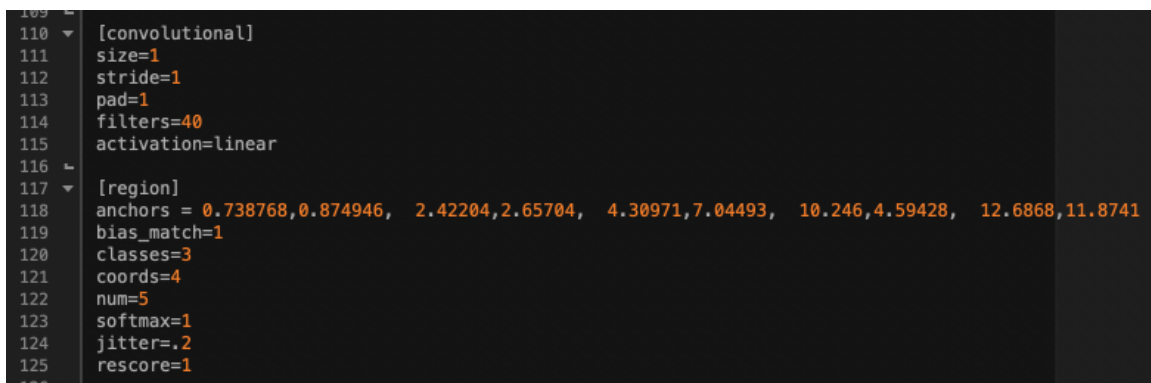
labelImg/data/predefined_classes.txt 을 위와 같이 수정하였다. 'trafficlightGreen' 은 보행자 신호가 초록불인 상태의 이미지를 의미하고, 'trafficlightRed'는 보행자 신호가 빨간불인 상태의 이미지를 의미하며, 'trafficlight'는 신호에 불이 들어오지 않은 상태의 이미지를 의미한다.

3.5.2.5 label.txt 수정

darkflow/label.txt 도 [Figure 20]와 동일하게 수정한다.

3.5.2.6 cfg 파일 수정

darkflow/cfg 디렉토리 내의 cfg 중 사용할 모델을 수정한다.



[Figure 21] 수정된 my-tiny-yolo.cfg 파일

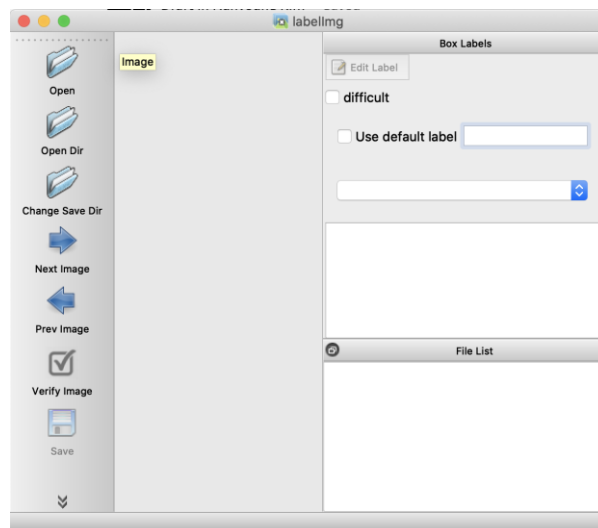
모바일 애플리케이션에 사용될 모델이기 때문에 tiny-yolo.cfg 를 이용하였고

해당 파일을 my-tiny-yolo.cfg 로 파일명을 바꾼 뒤 위와 같이 Line 114,120을 수정하였다.

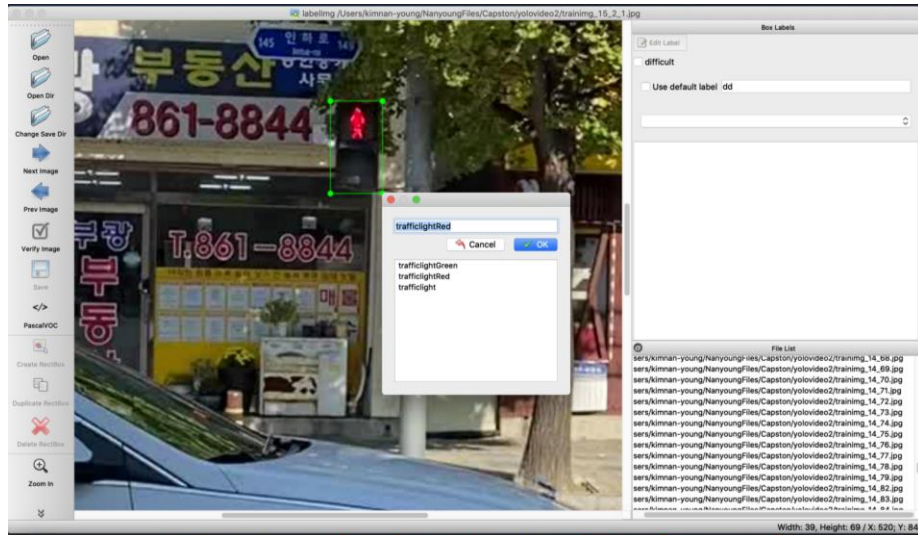
Line114의 Filters와 Line 120의 classes를 수정해준다. Supervised Learning 을 위한 총 라벨 수가 3개('trafficlightGreen', 'trafficlightRed', 'trafficlight') 이므로 'classes = 3'으로 지정한다. filters = (classes수 + 5) * 5 로 지정해줘야 하며 본 어플리케이션에서는 $(3 + 5) * 5 = 40$ 으로 지정하였다.

3.5.2.7 이미지 라벨링

YOLO 모델은 Supervised Learning 이므로 정답인 클래스 라벨을 지정한 뒤 학습시켜야한다. Object Detection의 정답 라벨은 각 객체의 Class Label과 Boundary-box의 쌍으로 구성되며 이를 Annotation이라 부른다. LabelImg를 이용해 트레이닝 이미지의 라벨을 지정함으로써 추출한 이미지에 Annotation을 할당하였다.



[Figure 22] labemImg 실행 화면



[Figure 23] labelImg 로 라벨 지정하기

이미지를 불러와서 신호등에 라벨을 지정한 상태이다.

| | |
|---|----------------------|
| w | Create rectangle box |
| d | Next image |
| a | Previous image |

[표 10] labelImg 단축키

3.5.2.8 모델 학습

```
kimnan-young@gimnan-yeong-ui-MacBook-Pro ~/NanyoungFiles python flow --model
./cfg/my-tiny-yolo.cfg --labels ./labels.txt --trainer adam --dataset ../yolovideo2/
--annotation ../yolovideo2 --train --summary ./logs --batch 3 --epoch 100 --save
50 --keep 5 --lr 1e-04
```

[Figure 24] 모델 학습을 위한 명령어

위의 명령어를 터미널에 입력하여 라벨링 한 데이터로 모델 학습을 시작한다. 초반에는 learning rate를 크게 해서 loss 값을 빠르게 낮춘뒤, 어느 정도 모델 훈련이 진행되면 learning rate 값을 작게하여 정확도를 높일 수 있도록 한다.

| | |
|----------|---|
| -trainer | Optimizer 설정 |
| -lr | Learning rate (1e-04 : 0.0001) |
| -gpu | Gpu 사용 여부 |
| -load | 이전 학습 가중치 사용 여부. -1 설정시 마지막 save 를 불러오며 특정 step 으로 설정 시 해당 step 부터 학습 이어서 시작. |

[표 11] 모델 학습 명령어 요소

```

trafficlightGreen: 304
trafficlightRed: 566
trafficlight: 118
Dataset size: 985
Dataset of 985 instance(s)
Training statistics:
    Learning rate : 0.0001
    Batch size    : 3
    Epoch number  : 100
    Backup every   : 50

```

[Figure 25] 학습 환경

3.5.2.9 학습 결과

```

Checkpoint at step 30432
step 30433 - loss 0.005979351699352264 - moving ave loss 0.056953319301833255
step 30434 - loss 0.0017323391512036324 - moving ave loss 0.051431221286770294
step 30435 - loss 0.005510204937309027 - moving ave loss 0.04683911965182417
step 30436 - loss 0.016725579276680946 - moving ave loss 0.043827765614309844
step 30437 - loss 0.0015978054143488407 - moving ave loss 0.03960476959431374
step 30438 - loss 0.20282043516635895 - moving ave loss 0.05592633615151826
step 30439 - loss 0.08628208190202713 - moving ave loss 0.05896191072656915
step 30440 - loss 0.037256091833114624 - moving ave loss 0.056791328837223695
Finish 5 epoch(es)
step 30441 - loss 0.005983212497085333 - moving ave loss 0.051710517203209856
step 30442 - loss 0.01450507715344429 - moving ave loss 0.047989973198233296
step 30443 - loss 0.0399073027074337 - moving ave loss 0.04718170614915334
step 30444 - loss 0.014412350952625275 - moving ave loss 0.04390477062950053
step 30445 - loss 0.3335641622543335 - moving ave loss 0.07287070979198383
step 30446 - loss 0.007187402341514826 - moving ave loss 0.06630237904693694
step 30447 - loss 0.00589241087436676 - moving ave loss 0.060261382229679916
step 30448 - loss 0.015630649402737617 - moving ave loss 0.05579830894698569

```

[Figure 26] 학습 결과

30,000번 이상의 step을 훈련한 모델의 loss 값을 확인할 수 있다. 초반의 loss 값은 100초반 대였으나 점점 0에 가까워진 모습이다.

```

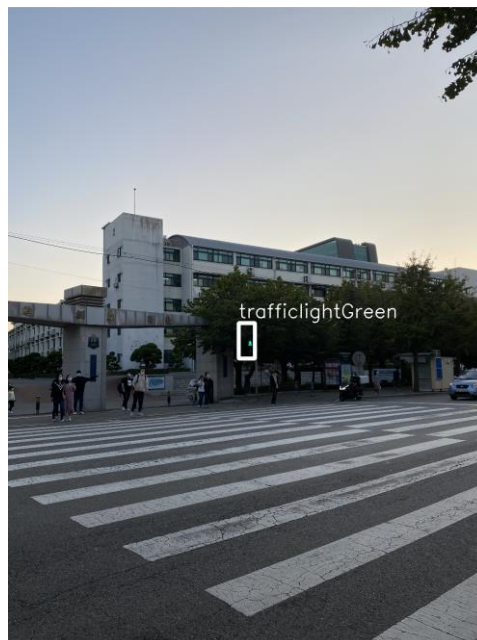
kimnan-young@gimnan-yeong-ui-MacBook-Pro ~/NanyoungFiles python flow --imgdir
../files/testset/folder --model ./cfg/my-tiny-yolo.cfg --load -1 --batch 1 --thre
shold 0.5

```

[Figure 27] 테스트 명령어

| | |
|-------------|---|
| --imgdir | 테스트할 이미지가 위치한 디렉토리 경로 |
| --model | Cfg 모델이 위치한 디렉토리 경로 |
| --threshold | Confidence 가 0.5보다 클 경우 boundary-box 를 수용한다는 의미 |

[표 12] 테스트 명령어 요소



[Figure 28] test data 를 넣은 결과

훈련한 모델에 훈련할 때 쓴 이미지가 아닌 테스트 이미지를 넣어 테스트 한 결과이다. 신호가 초록불인 상태를 올바르게 인식한 것을 확인할 수 있다.

3.5.3 Application

| | | |
|------------|--------------------------|--|
| Class Name | MapActivity | |
| 설명 | Map에 관련된 전반적인 기능을 하는 클래스 | |
| Method | void initArea() | 사전에 Firebase에 저장된 위험지역을 지오펜싱할 수 있도록 하는 메서드 |

| | | |
|--|------------------------------|---|
| | void addUserMarker() | 사용자의 현재 위치를 지도 상에 마커로 표시하는 메서드 |
| | void settingGeoFire() | geofire를 setting 하는 메서드 |
| | void buildLocationCallback() | 사용자의 현재 위치를 가져오고 저장하는 메서드 |
| | void buildLocationRequest() | 위치 정보를 요청하고 설정하는 메서드 |
| | void addCircleArea() | 지도 상에 위험지역 및 신호등, 회전 정보 등 지오펜싱을 할 지점을 표시하는 메서드 |
| | void FindPOI) | TMap API를 이용하여 destination과 일치하는 장소를 검색하고, 위치 정보를 FindPathData에 전달하여 경로를 찾을 수 있게 하는 메서드 |
| | void FindPathData() | FindPOI로 부터 전달받은 목적지까지의 경로를 탐색하고 경로를 지도상에 그려주는 메서드 |

[표 13] MapActivity

| | | |
|------------|---|--|
| Class Name | CalculateTL | |
| 설명 | 경로 상에 포함되는 신호등을 찾는 클래스 | |
| Method | int ReadDataSig(List<TrafficLightInfo> trafficLightInfo, TMapPoint point, int turntype) | firebase에 저장된 경로상의 횡단보도의 좌표를 가져오는 메서드 |
| | void CalculateMin(List<TrafficLightInfo> trafficLightInfo, TMapPoint point, List<MyLatLng> latLngList, int turntype) | ReadDataSig 메서드로 읽어온 좌표들과 신호등의 거리를 계산하여 경로 내에 위치한 신호등의 id를 리턴해주는 메서드 |

[표 14] CalculateTL

| | |
|------------|-------------------|
| Class Name | DetectorAcitivity |
|------------|-------------------|

| | | |
|--------|---|---|
| 설명 | 신호등 인식 모델을 적용하여 신호등을 찾고 정보를 음성 안내하는 클래스 | |
| Method | void processImg() | Confidence가 일정값 이상인 객체를 디텍션하여 해당 이미지를 처리하며, 디텍션 결과 클래스에 따라 신호등 색을 알려주는 음성을 출력하는 메서드 |
| | void onPreviewSizeChosen(final Size size, final int rotation) | MODE에 따라 객체 인식을 위한 변수를 초기화한다. 모델 훈련에 사용되는 이미지와 같은 크기로 input image를 crop하는 메서드 |

[표 15] DetectorActivity

| | | |
|------------|--|-----------------------|
| Class Name | FBLatLon | |
| 설명 | 경로 상의 turnPointer의 위도, 경도 및 turnType을 Firebase에 저장하는 클래스 | |
| Method | void writeNewLatLon(PathItem pathItem, int idx) | point의 Type을 저장하는 메서드 |
| | void RemoveAll() | 저장된 정보들을 모두 지우는 메서드 |

[표 16] FBLatLon

| | | |
|------------|---------------------------|--|
| Class Name | HelpActivity | |
| 설명 | 애플리케이션 도움말을 안내하는 초기화면 클래스 | |

[표 17] HelpActivity

| | | |
|------------|---------------------------|--|
| Class Name | FuncActivity | |
| 설명 | 지도 및 카메라 사용법을 안내하는 초기 클래스 | |

[표 18] FuncActivity

| | | |
|------------|--------------------|--|
| Class Name | CameraInfoActivity | |
|------------|--------------------|--|

| | |
|----|-------------------|
| 설명 | 카메라 사용법을 안내하는 클래스 |
|----|-------------------|

[표 19] CameraInfoActivity

| | |
|------------|------------------|
| Class Name | MapInfoActivity |
| 설명 | 지도 사용법을 안내하는 클래스 |

[표 20] MapInfoActivity

| | |
|------------|----------------------|
| Class Name | TutorialAcitivity |
| 설명 | 애플리케이션 사용법을 안내하는 클래스 |

[표 21] TutorialActivity

| | | |
|------------|--|---|
| Class Name | InfoGeofence | |
| 설명 | 경로 상의 좌/우회전, 신호등 등의 정보를 geofencing하기 위한 작업을 하는 클래스 | |
| Method | void initInfoArea(final List<TrafficLightInfo> trafficLightInfo) | 경로 상의 좌/우회전, 신호등을 geofencing 하는 환경을 만드는 메서드 |

[표 22] InfoGeofence

| | | |
|------------|---|-----------------------------|
| Class Name | MyLatLng | |
| 설명 | Firebase에 저장되는 위험요소, 신호등, 회전정보 등의 위/경도, id와 같은 정보를 용이하게 사용하기 위한 클래스 | |
| Method | int getId() | Firebase에 저장되는 요소 id의 get함수 |
| | void setId(int ID) | Firebase에 저장되는 요소 id의 set함수 |
| | double getLatitude() | Firebase에 저장되는 요소 위도의 get함수 |
| | void setLatitude(double latitude) | Firebase에 저장되는 요소 위도의 |

| | | |
|--|-------------------------------------|-------------------------------------|
| | | set함수 |
| | double getLongitude() | Firestore에 저장되는 요소 경도의 get함수 |
| | void setLongitude(double longitude) | Firestore에 저장되는 요소 경도의 set함수 |
| | int getPair_ID() | Firestore에 저장되는 신호등의 짝신호등 id get함수 |
| | void setPair_ID() | Firestore에 저장되는 신호등의 짝신호등 id의 set함수 |
| | double getDistance() | Firestore에 저장되는 횡단보도 거리의 get함수 |
| | void setDistance(double distance) | Firestore에 저장되는 횡단보도 거리의 set함수 |

[표 23] MyLatLng

| | | |
|------------|------------------------------------|---------------------------------------|
| Class Name | PathItem | |
| 설명 | 경로 상의 회전정보들의 정보를 저장하는 클래스 | |
| Method | boolean isNodeType() | Firestore에 저장되는 회전정보의 NodeType의 get함수 |
| | void setNodeType(boolean nodeType) | Firestore에 저장되는 회전정보의 NodeType의 set함수 |
| | int getTurnType() | Firestore에 저장되는 회전정보의 TurnType의 get함수 |
| | void setTurnType(int turnType) | Firestore에 저장되는 회전정보의 TurnType의 set함수 |
| | TMapPoin gettMapPoint() | Firestore에 저장되는 회전정보의 위치의 get함수 |
| | void settMapPoint(TMapPoint point) | Firestore에 저장되는 회전정보의 위치의 set함수 |
| | String toString() | 저장된 회전 정보들을 출력하는 메서드 |

[표 24] PathItem

| | |
|------------|---------------------------------|
| Class Name | PopupActivity |
| 설명 | 애플리케이션 실행 시 튜토리얼을 팝업창으로 띄우는 클래스 |

[표 25] PopupActivity

| | |
|------------|---|
| Class Name | PopupCheckDestActivity |
| 설명 | 사용자가 음성으로 입력한 목적지 정보가 맞는지 확인해주는 클래스 |
| Method | void TTS() TextToSpeech를 사용하기 위한 메서드 |

[표 26] PopupCheckDestActivity

| | |
|------------|--------------------------------|
| Class Name | SearchActivity |
| 설명 | 사용자가 목적지를 음성으로 검색할 수 있게 하는 클래스 |

[표 27] SearchActivity

| | | |
|------------|--------------------------------|-------------------------------|
| Class Name | TrafficLightInfo | |
| 설명 | 신호등의 정보들을 저장하는 클래스 | |
| Method | int getId() | Firebase에 저장된 신호등의 id의 get 함수 |
| | void setId(int ID) | Firebase에 저장된 신호등의 id의 set 함수 |
| | TMapPoint getPoint() | TMapPoint의 get함수 |
| | void setPoint(TMapPoint point) | TMapPoint의 set함수 |
| | int getTurnType() | turnType의 get함수 |
| | void setTurnType(int turnType) | turnTyep의 set함수 |

[표 28] TrafficLightInfo

| | | |
|------------|---|---|
| Class Name | TTSSTT | |
| 설명 | 음성을 텍스트로, 텍스트를 음성으로 변환할 수 있게 하는 클래스 | |
| Method | void TimeToClick(String text, Class mClass) | 한 번 터치 시, text 출력, 두 번 터치 시 mClass로 이동하는 메서드 |
| | void ClickFind(String text) | 한 번 터치 시, text 출력, 두 번 터치 시 FindPOI 메서드를 호출하는 메서드 |

[표 29] TTSSTT

3.6 시험/테스트 결과

3.6.1 1차 테스트

| | |
|--------|---|
| 날짜 | 2020년 11월 26일 목요일 |
| 테스트 내용 | 목적지 입력 후 경로 제공과 음성 안내 |
| 결과 | 목적지 및 경로 탐색은 정상적으로 작동하였으나, 보행 중 안내는 출력되지 않는 오류 발견 |

[표 30] 1차 테스트

3.6.2 2차 테스트

| | |
|--------|--|
| 날짜 | 2020년 12월 04일 금요일 |
| 테스트 내용 | 목적지 입력 후 경로 제공과 음성 안내 |
| 결과 | <ol style="list-style-type: none"> 1. 경로 상에 신호등이 포함되지 않는 경우, 지오펜싱이 작동하지 않는 오류 발견 2. 목적지 입력을 한 후 해당 목적지를 원하지 않을 경우에도 잘못 입력한 목적지 값이 그대로 들어가는 오류 발견 |

[표 31] 2차 테스트

3.6.3 최종 테스트

| | |
|--------|--|
| 날짜 | 2020년 12월 9일 금요일 |
| 테스트 내용 | 전반적인 애플리케이션 실행에 오류가 없는지 확인한다. |
| 결과 | 최종 테스트 결과, 실행, 음성 인식, 목적지 및 경로 탐색, 경로 안내 그리고 위험 지역 안내가 제대로 출력됨을 확인 |

[표 32] 최종 테스트

3.7 DEMO

시나리오를 기반으로 직접 시연한 것을 촬영한 영상이다. 아래 링크에 게시되어 있다.

<https://www.youtube.com/watch?v=c3CmbfIzA3Q&feature=youtu.be>

4. 프로젝트 활용 및 향후 계획

4.1 프로젝트 활용

인하대 후문 내에서 위험요소와 신호등 접근 시 음성 안내, 신호등을 탐지 후 정보 제공 및 안전한 경로 안내를 함으로써 시각장애인이 안전하게 길을 보행할 수 있도록 돕는데 활용할 수 있다.

4.2 프로젝트 향후 계획

추후 개발 인원이 충원된다면, 서비스 가능 지역을 늘릴 것이다. 또한, 시각장애인 이외의 사용자들의 참여를 유도하여 게시판과 같이 상황을 공유할 수 있는 플랫폼을 만든다. 이를 활용하면 인도의 위험요소 및 신호등 위치 등의 상황을 직접 조사하지 않고도 파악할 수 있을 것이다.

5. 일정 및 역할 분담

5.1 팀 역할 분담

| 팀원 이름 | 역할 |
|-------|---|
| 김난영 | 인하대학교 주변 신호등 이미지를 직접 수집하여 데이터 셋을 구축한다. Darkflow 기반으로 신호등을 디텍션하여 색을 실시간으로 파악하는 모델을 구현한다. 어플리케이션 시작 시 나타나는 Splash 화면 및 UI 디자인을 구현한다. |
| 김혜윤 | 전반적인 Frontend 부분을 담당하여 Activity 전환과 Activity간 데이터 전달 기능을 구현한다. MAP API를 활용하여 안전한 경로를 찾고 경로 내 지점들에 대한 정보를 Firebase에 저장한다. 저장한 정보와 실시간으로 변하는 사용자의 위치를 비교하여 우회전, 좌회전과 같은 상황을 처리한다. |
| 이영주 | STT API를 활용하여 사용자가 음성으로 입력한 데이터를 텍스트로 변환하여 활용할 수 있도록 한다. TTS API를 활용하여 어플리케이션 내 컴포넌트를 설명해주는 음성을 출력하며, 사용자가 위험지역 및 신호등에 접근 시 경고 음성을 출력하고, |

| | |
|--|--|
| | 우회전, 좌회전과 같은 상황을 안내한다. Geo-fencing과 Firebase를 활용하여 물체 범위를 지정하고 사용자가 일정 범위에 들어가는지 여부를 판단한다. |
|--|--|

[표 33] 팀원 간 역할

5.2 개발 일정

| 구분 | 세부 구분 | 추진 내용 | 추진일정(월) | | | |
|------|------------|-------------------------------------|---------|----|----|----|
| | | | 9 | 10 | 11 | 12 |
| 설계 | 개발 계획 & 환경 | 주변 탐색, 시스템 구조 설계 및 구성도 작성 | | | | |
| 개발 | 앱 UX/UI 구현 | 제스처 인식 기능 및 어플 디자인 | | | | |
| | DB 구현 | DB 생성 및 데이터 저장 | | | | |
| | API 적용 | MAP API와 SST & TTS API 적용 및 기능 구현 | | | | |
| | YOLO 모델 적용 | Darkflow 이용하여 Object Detection | | | | |
| 테스트 | 통합 테스트 | 시나리오 기반으로 개발자의 의도 및 프로토 타입 일치 여부 확인 | | | | |
| | 디바이스 테스트 | 길거리에서 시뮬레이션으로 기능 작동 확인 | | | | |
| 유지보수 | 최적화 & 디버깅 | 오류 여부 지속적인 확인 & 최적화 시도 | | | | |
| 종료 | 발표 | 최종 발표 | | | | |

[표 34] 개발 일정