

Yolov5를 이용하여 도로 위 위험요소를 인식하는 경로 안내 애플리케이션 개발

Route Guidance Application Development for Detecting Obstacle on the Road Using YOLOv5 model

서민영 (Minyoung Seo), 박가은 (Gaeun Park), 안예림 (Yelim An)

요약

경로를 검색하여 결과를 제공하는 플랫폼이 만연한 현대사회에, 이동제약자를 위한 길찾기 서비스는 찾기 어렵다. 이동제약자에게 미리 도로의 정보를 알려주어 이동성과 안전성을 높여주는 Flatroad 서비스를 개발하였다. Flatroad 서비스는 사용자가 앱으로 출발지와 목적지를 입력하면 목적지까지의 경로와 해당 경로에서의 위험요소를 알려준다. 이 서비스는 앱과 서버로 기능이 나누어져 있다. 앱은 사용자에게 정보를 입력받아 경로를 찾고 서버에서는 해당 경로에서의 위험요소를 찾는다. YOLO v5를 사용하여 볼라드를 탐색하기 위한 객체 탐색 모델을 구축하였고 이를 서버에서 사용한다. 로드뷰를 얻기 위한 Kakao Maps API, 경로를 얻기위한 Mapbox API, 스크린 샷을 위한 Grabzit API를 사용하였다. Flatroad 애플리케이션은 이동제약자들을 위해 위험요소를 지도에 표시해준다는 점에서, 일반적인 길찾기 애플리케이션과는 차별점을 두었다. 이는 이동제약자의 이동권 보장에 기여했다고 할 수 있다. 추후 위험요소 종류 증가와 로드뷰 이미지와 객체탐색모델의 정확도 개선을 통해 업데이트를 지속할 예정이다.

키워드: YOLOv5, Yolo Mark, Bollard, Mapbox, Kakao

Abstract

In modern society, where platforms that search for routes and provide results are prevalent, directions for movement constraints are difficult to find. Flatroad has been developed to improve mobility and safety by informing movement constraints of walking information in advance. The Flatroad service informs users of the route to and from the destination when they enter their origin and destination through the app. The service is divided into apps and servers. The app receives information from the user to find the path, and the server looks for obstacles on the road. We built an object detecting model for exploring bollards using YOLO v5 and used it on servers. It used the Kakao Maps API for obtaining roadviews, the Mapbox API for obtaining paths, and the Grabzit API for screenshots. Flatroad applications differentiate themselves from ordinary directions applications in that they map obstacles on the road for movement constraints. This can be said to have contributed to the guarantee of mobility rights for the mobility constrictor. It is planning to continue updating by increasing obstacles and improving accuracy of roadview images and object detection models in the future.

1.서론

도로에는 다양한 장애물들이 많다. 도로의 턱, 신호등, 가로수, 볼라드 등 도로를 구성하는 것뿐만 아니라 주차된 차들, 자전거 등 예상할 수 없는 장애물들도 있다. 신체가 건강한 사람이 일반적인 도로를 걸어가는 것은 전혀 어렵지 않다. 하지만 휠체어, 목발을 이용하거나 거동이 힘든 사람이라면 도로에 있는 작은 장애물이라도 이동하는 데 불편함을 줄 수 있다. 물론, 이동제약자를 위해 시설물을 교체하거나 저장버스를 운영하는 등 다양한 노력을 실행하고 있지만 실제로 그들의 불편함은 아직 나아지지 못하고 있다.

사람이 거리를 안전하게 돌아다니고 가고 싶은 곳을 가는 것은 일반적으로 당연하게 여겨지지만 이동제약자에게는 힘들고 불안한 일일 수 있다. 이동약자들은 도로의 기본적인 구조에도 영향을 받고 돌발사항이 벌어졌을 때 대처하기가 어렵고

위험도 커진다. 그래서 우리는 그들이 자신의 이동권을 자유롭게 안전하게 누릴 수 있도록 도와줄 수 있는 서비스를 제작하고자 한다. 만약 이동제약자들이 사전에 자신의 경로를 파악하고 도로에 있는 위험요소에 대한 정보를 얻어 대비할 수 있다면 더 안전하게 길을 나설 수 있을 것이다. 본 프로젝트는 사용자가 Flatroad 어플리케이션을 이용하여 기본적인 도로 경로 안내를 받고 해당 경로에서 위험요소를 볼 수 있는 서비스를 기획하였다. 사용자는 이 서비스를 통해 이동의 효율성과 안전성을 얻을 수 있다. Flatroad는 사용자가 이용하는 어플리케이션과 해당 경로에서 위험요소를 찾아주는 서버로 구성된다. 서버에서는 인공지능 모델을 사용하여 도로 이미지에서 위험요소를 찾아낸다. 본 논문에서는 Flatroad 서비스의 사용방법과 구조, 핵심모듈에 대하여 설명할 것이다.

2. 관련연구

2.1 Mapbox

Map box는 웹사이트 및 애플리케이션을 위한 맞춤형 온라인 지도를 제공하는 미국업체이다. 구글이 카메라와 레이더가 장착된 전용차량과 위성 데이터를 기반해 지도 데이터를 정밀하게 하는 것과는 다르게, Map box는 이용자들이 직접 입력하는 지도 정보와 GPS 데이터를 기반으로 지도를 업데이트 한다. 이런 장점으로 현재 Microsoft, Facebook 등 글로벌 기업들에게 디지털 지도 서비스를 제공한다. 데이터는 OpenStreetMap 및 NASA와 같은 오픈 데이터 소스 및 DigitalGlobe와 같은 독점 데이터 소스에서 획득한다.

본 프로젝트에서는 도로경로 길 찾기 결과를 제공하는 Direction API, 화면에 지도를 띄우기 위한 Maps SDK for Android, 주소를 좌표로 변환하기 위한 Geocoding API를 이용한다. [1]

2.2 카카오 로드뷰와 방위각

로드뷰는 도로를 360도 파노라마 이미지로 촬영한 것이다. 대표적으로 로드뷰를 제공하는 회사는 구글, 카카오, 네이버가 있다. 이 중에서 구글이 제공하는 Google Street View API는 한국에서 사용하기에 적절하지 않다. 국내에는 스트리트 뷰를 지원하지 않는 도시가 많고, 있더라도 업데이트를 하지 않아서 과거 이미지를 제공한다. 카카오는 구글에 비해 로드뷰가 최근 이미지이고 대부분의 지역에서 자세한 로드뷰를 제공해 Kakao Maps API를 사용하였다. Kakao Maps API는 로드뷰의

수평 각, 수직 각, 줌 정도를 속성으로 설정할 수 있다.[2]

방위각은 나침반의 정북 방향에서 시계방향으로 잰 각도이다. 천문학에서 별의 위치를 측정하거나 항법, 지도를 제작할 때 사용된다. 아래의 공식을 통해 방위각이 계산될 수 있다.

$$\theta = \text{atan2}(\sin \angle \lambda \cdot \cos \varphi_2, \cos \varphi_1 \cdot \sin \varphi_2 - \sin \varphi_1 \cdot \cos \varphi_2 \cdot \cos \angle \lambda)$$

φ_1 , λ_1 출발지 위도 경도 φ_2 , λ_2 목적지 위도 경도

카카오 로드뷰는 처음 실행 시 초기 수평 각이 북쪽을 향하도록 되어 있어 방향을 알기 어렵다. 따라서 본 프로젝트에서는 위 방위각 공식을 이용해서 로드뷰의 방향을 조절하여 도로가 보이도록 설정하였다.[3]

2.3 클라우드 컴퓨팅 - AWS

클라우드 컴퓨팅은 데이터를 개인 컴퓨터가 아닌 다른 장소의 클라우드 컴퓨터로 전송해서 처리하는 기술을 의미한다. 서비스의 규모가 확대되면 인프라를 유연하게 추가할 수 있고 트래픽이 갑자기 증가하면 쉽게 대응할 수 있다는 장점이 있다. 본 프로젝트에서는 IaaS 클라우드 서비스인 Amazon Web Services(AWS)를 사용한다. AWS는 전세계 25곳에 물리적인 컴퓨터를 두고 지리적 영역인 리전(region)을 가지고 있다. 가상화 기술을 사용해서 물리적인 서버를 가상 서버로 나누고 사용자에게 서버, 스토리지 등 자원을 제공한다.

2.4 Grabzit

Grabzit은 화면을 캡처해 이미지로 저장하는 API이다. 스크린샷 저장 방법은 동기식과 비동기식이 있다. 동기식은 URL을 Grabzit으로 전송하고 Grabzit에서 화면이 캡처되었다는 응답이 올 때까지 대기한다. 응답이 오지 않으면 받을 때까지 여러 번 호출해서 웹 어플리케이션에 적합하지 않다. 비동기식은 Grabzit으로 URL을 전송하고 응답을 기다리면서 서버에서 다른 프로세스를 작동시킬 수 있다. 기다리는 동안 새로운 요청을 전송할 수 있고 서버가 중지되지 않으면 시간이 오래 걸려도 응답을 받을 수 있다. 본 프로젝트에서는 경로 당 여러 개의 로드뷰를 캡처해야 하므로 비동기식 방법을 이용했다.

2.5 Data Labelling : Yolo Mark

데이터 라벨링이란 데이터의 형태를 인공지능이 스스로 학습할 수 있는 형태로 가공하는 과정이다. 학습하는 데이터와 그것이 어떤 의미인지 가르쳐주는 라벨이 필요한 것이다. Object Detection 모델인 YOLOv5를 학습시키기 위해서는 이미지와 그 이미지에서 탐지할 객체의 위치를 담은 텍스트 파일이 필요하다.

Yolo_mark는 Windows와 Linux에서 사용할 수 있는 GUI 프로그램으로 이미지에서 인식할 객체를 bounded box하여 데이터 라벨링을 할 수 있다. 이미지에서 객체를 bounded box 처리를 하면 그 위치에 대한 정보가 담긴 텍스트 파일이 생성된다. 본 프로젝트에서는 Yolo_mark를 사용해 데이터셋을 직접 구축하였다.

2.6 Image train model: YOLOv5

object detection이란 이미지나 동영상에서 사람, 동물, 차량 등 의미 있는 객체(object)의 종류와 그 위치(bounding box)를 정확하게 찾기 위한 컴퓨터 비전기술이다.

YOLOv5는 이미지에서 object detection(객체탐지)를 해주는 오픈소스 라이브러리이다. YOLOv5는 "You Only Look Once"라는 뜻을 가진 YOLO의 가장 최신 버전으로 빠른 속도와 높은 정확도를 보여준다.

인공신경망 기법인 CNN의 성능을 높여주는 CSPNet 기반의 backbone이 PyTorch로 구현되었다. 학습의 깊이와 너비 정도에 따라서 모델이 4가지로 나뉘어진다. YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x로 각각 small, medium, large, xlarge라고 생각하면 된다. s가 가장 빠른 대신 정확도는 떨어지고 x가 가장 느린 대신 정확도는 향상된다.[4]

3. 사용방법

Android 사용자들은 애플리케이션 설치를 통해 플랫폼을 사용할 수 있다. 위치 기반 어플리케이션이기 때문에 위치 허용을 해줘야 사용이 가능하다. 그림1의 경로 입력화면에서 사용자는 자신이 얻고자 하는 경로의 출발지와 도착지를 입력할 수 있다. 출발지를 현재 사용자의 위치로도 설정할 수 있다.



그림 1 홈 화면(좌) 경로 입력(우)

확인 버튼을 클릭하면, 그림2의 결과 지도 화면과 같이, 사용자가 입력한 출발지-도착지까지의 경로를 선으로 표현해주고 해당 경로에서 인식된 블라드의 위치를 마커로 표현해준다. 마커를 클릭하면 그림2의 로드뷰 화면으로 이동하고, 해당 좌표의 로드뷰 사진과 위험요소 이름을 사용자에게 알려준다.

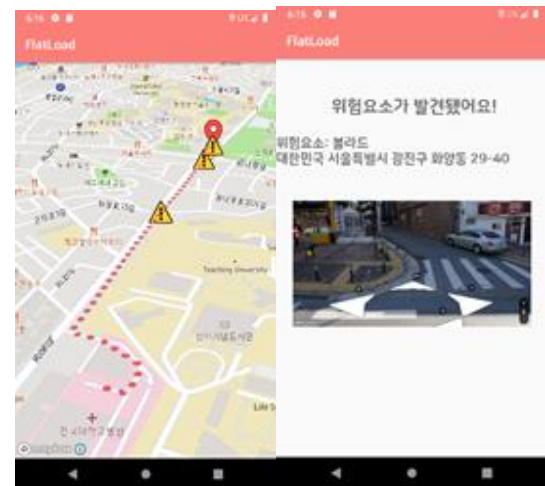


그림 2 결과 지도(좌) 결과 로드뷰(우)

4. 설계

본 장에서는 결과 애플리케이션 제작을 위해 구축된 서버, Image train model과 같은 기술과 자체적으로 설정한 도로 위 위험요소에 대해 설명한다. 본 프로젝트의 전체적인 구조는 그림3과 같으며, 크게 Android Application, Server, Image Model로 구성이 되어있다.

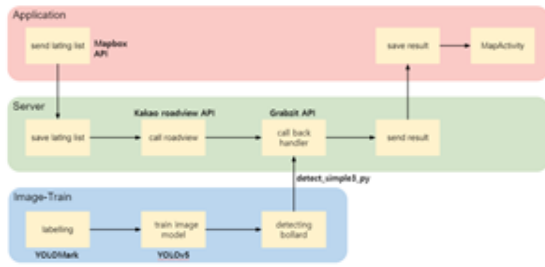


그림 3 플랫로드 시스템 구조

로드뷰 이미지 리스트를 응답 받는다. 이것으로 그림2의 결과 화면을 구성하였다.

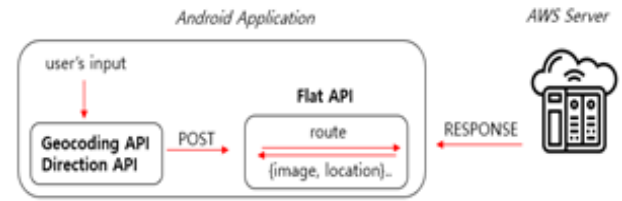


그림 5 애플리케이션-서버 통신

4.1 도로 위 위험요소 정의

Flatroad는 최종적으로 이동제약자를 위해 안전한, 위험요소가 적은 길 찾기 경로를 사용자에게 제공한다. 본 프로젝트에서 자체적으로, 이동 제약자가 이동할 때 가장 불편함을 주는 도로 위 위험요소로 블라드를 정의했다. 블라드(bollard)란, 자동차가 인도에 진입하는 것을 막기 위해 차도와 인도 경계면에 세워 둔 구조물이다. 도로 위에 고정되어있고, 이동제약자의 무릎쯤 오는 블라드가 대부분이다. 또한 횡단보도 이외에도 도로 한가운데 설치되어 있는 경우도 있어 그들의 보행에 방해가 가장 많이 될 것으로 예상되었다. 따라서 블라드를 위험요소로 설정하였고 사용자의 경로에서 찾아서 위치를 알려주기로 하였다. 블라드도 다양한 형태가 존재하는데 가장 대중적으로 도로에 많이 있는, 그림 4와 같은 블라드를 위험요소로 선정하여 프로젝트를 진행하였다.



그림 4 위험요소로 지정한 블라드 이미지

4.2 애플리케이션

서버와 데이터를 주고받아, 결과를 출력하는 애플리케이션을 제작하기 위해, 그림 5 와 같은 설계를 기반으로 통신 체계를 구축했다. 애플리케이션에서 사용자의 입력을 받고, 2.1의 Mapbox API를 바탕으로 경로를 얻는다. 경로는 Retrofit 통신 모듈을 이용해 자체제작한 Flat API를 통해 서버로 전송한다. 일정시간이 지난 후, 서버에서 4.1에서 정의한 위험요소의 위치와

4.3 서버

4.3.1 애플리케이션 간 데이터 통신

Node.js 웹 애플리케이션 프레임워크인 Express를 기반으로 서버가 구축된다. HTTP 통신요청에 대한 핸들러를 만들고, view의 렌더링 엔진과 결합하여 템플릿에 데이터를 넣은 응답을 만든다. 이러한 지식을 바탕으로 본 장에서 설명한다.

클라이언트 역할을 하는 안드로이드 애플리케이션과 Express 프레임워크로 통신하기 위해, 4.2에서 설명한 것처럼 클라이언트는 Retrofit 통신 모듈을 이용할 것이다. 서버에서는 클라이언트에서 POST요청을 받았을 때의 처리를 진행한다. 클라이언트에서 받은 경로 리스트를 저장하고 timeout 내장함수를 이용하여 일정시간이 지나면, 처리한 결과 정보를 클라이언트에게 응답한다. 이미지를 전송해야하기 때문에 base4 encoding 과정을 거쳐 bitmap으로 클라이언트에게 전송한다.

4.3.2 로드뷰 스크린 샷

서버에서는 앱에서 POST한 좌표를 경도, 위도로 순차적으로 저장하고 앞 좌표의 위도, 경도와 뒤 좌표의 위도, 경도를 위의 2.2 방위각 공식에 넣어 방위각을 계산한다. 카카오 로드뷰의 Javascript 라이브러리를 사용하고 서버에서 보낸 좌표와 계산한 방위각을 Query String으로 받아서 웹페이지에 로드뷰를 띄운다. 웹 페이지는 데이터 처리를 위해 Express의 템플릿 엔진 EJS를 사용해서 동적으로 구성하였다.

모든 과정이 완료되면 받은 좌표와 방위각으로 URI를 생성해 Grabzit 서버로 스크린샷 요청을 전송한다. Grabzit에서 스크린 샷이 준비되면 Handler를 호출해서 서버에 jpg 파일로 저장한다. 모든 좌표의 스크린샷이 저장되면 인공지능 모델을 작동시킨다.

Welcome to FlatRoad

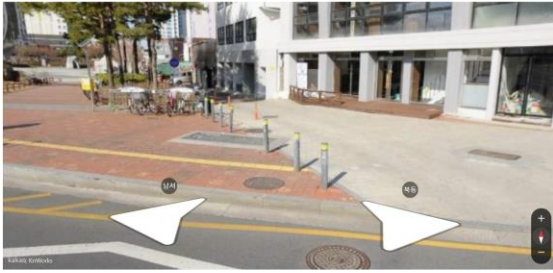


그림 6 Flatroad 웹 페이지

4.3.3 블라드 객체 탐색

로드뷰 이미지에서 블라드를 탐색해주는 인공지능 모델은 파이썬 파일에서 실행된다. 그래서 node.js에서 파이썬 파일을 실행할 수 있는 child_process 모듈을 사용하였다. child_process는 node.js의 기본 모듈로 자식프로세스 기능을 사용할 수 있게 한다. child_process 모듈의 spawn을 통해 파이썬 파일을 실행한다. stdout의 이벤트리스너를 통해 파이썬 파일의 실행 결과를 받는다.

인공지능 모델을 돌리는 파이썬 파일은 YOLOv5에서 제공하는 detect.py를 파일을 수정하여 사용하였다. 이 파일은 블라드가 검출된 '이미지파일의 이름' 과 객체 이름인 '블라드'를 디렉터리로 저장하여 출력한다. 블라드가 검출되었을 때 정확도가 0.6 이상이 나온 것만 실제 블라드를 인식해낸 것이라 판단하도록 하였다.

4.4 인공지능 모델

인공지능 모델은 서버에서 사용되어 로드뷰 이미지에서 블라드 유무를 판별해낸다. 데이터셋 구축부터 인공지능 모델 학습까지 직접 진행하였다. 이미지데이터 모으기, 데이터 라벨링, 인공지능 모델 학습으로 3가지 단계에 거쳐서 인공지능 모델을 구축하였다.

이미지 데이터는 직접 찍은 사진, 구글 크롤링 이미지, AIHub에서 지원받은 사진, 카카오 로드뷰를 캡처한 사진을 통해 총 563장의 이미지를 모았다.

데이터 라벨링은 Yolo_mark 프로그램을 이용하였다. 이미지 속 블라드를 찾아 bounded box 처리하였고 블라드의 위치 정보를 담은 텍스트 파일이 생성되었다.

이렇게 구축한 데이터셋을 가지고 Google colabatory(코랩)에서 블라드를 판별해내는

인공지능 모델을 만들었다. YOLOv5의 4가지 모델 중 YOLOv5s 모델을 사용하여 학습시켰다. 563장의 이미지를 50 epoch로 학습시켰고 약 4시간 30분이 소요되었다.

학습된 모델은 블라드를 잘 인식하였고 정확도도 70프로 이상으로 괜찮은 성능을 보여주었다. 이 모델은 best.pt 라는 파일로 저장되었고 이를 가지고 서버에서 저장한 로드뷰 이미지에서 블라드의 유무를 판별하도록 하였다.



그림 7 학습된 인공지능모델 성능평가 결과

5. 결론 및 향후연구

시대가 발전하면서 사람들은 더 좋은 복지와 편리함을 누리고 있다. 하지만 관심이 닿지 않는 곳에서는 누군가 아직 불편함을 가진채 살아가고 있다. 이동 제약자들에게 도로를 이용하는 자체가 하나의 불편함이고 일반 사람을 기준으로 한 도로를 이동하는것에는 어려움이 존재한다. Flatroad를 통해 이동 제약자들이 경로에 대한 정보를 얻는다면, 그들이 가고자 하는 경로의 위험요소 위치를 파악하고, 위험성을 인지하여 대비할 수 있다. 하지만 현재 개발된 Flatroad는 위험요소를 블라드로만 정의했다는 점이 한계점이다. 다양한 위험요소를 탐색해낼 수 있도록 데이터셋을 더 구축하고 인공지능의 성능을 높여, 보다 정확한 정보를 사용자에게 제공할 수 있도록 할 것이다. 또한 최신 도로 정보 수집을 위해 사용자들이 직접 위험요소의 사진을 업로드 하고 위험요소를 지정할 수 있게 하여, 맞춤형 정보를 제공할 수 있도록 할 예정이다.

참고문헌

- [1] Mapbox Developer Documentation – Maps SDK for Android, DirectionAPI, GeocodingAPI
- [2] Kakao Developers, Kakao Maps API
- [3] Movable Type Scripts(2010), Calculate distance, bearing and more between Latitude/Longitude points
- [4] Glenn Jocher (2020/5/18), YOLOv5 github