



출품 분야 : 임베디드 SW, 윈도우, 모바일SW, 기타()

1. 프로젝트 이름

음성 기반의 딥 러닝을 활용한 시각장애인 보조 에이전트

2. 팀 이름

라이썬!(Light you)

3. 팀 구성

유형열(컴퓨터학부)

박정훈(컴퓨터학부)

최형석(컴퓨터학부)

현석우(컴퓨터학부)

황태관(전자정보공학부)

4. 담당교수님

최지웅 교수님

5. 기획의도

한국 국민들의 소득수준은 불과 몇 년 사이에 수십 배 이상 상승했고 삶의 질은 높아져왔다. 이러한 물질적 풍요는 사회 전반에 걸쳐 보편적인 복지가 실현될 수 있게 하였다. 하지만 사회적 소외계층에 대한 복지는 여전히 미흡한 실정이다. 그 중 시각장애를 가진 국민에 대한 처우는 오히려 시대의 흐름에 역행하는 모습도 보이고 있다. SBS 8뉴스 리포트에 따르면 한 행정구에서는 시각장애인용 보도블록을 기존의 입체적인 벽돌 형태에서 외관상으로만 구분 가능한 스티커로 교체하기도 하였다. 이밖에도 현대 한국사회는 시각장애인들이 살아가는 데 있어서 많은 어려움을 겪을 수밖에 없는 모습을 띈다. 물론 이들의 처우 개선을 위한 노력도 상당 부분 존재한다. 시각장애인들을 위한 스마트폰 기능, 장애물이 있을 경우 경고를 해주는 지팡이 등 미약하지만 이들의 불편을 이해하고 해결하기 위한 노력들이 있

다. 그럼에도 불구하고 이와 같은 기능들은 시각장애인들의 생활 보조에 있어서 뚜렷한 해결책을 제시하지는 못하고 있다. 종합해보면 현대사회를 살아가는 시각장애인들은 여전히 수많은 난관들을 겪고 있다. 때문에 이들의 생활 전반을 보조할 수 있는 서비스를 만들기 위해 본 프로젝트를 기획하게 되었다.

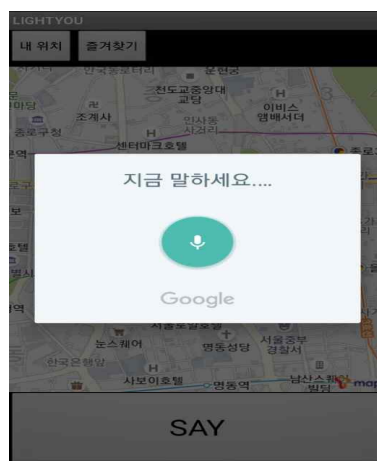
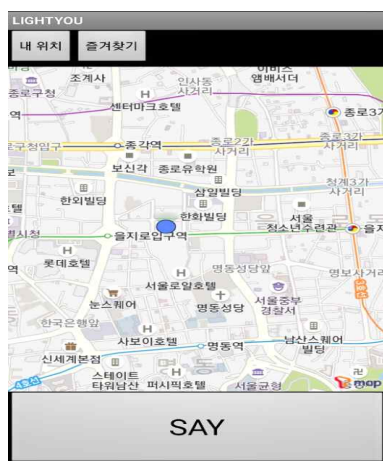
6. 프로젝트 및 작품 설명

가. 작품 개요

라이츄!(Light you)는 음성 기반의 딥 러닝을 활용한 시각장애인 보조 에이전트이다. 목표 사용자인 시각장애인이 겪을 법한 장애 요소에 주안점을 두어 일상생활에서 가장 흔하게 필요한 기능을 구현하였다. 음성을 통해 실행하고 응답을 들을 수 있는 서비스로 시각장애인의 일상에 도움을 주어 삶의 편의를 크게 개선할 수 있도록 한다.

나. 개발 목표 및 활용 방안

음성을 통해 애플리케이션과 상호작용하고 사용자가 원하는 기능을 수행할 수 있도록 하는 시스템을 제작한다. 타겟 사용자가 시각장애를 가졌기 때문에 애플리케이션 내부에서 자유로운 터치 이벤트를 작동시키기에는 많은 제약이 있다. 따라서 이러한 부분을 제일 먼저 고려하여 시각적인 한계가 있더라도 손쉽게 애플리케이션을 사용할 수 있도록 하기 위해 사용자 인터페이스(UI)는 최대한 간결하게 만들었다. 다만 보호자도 본 기능이 제대로 수행되고 있는지 확인할 수 있도록 하기 위해 추가로 지도와 즐겨찾기 목록을 볼 수 있도록 하였다.



이를 통해 보호자는 시각장애인이 정상적으로 애플리케이션을 동작시키고 있는지 확인할 수 있다.

애플리케이션의 하단에 위치한 버튼을 누른 후 음성 명령을 내리면 특정 기능을 수행할 수 있다. 사용가능한 기능과 음성의 예는 다음과 같다.

[그림 1] Light you 초기화면 [그림 2] Light you 음성인식

- 1) 지정한 위치 즐겨찾기 추가 : “즐거찾기 추가 해줘”
 - A라는 장소를 B라는 장소로 즐겨찾기하는 기능이다.
 - 본 기능은 데이터베이스에 저장되어 차후 길 안내 기능에서 활용될 수 있다.
 - 예시) ‘숭실대학교’를 ‘우리학교’로 저장한다.
- 2) 내 위치 확인 : “내가 지금 어디에 있지?”
 - GPS 기능을 통해 현재 내가 위치한 장소를 알려주는 기능이다.
- 3) 원하는 목적지로 길 안내 : “길 안내 해줘”
 - 목적지를 입력받으면 현재 내가 위치한 장소로부터 목적지까지의 보행 경로를 알려주는 기능이다.

4) 전방 인식 : “전방 인식 시작할게”

– 카메라 모듈을 통해 전방에 있는 사물을 식별하고 그 사물에 대한 정보를 알려주는 기능이다.

5) 특정 물건 찾기 : “OOO 어디에 있어?”

– 특정 사물이 어디 있는지 물어본 뒤 카메라 모듈을 통해 인식된 사물이 찾고자 하는 사물과 일치하면 발견했다는 메시지를 말해주는 기능이다.

6) 현재 시각 확인 : “지금 몇 시야?”

– 현재 시각을 알려주는 기능이다.

다. 관련 기술

1. 라즈베리파이, 파이카메라

라즈베리파이는 초소형, 초저가 PC이다. 라즈베리파이에서는 리눅스 환경에서 Python으로 개발을 진행한다. 데비안(Debian) 계열의 raspbian-jessie라는 운영체제를 사용하여 개발을 진행하였다. 사진을 찍기 위한 카메라는 라즈베리파이 카메라 모듈 v2를 이용하였다. 본 프로젝트에서는 Python에서 카메라를 제어할 수 있는 모듈 picamera를 사용하였고, 이를 서버에 전송하기 위해 socket 모듈을 사용하였다.

2. 텐서플로우(Tensorflow)

텐서플로우란 다차원의 배열(텐서)들이 그래프에서 이동(플로우)한다는 의미의 구글의 딥러닝 프레임워크이다. 학습한 데이터와 연산 결과를 다차원의 배열로 저장하여 그래프로 연결하는 데 강점이 있다.

3. Inception-v3

Inception-v3란 Image Object Detection 분야에서 사용되는 구글의 최신 모델이다. 이 모델은 여러 항목(label)과 각각 그에 해당하는 데이터들이 있을 때, 입력된 이미지에 대하여 각 벡터의 가중치를 계산하는 softmax regression (multinomial regression) 알고리즘을 이용한다.

4. GCP(Google Cloud Platform)

구글의 데이터센터 인프라를 기반으로 실제 사용하는 컴퓨팅, 스토리지 등의 기술들을 외부 개발자들이 활용할 수 있게끔 한 구글의 플랫폼으로서 스냅챗, 앵그리버드 등 수 백만 개의 애플리케이션이 호스팅 되어 있는 퍼블릭 클라우드다. 자신이 사용한 가상머신에 상응하는 가격을 지불함으로써 간편하고 유연한 서버의 구축이 가능하다. 서버는 다양한 지역에 위치하여 사용자가 선택할 수 있게끔 되어있으며, 본 프로젝트에서는 극동아시아 지역의 서버를 이용하였다.

5. TMAP API

TMAP API는 SK Planet에서 제공하는 지도 API 기능으로서, 실제 휴대폰이나 네비게이션에서 제공되는 T MAP의 정보를 해당 API를 통해 받아 이용을 할 수 있다. 본 프로젝트에서는 목적지 통합검색 기능(경로검색), POI 기능을 주로 사용하였다.

6. POI(Point Of Interest)

POI란 사용자가 관심을 갖는 지점(point)을 말한다. 이는 우리가 알고자 하는 지점에 대한 정보를 얻기 위해 사용한다. 본 프로젝트에서는 내 위치, 길 안내 기능 중 목적지에 대한 정보를 얻고자 할 때 활용되었다.

7. 위도, 경도를 이용한 거리 및 반경 계산

두 점의 위도 경도를 이용하여, 두 위치 사이의 거리를 구한다.

- 1) LongDistance : 두 경도간의 각도(도, 분, 초) 차이 값에 각 거리를 곱하여 두 점 간 수평거리를 계산한다.
- 2) LatDistance : 두 위도간의 각도(도, 분, 초) 차이 값에 각 거리를 곱하여 두 점 간 수직거리를 계산한다.
- 3) Distance : 직각 삼각형의 빗변의 거리를 구하는 공식을 이용한다.

$$Distance^2 = LongDistance^2 + LatDistance^2$$

위도 경도를 이용하여 반경을 구한다.

- 1) 반경 내의 위도 차이를 구한다.
위도 차이 = 반경 * 360 / 2 * π * 지구반지름
- 2) 반경 내의 경도 차이를 구한다.
경도 차이 = 반경 * 360 / 2 * π * 지구반지름 * Cos(Latitude * π / 180)
→ 반경 및 지구반지름은 m단위

반경 내의 위도 범위 = 해당 지점의 위도 ± 위도 차이

반경 내의 경도 범위 = 해당 지점의 경도 ± 경도 차이

8. STT(Speech to Text)

음성 인식(Speech Recognition)이란 사람이 말하는 음성 언어를 컴퓨터가 해석해 그 내용을 문자 데이터로 전환하는 처리를 말한다. STT(Speech-to-Text)라고도 한다. 키보드 대신 문자를 입력하는 방식으로 주목을 받고 있다. 대표적인 알고리즘은 HMM(Hidden Markov Model)으로서, 다양한 화자들이 발성한 음성들을 통계적으로 모델링하여 음향모델을 구성하며 말뭉치 수집을 통하여 언어모델을 구성한다. 본 프로젝트에서는 사용자의 음성 명령을 인식하고 이를 텍스트로 변환하여 특정한 기능을 수행하도록 하기 위해 사용되었다.

9. TTS(Text to Speech)

음성 합성(Speech Synthesis)은 말소리의 음파를 기계가 자동으로 만들어 내는 기술로, 간단히 말하면 모델로 선정된 한 사람의 말소리를 녹음하여 일정한 음성 단위로 분할한 다음, 부호를 붙여 합성기에 입력하였다가 지시에 따라 필요한 음성 단위만을 다시 합쳐 말소리를 인위로 만들어내는 기술이다. TTS(Text-to-Speech)라고도 한다.

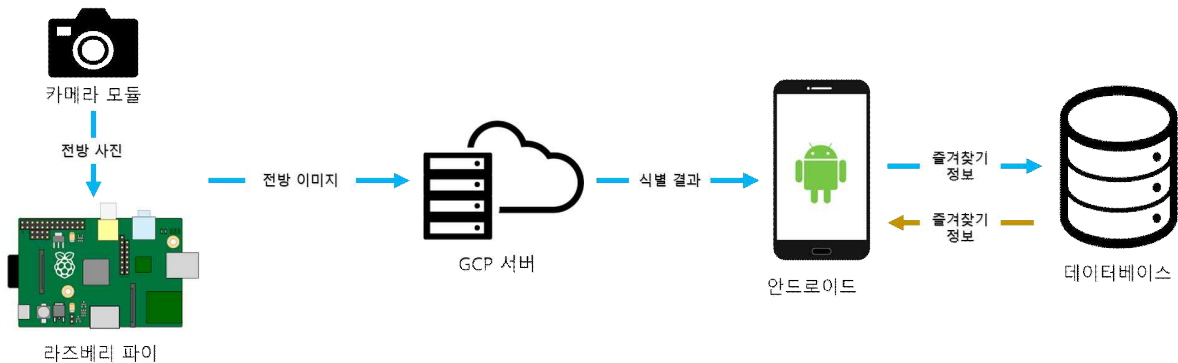
라. 시스템 구성도

1) 시스템 구성도(요약)

카메라 모듈은 전방의 모습을 실시간으로 촬영한다. 촬영된 이미지는 라즈베리 파이를 통해 GCP서버로 전송되게 된다. 서버에서는 전송된 이미지를 모델링하여 식별한 뒤 해당 이미지

가 나타내는 사물의 이름을 Android Device에 전송하게 된다.

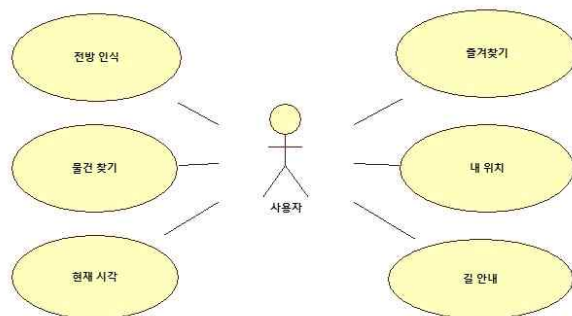
데이터베이스는 즐겨찾기 기능을 위한 것으로서, 즐겨찾기 정보를 저장하고 이 정보를 바탕으로 새로운 길 찾기 기능을 수행하거나 현재 데이터베이스에 저장된 즐겨찾기 정보를 확인할 때 사용된다.



[그림 3] Light you 시스템 구성도

2) Use Case Diagram

사용자가 실제 애플리케이션을 이용할 때 사용할 수 있는 기능을 정리하면 [그림4]와 같다.



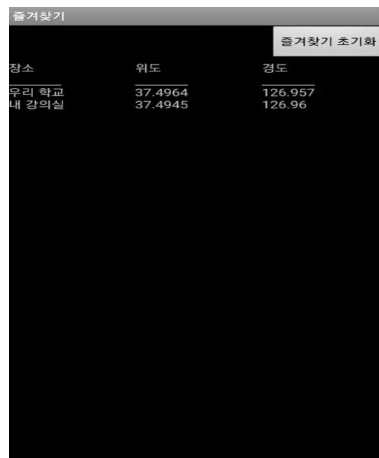
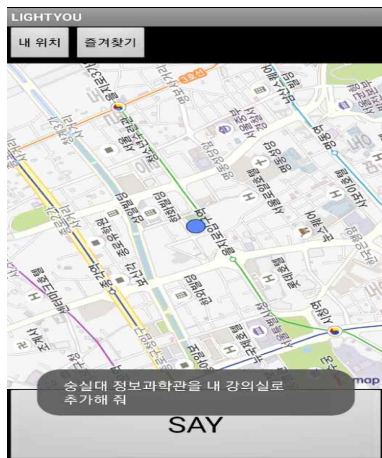
[그림 4] Use Case Diagram

마. 구현기능

1) 애플리케이션의 기능별 서술

1. 즐겨찾기

즐거찾기 기능은 사용자가 길 안내 기능을 이용할 때 빈번하게 입력하는 목적지에 대해서 사용자가 설정한 명칭으로 대체하여 입력할 수 있도록 하는 기능이다. 이러한 기능은 이름이 긴 목적지 주소를 사용자가 직접 간단한 명칭으로 설정할 수 있게끔 한다는 장점이 있다. 구체적인 구현 내용은 다음과 같다. 먼저 즐겨찾기 기능은 다음과 같은 형식에 맞춰 입력한다. 사용자가 SAY 버튼을 누른 후 “송실대를 우리학교로 추가해줘”와 같이 음성 명령을 내리면 입력한 음성을 인식하여 문자열로 변환한다. 이렇게 문자열로 변환된 음성은 StringTokenizer 클래스의 split() 메소드를 호출하여 분리한다. split() 메소드는 정규 표현식을 구분자로 해서 문자열을 분리한 후, 배열에 저장하고 반환한다. 구체적으로 즐겨찾기



기능에서는 “을|를|으로|로”를 구분자로 해서 문자열을 나누는데, 이렇게 나누게 되면 “송실대를 우리학교로 추가해줘”와 같은 문자열에서 ‘송실대’와 ‘우리학교’로 분리하여 배열에 저장할 수 있다. 이렇게 나눈 문자열 중 목적지 주소는 그 주소의 위도와 경도를 찾는 데

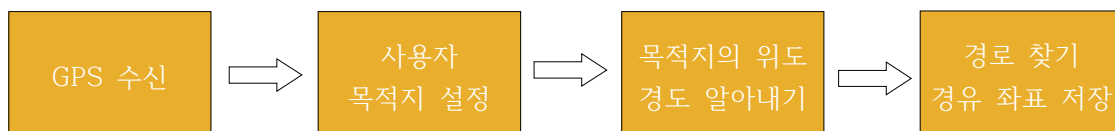
[그림 5] 즐겨찾기 추가 화면 [그림 6] 즐겨찾기 목록 확인 사용되고, 사용자가 지정한 명칭은 앞서 구한 위도와 경도와 함께 DB에 저장된다. 결과적으로 즐겨찾기 기능을 사용하면 앞으로 길 안내 시 목적지 주소로 길 안내를 요청하지 않고 사용자가 직접 지정한 명칭으로 요청할 수도 있는 것이다.

2. 내 위치

본 애플리케이션은 처음 실행될 때 GPS로 현재 위치정보를 얻어온다. 현재 위치에 대한 위치정보는 현재 위치가 변화 할 때마다 갱신된다. 위치 정보를 통해 위도, 경도를 알아낸 뒤 두 데이터를 이용해 현재 주소를 확인할 수 있게 된다. 현재 주소에 대한 정보를 텍스트로 알아낸 후 음성을 통해 최종적으로 현재 나의 위치를 안내해준다.

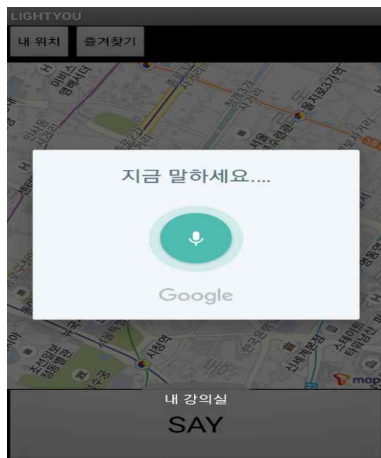
3. 길 안내

길 안내 기능의 시나리오는 다음과 같다.

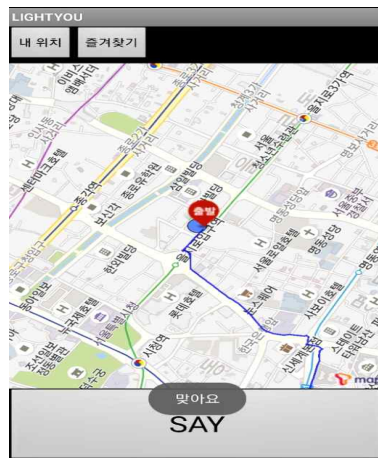


[그림 7] 길 안내 기능 시나리오

먼저, GPS를 통해 현재 위치를 인식한다. 이어서 현재 위치에 대한 위도 경도를 변수에 저장한다. 그 다음, 사용자가 음성 인식을 통해 목적지를 입력하면 입력된 음성에 가장 비슷한 문자열로 변환 한다. 목적지가 입력된 후에는 목적지에 대한 위도, 경도를 알아낸다. 목적지에 대한 위도, 경도 역시 변수에 저장한다. 현재 위치 - 목적지 간 경로를 탐색하여, 경유좌표를 리스트에 저장한다. 이 때, 경유좌표에서 어디로 가야하는지에 대한 방향도 함께 저장한다. 이 정보로 적절한 시기에 안내를 한다. 목적지에 도달할 때까지 현재 위치가 다음 경유 좌표의 근방에 도달할 때마다 다음 방향을 안내한다. 경로 탐색 시 안내에 대한 정보를 음성으로 변환하여 사용자에게 알려준다.



[그림 8] 목적지 입력 화면



[그림 9] 길 안내 화면

4. 전방인식

본 기능은 사용자 전방 이미지를 추론해 음성으로 안내해주는 기능이다. 우선 서버는 사전에 학습된 모델을 이용하여 softmax regression 알고리즘을 적용해 학습된 항목 중 가장 유사한 사물의 정보를 Android Device에 전송한다. 안드로이드는 이렇게 받은 정보를 사용자에게 음성으로 전달한다. 사용자는 "전방", "전방안내" 등의 키워드로 해당 기능을 실행할 수 있다. 또한 안드로이드 어플리케이션을 시작 할 때 서버와 이미 연결이 되어있으므로 서버가 지속적으로 보내오는 전방의 물체에 대한 정보는 버퍼에 저장되어 있다. BufferedReader 클래스의 readLine 매서드는 일정시간마다 이렇게 버퍼에 쌓인 데이터를 전부 비우고 새로 들어오는 정보를 읽음으로써 서버로부터 전송된 실시간의 정보를 수신한다. 수신된 정보는 영어이기 때문에 translator() 매서드를 이용해 한글로 변환해서 사용자에게 안내한다. 해당 기능은 스레드로 구현되어 있으며, "이제 됐어" 키워드로 기능을 종료할 수 있다.

5. 물건 찾기

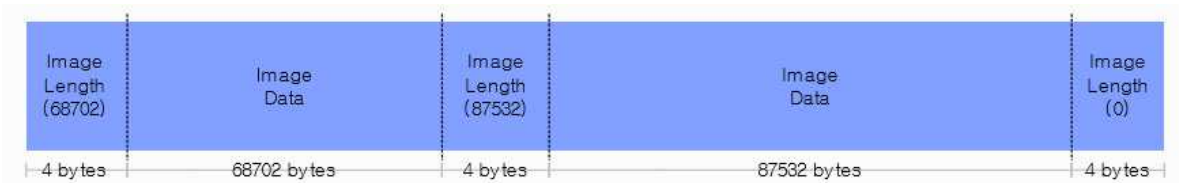
"찾아", "어디있어" 등의 키워드로 사용자는 물건 찾기 기능을 실행할 수 있다. 추가적인 인식 수단이 없이 카메라 모듈을 이용한 탐색이므로 위치에 대한 상세 안내는 어렵다. 그러나, 전방에 찾으려는 물체로 추정되는 사물이 인식 될 때 즉각 사용자에게 알려주는 기능은 유용하게 활용될 수 있다. 기본적으로 위에 서술된 전방인식 기능과 유사하게 스레드를 새로 만들어 실행되고 스레드를 닫음으로써 종료된다. 기능을 종료하는 키워드는 "그만 찾기"이다.

6. 현재 시간

현재 시각을 말해주는 기능이다. 자바의 Calendar 클래스를 활용해서 현재 시각 정보를 텍스트로 받아온 뒤 음성을 통해 알려준다.

2) 라즈베리 파이, 파이카메라를 통한 사진 촬영, 이미지 전송

Raspberry Pi 3 model B device에 카메라 모듈을 연결하여 사진을 찍을 수 있는 물리적 환경을 구축하였다. 우선 socket을 이용해 Google Cloud Platform 상의 instance에서 실행한 서버 프로세스에 TCP connection을 생성한다. python에서 카메라 모듈의 편리한 사용을 돕는 picamera 모듈을 import하여 PiCamera() instance를 생성하고, 용량을 줄이기 위해서 사진의 크기를 320 X 240 로 설정한다. 그리고 사진을 연속적으로 보내기 위하여 간단한 프로토콜을 사용하는데, 처음에는 4 Bytes에 이미지의 크기를, 그 후 이미지의 크기만큼 이미지 데이터를 전송하고, 다시 4 Bytes에 이미지의 크기를 전송하는 방식이다. 이 프로토콜을 사용하여 앞서 생성한 TCP connection에 사진을 연속적으로 전송한다.



[그림 10] 이미지 송신 프로토콜

3) 서버 기능

Google Cloud Platform 상에는 사전에 모아둔 Training Image Set으로 multinomial classification 중 하나인 softmax regression 알고리즘을 활용하여 이미지를 분류하는 모델이 사전에 학습되어 있다. 이 서버는 실제 구동 시 Raspberry Pi와 Android device에서의 연결 요청을 받아 TCP connection을 만든다. 앞서 서술한 연속 사진 전송 프로토콜을 사용하여 Raspberry Pi에서 사진을 받아온다. 학습된 모델을 바탕으로 Raspberry Pi에서 받은 사진을 판별하여 전방의 물체가 무엇인지 예측하고, 학습된 항목(label) 중 가장 유사하다고 여겨지는 항목의 문자열 데이터로 저장된다. 그리고 Android Device가 서버와 연결되어 있다면, 이 문자열을 Android Device에 전송한다. 서버는 멀티쓰레드로 구성되어 이미지 수신, 대상 식별(Object Detection), 결과 전송을 동시에 수행할 수 있게끔 구현되었다. 본 서버를 통해 데이터 송수신 및 이미지 모델링을 하는 데 걸리는 시간은 약 1초(1sec)정도이기 때문에 본 애플리케이션의 사용자는 실시간에 가깝게 전방 물체를 인식할 수 있게 된다.

7. 개발 환경

하드웨어 \ 소프트웨어	OS/Platform	Language
Galaxy S6	android-7.0	java
Google Cloud Platform	Ubuntu LTS 16.04.3	python
Raspberry Pi 3 Model B	raspbian-jessie	python
Raspberry Pi Camera V2.1	-	-

[표 1] 개발 환경

8. 참고문헌

- [1] https://www.tensorflow.org/get_started/
- [2] https://www.tensorflow.org/tutorials/image_recognition
- [3] <https://cloud.google.com/why-google/>
- [4] <https://developers.skplanetx.com/apidoc/kor/tmap/>
- [5] https://en.wikipedia.org/wiki/Speech_recognition
- [6] https://en.wikipedia.org/wiki/Speech_synthesis