

공개 소프트웨어 프로젝트 최종보고서

과제 수행원 현황						
수행 학기	<input checked="" type="checkbox"/> 2019년 3월~2019년 6월 <input type="checkbox"/> 2019년 9월~2019년 12월					
프로젝트명	CANE (Co-Assistant system for Neighbor's Eyes)					
팀명	자료9조					
	학과	학번	성명	성별	연락처	E-mail
팀장	컴퓨터공학과	2015112082	김현도	남	010-2304-9068	kk090303@naver.com
팀원	컴퓨터공학과	2015112149	연정민	남	010-9943-8426	jryoun1@naver.com
	컴퓨터공학과	2015112150	최연호	남	010-8645-2275	dusgh4321@naver.com
	컴퓨터공학과	2015112104	이정우	남	010-4798-1087	dwd4798@gmail.com
지도교수	교과목명	공개 소프트웨어 프로젝트				
	소속	<input checked="" type="checkbox"/> 컴퓨터공학전공 <input type="checkbox"/> 정보통신공학전공 <input type="checkbox"/> 멀티미디어공학전공 <input type="checkbox"/> 융합소프트웨어연계전공				
	성명	손 윤 식				

# Key Words	#베리어프리	#시각장애인	#안드로이드	#텐서플로우	#보행보조 어플리케이션
1.개발동기/ 목적/필요성 및 개발 목표	<p>개발동기 및 목적</p> <p>우리 조원들은 신공학과 3층에 있는 알파실을 자주 이용하곤 하는데, 알파실 앞에 붙어있는 게시물들 중에 베리어 프리라는 사회적 약자계층을 위한 공모전과 관련된 게시물을 보게 되었다. 이에 관심이 생겨 자료를 찾아보게 되었고, 그 중에서도 시각 장애인에 대한 자료를 보게 되었는데, 조사해본 결과 전국 1.2급 시각장애인 약 4만명 중 안내견의 도움을 받는 시각 장애인은 2%에 못 미치는 수준이었고, 대부분의 시각 장애인이 도보 스틱과 주변 사람들의 도움을 받아 생활을 하고 있는 것으로 알려졌다. 안내견 한 마리를 훈련시키는데 들어가는 비용 1억원에 대한 현실적인 비용 문제와, 사회적인 인식 부족 등의 많은 문제가 있었는데 우리는 이러한 상황 중, 현실적인 비용 문제를 조금이나마 해결할 수 있도록 우리는 경고음을 이용하여 시각장애인 보행 보조 프로그램을 주제로 선정하였다.</p> <p>사용대상</p> <p>사용 대상은 평소 보호자 또는 안내견의 도움 없이 케인을 사용하여 보행을 해야</p>				

	<p>하는 시각 장애인이다. 여기서 사용 대상은 우리가 개발한 프로그램에만 의존하는 것이 아니라 케인이라는 물리적 보호 장치를 수반하여 프로그램을 사용한다.</p> <p>개발목표</p> <p>위에 언급한 사용대상자들이 케인과 우리 프로젝트 결과물을 통해서 일상생활에서의 큰 부분을 차지하는 보행이라는 부분을 부담 없이 해결할 수 있도록 한다. 또한 긴급하게 도움이 필요한 경우 어플리케이션을 사용하는 도중에도 빠르게 연락을 할 수 있도록 한다.</p>
<p>2.최종 결과물 소개</p>	<p>1. 결과물에 대한 아이디어 추가</p> <div data-bbox="699 680 1066 837" data-label="Image"> </div> <p>그림 1 빅스비를 통한 어플 사용</p> <p>어플리케이션을 실행할 때 사용대상이 직접 어플리케이션의 위치를 찾아 실행시키는 것이 어렵기 때문에 삼성에서는 "빅스비", LG에서는 "씽큐" 등 기본적으로 설치되어있는 음성인식기반 어플리케이션을 통해 CANE(케인) 어플리케이션을 부를 수 있도록 이름을 한글로 설정해 주었고 실제 실험을 통해 어플리케이션이 음성인식으로 실행되는 것을 확인하였다.</p> <p>2. 최종 결과물 소개</p> <div data-bbox="485 1238 890 1935" data-label="Image"> </div> <p>그림 2 어플 기본 설정 번호 112</p> <div data-bbox="943 1238 1348 1935" data-label="Image"> </div> <p>그림 3 사용자가 번호 입력한 경우</p>

이 어플리케이션의 사용자가 object detection 기능을 사용하던 도중 전화를 걸기 위해 전화번호를 입력하거나 연락처를 찾는 등의 일은 굉장히 어려울 것이다. 그렇기 때문에 사용자가 위급상황이나 꼭 전화를 해야 하는 경우 더블 탭을 통해 그림 4처럼 전화를 걸 수 있도록 하였고 보행을 시작하기 전에 처음에 지정해둔 전화번호로 편하게 전화를 걸 수 있도록 하였다.

어플리케이션을 처음 다운받은 경우 그림 2처럼 default number는 112 (경찰서) 이고 사용자가 실제로 사용하게 되면 그림 3처럼 번호를 입력 가능하다. 이후에 다시 어플리케이션을 실행 할 경우 가장 최근에 설정한 값으로 번호를 설정되어있다. 어플리케이션을 사용하던 도중에 전화를 걸고 전화통화를 마치게 되면 다시 어플리케이션으로 돌아와서 계속하여 object detection이 가능하다.

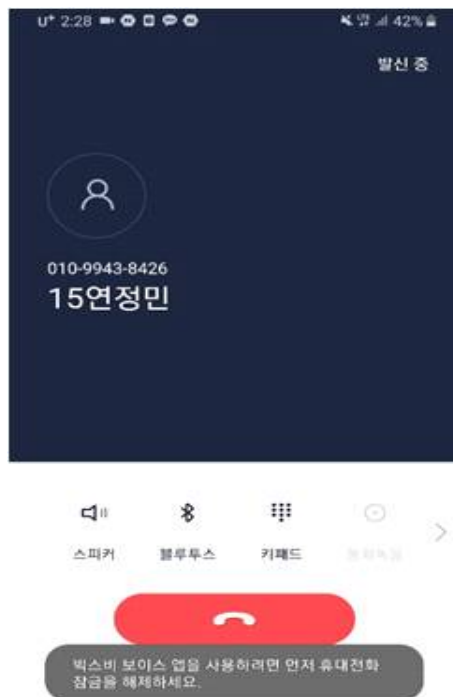


그림 4 더블 탭 한 경우 전화 걸림



그림 5 어플을 통한 트럭 인식

Object Detection을 하는 사진을 그림 5, 6, 7을 통해 제시한다. 다음 세 가지 외에도 추가적으로 자전거와 신호등까지 인식한다. 제외한 물체들은 시야에서 물체를 인식하는 네모 박스를 기준으로 상대적으로 거리를 계산하여 위험 하다고 판단될 시에 신호음을 통해 경고해준다. 자동차, 버스, 트럭과 자전거, 오토바이 두 분류에 따라 위험 신호 소리를 다르게 해주었다. 상대적인 거리 계산이기 때문에 물체가 처음부터 가까이 있다면 당연히 박스의 크기가 크고 소리가 울리게 된다. 어플리케이션을 켜올 당시 가까이 있는 물체는 어플리케이션 사용자가 Cane(지팡이)을 사용함으로 어느 방향에 물체가 있고 어느 크기의 물체인지 가늠할 수 있다. 이러한 이유로 사용자는 시야에 소리가 나지 않을 때까지 물리적으로 방향을 설정하고 소리가 나는 경우 피해서 보행 할 수 있다.



그림 6 어플을 통한 오토바이 인식



그림 7 어플을 통한 자동차 인식

3.프로젝트 추진 내용

우리 조는 주제를 두 번이나 바꾸게 되어서 기존에 진행하였던 부분들은 최종 발표에서만 발표하고 최종 보고서에는 최종 결과만 작성하겠다. 기존의 진행했던 부분들은 1-15주차 보고서를 확인하면 된다.

다음의 표는 보행 보조 프로그램인 CANE을 만드는데 있어서의 7주차부터 16주차까지의 수행과정을 차트로 나타내 보았다.

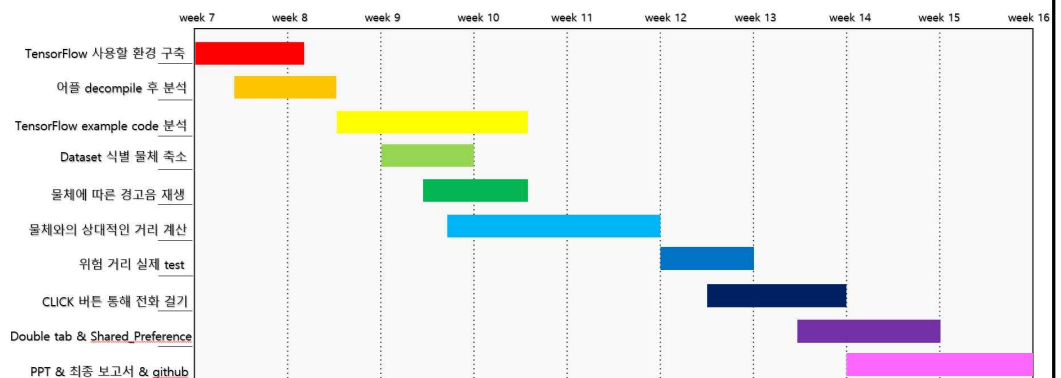


표 1 7-16주차까지의 수행과정 차트로 작성

두 번의 주제 선정과정에 이어서 노트북으로 영상을 받아와서 Tensorflow를 활용해왔던 프로그램을 어플리케이션으로 다시 구현하는 과정으로 인해 굉장히 시간이 부

족했다. 또한 어플리케이션 개발 이전에 Tensorflow식별 대상의 상대적인 거리 계산식과 경고음 발생 부분을 어플리케이션으로 만드는 데에 적극 활용 하였다. 추가로 기존에 없었던 더블 탭과 전화번호를 한번 입력받고 계속 해서 사용할 수 있는 기능을 추가해준 뒤에 짐벌 셀카봉을 사용하여 스마트폰의 후면 카메라를 최대한 시각장애인의 시야에서 볼 수 있도록 설정하려고 노력했으며 어플리케이션을 설치한 뒤 실제로 테스트를 지속적으로 하여 값들을 수정하고 반영시켜 최종 결과물을 완성하였다.

개념설계

“시각장애인들을 위한 보행보조 프로그램”이라는 주제를 실제로 구현하기 위해 우선적으로 지팡이를 활용한 시각장애인의 보행에서 안내견과 사회복지사를 제외하였을 경우에 어떤 기능이 필요한지, 어떤 오픈 소스를 사용해야 할지, 프로그램을 개발하기 위해 어떤 언어를 사용해야 할지, 어떤 개발 환경에서 개발해야 할지를 결정하기 위해 고민하였고 흐름도를 작성하여 우리가 개발할 프로그램의 전체적인 구조를 설계하였다.

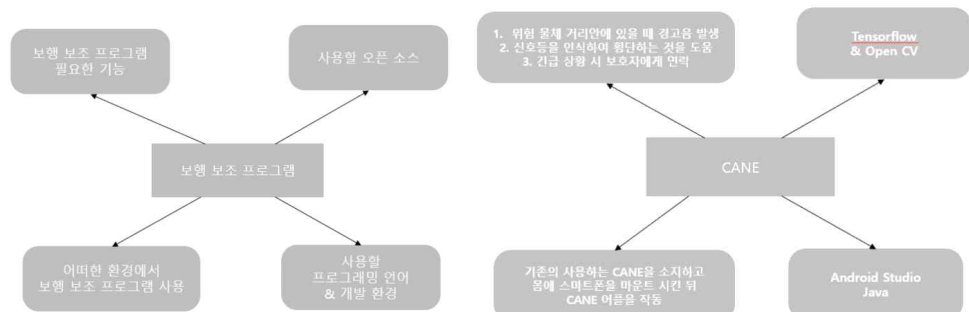


그림 8 개념설계에서의 흐름도

그림 8 흐름도에 대한 결과

상세설계

시각장애인들을 위한 보행보조 프로그램을 제작하기 위해서는 여러 가지 문제점들이 발생하였는데, 대표적으로 두 가지를 꼽자면 다음과 같다.

1. 시각장애인의 실제 도보 보행 어플리케이션을 사용 시 즉각적으로 반응을 해야 한다는 점과 영상을 촬영할 때 시야가 흔들리는 것을 방지하는 장비

2. 영상에서 인식한 물체가 실제로 어떤 경우에 위험한지를 판단

첫 번째 문제점인 즉각적인 반응을 해야 한다는 점과 장비에 대한 문제를 해결하기 위해 기존의 노트북으로 영상을 받아와서 인식하는 방법이 아닌 대부분의 사람들이 소지하고 있는 스마트폰의 카메라 기능을 통해 이를 해결하였고 또한 시야가 흔들릴 수 있다는 장비에 대해서는 몸에 마운트 시키는 장비들을 인터넷에서 구매하면 되지만 가격이 비싸 우리는 짐벌 셀카봉을 몸에 마운트 시켰다고 가정하고 사용하였다. 또한 어플리케이션으로 제작하여 스마트폰의 카메라를 통해 영상을 받아온 뒤 즉시 연산하여 반응할 수 있도록 하기 위해서 android studio를 개발환경으로 선택하였다. 두 번째 문제점인 영상에서 얻어온 물체가 위험한 경우를 판단하기 위해, TensorflowLite의 Object Detection 예제 코드를 참고하고 상대적인 거리를 계산하는 방법을 고안해

내어 물체가 시각 장애인의 보행에 위험한 물체인지를 판단하고 경고음을 내주게 하였다. 다음의 순서도는 상세 설계과정을 나타낸 순서도이다.

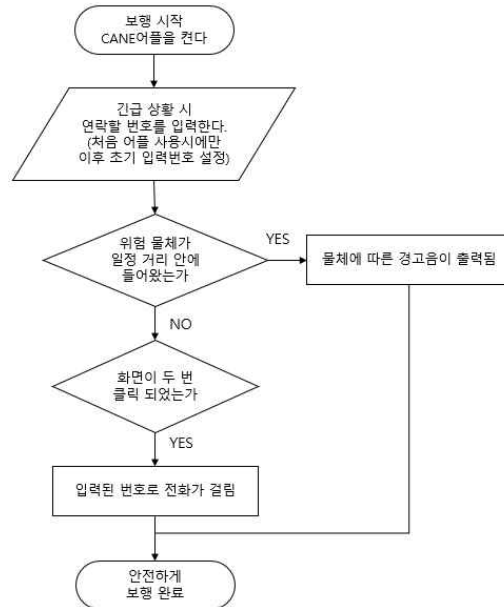


그림 9 상세 설계를 위한 순서도



그림 10 기존의 TensorFlowLite
어플리케이션 실행 결과

구현 (1, 2, 3 존재)

1. 스마트폰 영상에서 불러온 영상의 물체인식

보행 보조 어플리케이션을 구현하기 위해 TensorFlow 공식 홈페이지에 게시되어 있는 Object Detection sample code를 직접 스마트폰에 설치하여 사용해보고 코드들을 분석하며 이 어플리케이션이 어떤 기능들을 가지고 있고, 어떻게 설계되어 있는지 확인하는 과정을 거쳤다. 그림 10과 같이 이 sample code는 구축되어있는 dataset 파일을 read 하여 dataset에 포함되어 있는 카테고리가 영상에 나타난다면, 영상의 일치율과 이름을 사각형 박스를 통해 알려주는 기능이 구현되어 있었다. 우리가 필요하지 않은 정보도 이 dataset에 들어있어, 우리는 dataset을 새로 구축하려 했으나, 이를 위해서는 시간적, 금전적으로 많은 비용이 발생하여 이미 구축되어있는 dataset을 이용하여 우리가 필요한 label들에 대해서만 인식하게 하였다.

2. 어플리케이션 사용자와의 상대적인 거리 계산 방법

물체가 멀리 있는 경우
박스의 크기 작고 일치율 낮음



물체가 가까이 있는 경우
박스의 크기 크고 일치율 높음

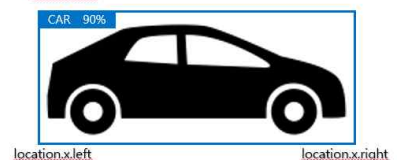


그림 11 물체와의 상대적인 거리 계산을 설명해주는 그림

차량이나 사용자의 위치를 실시간으로 GPS로 받아와서 확인하는 것이 아니기 때문에 상대적인 거리계산을 해야만 했다. 그래서 우리가 생각해낸 방법은 사물을 인식할 때 생기는 박스의 크기를 이용하는 것이다. 먼저 그림과 같이 차량이 있다고 할 때 파란색 박스가 생성되게 된다. 이때 물체의 종류와 일치율이 박스 왼쪽 상단에 표시 된다. 물체가 멀리 있다면 박스의 크기도 작고 일치율도 낮음으로 저희는 1차적으로는 일치율이 70%이상인 경우에만 박스가 생성되도록 하였다. 다음으로는 상대적인 거리를 계산할 것인데 먼저 우리는 어플리케이션에서 화면이 가로 300 세로 300의 크기 안에서 인식을 하기 때문에 박스가 생성 되어있는 경우 **그림 11**에서 표시된 `location.x.right`에서 `location.x.left`을 300으로 나눠 주었다. 나눠준 `right`에서 `left`를 빼 준 값을 `x_length`라고 하고 빼 준 결과 박스의 크기를 비율로 계산해 줄 수 있었다. 그리고 이 값을 더욱 정확하게 해주기 위해서 1에서 `x_length`를 빼고 4제곱해주었다. `1-x_length`를 통하여 물체로부터 거리가 가까우면 숫자가 점점 작아지게 하였고 제곱을 함으로써 값을 더욱 정확하게 해주었다. 다음은 2차적으로 이제 이러한 상대적인 거리를 가지고 어느 정도 거리 안에 있어야 위험한지를 직접 어플리케이션을 켜서 **그림 12, 13**과 같이 자동차와 오토바이 트럭을 기준으로 실험을 통하여 0.03라는 수치를 얻어냈다. 0.03정도의 거리는 평균적인 케인(1.2m)의 길이로 앞을 인지하면서 간다고 할 때 물체를 인식하기 전의 3보 정도의 거리이다.

따라서 1차적으로 일치율이 70%이상 그리고 2차적으로 물체가 0.03이라는 상대적인 거리 안에 들어왔을 때 경고음을 내주도록 했다.

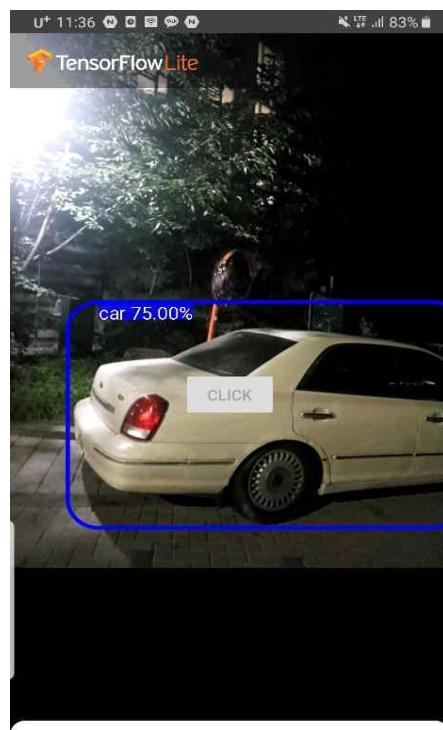


그림 12 0.03값을 얻어내는 실험과정



그림 13 0.03값을 얻어내는 실험과정

3. 어플리케이션에서 더블 탭을 통한 입력된 번호로 전화 연결

처음에는 **그림 12, 13**처럼 CLICK버튼을 구현하여 시각장애인이 전화를 할 일이 생긴

	<p>다면 CLICK 버튼을 누르게 하려고 했지만 시각장애인에게 버튼을 찾는다는 것은 현실적으로 불가능할 것이라고 생각하여 다른 방법들을 고안하다가 화면을 두 번 터치하게 하는 더블 탭하는 경우 전화를 해주는 방법을 떠올렸다. 또한 번호를 입력받는 것에 있어서 처음에는 Data Base를 사용하여 여러 개의 번호를 입력받으려고 했지만 각 번호에 대해서 어느 상황에 어떤 번호로 연락을 해야 할지에 대한 문제가 발생하여 기존의 하나의 번호를 입력받고 그 번호로 전화가 가게 하였다. 처음에 번호를 입력해주지 않는다면 default는 112로 되어있으며 한번 번호가 입력되면 다시 어플리케이션을 실행시켰을 때는 재입력하지 않고 마지막으로 입력 받은 번호로 저장이 되어 있게 하였다.</p>
4.기대효과	<p>개발 동기에서도 언급하였지만 현재 한국에는 25만 3천여 명, 서울에만 4만 3천여 명의 시각장애인이 있으며, 전체 중에서도 1,2급 시각장애인은 약 4만명이다. 그 중 안내견의 도움을 받는 시각장애인은 0.2%에도 못 미치는 수준이며, 외에 다른 사람들은 주변 동행자나 사회 복지사의 도움을 받아 생활하는 것으로 조사되었다. 이렇게 현재저하게 복지가 부족한 현실의 가장 큰 원인은 안내견을 훈련시키는 데에 필요한 경제적, 시간적인 비용 문제가 있다. 우선 미국과 일본과 달리 한국의 안내견 실정은 턱없이 부족하다. 국내에서 세계 안내견 협회에 정식 등록된 기관은 삼성화재 안내견학교가 유일하며, 1년에 약 12마리 정도만 배출되어 시각장애인과 매칭 되는데 시각장애인에 비하면 극소수에 불과하다. 안내견은 사회화 과정과 학교에서 본격적인 훈련을 받는 7~8개월 동안 체력, 품행, 공격성 등의 항목이 포함된 세 차례의 시험을 통과해야만 하는데, 이처럼 약 한 마리의 안내견이 탄생하기에는 2년간의 훈련기간이 필요하며 그마저도 합격률이 30%가 되지 않아 많은 시각장애인이 안내견의 도움을 받을 수가 없는 게 현실이다. 추가적으로 안내견 한 마리를 교육하는 데에 드는 사육비와 진료비 등은 1~2억이 소모된다. 경제적, 시간적인 측면에서 보았을 때 이러한 문제점을 우리의 프로젝트 결과물의 사용으로 개선할 수 있다. 시각장애인은 갑작스러운 사고에 대한 대처능력이 부족할 수밖에 없는 상태이다. 시각장애인을 대상으로 한 설문을 보면 그들에게 있어 가장 불편한 점은 독립적으로 보행할 때에 환경적인 영향을 받는 것으로, 보행도로에서는 자전거, 오토바이 건널목이나 차도에서는 자동차와 같은 이동수단에 대한 위험요소가 가장 컸다. 프로젝트 결과물의 도움을 받으면 이러한 위험요소에 대한 경고를 미리 받음으로써 사고예방에 도움을 줄 수 있고, 시각장애인은 휴대전화를 자유롭게 다루기가 불편하므로 두 번 터치를 통한 비상연락을 통해 위급상황 시에는 도움을 요청할 수 도 있다. 주변인의 영향을 받을 수 있는 안내견과 달리 어플리케이션의 이용은 보다 보장된 안정성을 확보할 수 있으며 경제적, 사회적 측면에서도 보다 효과적인 기대효과를 예상할 수 있다. 결론적으로 우리 프로젝트 결과물은 경제적, 사회적인 측면에서의 기대효과를 예상할 수 있다.</p>
	<p>발생 문제점 (1-7)</p> <p>1) 어플리케이션 전체적인 구상에서의 문제 발생</p> <p>모바일 환경에서 어플리케이션의 전체적인 디자인과 Tensorflow를 활용한 전반적 구성에 있어서 프로젝트 기간적 제한에 따른 구현 현실성에 대한 문제가 발생하였다.</p>

2) 기존 Tensorflow어플 de-compile을 통한 코드 분석에서의 문제 발생

현존하는 이미 개발되어있는 Tensorflow를 활용한 여러 어플리케이션을 디컴파일 하여 코드를 분석하고 프로젝트 구현 시 필요한 정보를 사용하려고 시도하였으나 전반적으로 모든 어플리케이션은 코드가 난독화 되어있거나 개발자가 아닌 이상 해석이 불가능한 점들이 많아서 필요 부분을 해석하거나 수정할 수 없는 상황이 발생하였다.

3) tesnsorflow example code사용 시 모든 dataset 물체 인식의 문제 발생

Tensorflowlite 사이트 제공 example code를 사용하여 물체 인식을 해보았을 때 이미 주어진 dataset에 들어있는 모든 물체를 인식하여 박스로 표시해주는데 이는 불필요한 물체의 인식으로 인해 개발 목적에 맞지 않는 기능으로, dataset의 수정을 통해 필요한 물체만 인식하도록 수정이 필요하였다.

4) 추가 기능인 비상연락 구현 시 버튼 클릭 형 기능의 문제 발생

위험 물체를 인식하여 경고음을 출력해 주는 기능 외에도 시각장애인에게 도움을 제공할 수 있는 다른 기능을 구현함에 있어서 버튼을 누르면 저장되어 있는 번호로 전화 발신을 가능하게 해주는 기능을 생각하였으나, 시각장애인이 실제로 어플리케이션 내에서 버튼을 찾고 클릭을 하는 것은 비현실적인 문제였다.

5) 비상연락처 등록에서의 문제 발생

추가 기능인 비상연락 구현에 있어서 우선적으로 고려되어야 할 사항이 비상연락처의 등록인데, 시각장애인은 직접 비상연락처를 등록할 방법이 없다는 문제가 발생하였다.

6) 시각장애인 대상 프로그램으로 영상을 찍는 기기 선정문제 발생

프로그램을 사용할 때 영상을 찍는 기기의 종류와 위치에 대한 문제가 고려되어야 하는 상황이 발생하였는데, 이는 시각장애인이 실제로 프로그램을 사용할 때에 신체의 어느 위치에서 어떤 기기를 사용해야 적합한지에 대한 문제이다.

7) 사용 대상의 보행 상황 및 시야의 무일관성에 대한 고려 문제 발생

프로그램 사용 대상인 시각장애인의 보행 상황은 예측 불가하며 동선에 따라서 시야가 일관성이 없으며 흔들림이 발생할 수 있다. 이러한 상황을 모두 고려해보면 영상을 판단하기가 어려워 어플리케이션 구동에 있어 문제가 발생한다.

문제점 해결

1) 어플리케이션 전체적인 구상에서의 문제 발생

모든 사항을 개발하기에는 시간적인 제한이 있다고 판단하여 기존에 널리 구현되어 있는 Tensorflow를 활용한 어플리케이션을 직접 사용함으로 문제를 해결하였다.

2) 기존 tensorflow어플 de-compile을 통한 코드 분석에서의 문제 발생

github과 구글 검색을 통하여 해결책을 찾으려 하였고, tensorflowlite 사이트에서 자체적으로 제공해주는 example code를 사용하여 어플리케이션 디컴파일을 통한 분석이 아닌 기존의 코드를 분석하고 수정하는 방안을 택하여 문제를 해결하였다.

3) Tesnsorflow example code사용 시 모든 dataset 물체 인식의 문제 발생

dataset에 포함되어 있는 물체들의 목록에서 프로젝트 진행에 있어 목적에 맞는 물체들만 인식하도록 하기 위해 불필요한 data들은 모두 제외하였다. 현실성 없는 동물들부터 보행 상황에 있어 위험요소라고 판단되지 않는 사항들이 제외의 대상이었으며 dataset 수정 결과 어플리케이션 목적에 맞게 필요한 물체만 인식하도록 구현을 성공하였다.

4) 추가 기능인 비상연락 구현 시 버튼 클릭 형 기능의 문제 발생

시각장애인이 시각적으로 버튼을 찾아서 누르는 것은 불가능한 비현실적 문제이기 때문에 대신 화면을 터치하는 대책 방안을 생각하였고, 한 번의 터치는 비상상황이 아님에도 잘못된 터치로 인해 전화가 걸릴 수 있다고 판단하여 두 번의 터치를 통해 비상연락 기능이 구동되도록 하였다.

5) 비상연락처 등록에서의 문제 발생

비상연락처를 시각장애인이 직접 등록할 수 없기에 우선 디폴트값으로 경찰서의 번호인 112를 저장하였고, 비상연락처의 수정이 필요할 경우에는 주변 지인이나 가족의 도움을 받을 수 있다고 가정하여 번호를 수정할 수 있도록 구현하였다.

6) 시각장애인 대상 프로그램으로 영상을 찍는 기기 선정문제 발생

영상을 기기로 촬영하고 실시간으로 노트북으로 받아서 프로그램을 구동하려는 점에서 실시간 영상 전송이 가능한 프로그램을 찾기도 어려웠고, 널리 사용되는 Team Viewer와 같은 프로그램들도 영상 송출 지연 문제가 발생하여 현실적으로 많은 문제가 있다고 판단하였으며 어플리케이션의 개발로 방향을 잡은 시점에서 어플리케이션의 사용이 가능하고 선명한 영상을 통해 전방을 인식해야한다는 점 등을 고려하였을 때, 스마트폰이 가장 적합하다고 판단하여 스마트폰을 영상촬영기기로 선정하였다. 신체 부위에 장착이 가능할 정도의 휴대성이 보장되며, 전화가 가능하다는 점에서 효율적이다.

7) 사용 대상의 보행 상황 및 시야의 무일관성에 대한 고려 문제 발생

보행 상황에 따른 좌우 흔들림을 방지하고 전방 물체를 효과적으로 인식하기 위한 영상을 위한 고려를 했고, 자이로스코프를 사용하여 정립 상태로 스마트폰을 유지해주는 보조 장치인 짐벌의 사용을 결정하였다. 이로 보행 상황에 따른 영향을 최소화하여 안정적인 영상을 통해 어플리케이션 사용 효과를 극대화 하였다.

6.참고문헌

<https://github.com/sentdex/pygta5> (Object-detection GTA5에서 사용방법)
<https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android>
(TensorFlowLite example code 올라온 github)
<http://android-journey.blogspot.com/2010/01/android-gestures.html>
(double tap 활용하는 방법)
<https://blog.naver.com/eo930827/221499624304>
(Shared_Preference 활용하는 방법)