



국민대학교
전자정보통신대학
컴퓨터공학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	Be Eyes
팀 명	ViewTopia
문서 제목	중간보고서

Version	1.5
Date	2019-04-18

팀원	박 병훈 (조장)
	김 상민
	안 은영
	서 준교
	이 옥걸
지도교수	이 시윤 교수

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18


CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 "BeEyes"를 수행하는 팀 "ViewTopia"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "ViewTopia"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


Filename	중간보고서-BeEyes.doc
원안작성자	안은영
수정작업자	박병훈, 김상민, 서준교

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-04-09	안은영	1.0	최초 작성	프로젝트 목표, 계획서 상의 연구 내용 추가
2019-04-15	김상민	1.1	내용 추가	수행내용, 수정된 연구내용 및 추진방향 추가
2019-04-15	서준교	1.2	내용 추가	향후 추진 계획, 고충 및 건의사항 추가
2019-04-16	박병훈	1.3	내용 추가	개발 목표 구체화 및 연구 개발 내용 추가 작성
2019-04-17	박병훈	1.4	내용 수정	수행 내용 수정 및 향후 추진계획 수정
2019-04-18	서준교	1.5	내용 수정	계획서 상의 연구내용 수정 및 향후 추진계획 추가

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

목 차

1	프로젝트 목표	4
2	수행 내용 및 중간결과	4
2.1	계획서 상의 연구내용	4
2.1.1	데이터 훈련 및 인식	4
2.1.2	TTS (Text-To-Speech) 기능	5
2.1.3	OpenCV 를 이용한 손가락 인식 및 ITT (Image-To-Text)	5
2.2	수행내용	5
2.2.1	데이터 훈련 및 인식	5
2.2.2	TTS(Text-to-Speech) 기능	10
2.2.3	OpenCV 를 이용한 손가락 인식 및 ITT(Image-To-Text) 기능	10
2.2.4	계획서 진도 비교 분석	11
3	수정된 연구내용 및 추진 방향	12
3.1	수정사항	12
3.1.1	기술적 요소 변경	12
4	향후 추진계획	15
4.1	향후 계획의 세부 내용	15
5	고충 및 건의사항	17

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

1 프로젝트 목표

본 프로젝트는 시각장애인들이 안전하게 일상생활을 할 수 있도록 주변의 물체 및 텍스트를 인식하여 음성으로 전달해주는 스마트 안경을 개발하는 것을 목표로 한다.

- 1) 시각장애인 전방의 장애물의 정보를 알려준다. 장애물의 물체 정보를 인식하므로 시각 장애인은 상황에 따른 구체적인 대처가 가능하다. 예: 볼라드, 계단, 벽돌 등
- 2) 시각 장애인이 손가락으로 가리키는 영역의 문구를 음성으로 전달한다. 텍스트의 정보를 인식한 시각장애인은 안전한 약 복용 및 음식 섭취가 가능하다. 예: 약 표시 정보, 식료품 (우유, 과자봉지) 등


위 2 가지 기능을 통하여 시각장애인들의 안전한 일상생활에 도움이 되고자 한다.

2 수행 내용 및 중간결과

2.1 계획서 상의 연구내용

2.1.1 데이터 훈련 및 인식

본 프로젝트는 뉴럴 네트워크 프레임워크인 darknet 의 신경망 모델 중 하나인 YOLO 를 이용한다. YOLO 는 실시간으로 객체를 탐지하는데 유용한 신경망이다. 'Be Eyes'는 시각장애인이 일상생활을 하며 즉각적으로 마주칠 장애물들에 대한 정보를 실시간으로 알려줘야 하기에 오브젝트 디텍션 모델로 YOLO 를 선택했다. YOLO 를 이용해 데이터 전처리 및 학습을 실행한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

2.1.2 TTS (Text-To-Speech) 기능

학습한 장애물이 인식되면 input 파일 내에 인식한 장애물의 이름을 적는다. 이 때 TTS 기능이 들어간 파이썬 파일이 이를 확인하고 음성으로 객체의 정보를 전달한다.

2.1.3 OpenCV 를 이용한 손가락 인식 및 ITT (Image-To-Text)

손가락 모양의 데이터를 미리 학습시킨다. 영상 처리를 통해 카메라가 손가락을 인식했을 때, 이미지를 저장한다. 이미지에서 손가락의 끝을 기준으로 boundary 를 잡아 Cloud Vision API 를 이용해 텍스트로 변환한다. 그 후, 텍스트의 내용을 구글에서 지원하는 Text To Speech API 를 이용해 음성으로 말해준다. 모션 인식 기능은 시각장애인들의 주된 정보 입력 방식을 '음성'에서 '동작'으로 바꾼 것에 의의가 있다.

2.2 수행내용

2.2.1 데이터 훈련 및 인식

프로젝트에서 사용할 모델은 TINY YOLO V3 이다. 일반적인 YOLO 모델이 요구하는 computing power 가 프로젝트 구현에 사용할 라즈베리파이의 성능보다 훨씬 높기 때문에 요구 성능도 낮고 용량도 훨씬 적은 TINY YOLO V3 모델이 적합할 것이라 판단되어 해당 모델을 채택하였다. (참고. YOLO V3 모델은 최소 4GB 의 RAM 을 요구하고, TINY YOLO V3 모델은 1GB 의 RAM 을 요구한다.)

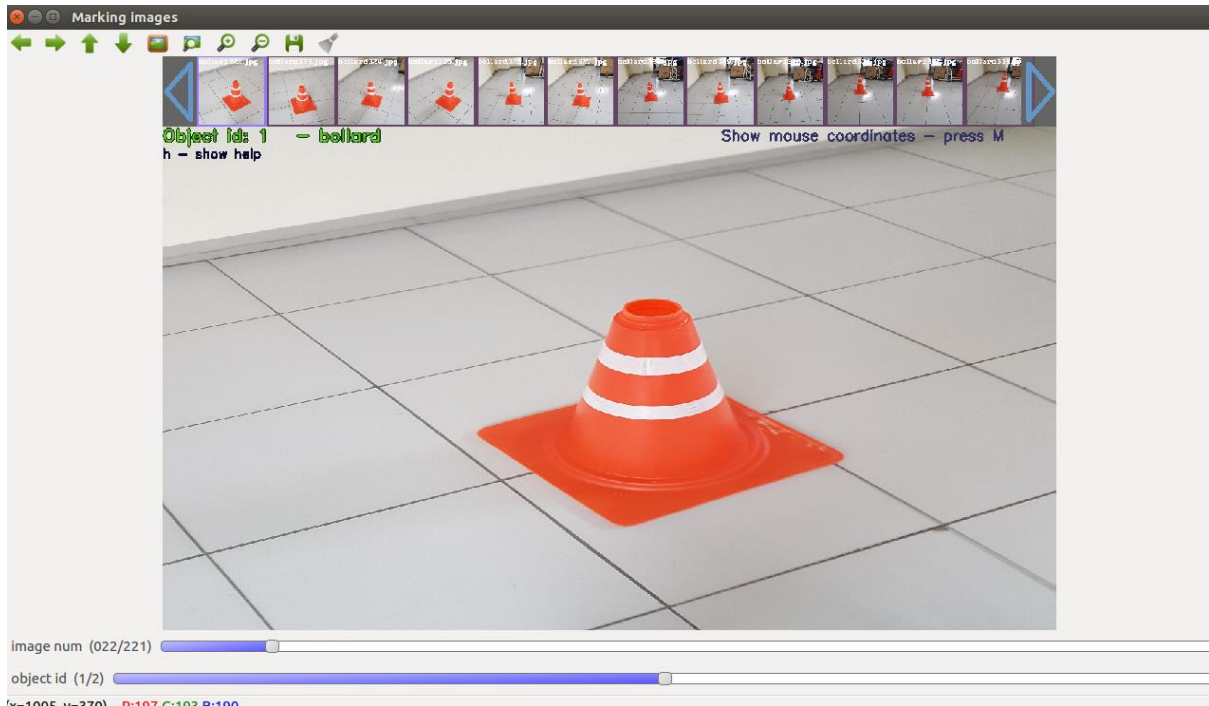
YOLO 모델에 이미지 데이터를 학습하는 과정에 시각 장애인이 일상생활을 하며 장애가 될 만한 요소들을 선별한다. 이 때, 프로젝트 실제 구현 환경의 제약에 따라 장애물과 유사한 물체들로 대체한다. (예: 볼라드 -> 라바콘) 또한 실제 시각 장애인의 눈높이를 고려하여 다양한 각도에서 이미지를 수집하였다.

이미지 수집 후, 모든 데이터의 확장자를 jpg 로 수정 후 Yolo mark 라는 프로그램을 이용하여 바운딩 박스를 직접 그려 넣어 라벨링 작업을 한다. 라벨링 작업이 완료되면 annotation file 이 생성되고 각 이미지 파일에 대한 라벨링된 id 와 바운딩 박스의 좌표값이 텍스트파일 형식으로 생성된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

생성되는 텍스트파일의 내용은 다음과 같다.

```
<object-class> <x_center> <y_center> <width> <height>
```

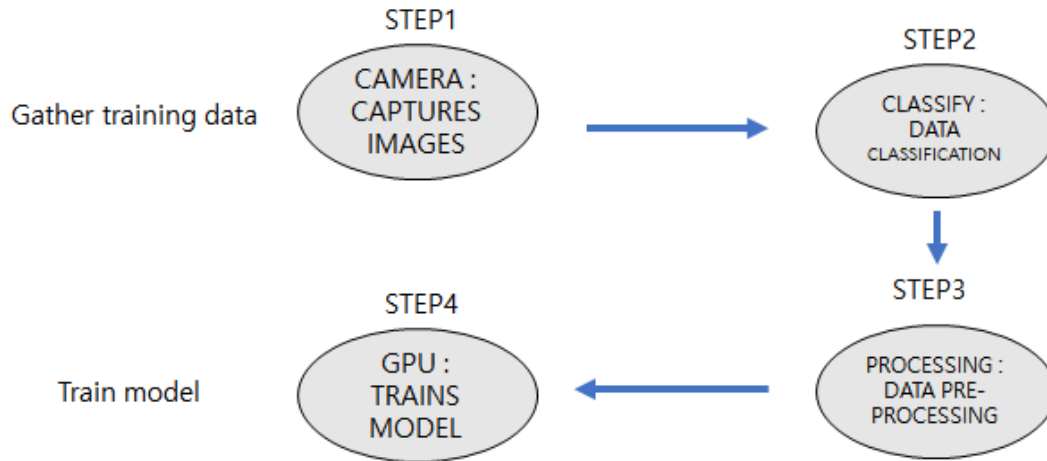


<그림> Yolo mark 를 통해 학습시킬 이미지 데이터를 라벨링하는 과정

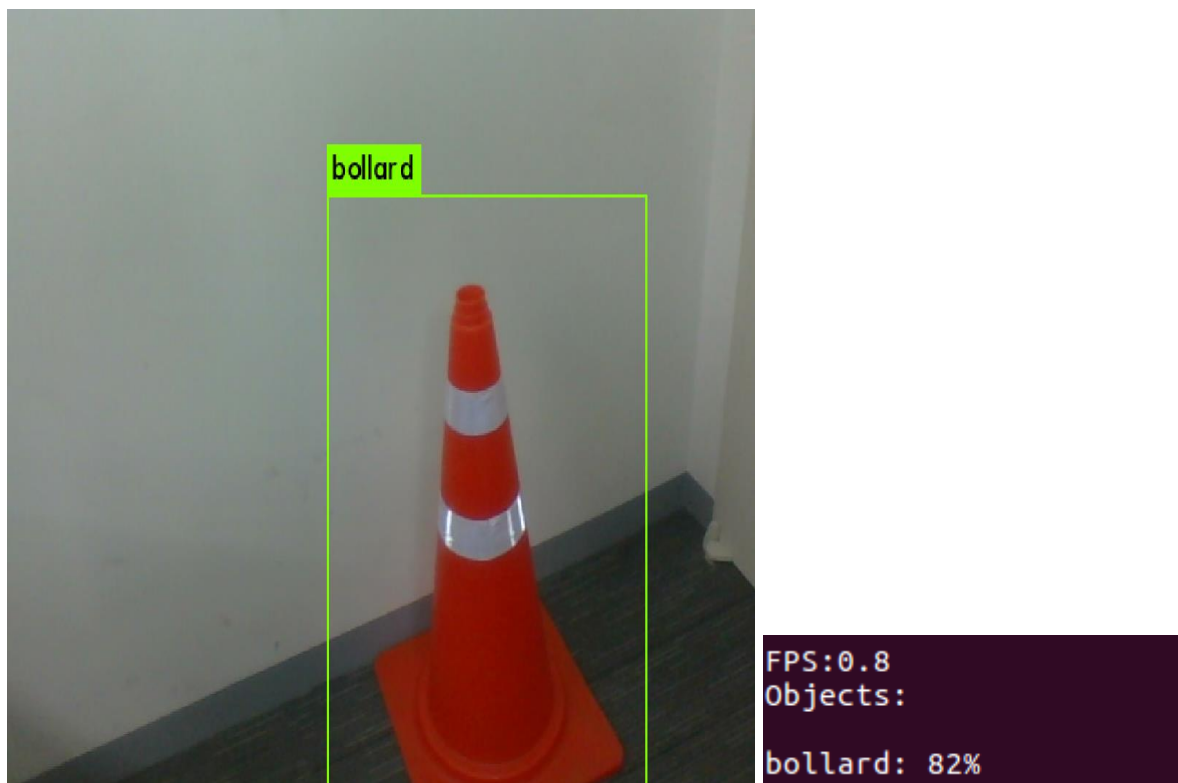
데이터 학습을 위해 충분한 수의 이미지를 수집하고 (예: 라바콘 400 장) GTX1080 8GB 환경에서 몇 일간 학습을 진행한다. 실제로 10 만 번의 학습을 진행하는 동안 약 10 시간이 소요되었다. 학습이 끝난 후 생긴 weight 파일을 이용하여 객체 탐지를 진행한다. 이 때, 생성된 여러 가중치 파일을 비교하여 가장 높은 인식률을 갖는 가중치 파일을 채택하였다.


현재 10 만번의 학습 결과, 오차율 4% 이내에서 웹캠을 통해 볼라드(라바콘) 인식이 가능하다. 추가적으로 다른 장애물의 이미지 수집 및 학습을 진행한다.

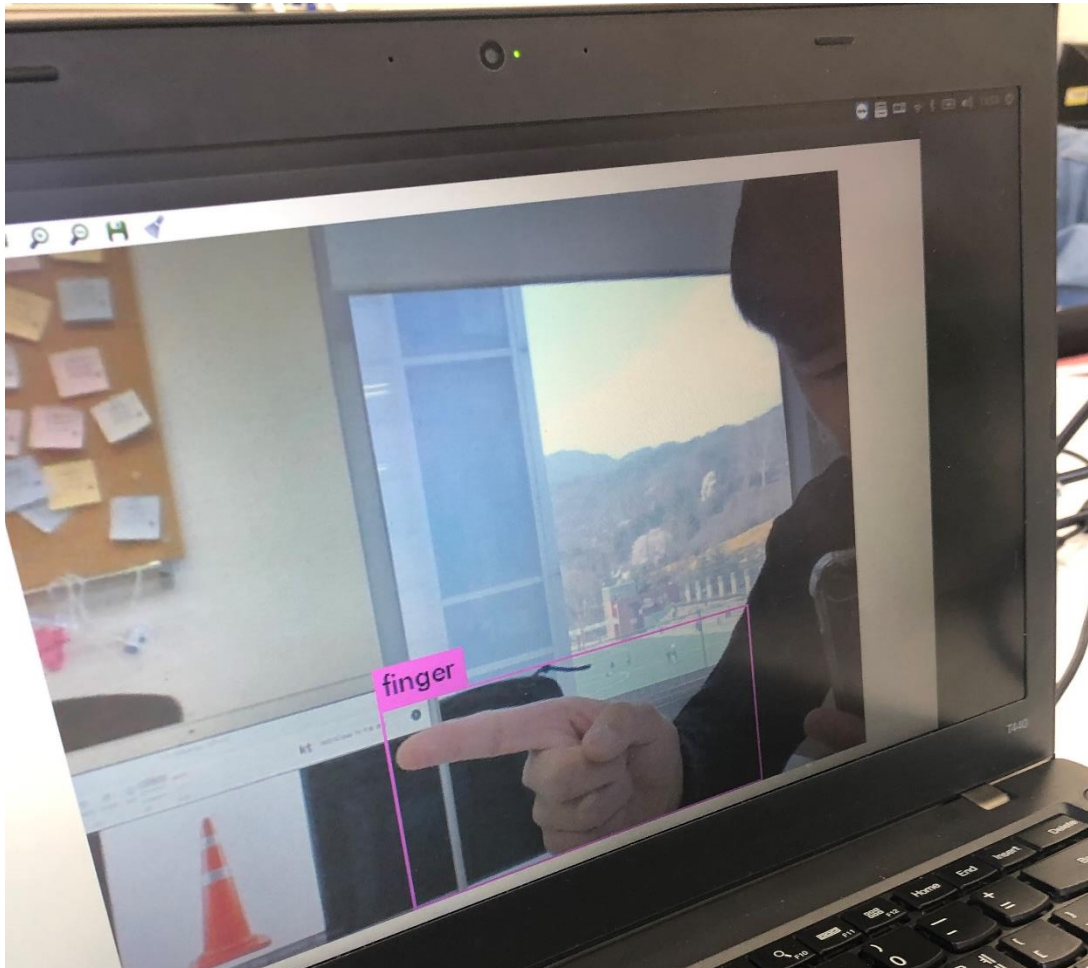
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18



<그림> 데이터 수집 및 학습 흐름도



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18



<그림> Ubuntu 16.04-LTS 환경 Webcam, 학습 데이터 인식 및 인식을

901: 0.814945, 0.814945 avg, 0.000659 rate, 0.860667 seconds, 21624 images

10001: 0.151525, 0.151525 avg, 0.001000 rate, 3.888302 seconds, 240024 images

100001: 0.025104, 0.025104 avg, 0.001000 rate, 1.340743 seconds, 2400024 images

150005: 0.030136, 0.030757 avg, 0.001000 rate, 0.346696 seconds, 3600120 images

<그림> 각각 900 번 학습 후, 만 번 학습 후, 10 만 번 학습 후, 15 만 번 학습 후의 손실 함수값




$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

<그림> YOLO 에 사용되는 손실 함수¹

머신 러닝에서는 신경망의 성능을 나타내는 지표로 손실 함수라는 개념을 사용한다. 손실 함수란, 현재 신경망이 훈련 데이터에 대해 얼마나 잘 처리하지 못하느냐를 보여준다. 학습은 2 일간 15 만 번까지 진행하였다. 위 그림에서 900 번 학습 당시의 손실 함수값은 약 0.815 이었지만, 학습을 시키는 과정에서 지속적으로 감소하는 모습을 보였고, 10 만 번 학습하였을 때 약 0.025 까지 값이 하락한 것을 확인할 수 있었다. 하지만 15 만 번 학습하였을 때에는 손실 함수값이 되려 늘어나는 overfitting 현상이 발생한 것으로 추정되었다. 이후에도 손실 함수값은 크게 변하지 않았다. 따라서 10 만 번 학습시켰을 때의 가중치 파일이 프로젝트에 사용하기 가장 적합하다고 판단되어 해당 weight 를 채택하였다.

사진에서 라즈베리파이가 아닌 ubuntu 에서 실행한 이유는 라즈베리파이 내에선 computing power 가 약해 YOLO 가 실행되지 않았기 때문이다. 따라서 밑의 수정된 연구내용에 해결방법을 기술하였다.

¹ YOLO 에 사용되는 손실 함수, <http://research.sualab.com/practice/2018/05/14/image-detection-deep-learning.html>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

2.2.2 TTS(Text-to-Speech) 기능

시각장애인에게 음성으로 장애물의 정보를 전달한다. 음성을 출력하기 위해서는 텍스트를 음성으로 변환하는 작업이 요구된다. 따라서 Google Cloud Platform 의 Text-To-Speech API 를 사용한다. 전체 흐름은 카메라에서 객체를 인식하면 이 객체에 대한 메시지를 input 파일에 입력하고 입력된 텍스트를 음성으로 변환한다.

[환경설정]


- 먼저, Google API 를 사용하기 위한 Google Cloud Platform Project 를 생성한다.
- TTS 기능을 사용하기 위해서는 Key 역할을 하는 JSON 파일이 필요하다. 구글 콘솔에서 프로젝트를 생성하면서 이 파일을 얻을 수 있다.
- 문서를 참고하여 Python 코드로 작성하고 코드 안에서 JSON 파일의 경로 및 이름을 설정한다. TTS 코드를 실행하면 텍스트를 음성으로 변환한 mp3 파일을 생성해준다.
- mp3 파일을 자동으로 재생해주는 pygame 모듈을 사용하여 변환된 파일을 재생한다.

현재 영상 처리를 통해 인식한 장애물의 이름이 input 파일에 적히면 음성으로 전달해주는 기능이 구현되어 있다.

2.2.3 OpenCV 를 이용한 손가락 인식 및 ITT(Image-To-Text) 기능

다양한 손가락 이미지를 수집하고 학습하여 카메라가 손가락을 인식하면 화면을 스크린샷으로 캡처하여 이미지 속의 글자를 추출하는 작업을 진행한다. input 파일에 'finger'가 적히면 실행중인 python 파일이 이를 확인하고 저장된 이미지 파일을 분석해 Google Cloud Platform 의 Vision API 를 활용한 Image-To-Text 기능을 사용한다. 그 후 추출한 텍스트를 음성으로 전달한다. 이를 통해 시각장애인이 손가락이 가리키는 이미지 내의 텍스트의 내용 전달이 가능하다.

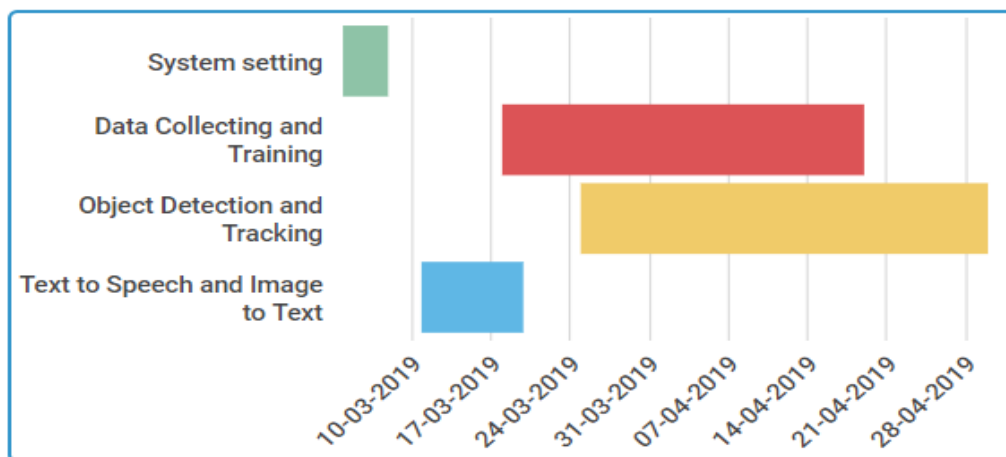
현재 손가락을 대신할 기존 YOLO 모델에서 제공하는 인식 가능한 클래스를 임의로 하나 정하고

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

텍스트가 적힌 이미지를 가리켰을 때, 해당 이미지가 캡처되고 음성으로 전달이 가능한 상황이다. 손가락 이외에 피부색이 유사한 물체들도 손가락으로 인식하기 때문에 추가적인 데이터 수집과 학습이 필요하다.

2.2.4 계획서 진도 비교 분석

'Be Eyes' Project Plan



System setting : ~ 2019.03.08


Data Collecting and Training : 2019.03.08 ~ 2019.04.19

Object Detection and Tracking : 2019.03.25 ~ 2019.04.30

Text to Speech and Image to Text : 2019.03.11 ~ 2019.03.20

<그림> 수행계획서 예측 사항

현재 시스템 세팅과 TTS, ITT 기능은 구현한 상태이다. 데이터 수집 및 훈련은 YOLO 모델이 요구하는 computing power 가 높은 환경을 찾는데 많은 고충 사항이 있었기에 진행 중이다. 현재는 블라드(라바콘)만 인식 및 추적이 가능한 상태이고 추가적인 이미지 수집 및 학습을 진행하여 최종 발표 전까지 완성할 계획이다. 데이터를 수집하고 학습하는 과정이 완료되면 객체 탐지 또한 완료된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

3 수정된 연구내용 및 추진 방향

3.1 수정사항

3.1.1 기술적 요소 변경


1. 하드웨어 성능의 한계 개선

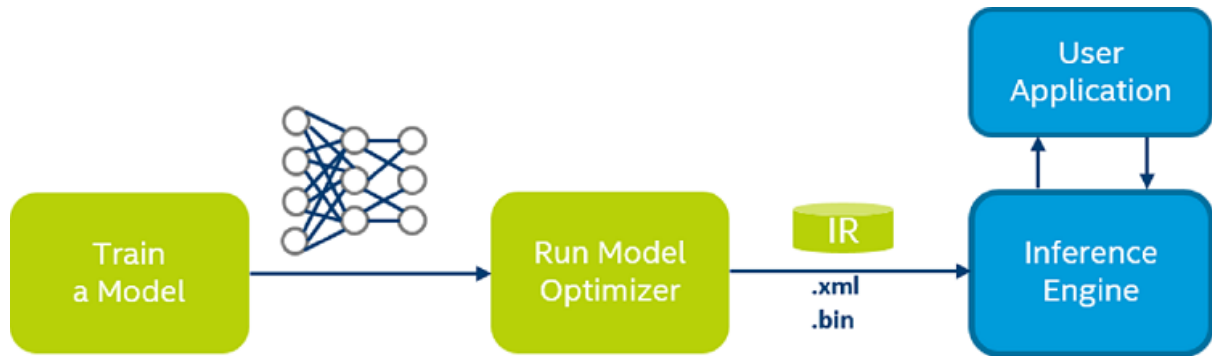
본 프로젝트에서는 대표적인 단일 단계 방식의 객체 탐지 모델인 YOLO 를 통해 객체 인식 및 추적을 구현한다. 기존에는 darknet 프레임워크를 통해 기능 구현을 진행했다. 그러나 라즈베리파이의 환경에서 darknet 은 전력 소모가 크고 딥러닝 모델을 정상적으로 실행하지 못하였다.



<그림> USB 형 딥러닝 장치, Neural Compute Stick 2

따라서 위 그림과 같이 Movidius™ VPU(비주얼 처리 장치)에 의해 저전력, 고성능으로 구동되는 딥러닝용 USB 드라이브인 Intel® Movidius™ NCS2(Neural Compute Stick 2)를 사용한다. 이 USB 는 개방형 시각 추론 및 신경망 최적화 도구 모음인 OpenVINO 라는 Toolkit 을 제공한다. 그러므로 데이터 학습은 darknet 에서 진행하고 학습된 모델을 OpenVINO 도구를 통해 구동한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18



<그림> 라즈베리파이에서 객체 탐지가 가능한 파일 변환 과정

모델의 학습은 성능이 좋은 데스크탑(GPU : GTX1080 8GB)에서 수행한다. darknet 에서 제공하는 학습 도구를 사용하여 학습 모델 파일(weight)을 생성한다. 여기서 학습된 모델 파일은 NCS2 USB 를 사용할 수 있는 OpenVINO 도구에서 실행되지 않는다. 따라서 위의 그림과 같이 OpenVINO 에서 사용할 수 있도록 최적화된 학습 모델로 변환하는 작업이 필요하다. Model Optimizer 란 최적의 실행이 가능하도록 딥러닝 모델을 자동으로 조정해주는 OpenVINO 의 cross-platform 명령 도구이다. 이 과정을 통해 학습한 모델을 Intermediate Representation(IR)인 xml 과 bin 파일로 변환한다. 이렇게 변환한 모델 파일을 사용하여 라즈베리파이 환경에서 OpenVINO 를 활용한 실질적인 YOLO 알고리즘 구현이 가능하다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18



<그림> 라즈베리파이 + Neural Compute Stick 2, 학습 데이터 인식

FPS(Frame Per Second, 초당 프레임)는 초당 처리할 수 있는 프레임 수를 나타내며 이 수치가 높을수록 물체를 인식하는 속도가 향상된다. 위 그림과 같이 학습한 볼라드 데이터를 약 3~5 FPS 로 인식한다. 라즈베리파이에서 darknet 을 사용했을 때는 화면이 멈추는 오류 현상 및 전력 부족 표시가 나타나면서 전혀 실행이 되지 않았다. 또한 ubuntu 16.04 LTS 환경에서 darknet 을 사용했을 때는 FPS 의 범위가 0.6~0.8 이었는데 이와 비교했을 때 약 5~7 배 정도의 성능이 향상된 것을 볼 수 있다.

2. 실시간 객체 탐지 모델 YOLO 의 방향성 구체화 및 손가락 인식 방법 변경

본 프로젝트의 목표 중 하나는 글자를 읽지 못하는 시각 장애인에게 텍스트 정보를 음성으로 전달해주는 것이다. 따라서 추가적으로 데이터 학습이 가능한 YOLO 모델에 finger 클래스를 만든 후 데이터를 수집하고 학습하여 손가락을 인식하게 하려했다. 하지만 YOLO 모델 사용 목적은 실시간 객체 탐지를 위한 것인데 손가락을 인식하여 텍스트를 읽는 것은 실시간일 필요가 없으므로 적합한 방법이라고 생각하지 않았다. 따라서 OpenCV 를 이용한 손가락 인식 알고리즘을 사용하기로 결정하였다. 손가락이 가리키는 끝점을 구하여 그 점으로부터 boundary box 를 만들고 그 영역 내의 텍스트를 읽기로 결정하였다. 향후 알고리즘 구현에 대한 추가적인 정보 수집이 요구된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

4 향후 추진계획

4.1 향후 계획의 세부 내용

1. 데모 환경 구성

최종 발표 시, 프로젝트를 시연할 환경의 제약이 존재하므로 실제 장애물과 유사한 물체로 대신한다. 따라서 데이터를 실제 최종 발표에서 보일 대체 물품들로 학습할 계획이다. 데모 시나리오를 고려하여 블라드는 라바콘으로 대체, 계단과 보도블럭은 직접 모형으로 제작한다.

2. 학습 모델의 확장

시각장애인들이 일상 생활에서 자주 마주치게 되는 장애물들을 선별하고 데이터를 모아 추가적으로 학습시킬 예정이다. 제안서에서 블라드(라바콘)만을 학습할 장애물의 항목이라고 기술하였다. 하지만 시각 장애인이 현실에서 마주치게 될 다양한 장애물들을 고려하고 완성도 있는 데모 환경을 구성하기 위해 장애물로 판단될 만한 요소를 추가하여 보다 구체적으로 프로젝트를 진행할 예정이다. 추가적인 데이터의 종류로는 계단, 보도블럭 등이 있다.

3. 데이터 인식을 향상 메커니즘 구현

외부에서 인식한 이미지 데이터를 일정 주기로 메모리카드에 있는 데이터를 서버에 전송하여 해당 이미지를 모델에 다시 학습시킴으로써 훈련 데이터에 없는 이미지 데이터에 대한 인식률을 높이는 방식을 구현할 예정이다. 특정한 인식률을 threshold로 설정하고, 해당 threshold를 넘는 이미지를 저장한 뒤, YOLO 모델에 해당 이미지를 다시 학습시키는 것이다. 해당 인식 과정을 거치면, 특정 클래스에 대한 이미지의 인식률이 높아질 뿐만 아니라 클래스와 유사한 이미지에 대해서도 적응적으로 학습하여 인식률을 높이는 효과를 기대할 수 있다.

4. 프로젝트 최적화

학습된 여러 장애물 데이터가 동시에 인식되면 일방향적으로 한 물체만 음성으로 전달하는 경우가 발생한다. 따라서 이를 해결하기 위해, 정해진 시간 내에 탐지했던 물체가 다시 나타나면 음성을 내지 않고 화면에 처음 잡힌 다른 객체 정보를 전달하는 방식으로 구현할 것이다. 프로그램의 오류를 줄이고 시각장애인에게 혼동이 될 만한 정보를 줄여 정확한 정보전달이 가능하도록 할 것이다.

5. 프로젝트 환경 개선

본 프로젝트에서 사용하는 라즈베리파이와 NCS2 USB 는 실시간으로 객체를 인식하는데 원하는 효율을 제공하지 못한다. 아래 자료는 YOLO tiny 모델을 사용하여 Movidius Myriad NCS2 와 NVIDIA P100 GPU 의 성능을 테스트하여 비교한 자료이다.


Table 1. Performance results

	1 x Movidius Myriad 2	2 x Movidius Myriad 2	1 x NVIDIA P100 GPU
Time [s]	123.1	69.8	23.3
Average FPS factor	4.05	7.16	21.3

<그림> Movidius Myriad2 와 NVIDIA P100 GPU 의 성능 차이 ²

또한 NVIDIA 의 작은 디바이스 크기를 가진 Jetson Nano 는 본 프로젝트에서 사용하고 있는 TINY YOLO V3 모델을 darknet 프레임워크를 사용하여 25FPS 로 지원한다. 현재 프로젝트 구현 사항과 비교하여 약 6 배 정도의 좋은 성능을 보인다. 이 모듈은 가격, 성능 등 다양한 측면에서 라즈베리파이 + NCS2 보다 이 프로젝트의 목표와 부합하고 프로젝트를 진행하는 데 있어 더 효율적이다. 이처럼

² Movidius Myriad2 와 NVIDIA P100 GPU 차이, <https://www.slideshare.net/byteLAKE/benchmark-of-common-ai-accelerators-nvidia-gpu-vs-intel-movidius-106453618>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

추후에 프로젝트 환경을 더욱 개선할 수 있는 방안을 고려할 것이다.

5 고충 및 건의사항

1. 원격 작업으로 인한 프로젝트 진행 지연


수백 장의 이미지 데이터를 수만에서 수십만 번 학습을 시키려면, 고성능의 GPU가 필요하고 많은 시간이 소요된다. 대량의 이미지 데이터를 학습시키기 위해 Amazon의 AWS 인스턴스 서버와 교내 KMU 딥러닝 클라우드 서버에서 학습을 시도하였으나, 파일 업로드 제한, 서버의 용량 또는 비용적인 문제로 팀원의 자택에 있는 데스크탑에 듀얼 부팅으로 ubuntu 16.04 LTS 환경을 구축하여 학습을 진행하고 있다. 그러다 보니, 모델에 학습시킬 데이터를 추가하거나, 학습 과정에 발생하는 오류를 수정할 때에는 원격 제어 프로그램을 이용하여 작업을 수행하고 있다. 원격으로 작업을 하다 보니, 직접적인 소통이 되지 않아 문제를 해결하는 데에 시간이 지체되는 경우가 종종 있었다.

2. 참고 자료의 부족

YOLO를 실행하기에 라즈베리파이는 부족한 computing power를 가지고 있다. 따라서 이를 극복하기 위해 딥러닝 USB인 Intel® Movidius™ NCS2 (Neural Compute Stick)를 추가 구입하여 사용하였다. 이 USB의 사용목적은 객체를 인식할 때 빠른 연산이 가능하며 저전력으로 라즈베리파이 내에서 객체 탐지가 가능하게 한다는 것이다. 그런데 출시된 지 얼마되지 않은 제품이어서 설치 과정에 발생하는 오류에 대한 해결책을 찾는 것이 매우 어려웠다. 그리고 NCS2를 설치하기 위해서는 플랫폼인 OpenVINO 툴킷을 설치해야 했는데, 해당 플랫폼을 구축하는 과정 자체도 복잡하고, 참고할 만한 자료도 부족하여 많은 시간이 소요되었다.

3. 데모환경 제작의 어려움

장애물로 선별한 물체들을 발표 현장에 가져올 수 없는 한계가 존재하여 대체품 구하거나

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	BeEyes	
	팀 명	ViewTopia	
	Confidential Restricted	Version 1.5	2019-APR-18

장애요소들을 직접 제작한다. (예: 계단, 보도블럭) 따라서 데모를 위한 장애물을 제작하기 전까지 데이터 수집 및 학습이 불가능하여 프로젝트 진행에 어려움이 따른다.