

## 시각장애인을 위한 모바일 기반 도보 위 위험 알림 시스템

조수형<sup>0</sup>, 김호진<sup>\*</sup>, 박상순<sup>\*</sup>, 최유준<sup>\*</sup>, 이수원(교신저자)<sup>\*</sup>

<sup>0</sup>숭실대학교 소프트웨어학부,

<sup>\*</sup>숭실대학교 소프트웨어학부

e-mail: ch5951130@naver.com<sup>0</sup>, ghwls328@gmail.com<sup>\*</sup>, paskss9677@gmail.com<sup>\*</sup>,  
cyj@gmail.com<sup>\*</sup>, swlee@ssu.ac.kr<sup>\*</sup>

## A Mobile-based Walking Danger Notification System for Visually Impaired People

Suhyeong Cho<sup>0</sup>, Hojin Kim<sup>\*</sup>, Sangsun Park<sup>\*</sup>, Yujun Choi<sup>\*</sup>, Soowon Lee(Corresponding Author)<sup>\*</sup>

<sup>0</sup>School of Software, Soong-Sil University,

<sup>\*</sup>School of Software, Soong-Sil University

### ● 요약 ●

도보 위 위험 알림이란 사람이 지나다닐 수 있는 길을 파악하고 길 위에서 사용자에게 접근하는 위협적인 장애물들을 탐지하고 알려주는 것이다. 본 연구에서는 Computer Vision의 Semantic Segmentation을 이용하여 사람이 다닐 수 있는 길을 구분하고 YOLO 사물 인식 알고리즘을 이용하여 시각장애인에게 접근하는 위협적인 장애물들을 탐지하여 알려줄 수 있는 시스템을 제시한다. 해당 시스템은 실용성을 고려하여 모바일 환경에서 이용할 수 있도록 구현하였으며, 서버와의 연동을 통해 실시간으로 사용자에게 사물 인식의 결과를 알려준다.

**키워드:** 시각장애인(impaired people), 장애물 탐지(obstacle detection), 위험 알림(danger notification)

### I. Introduction

시각장애인들을 위한 여러 시스템들이 개발되고 있지만, 현재 출시되어 있는 시스템들은 가격이 높고 필요한 주변기기들이 많거나 구하기 어려워 시각장애인들이 사용하기에 어려움이 존재한다. 따라서 사람들이 보편적으로 이용하고 있는 모바일 기기를 활용하여 시각장애인들을 대신하여 도보 위의 위험을 소리 혹은 진동으로 알려줄 수 있다면 누구나 쉽게 사용할 수 있을 것이다.

기존 시각장애인을 위한 모바일 기반 장애물 탐지 연구 중, [1]에서는 YOLO[2] 사물 인식 기술을 활용하여 차량, 자전거, 전동 킷보드, 사람 4종류의 장애물을 인식하고 장애물과의 거리를 계산한 뒤 모바일 기기를 통해 사용자에게 알려주는 기능을 제시하였다. 그러나 일반적으로 사람이 다닐 수 없는 길인 차도 위의 장애물까지 인식하기 때문에 실제적인 위험이 되지 않는 장애물도 포함되는 문제점이 있다. 또한 장애물과의 거리를 계산하는 과정에서 사진 속 장애물이 수직으로 서있는 경우가 아니라면 장애물의 높이가 왜곡되어 거리 계산에 부정적인 영향을 주는 문제가 발생한다.

이러한 배경에서 본 연구에서는 위 문제점들을 보완하여 인도 및 골목길 등 사람이 다닐 수 있는 길과 차도 등의 길을 구분하고 이미지 회전을 통해 보다 정확하게 장애물을 인식하고 판별하는 시스템을 제시한다. 제안 시스템은 먼저 시각장애인이 스마트폰 카메라

를 이용하여 촬영한 전방의 이미지와 스마트폰의 회전 정보를 서버로 송신한다. 서버에서 이미지를 수신하면 시각장애인이 스마트폰을 기울였을 경우 이미지를 정방향으로 회전시키는 작업을 수행한다. 그 후, YOLO알고리즘을 이용하여 전방에 있는 사람, 자전거, 전동 킷보드, 차량 4종류의 장애물을 탐지하고 각 장애물까지의 거리를 계산한다. 또한, 구글 Deeplabv3+[3]을 활용한 Segmentation-Selectstar라이브러리를 사용하여 사람이 다닐 수 있는 길과 다닐 수 없는 길을 구분한다. 그 후 다닐 수 있는 길 위의 장애물의 종류 및 거리를 종합하여 장애물의 위험 등급을 정한 후 스마트폰으로 그 결과를 전송한다. 마지막으로 스마트폰에서는 해당 정보를 토대로 사용자에게 소리 및 진동을 이용한 실시간 알람을 보낸다.

### II. Related Works

[4]와 [5]는 모바일 기기를 통해 장애물을 인식하는 연구이다. [4]의 경우, HOG[6] 알고리즘을 이용하여 사람만을 장애물로 인식하고 거리를 측정한다. [5]의 경우에도 HOG 알고리즘을 이용하며 사람 뿐 아니라 현재 사용자가 위치한 도보 위를 제외한 모든 장애물을

인식하지만 거리를 측정하지는 않는다.

본 시스템은 HOG 알고리즘이 아닌 YOLO 알고리즘을 사용하여 장애물 인식을 높이고, 알고리즘이 요구하는 많은 연산량을 서버를 활용하여 해결한다. [Table 1]은 본 시스템의 제안 방법과 [4]와 [5]의 연구를 비교한 표이다.

Table 1. Comparison with related works

	알고리즘	거리 측정	차도 구분	장애물	구동 방식
[4]	HOG	O	O	사람	스마트폰
[5]	HOG	X	X	모든 장애물	스마트폰
제안 방법	YOLO	O	O	사람, 자전거, 전동 키포드, 차량	스마트폰 + 서버

또한, 영상에서 실시간 Semantic Segmentation을 적용하는 연구로 [3]과 [7] 등이 있다. [3]의 연구는 구글에서 제안한 DeepLab 모델의 심화 버전이며, [7]의 연구는 기본적인 Segmentation 모델이었던 FCN의 낮은 정확도를 보완하기 위해 고안된 모델이다.

본 시스템에서는 [3]의 연구에서 제시한 Segmentation-Selectstar 라이브러리를 사용한다. [7]의 연구와 비교하였을 때 mIOU의 값이 높은 [3]의 연구가 더 높은 정확도를 보였다. 또한 [7]의 연구는 1.5FPS로 성능에 비해 수행 속도가 느리기 때문에 실시간으로 서버와 클라이언트가 통신하여 사용자에게 위험을 알려줘야 하는 본 시스템에는 부적합하다.

### III. The Proposed Scheme

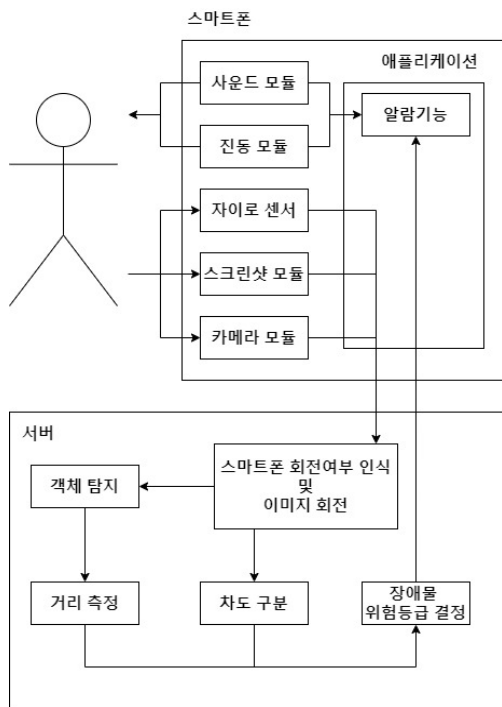


Fig. 1. System architecture

본 시스템에서 제안하는 시스템의 구조도는 [Fig. 1]과 같다. 본 시스템은 모바일 기기인 스마트폰과 기기 내에서 동작하는 애플리케이션, 그리고 서버로 구성되어 있다.

애플리케이션에서는 사용자의 스마트폰의 자이로 센서, 스크린샷 모듈, 카메라 모듈들을 이용하여 실시간으로 진행 방향의 길거리 화면을 촬영하고 사진 정보를 base64로 변환한다. 변환된 사진 정보는 스마트폰의 회전 정보와 함께 서버로 전송한다. 이후 서버가 정보를 처리하여 처리된 정보를 받으면 그것을 토대로 사용자에게 사운드 모듈과 진동 모듈을 통해 알람을 준다.

서버는 base64로 변환되어 받은 사진 정보를 먼저 디코딩을 통해 다시 jpg파일로 변환하고 스마트폰 회전 정보를 바탕으로 정방향으로 회전시킨다. 이후 YOLO를 통해 이미지 내에 있는 객체를 탐지하여 사용자와의 거리를 계산한다. 또한, Segmentation-Selectstar 라이브러리를 사용하여 사람이 다닐 수 있는 길을 파악하여 탐지된 객체가 사용자에게 위험이 되는지 확인하고 장애물들의 위험등급을 결정한다. 서버는 분석된 위험 정보를 스마트폰으로 전송한다.

#### 1. 스마트폰 회전 여부 인식 및 이미지 회전

애플리케이션으로부터 받은 이미지는 스마트폰의 회전 여부와 상관 없이 촬영되어 주어지므로 본 시스템에서는 이미지를 일괄적인 방향으로 처리할 수 있도록 이미지를 회전시킨다.

안드로이드에서 제공하는 SensorEvent 클래스의 TYPE\_ARRCELEROMETER을 사용하면 스마트폰에 작용하는 중력 가속도 값을 알 수 있다. 중력 가속도 값은 x축, y축, z축의 가속도를 m/s<sup>2</sup>의 단위로 표현한다. 스마트폰을 관통하는 축은 z축이며, 식 (1)은 z축의 각속도(yaw)를 계산하는 식이다.

$$yaw = \arctan(z/x) \tag{1}$$

z축의 각속도(yaw)를 단위 시간으로 적분하면 z축의 회전각을 알 수 있다. 이 회전각 만큼 이미지를 회전시킨 후 원본 이미지를 포함하는 최소 크기의 직사각형을 그리고, 원본 이미지에 포함되지 않는 직사각형의 내부 공간을 무색으로 설정하여 회전한 이미지를 만든다.

#### 2. 객체 탐지

YOLO 알고리즘은 회전한 이미지를 동일한 크기의 구역으로 나누고, 각 구역의 중심을 기준으로 미리 정의된 형태의 경계 박스의 개수를 예측한 후, 각각의 박스가 예측하는 객체와 일치하는지를 계산한다. 본 시스템에서는 사람, 자전거, 전동 키포드, 차량 4종류의 객체를 탐지한다.

#### 3. 거리 측정

본 시스템에서 사용하는 거리 측정 알고리즘은 실제 객체가 카메라의 센서를 지나 화면에 보이는 과정을 확장한 알고리즘이다.

안드로이드에서 제공하는 CameraCharacteristics 클래스의 LENS\_INFO\_AVAILABLE\_FOCAL\_LENGTH 와

SENSOR\_INFO\_PHYSICAL\_SIZE을 이용하여 초점 거리와 센서 높이 값을 알 수 있다. 초점 거리와 센서 높이 값을 이용하여 카메라와 장애물과의 거리를 구하는 수식은 식 (2)와 같다.

$$Distance = \frac{\text{초점거리} \times \text{실제객체높이} \times \text{이미지높이}}{\text{이미지상의객체높이} \times \text{센서높이}} \quad (2)$$

거리를 계산하는 식에 사용하는 ‘실제객체높이’ 값으로는 실제 장애물의 높이를 알 수 없기 때문에 각 장애물 별 평균 높이를 사용한다.

#### 4. 차도 구분

본 시스템에서는 Segmentation-Selecstar 라이브러리를 사용하여 사람이 다닐 수 없는 길인 차도와 사람이 다닐 수 있는 길인 인도, 그리고 횡단보도를 구분한다. Segmentation-Selecstar 라이브러리는 구글에서 발표한 DeepLab V3+을 이용하여 사진에 있는 모든 픽셀을 7개의 클래스로 분류하는 작업을 한다. 7개의 클래스는 배경, 자전거도로, 주의구역, 횡단보도, 가이드 블록, 차도, 인도이다.

#### 5. 장애물 위험등급 결정 및 알람

본 시스템은 도로 위의 장애물 중 사람, 자전거, 전동 킥보드, 차량을 탐지하고, 그 결과에 따른 위험등급을 4단계로 구분한다. 1단계의 위험등급은 가장 높은 위험도를 나타내며, 단계가 증가할수록 위험도가 감소함을 나타낸다.

서버는 각 장애물의 종류, 장애물과의 거리, 장애물이 위치한 길을 고려하여 위험등급을 결정하고 이를 애플리케이션에 전달한다. 본 시스템에서 설정한 위험등급 결정 기준은 [Table 2]와 같다. [Table 2]의 “사람”, “자전거/킥보드”, “차량” 이라는 3가지 열의 해당하는 조건 중 하나라도 만족하면 해당 위험등급으로 결정한다. 만족하는 위험등급이 중복된다면, 더 높은 위험등급을 갖도록 결정한다.

Table 2. Danger class by obstacle situation

위험 등급	알림	사람	자전거 / 킥보드	차량
1	비프음 1회	1.5m 이내	3m 이내	4m 이내
2	진동 3회	-	-	8m 이내
3	진동 2회	-	8m 이내	-
4	진동 1회	6m 이내	-	-

### IV. Experiments

#### 1. 실험 평가 방법

본 시스템에서는 두 가지 실험 평가를 수행하였다. 첫 번째는 회전 여부에 따른 장애물 탐지 인식을 비교이다. III장에서 기술하였듯이 장애물 탐지 및 거리 측정, 차도 구분을 위한 이미지는 스마트폰의 회전 여부와 상관없이 촬영되어 주어지므로 본 시스템에서 이미지를 일괄적인 방향으로 처리할 수 있도록 이미지를 회전시킨다. 이 때 회전하지 않는 원래의 이미지와 본 시스템에서 제안한 회전한 이미지의 각 객체 탐지의 정확도를 비교한다. 정확도는 임의의 1000장의

사진을 통해 진행하며 식 (3)에 따라 계산한다.

$$\text{정확도} = \frac{\text{탐지된 객체수}}{\text{실제 사진에 존재하는 객체수}} \times 100\% \quad (3)$$

두 번째는 장애물의 실제 높이에 따른 탐지된 장애물과의 거리와 실제 거리의 오차 평균 계산이다. 오차 평균의 값은 예측 값과 실제 값 오차의 절대값 평균 지표인 MAE를 사용한다. 장애물과의 거리를 계산할 때 실제 객체의 높이가 필요하나, 이를 알 수 없기 때문에 각 장애물 별 평균 높이를 사람은 170cm, 자전거는 90cm, 킥보드는 120cm, 차량은 150cm로 설정하였다. 본 평가에서는 각 장애물 별로 평균 높이, 평균 높이 ± 10cm 총 3가지의 값에 대해 MAE 값을 계산하였다. MAE 값이 0에 가까울수록 실제 거리와의 오차가 적은 것이므로 어떠한 값이 가장 정확한지 알 수 있다.

#### 2. 실험 평가 결과

[Table 3]은 이미지의 회전 여부에 따른 객체 탐지 정확도를 비교한 표이다. 1000장의 임의의 촬영된 이미지를 회전없이 본 시스템이 제안하는 객체 탐지를 수행하였을 때의 정확도는 75.2%로 측정되었다. 그러나 본 시스템에서 제안한 방법 즉, 회전 여부를 인식하고 이미지를 회전한 뒤 객체 탐지를 수행하였을 경우 정확도는 82.8%로 7.6%만큼 증가하였다.

Table 3. Comparison of object detection accuracy

회전 여부	정확도
회전 X (촬영된 이미지)	75.2%
회전 O (본 시스템의 제안 방법)	82.8%

[Table 4]는 각 장애물 별 높이에 따른 거리 측정에 대한 MAE 값을 비교한 결과이다. [Table 4]의 값들은 해당 장애물의 높이로 설정한 값과 그 값을 이용했을 때의 거리 측정 MAE 값이다. 평가 결과, 사람, 자전거, 전동 킥보드, 차량 모두 본 시스템이 설정한 평균 높이에서의 MAE가 10cm 오차를 준 높이에서의 MAE보다 작은 것으로 나타났으므로, 본 시스템이 설정한 각 장애물 별 평균 높이의 정확도가 더 높음을 알 수 있다.

Table 4. MAEs according to the height of each obstacle

사람	자전거	킥보드	차량
160cm / 0.52	80cm / 0.62	110cm / 0.39	140cm / 0.64
170cm / 0.35	90cm / 0.41	120cm / 0.31	150cm / 0.55
180cm / 0.37	100cm / 0.85	130cm / 0.79	160cm / 0.72

### V. Conclusions

본 연구에서는 시각장애인을 위한 모바일 기반 도로 위 위험 알람 시스템을 제안하였다. 제안 시스템은 서버와의 연동을 통해 YOLO 라이브러리를 사용한 객체 탐지 모듈과 핀홀 카메라의 기본 원리를

활용한 거리 측정 모듈, Segmentation-Selectstar를 이용한 차도 구분 모듈, 그리고 위험등급 결정 모듈로 구성된다. 본 시스템은 모바일 환경으로 실행되기 때문에 사용자의 휴대성이 간편하고 접근성이 높다는 장점이 있다. 또한, 시각장애인에게 위험에 대해 능동적으로 대처할 수 있는 수단을 제공하여 안전한 보행에 도움이 될 수 있을 것으로 기대된다.

본 기술은 보행 시각장애인의 안전과 직결된 기능인만큼 서버와의 통신 및 보안성을 더욱 강화할 필요가 있다. 또한, 축적된 데이터를 기반으로 자체적인 학습을 통해 시각장애인에게 안전한 상황으로 유도할 수 있는 추가적인 기능이 필요하다. 아울러, 이 기능을 소형 탈것에 적용하여 걸음이 불편한 시각장애인이 안전한 길로 이동할 수 있도록 도와주는 기술에 대한 연구가 필요하다.

Recognition(CVPR'05), 886-893, 20-25 June 2005.

- [7] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang and Jiaya Jia, "Pyramid Scene Parsing Network", CVPR 2017, 2881-2890, 27 Apr. 2017.

## ACKNOWLEDGEMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT 연구센터지원사업의 연구결과로 수행되었음 (IITP-2021-2018-0-01419)

## REFERENCES

- [1] Suhyeong Cho, Hojin Kim, Sangsun Park, Yujun Choi and Soowon Lee, "A Study on Mobile-based Obstacle Detection for Blinds", KIPS 2021, Mar. 2021.
- [2] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016, 779-788, 9 May 2016.
- [3] Liang-ChiehChen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation", ECCV 2018, 833-851, 22 Aug. 2018.
- [4] Fernando Garcia, Jesus Urdiales, JuanCarmona, David Martin and Jose Maria Armingol, "Mobile based Pedestrian Detection with Accurate Tracking", 2016 IEEE Intelligent Vehicles Symposium(IV), 44-48, 19-22 June 2016.
- [5] Ruxandra Tapu, Bogdan Mocanu, Andrei Bursuc, Titus Zaharia, "A SmartPhone-Based Obstacle Detection and Classification System for Assisting Visually Impaired People", 2013 IEEE International Conference on Computer Vision Workshops, 444 - 451, 2-8 Dec. 2013.
- [6] Navneet Dalal, Bill Triggs, "Histograms of oriented gradients for human detection", 2005 IEEE Computer Society Conference on Computer Vision and Pattern