

시각 장애인의 보행 보조를 위한 점자블록 인식 정확도 향상 방안 :

YOLOv5와 꼭짓점 좌표 분석 활용

강준구¹ · 바제네자 발렌틴¹ · 안소명² · 성민우² · 이영석³

¹강남대학교 IoT전자공학과 대학생, ²강남대학교 소프트웨어응용학부 대학생,

³강남대학교 KNU참인재대학 교수

{engineerJKK, bajeneza13, sso.mmyeong, okekdma2}@gmail.com, yslee38@kangnam.ac.kr

A Method to Enhance the Accuracy of Braille Block Recognition

for Walking Assistance of the Visually Impaired :

Using YOLOv5 and Analysis of Vertex Coordinates

Junekoo Kang¹ · Valentin BAJENEZA¹ · Somyeong Ahn² · Minwoo Sung² · Youngseok Lee³

¹Undergraduate student, Department of IoT Electronic Engineering, Kangnam University,

²Undergraduate student, School of Software Application, Kangnam University,

³Professor, KNU College of Liberal Arts and Sciences, Kangnam University

본 논문에서는 시각장애인의 보행에 도움을 주기 위하여 점자블록을 카메라가 정확하게 인식할 수 있는 정확도 향상 방안을 제안한다. YOLOv5(You Only Look Once Version 5)는 객체탐지에서 평균 정확도와 실행 속도가 적합한 알고리즘으로, 제안하는 정확도 향상방안은 YOLOv5와 꼭짓점 검출 알고리즘을 조합하여 점자블록을 실시간 검출하여 관심 영역을 추출한다. 정확하고 효율적인 이진화 작업을 수행할 수 있도록 프레임마다 추출된 관심 영역 속 이진화된 영상에서 꼭짓점을 검출하여 점자블록의 정보를 추출한다. 검출 과정을 실험한 결과, YOLOv5와 이진화 패턴 연구 기법보다 정확한 점자블록 패턴인식 정보를 제공하여, 성능이 가장 좋게 나타났다. 제안하는 정확도 향상방안을 기반으로 점자블록의 상태를 판단한다면, 시각장애인의 보행에 도움을 줄 수 있는 보조 장치 개발에 기여할 수 있을 것이다.

1. 서 론

점자블록은 시각장애인이 주의해야 할 위치나 유도 대상 시설 등의 상황 확인을 위해서 설치하는 것으로 위치 감지용 점형블록과 방향 유도용 선형블록이 있다[1].

점자블록 인식을 위해, 컴퓨터 비전 기술을 이용하여 시각장애인의 눈이 되어주기 위해 연구했던 방법으로는 다음과 같다.

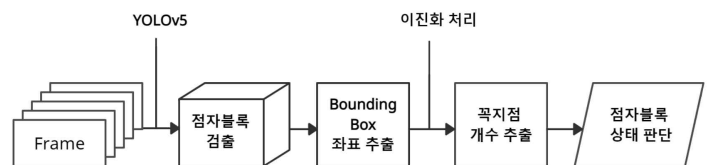
첫 번째로 노란 점자블록을 학습시켜 만든 YOLO(You Look Only Once) 모델[2,3]은 빠른 객체 인식을 통해 정보를 제공해 줄 수 있다. 하지만 점자블록 특성상 삼거리 안에서도 정지와 직진이 함께 검출되기에 사용자에게 어떠한 정보를 우선 제공할지 기준이 모호하다.

두 번째로 이진화 처리를 통해 점자블록의 패턴 정보를 인식하는 연구[4]의 경우 이상적인 환경일 때에는 인식 정확도가 높다. 하지만 점자블록 설치 후 노후화로 인하여 색이 변질되거나 주변 노이즈가 있을 경우, 정확도가 낮아지는 문제가 있다.

따라서 본 연구에서는 점자블록 인식의 정확도 향상을 위해서 YOLOv5 알고리즘을 활용하여 실시간 객체 검출을 하고, 정확성을 향상시키면서 논리적인 판단이 가능하도록 꼭짓점 좌표 분석을 활용하고자 한다.

2. 점자블록 인식 향상도 방안 설계

실시간 점자블록 인식 시 정확도를 높이기 위해 YOLOv5 알고리즘에 이진화 처리를 통한 꼭짓점 인식을 추가한 시스템의 알고리즘을 제시하고자 한다. 본 연구에서 제시한 방안을 설계한 구조도는 <그림 1>과 같다.



<그림 1> YOLOv5와 꼭짓점 좌표를 활용한 점자블록 인식 방안

제안하는 방법을 구현하기 위한 구체적인 절차는 다음과 같다. 첫째, 데이터 프레임 수집 및 전처리하여 YOLOv5모델을 통해 점자블록에 Bounding Box 처리한다. 둘째, Bounding Box 처리한 좌표를 구한다. 셋째, 좌표를 나타내는 사각형 내부를 이진화한 뒤 꼭짓점 개수를 추출한다. 넷째, 꼭짓점 개수에 따른 점자블록 상태 판단을 사용자에게 알려준다.

2.1. YOLO(You Only Look Once)

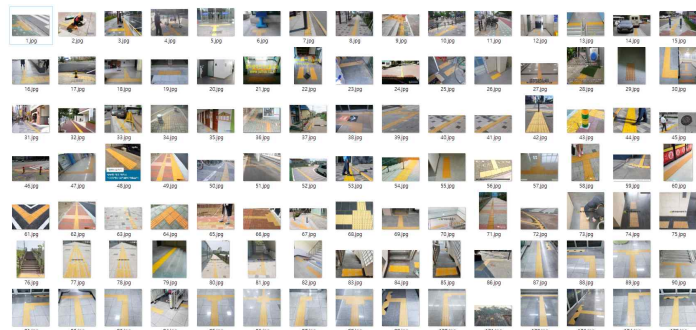
본 연구에서는 YOLO(You Only Look Once) 알고리즘을 활용하여 노란색 점자 블록을 실시간 탐지하는 시스템을 구축하였다. 이전 연구[3]에서 사용됐던 YOLOv3는 45 FPS(frame per seconds)와 70mAP(mean average precision)인 반면 YOLOv5는 140 FPS와 89.5mAP로 보다 뛰어난 성능을 발휘한다. 빠르고 정확한 영상처리가 안전과 직결된 만큼 본 연구에서는 YOLOv5를 활용하고자 한다.

2.2. 데이터 학습을 위한 환경설정

이미지 데이터 Labeling 작업은 ‘makesense.ai’에서 수행하였으며[5], 생성된 데이터 세트를 이용하여 학습을 진행하였다. 학습은 Google Colab 환경에서 런타임 유형은 Python 3, 하드웨어 가속기는 GPU를 사용하였다.

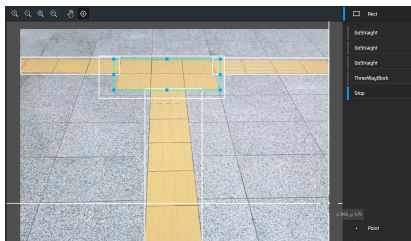
2.3. 이미지 데이터 수집과 전처리 과정

학습에 적합한 이미지 산출을 위해 일반적인 낮, 밤 환경과 지하철 환경에서의 점자블록 이미지 총 500장을 Google 지도와 직접 촬영을 통해 확보하였고, 예시 이미지는 <그림 2>와 같다. 총 학습 횟수(epochs)는 60회로 설정하였다.



<그림 2> 이미지 데이터 수집 예시

학습할 이미지는 Labeling 작업이 필요한데, 이는 탐지할 객체에 대해서 이미지 화면에 좌표를 Labeling 하는 전처리 과정으로 <그림 3>과 같다.



1 0.495707 0.639330 0.185194 0.721340
1 0.160887 0.170194 0.320103 0.079365
1 0.811322 0.171958 0.377355 0.075838
0 0.457688 0.223986 0.405954 0.197531
2 0.474858 0.211640 0.344632 0.144621

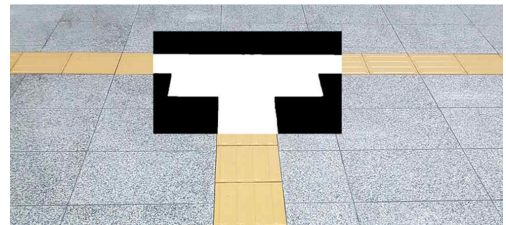
<그림 3> 이미지 데이터 라벨링 작업

Labeling 작업 후 각각의 이미지별로 생성된 텍스트 파일에는 객체 분류 번호와 학습 이미지의 좌표 지점을 기록하였다. 생성된 텍스트 파일에 기록되는 형태는 앞의 첫 번째 자리 숫자가 객체 분류 번호, 두 번째 자리 숫

자부터 좌표 지점이 소수점 자릿수로 처리되도록 하였다. 예를들어 1 0.49~ 0.63~ 0.18~ 0.72~에서 1은 ‘Go Straight’, 049~는 중앙 좌표 X값, 0.63~는 중앙 좌표 Y 값, 0.18~은 너비, 0.72~는 높이로 기록된다.

2.4. Bounding Box 안에서의 이진화 처리

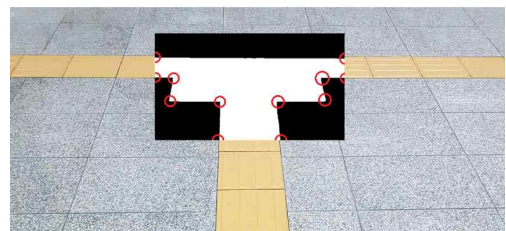
YOLOv5 에서 검출한 노란 점자블록 안에서 이진화 처리를 해주면 불필요한 외부 환경의 노란색으로부터 방해를 받지 않을 수 있다. 따라서 YOLOv5 모델에서 검출한 객체의 좌표를 활용하여 그 안에서만 이진화를 처리한다. 지하철 환경과 외부 바깥 환경, 그리고 노후화로 인한 노란 점자블록의 색상은 모두 다르므로 Bounding Box 중앙 좌표에서 20 pixel by 20 pixel 크기의 구역을 한 번 더 만들어주고 그 안에서 검출된 색상의 평균 BGR(Blue, Green, Red)값에서 각각 -20, +20의 범위를 설정한 뒤 이진화 처리를 수행한다. 이에 관한 결과는 <그림 4>와 같다.



<그림 4> Bounding Box 안에서 이진화 처리 작업

2.5. 이진화 영상의 꼭짓점 개수 검출

본 연구에서 시각장애인에게 정보를 제공하기 위해 해당 이미지가 삼거리인지 사거리인지 알려준다. 본 연구에서는 Douglas - Peucker 알고리즘[6]을 활용하여 꼭짓점을 검출한다. 예를 들어 삼거리의 경우 꼭짓점이 12개이며, 사거리의 경우 20개, 직진의 경우 4개이다. 실험적으로 처리한 삼거리 사례는 <그림 5>와 같으며, 꼭짓점이 12개로 나타나서 삼거리라고 판단할 수 있다.



<그림 5> 꼭짓점 검출 이미지


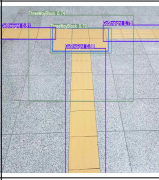

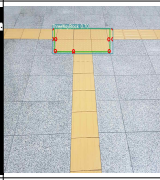



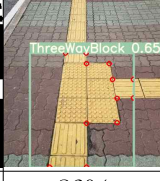
2.6. 실험 결과

본 연구에서 제안하는 방법의 성능평가를 하기 위한 정확도 계산에는 <표 1>과 식 (1)을 사용하였으며, 기존 연구와 비교 분석한 결과는 <표 2>와 <그림 6>과 같다.

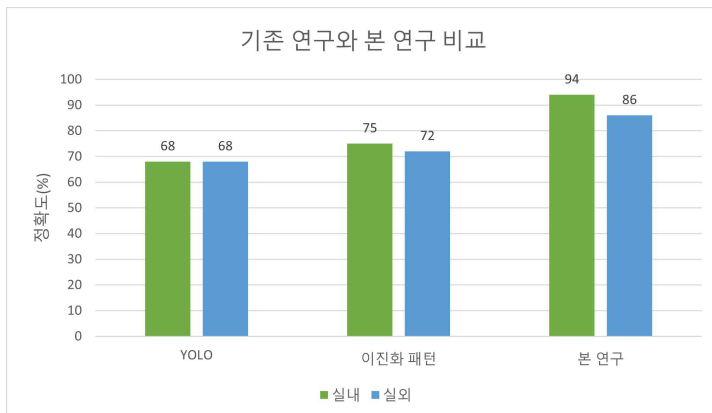
		Predict result	
		Positive	Negative
Real	Positive	TP (True positive)	FN (False negative)
	Negative	FP (False positive)	TN (True negative)

<표 1> 정확도 계산을 위한 분류표

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{All\ Positive\ Detections} \quad (1)$$

	A	B	C	D
	이미지	YOLO	이진화 패턴	본 연구
실 내				
	정확도	68%	75%	94%
실 외				
	정확도	68%	72%	86%

〈표 2〉 기존 연구와의 비교분석 실험결과



〈그림 6〉 실내외 점자블록 인식 정확도

기존 이미지인 A를 기준으로 실내와 실외 상황을 비교하였다. (B)YOLO의 경우 한 객체 안에서 직진, 정지, 삼거리 등 다양하게 인식하므로 선형블록과 점형블록의 구분에 어려움이 있어서 정확성이 낮다. (C)이진화 패턴의 경우 이상적인 지하철 환경에서는 비교적 잘 검출하지만 실외에서 주변 노이즈들이 함께 검출되므로 정확도가 낮다. 본 연구에서는 YOLOv5로 관심 영역의 범위를 좁혀준 뒤 그 안에서 이진화 처리 후 꼭짓점 개수를 검출한다. YOLOv5와 이진화 처리, 꼭짓점 검출 알고리즘을 결합하여 더욱 정확한 점자블록의 방향 안내 정보를 제공해 주므로 성능이 가장 좋게 나타났다.

3. 결론 및 향후 연구

본 연구에서는 시각장애인의 보행 보조를 위한 점자블록의 인식에 대한 정확도를 향상시킬 수 있는 방안을 제시하였다. 제안하는 방법의 성능을 분석하기 위해 실내와 실외 이미지 약 500장의 데이터 세트를 구축한 뒤 학습시켜 YOLOv5모델을 만들었다. 제작한 YOLOv5 모델을 활용하여 검출한 점자블록만을 이진화 처리한 뒤, 각각의 패턴이 가지는 꼭짓점 개수를 활용하여 정확한 방향을 안내하는 정보를 제공할 수 있도록 하였다.

실험 결과, 기존 연구 기법과 비교하여 노이즈에 강하고 높은 정확도를 보였음을 확인했다. 하지만 그 과정에서 훼손되거나 색이 바랜 점자블록의 경우 이진화 처리를 수행하는 데 어려움을 겪었다. 이를 해결하는 방안으로써 반복적인 실험을 수행한 결과 검출된 객체의 평균 BGR에서 각각 -20, +20 범위값을 정한 뒤 이진화 처리를 함으로써 해결하였다. 프레임마다 새롭게 BGR 범위를 지정해주기 때문에 조명의 영향을 받지 않는다는 장점이 있다. 따라서 자체 실험 결과 HSV, YCrCb 색상모델을 사용하지 않으면서 더 높은 정확도를 얻을 수 있었다.

또한 다양한 기상환경과 시간대의 점자블록 인식을 위한 추가적인 데이터 학습과 점자블록의 다양한 패턴으로 인하여 다르게 나올 꼭짓점 개수 변수에 대비할 필요가 있어 앞으로 추가적인 연구가 진행된다면 시각장애인의 보행 보조 개발에 기여할 수 있을 것이다.

참 고 문 헌

- [1] 시각장애인편의시설지원센터, “시각장애인 편의시설 설치 매뉴얼”, 한국시각장애인연합회, 2017
- [2] 이창석, 김민수, 강민희, 송재인, 황기연, “보행자 교통사고 예방을 위한 YOLO기반의 보행자 검출 방안 연구”, 한국ITS학회 추계학술대회, pp 53-57, 2020
- [3] 오세량, 배영철, “YOLO V3 기반의 시각장애인을 위한 유도 블록 인식 알고리즘”, 한국지능시스템학회 논문지, pp 60-67, 2021.2
- [4] 김경빈, 정경훈, “시각장애인 보행 유도를 위한 점자블록 패턴 분류”, 대한전자공학회 학술대회, pp 356-359, 2020.11
- [5] Piotr Skalski. 2019. makesense.ai. <https://www.makesense.ai/>
- [6] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a line or its caricature. The Canadian Cartographer, 10(2):112-122, 1973