

실내 환경 자율주행 로봇을 위한 객체 인지 모듈 개발

김명현, 김영인, 최인훈, 허의남*
경희대학교

{freckie, rladuddls3, inhun321, *johnhuh}@khu.ac.kr

An Implementation of Object Recognition Module for Indoor Self-Driving Robot

Myung-Hyun Kim, Yeong-In Kim, In-Hun Choi, Eui-Nam Huh*
Department of Computer Science and Engineering, Kyung Hee University

요 약

최근 비약적으로 발전한 기계 학습 기술은 자율주행 자동차를 실현하는데 큰 축을 담당하고 있다. 다만 실내 주행의 경우, 여러 환경적 제약 조건에 따라 자동차의 현재 위치를 파악하기에 어려움이 있다. 이러한 문제점에 주목하여 본 연구에서는 실시간 영상에서 실내 복도의 문패를 인식해 실내 위치 정보를 제공하는 모듈 개발을 목표로 한다.

1. 서론

미국의 아마존(Amazon)에서는 물류센터에서 물건을 옮기는 자율주행 로봇이 이미 운영 중이다. 다만 이는 실외나 다름없는 거대한 환경에서의 운행으로, 건물 복도와 같은 협소한 실내 환경과는 거리가 많다. 따라서 본 논문에서는 좁은 통로가 존재하고 사람들의 왕래가 잦으며, 차선 등 가이드라인이 없는 실내 환경에서의 개발을 진행했다.

또한 본 논문에서 개발을 목표로 하는 자율주행 로봇은 실내 운행용으로, GPS 등의 도움을 받기 어려워 정확한 현재 위치를 획득할 수 없다. 대신, 복도를 주행하며 로봇에 부착된 카메라로 실시간 영상을 분석, 복도의 문에 붙어 있는 문패(Door Plate)를 인식해 위치 정보를 획득하는 것을 목표로 한다.

2. 본론

2.1. 객체 인식 (Object Detection)

객체 인식은 입력 영상에서 여러 객체에 대한 Classification 과 Localization 을 진행한다. 객체 인식을 수행하는 딥 러닝 아키텍처인 R-CNN(Regions with Convolutional Neural Networks)와 YOLO 를 먼저 살펴보고 본 연구에 반영했다.

R-CNN 은 CNN 과 같은 딥 러닝 기반 이미지 분류기를 특징 추출기(Feature Extractor)로 사용하여 객체 인식에 높은 성능을 보인 예시이다. R-CNN 은 입력 영상에서 객체가 있을 것이라고 예측되는 위치를 찾고, 각 객체를 CNN 모델에 통과시켜 특징 벡터를 얻은 후 이를 SVM Classifier 에 통과시키는 2-Stage Object Detection 방식을 수행한다.[1]

다만 이 아키텍처는 비교적 정확하지만 매우 느리다는 단점이 있어, 이를 실시간에서 사용할 수 있게 개선한 모델인 YOLO 를 선택하였다.[2]

본 연구는 문패 속 숫자를 판별하는 것이 목표이므로, YOLO 모델을 숫자 인식 용도로 제한하여 학습했다. 데이터셋으로는 각 숫자의 경계 박스(Bounding Box) 좌표 데이터와 수 데이터를 포함하고 있는 SVHN (Street View House Numbers) 데이터셋¹을 사용하였다. YOLO 는 자전거, 강아지 등 80 종류의 실생활 객체를 포함한 coco 데이터셋²을 사용하였지만, 본 연구에서는 0~9 한 자리의 숫자를 10 종류의 객체로 사용하였다.



[그림 1] SVHN 데이터셋에 경계 박스를 표시한 그림

2.2. 문패 이미지 획득과 숫자 추출 과정의 분리

YOLO 의 실시간 영상 분석은 NVIDIA Titan X GPU 를 사용하여 초당 30 프레임의 처리 속도[3]를 보여준다. 앞서 제시된 R-CNN 보다 처리 속도가 빠르지만, 데스크탑 용 GPU 를 공간 제약 상 사용하기 힘든 자율주행 로봇의 환경에서 모든 프레임을 YOLO 모델로 처리하기에는 큰 어려움이 있다. 따라서 최대한 경량화한 모델 기반으로 구현하며, 문패 이미지를 먼저 획득한 후 YOLO 모델로 처리하도록 알고리즘을 구현하였다.

또한, 라즈베리 파이³에서 YOLO 의 처리 속도가 매우 느리므로 사실상 모든 프레임을 YOLO 모델로 처리가 불가능

¹ [http://ufldl.stanford.edu/housenumbers \(format 2\)](http://ufldl.stanford.edu/housenumbers (format 2))

² <https://cocodataset.org/#home>

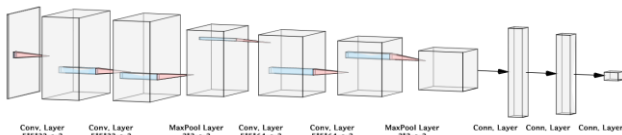
³ Raspberry Pi 4 Model B 4GB 기준

하다. 따라서 본 연구에서는 문패 이미지 획득 과정과 숫자 추출 과정을 분리하여 별개의 연구 목표로 설정하였다.

문패 이미지 획득 과정은 최대한 실시간에 가까운 처리 속도가 요구되며, 숫자 추출 과정은 최대한 높은 정확도가 요구된다. 따라서 문패 이미지 획득 과정은 초당 15 프레임 이상 처리 속도와 80% 이상의 정확도를, 숫자 추출 과정은 500ms 이하의 처리 시간과 95% 이상의 정확도를 목표로 설정하였다.

2.3. 문패 이미지 획득 과정

문패 이미지 획득 과정은 후보 이미지 추출 단계와 문패 이미지 여부 판별 단계, 총 두 단계로 구성하였다. 후보 이미지 추출 과정은 OpenCV 라이브러리를 이용하여 Canny Edge 검출, Contour 넓이 비교, Contour 비율 비교 등의 과정을 거쳐 후보 이미지를 생성한다. 인식된 Contour는 가로와 세로 비율이 1.0 ~ 3.0 인 이미지로 제한한다. 이렇게 생성된 이미지는 48*48 Grayscale 이미지로 리사이즈되어, 문패 이미지 판별 모델로 전달된다.



[그림 2] 문패 이미지 판별 모델 아키텍처

문패 이미지 판별 모델은 48*48*1 Grayscale 이미지를 입력으로 받아 문패 이미지인지 여부를 출력하는 이진 분류 모델(Binary Classification)이다. 모델 출력이 1 이 되어 문패 이미지로 판별되면 해당 이미지가 크롭된 프레임 이미지 전체를 YOLO 기반 숫자 추출 모델로 전달한다.

2.4. 숫자 추출 과정

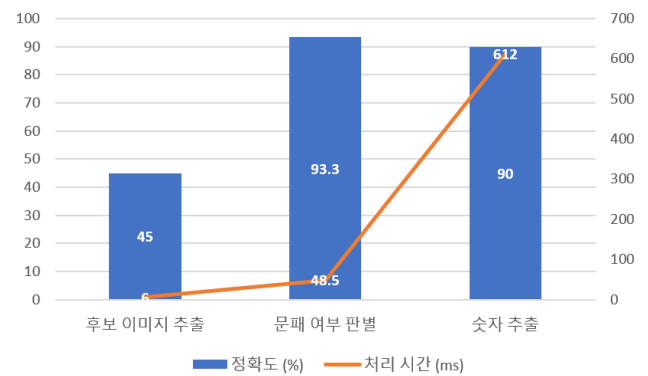
숫자 추출 과정은 프레임 이미지를 YOLO 기반 숫자 추출 모델에 통과하는 단계와 모델의 결과를 처리하여 현재 위치를 문자열로 추출하는 단계로 구성하였다. 프레임 이미지가 숫자 추출 모델에 입력되면, 모델은 인식된 각 숫자들을 박스 좌표와 함께 예측 값을 반환한다. 숫자 추출 모델은 경량화를 위해 YOLOv5s 모델을 SVHN 데이터셋으로 학습하였다.

모델을 통해 예측된 숫자 박스들을 좌표 기준으로 좌상단부터 정렬하여 예측 값을 조합해 숫자 문자열을 최종적으로 반환한다.

그러나 노이즈가 100%로 걸러질 수 없기 때문에 정확한 예측이 실패하는 경우도 있다. 이를 해결하기 위해 최근 20 프레임의 예측 값 중 가장 빈도가 높은 값을 선택해 출력하는 알고리즘을 추가로 구현하였다.

2.6. 성능평가

[그림 3]은 각 과정 별 평균 처리 시간과 정확도를 나타낸 그래프이다. 비교를 위해 문패 이미지 획득 과정의 두 단계를 모두 나타냈다. 또한 숫자 추출 과정의 정확도는 실제 문패의 정보와 예측한 문자열이 정확히 같은 경우 성공하였다고 설정했다.



[그림 3] 각 과정 별 평균 처리 시간 및 정확도

한 프레임의 처리에서 후보 이미지 추출 과정은 6ms 이하의 평균 처리 시간을 보여 전체 처리 과정에 유의미한 영향을 주지 않았다. 그러나 정확도가 45%로 크게 낮아 문패 여부 판별 모델로 한 번 더 필터링을 수행할 필요가 있다.

문패 여부 판별 모델은 48.5ms의 평균 처리 시간을 보여 주어 초당 약 20 프레임을 처리할 수 있다. 정확도 또한 93.3%로 목표치인 80%를 초과하는 성능을 보였다.

숫자 추출 모델은 약 90%의 정확도를 보였지만 612ms의 평균 처리 시간을 보여 사실상 실시간으로 처리가 어려운 것을 알 수 있다. 따라서 앞의 문패 이미지 획득 과정의 정확도를 최대한 높여 정확한 문패 이미지만이 숫자 추출 모델에 전달되도록 해야 할 것이다.

3. 결론 및 향후 연구

본 연구는 상황에 따라 유연한 대응을 하는 실내 자율주행을 위해, 실내 복도의 문패를 인식해 자율주행 로봇의 위치 정보를 획득하는 모듈을 개발하였다. 약 93.3%의 정확도를 보이는 문패 여부 판별 모델, 90%의 정확도를 보이는 숫자 추출 모델을 이용해 우발적으로 변화하는 실내 환경에도 안정적인 주행을 기대할 수 있다.

향후 연구로, 문패 이미지 획득 과정의 전체적인 정확도를 98% 이상으로 올려 노이즈를 최대한 배제하여 보다 더 최적화된 모듈을 구현할 계획이다. 또한 모듈을 계속 구동하는 방법보다, 문이 인식되었을 때만 모듈을 가동하는 방법으로 라즈베리 파이의 컴퓨팅 자원을 절약할 계획이다.

ACKNOWLEDGMENT

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2017-0-00294, (대학 ICT 기초연구실) 서비스 이동 지원을 위한 분산형 클라우드 핵심원천기술 연구)

참 고 문 헌

- [1] Shaoqing Ren, ed., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", 2015.
- [2] Joseph Redmon, ed., "You Only Look Once: Unified, Real-Time Object Detection", pp. 5-6, 2015.
- [3] *ibid.* p. 1.