

## 자율 주행 자동차를 위한 도로 및 신호정보 자동 해석 기술 연구

오지현, 문초이, 이유진, 허금조, 심동규

광운대학교

ddwwgg01@naver.com, dgsim@kw.ac.kr

Research on A System for Detecting and  
Interpreting Driving Environment for Autonomous Cars.Jihyeon Oh, Choyi Moon, Yujin Lee, Geumjo Heo, Donggyu Sim  
Kwangwoon University

## 요 약

본 논문에서는 차량, 보행자, 신호등, 표지판에 대한 정확하고 신속한 탐지, 해석을 위한 인공신경망 기반의 영상처리 기술 방법을 제안한다. 본 논문에서 제안하는 기술은 객체 인식 모듈과 차선 인식 모듈로 구성되며, 각 모듈에서 인식된 객체와 차선을 이용하여 도로 및 신호 정보를 해석한다. 제안하는 기술과 함께 일반적으로 자율주행 자동차에서 많이 사용되는 센서를 사용함으로써 자율주행 자동차에서 사용하는 카메라에서의 인식 성능을 높일 수 있다. 제안하는 기술은 인식된 표지판, 신호등의 객체를 통해 의미를 해석하고 이를 이용하여 자율주행 자동차가 위험을 감지하는데 도움을 줄 수 있다.

## 1. 서론

운전자의 운전 법규 위반으로 인한 교통사고의 발생을 줄이기 위해 자율 주행 자동차의 개발이 세계 각지에서 활발하게 이루어지고 있다. 인지능력 구현은 자율주행 차량 개발에서 핵심적인 부분이며, 이를 위해 카메라, 레이더, 라이다 등 다양한 종류의 센서를 조합하여 활용하고 있다. 최근에는 인지 기술의 신뢰성과 안전성을 향상시킬 수 있는 V2X(Vehicle to Everything, 무선망을 통하여 다른 차량 및 도로 등 인프라가 구축된 사물과 정보를 교환하는 통신 기술)나 고정밀지도 제작 관련 기술 연구가 활발하다.

다만 높은 수준의 자율주행 기술 구현을 위한 핵심부품으로 꼽히는 라이다 등은 여전히 타 센서 대비 상대적으로 높은 가격대를 형성하고 있어 아직 자율 주행 기술 상용화 및 대중화에 장애 요소라는 시각이 많다. 차량 전방의 긴급한 상황(비자율주행 차량, 임시 표지판, 공사 및 사고 현장 등)을 감지하는 경우 V2X 나 정밀 지도로는 큰 효과를 보기 힘들다. 다음과 같은 상황에서는 카메라와 같은 가장 기본적인 저렴한 센서가 핵심적인 역할을 할 수 있다.

이에 본 논문에서는 딥러닝과 영상처리 알고리즘을 이용하여 자동차 전방의 객체 인식 시스템을 제안한다. 특히 딥러닝 알고리즘을 사용하는데 있어, 해외 도로정보 데이터셋으로 학습된 기존의 실시간 객체 검출 알고리즘을 국내 도로정보 검출에 활용하기 위해서 국내 도로정보 데이터셋을 직접 구축하고 알고리즘을 추가로 학습시켰다.

본 논문에서 제안하는 시스템을 통하여 도로 및 신호 정보를 자동으로 해석할 수 있다. 객체 인식과 해석의 단계를 분리하여

객체 인식 속도와 정확도를 보다 향상시켰으며, 이 시스템을 국내 자율 주행 자동차를 위한 학습용 데이터를 추가 생성하는데 사용할 수 있을 것으로 기대된다.

본 논문의 2 장에서는 YOLO 를 이용하여 7 가지 객체에 대해 모델을 학습시키고, 학습된 모델로 객체를 탐지하는 방법에 대하여 설명한다. 3 장과 4 장에서는 표지판과 신호등의 분류 방법을 설명한다. 5 장에서 OpenCV 를 이용하여 차선을 인식하는 방법을 설명하고 6 장에서는 전체 결과와 성능을 평가한다.

## 2. 전체 시스템

본 논문에서 제안하는 시스템은 크게 두 모듈로, 딥러닝을 이용한 객체 인식 모듈과 영상처리를 이용한 차선 인식 모듈로 구성된다.

객체 인식 모듈은 YOLO v3(이하 YOLO) 모델을 사용하여 주행 영상에서 객체를 탐지한다. 객체에는 자동차, 사람, 오토바이, 버스, 트럭, 표지판, 신호등이 포함되고, 그중 표지판은 규제, 주의, 지시로 분류하고, 신호등은 빨간불, 초록불로 분류한다. 차선 인식 모듈에서는 OpenCV 를 사용하여 차선을 인식하며, 두 모듈의 결과를 통해 최종적으로 자율 주행 차량에게 필요한 도로 및 신호 정보의 탐지 및 해석이 완료된다.

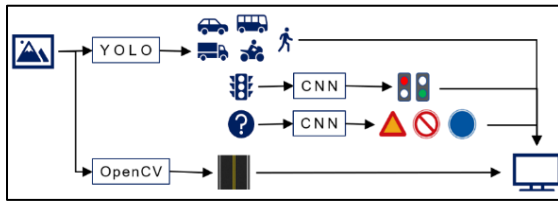


그림 1. 전체 시스템 개요

### 3. 객체 인식 모듈

본 논문에서 탐지하고자 하는 객체는 총 7 종류로 자동차, 사람, 오토바이, 버스, 트럭, 표지판, 신호등이다. 학습 데이터를 수집하여 YOLO 모델을 학습시키고, 학습된 YOLO 모델로 객체를 탐지한다. 탐지된 표지판과 신호등은 이후 Traffic sign CNN 모델과 Traffic light CNN 모델의 입력 영상이 되어 분류된다.

#### 3.1 학습 데이터 수집

차량 전방 블랙박스에 녹화된 동영상을 학습 데이터로 사용했다. 블랙박스 동영상을 40프레임 당 1프레임씩 이미지로 저장하고, YOLO\_MARK 를 사용하여 각 이미지에서 탐지하고자 하는 7 개 클래스의 객체에 대해 라벨링(Labeling)하여 학습용 데이터를 수집했다.



그림 2. 블랙박스 영상 라벨링 예시

한양대학교의 ACE Lab 에서 제공하는 Traffic Light Dataset[1]의 일부도 학습 데이터로 사용하였다.

표 1. YOLO 학습용 데이터셋 구성

구분	용도 별 데이터 개수 (단위: 개)		
	학습	검증	시험
사람	1,218	243	531
자동차	8,317	2,094	4,607
트럭	1,871	251	691
버스	920	219	566
오토바이	329	104	203
표지판	3,196	700	945
신호등	2,689	744	829
합계	19,540	4,355	8,372

#### 3.2 YOLO v3 모델 학습

YOLO 는 초당 프레임 처리 속도가 빠르면서도 정확도 측면에서 합리적인 성능을 보이는 실시간 객체 검출 알고리즘이다. 사용한 모델은 YOLO v3 의 Darknet-53[2]이다.

COCO 데이터셋으로 이미 학습된 가중치 파일에 앞서 수집한 학습용 데이터를 추가 학습시켰다.

### 4. 표지판 분류

YOLO 모델에 의해 검출된 교통 표지판을 분류하기 위한 단계이다. 한국의 교통 표지판은 그림 3 과 같이 규제, 주의, 지시, 보조 표지판으로 구분할 수 있다. 본 논문에서는 규제 표지판, 주의 표지판, 지시 표지판을 대상으로 한다.

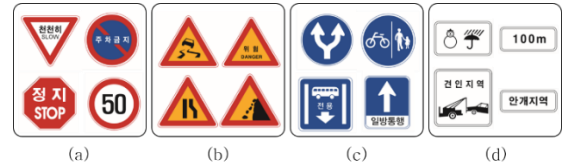


그림 3. 한국 교통 표지판 (a) 규제 표지판, (b) 주의 표지판, (c) 지시 표지판, (d) 보조 표지판

#### 4.1 학습용 표지판 데이터 수집

표지판 분류를 위하여 표지판 데이터를 수집하고, 추가로 Keras 의 ImageDataGenerator 클래스를 사용하여 원본 이미지를 변형시키는 방법을 이용하여 학습 데이터를 구성하였다.

규제, 주의, 지시에 해당되는 데이터를 수집했지만 인터넷상에 제공되는 한국 교통 표지판 이미지가 부족했기 때문에 그림 4 와 같이 ImageDataGenerator 클래스를 사용하여 원본 이미지에 임의로 변형을 가해 부족한 데이터를 보충했다[3]. 다만 표지판은 상하좌우의 방향이 존재하기 때문에 대칭이나 큰 회전 변화는 주지 않았다.

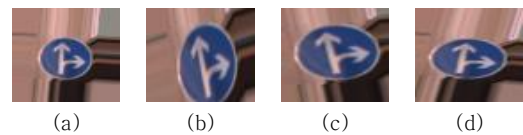


그림 4. 데이터 보충 예시 (a) 원본 이미지, (b) (c) (d) 변형된 이미지

#### 4.2 CNN 모델 생성 및 학습

상기 구성된 학습 데이터를 이용하여 인식된 표지판을 규제, 주의, 지시에 해당하는 3 개의 클래스로 분류하는 CNN 모델을 생성하고 수집한 데이터로 학습시킨다.

CNN 모델은 Keras 에서 제공하는 시퀀스 모델을 사용하여 레이어를 쌓아 생성했다. 총 3 개의 층으로 구성했으며, 활성화 함수로는 relu, softmax 함수를 적용했다. 또한 dropout 을 적용하여 과적합을 방지했다. 생성한 모델을 앞에서 수집한 총 12000 개의 표지판 데이터로 학습시켰다. 최종 분류될 클래스는 규제, 주의, 지시로 지정했다[4].

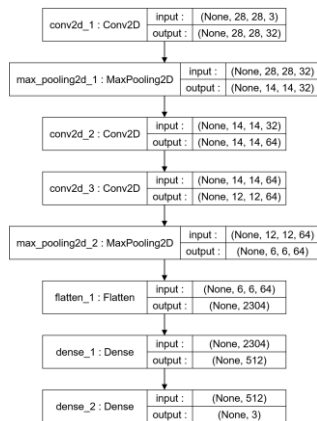


그림 5. Traffic sign CNN 모델

## 5. 신호등 분류

YOLO 모델에 의해 검출된 교통 신호등을 분류하기 위한 단계이다. 한국의 교통 신호등은 빨간불, 초록불 외에도 노란불, 화살표를 이용한 좌회전, 우회전 신호도 존재하지만 본 논문에서는 빨간불, 초록불 분류를 목적으로 한다.

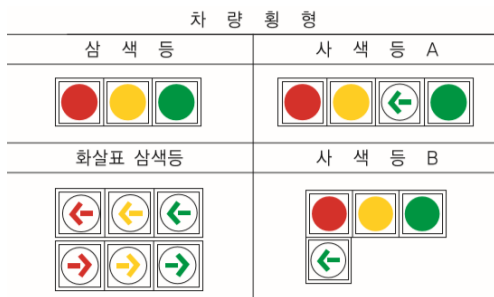


그림 6. 한국 교통 신호등

### 5.1 학습용 신호등 데이터 수집

앞서 만든 YOLO 모델을 이용하여 탐지된 신호등 이미지를 검은색을 기준으로 크롭(crop)하여 색 분류에 방해가 되는 배경색을 최소화한다. 이후 신호등 이미지를 좌, 우 반으로 분리하여 좌측에서 RGB 중 빨간색 화소가 60% 이상을 차지하면 빨간불로, 우측에서 초록색 화소가 60% 이상을 차지하면 초록불로 분류하였다.

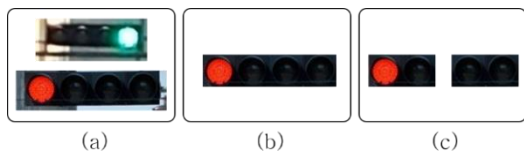


그림 7. 데이터 수집 과정 (a) yolo가 탐지한 이미지, (b) 검은색 기준 크롭, (c) 반으로 분리

이 신호등 구별 알고리즘은 YOLO 모델이 탐지한 신호등의 화질과 상태에 따라 인식률이 떨어지기 때문에 최종 신호등 분류 알고리즘으로 사용하지는 못했으나 학습용 데이터를 수집하기에는 충분한 인식률을 가졌다.

### 5.2 CNN 모델 생성 및 학습

CNN 모델은 표지판과 마찬가지로 Keras 에서 제공하는 시퀀스 모델을 사용하여 레이어를 쌓아 생성했다. 활성화 함수로는 relu, softmax 함수를 적용했고 adam optimizer 을 사용했다. 생성한 모델을 앞에서 수집한 총 8000 개의 표지판 데이터로 학습시켰으며, 학습 횟수는 loss 가 최대한 작은 숫자에서 더이상 줄어들지 않을 때까지 진행했다. 최종 분류될 클래스는 빨간불, 초록불로 지정했다.

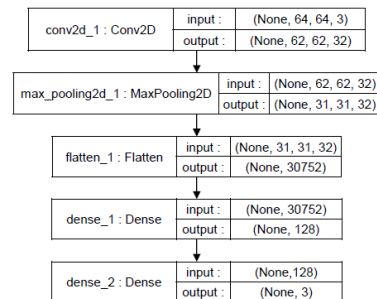


그림 8. Traffic light CNN 모델

## 6. 차선 인식 모듈

OpenCV 를 이용하여 차량 전방 블랙박스에 녹화된 동영상에 대하여 차선 인식을 수행한다. 차선 인식은 입력 영상에 대하여 전처리를 수행한 뒤, edge 추출, 직선 검출, 최적 직선 도출을 통해 차선을 검출한다.

### 6.1 전처리

주행 차량의 전방에서 차선이 탐지될 수 있는 관심 영역을 사각형으로 ROI 설정해준다. 흰색과 노란색인 차선은 그레이스케일에서 이진화 하기에 적합하기 때문에 설정된 영역을 Gray Image 로 변경한다. 그리고 가우시안 필터를 적용하여 잡음을 제거한다.

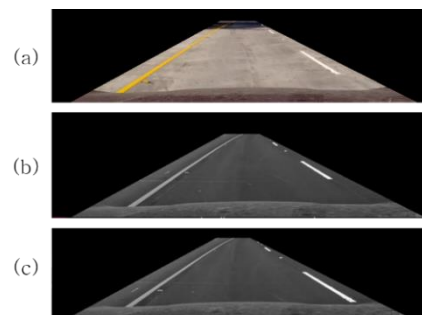


그림 9. 전처리 과정 (a) ROI 설정, (b) 그레이스케일, (c) 가우시안 필터

### 6.2 차선 검출

전처리한 영상에서 Canny 함수를 이용해 차선의 edge 를 추출한다. 그리고 최소 직선 길이를 지정하고 기울기를 필터링하여 몇 개의 직선으로 정리한다. RANSAC(Random Sample Consensus) 방식을 사용하여 검출된 직선들 중 최적의 직선을 선택하고 현재 프레임에서 구한 직선이 이전

프레임의 직선과 자연스럽게 연결될 수 있도록 프레임들 모아 평균 선을 출력한다.

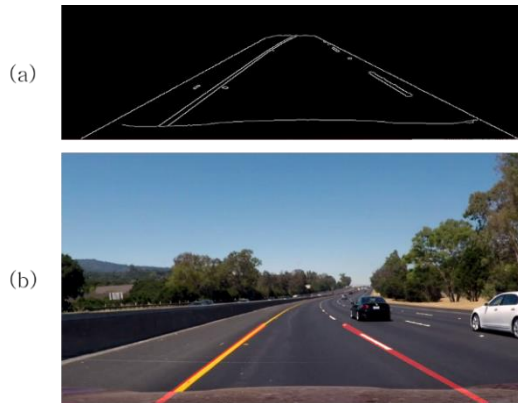


그림 10. 차선 검출 과정 (a) Edge 추출, (b) 직선 검출

## 7. 실험 및 결과

YOLO 모델은 해상도 FHD 수준의 시험용 동영상을 사용하여 실험했다. 성능 평가 지표로는 AP 를 사용했다. 시험용 동영상에 대하여 20fps 정도의 처리 속도를 보였으며, 동영상 속 객체 8372 개에 대해서는 표 2 와 같은 인식률을 보였다. 학습 이전의 COCO detection weight 모델과 비교했을 때, 학습 이후 모든 종류의 객체에서 인식률이 향상되었다.

표 2. YOLO 모델 성능 평가

구분	AP (단위: %)	
	학습 이전	학습 이후
사람	1.81	28.12
자동차	2.14	75.81
트럭	0.4	37.86
버스	16.97	32.55
오토바이	9.51	23.99
표지판	0	36.52
신호등	0.24	46.51
thresh = 0.25, F1-score = 0.57		

Traffic sign CNN 모델과 Traffic light CNN 모델은 15 분 30 초의 차량 블랙박스 주행 영상에서 YOLO 모델을 통해 검출된 표지판과 신호등 이미지를 각 클래스로 정확히 분류하는지 인식률을 계산하는 방법으로 실험했다. 실험 결과 표 3 과 같이 표지판은 94% 이상의 인식률을 보였으며, 신호등은 98% 이상의 인식률을 보였다.

표 3. 표지판과 차선 분류 성능 평가

모델	클래스	인식률 (단위: %)
Traffic sign CNN	규제	94.84
	주의	94.61
	지시	95.00
Traffic light CNN	빨간불	99.53
	초록불	98.95

차선 탐지는 여러 화질의 블랙박스 주행 영상을 사용하여 실험했다. 화질이 좋고 차선이 뚜렷한 곳이나 고속도로, 각도가 큰 곡선이 아닌 구간에서는 문제없이 동작하지만, 차량이 많아 차선이 가려지거나 흐린 곳, 유턴 구간과 횡단보도에서는 검출 차선이 정확하지 않은 결과를 보인다.

## 8. 결론

본 논문은 YOLO, CNN 과 OpenCV 를 사용하여 차량이 전방의 기본적인 도로정보와 객체를 탐지 및 해석하는 기술을 제안하였다. 보다 향상된 객체 검출 결과를 보였으며, 표지판과 신호등을 분류하는데 있어 상당히 높은 인식률을 보였다.

향후 연구 계획은 학습용 데이터셋을 보충하고 YOLO 모델을 추가 학습하여 객체 인식률을 높일 계획이다. 또한 표지판과 신호등의 의미를 더욱 다양하게 해석할 수 있도록 클래스를 확장할 계획이다. 차선 인식은 고속화[5] 방면으로 연구하고 외부 환경 변화에 강인한 에지 검출[6]을 사용하여 보다 정확한 차선 검출 방법을 개발할 계획이다.

## ACKNOWLEDGEMENT

"본 논문은 2019 년도 교육부의 대학혁신지원사업 사업비를 지원받아 수행된 연구임"

## 9. 참고문헌

- [1] <https://www.acelab.org/traffic-light-dataset>
- [2] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018
- [3] "작은 데이터셋으로 강력한 이미지 분류 모델 설계하기", accessed Sept. 19, 2019, [https://keraskorea.github.io/posts/2018-10-24-little\\_data\\_powerful\\_model/](https://keraskorea.github.io/posts/2018-10-24-little_data_powerful_model/)
- [4] Simple Image Classification using Convolutional Neural Network — Deep Learning in python, Venkatesh Tata, accessed Sept. 19, 2019, <https://becominghuman.ai/building-an-image-classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8>
- [5] 강병찬, 정차근. (2005). A Study on high speedization of lane detection using Hough Transform. 학술저널. Vol.2005 No.- [2005]. 195-198(4 쪽)
- [6] 권보철, 신동원. (2012). 외부 환경 변화에 강인한 에지 검출을 통한 차선의 스플라인 생성(Lane Spline Generation Using Edge Detection Robust to Environmental Changes). 학술저널. Vol.17 No.6 [2012]. 1069-1079(11 쪽)