

# 도로 내 장애물 인식에 관한 연구

김슬기<sup>§</sup>, 이재민<sup>§</sup>, 표성민<sup>\*</sup>, 방인규<sup>■†</sup>, 김태훈<sup>■§</sup>

<sup>§</sup>한밭대학교 컴퓨터공학과, <sup>\*</sup>한밭대학교 정보통신공학과, <sup>†</sup>한밭대학교 지능미디어공학과  
{20211870, 20212127, spyo}@edu.hanbat.ac.kr, {ikbang, thkim}@hanbat.ac.kr,

## A Study on the Obstacle Recognition on the Road

Seul Ki Kim<sup>§</sup>, Jae Min Lee<sup>§</sup>, Sungmin Pyo<sup>\*</sup>, Inkyu Bang<sup>■†</sup>, Taehoon Kim<sup>■§</sup>

<sup>§</sup>Department of Computer Engineering, Hanbat National University

<sup>\*</sup>Department of Information and Communication Engineering, Hanbat National University

<sup>†</sup>Department of Intelligence Media Engineering, Hanbat National University

### 요 약

본 논문에서는 OpenCV와 YoloV3를 이용해 차선 인식 및 객체 탐지 기술을 구현하며, 자율 주행의 기본인 장애물 회피를 위해 차선 인식 기술을 기반으로 특정 영역 내의 장애물 인식 기술을 제안하고 구현한다.

### I. 서 론

도로교통공단 시스템에서 제공하는 통계에 따르면 2021년 가해운전자의 사고 건수가 20세 이하면 6,960건, 21~30세면 29,076건, 31~40세면 30,304건, 41~50세면 36,480건, 51~60세면 46,938건으로 연령이 높아짐에 따라 사고율이 높아나는 것을 확인할 수 있다 [1]. 또한, 미국 도로교통안전국의 통계에 따르면 도로에서 발생하는 94%의 교통사고는 운전자의 부주의가 주원인인 것으로 밝혀졌다 [2].



그림 1 2021년 가해운전자 연령층별 교통사고 [1]

최근 4차 산업 혁명으로 자율주행차가 각광 받음에 따라 자율 주행 기술의 필요성 또한 주목받고 있다. 자율 주행은 운전자 부주의 또는 도로의 상태 등 다양한 이유로 발생하는 교통사고 예방, 노약자의 이동성 향상, 운전자의 운전 부담을 줄여 차내에서 다른 활동을 가능하게 해 주는 등 다양한 측면에서 유용하게 이용될 것으로 예상된다.

안전한 자율 주행을 위해서 주변 장애물에 대한 빠른 탐지와 신속 대응에 대한 문제가 우선인데 [3], 장애물 회피를 위해서 일반적으로 차선 인식을 통해 관심 영역을 식별해야 하며 관심 영역 내의 객체 탐지를 수행해야 한다. 본 논문에서는 OpenCV와 YoloV3를 이용해 차선 인식 및 객체 탐지 기술을 구현하며, 장애물 회피를 위해 차선 인식 기술을 기반으로 특정 영역 내의 장애물 탐지 기술을 제안하고 구현한다.

### II. 본 론

#### 가. 차선 인식

차선 인식은 도로에 표시된 차선을 검출하여 차선 정보를 획득하는 것을 의미한다. 실험은 OpenCV와 Jupyter Notebook을 통해 수행하였으며 차선 인식 절차는 그림 3과 같다.

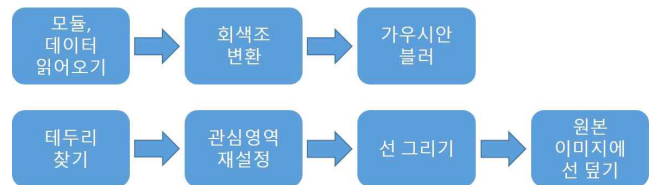


그림 3 차선인식 절차

첫 번째로 필요한 모듈 및 패키지를 설치하고 데이터를 읽어온다. 두 번째로 카메라를 통해 들어온 컬러 이미지는 복잡한 형태의 데이터이기 때문에, 단순한 형태인 회색조 이미지로 변환을 해준다. 세 번째로, 잡음을 제거해 주기 위해 가우시안 블러(blur) 처리를 해준다. 네 번째로, 적절한 임계값을 기준으로 어두우면 검은색, 밝으면 흰색으로 바꿔 주어 테두리를 검출해준다. 다섯 번째로, 검출된 이미지 중에서 차선으로 인식이 되고자 하는 부분만을 재설정 해 주면 관심 영역이 하얗게 나오는데 그 영역에 들어온 그림만을 대상으로 선을 찾는다. 그리고 여섯 번째로, 윤곽선 중 차선으로 적합한 직선을 골라 HoughLinesP 함수를 이용해서 선을 그려준다. 마지막으로, 원본 이미지에 선을 덮으면 차선 검출이 완료된다 (그림 4 참고).

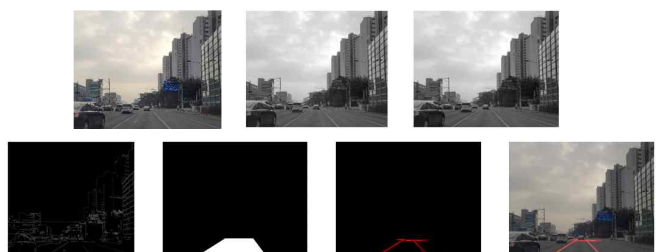


그림 4 차선인식 결과

### 나. 객체 탐지

객체 탐지는 이미지 내 사용자가 관심 있는 객체를 검출하는 기술이다. 객체 탐지 과정에서 분류(classification)과 위치 식별(localization)을 동시에 수행하는데, 객체라고 판단되는 곳에 box 표시를 해 주고(localization) box 내부의 객체가 무엇인지 분류해준다. 실험은 OpenCV와 YOLOv3, Jupyter Notebook을 통해 수행하였고 그 절차는 그림 5와 같다.

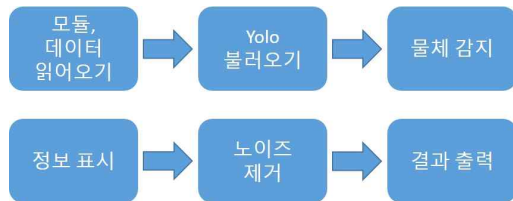


그림 5 객체 탐지 절차

첫 번째로 필요한 모듈 및 패키지를 설치하고 데이터를 읽어온다. 두 번째로 YOLOv3 모델을 불러온다. 세 번째로, 이미지를 blob 객체로 처리해서 setInput 함수에 적용하고 네트워크 순방향으로 실행해 준다. 네 번째로, 물체 인식에 대한 신뢰도가 특정 값을 넘으면 물체라고 인식하도록 해 주고, 물체의 범위를 사각형으로 그린 후 물체가 무엇인지 이름을 붙여 준다. 다섯 번째로, 같은 물체에 대해 여러 개의 사각형이 그려진 부분들을 제거한다. 마지막으로, 모든 정보를 추출해서 글씨체와 사각형의 두께, 글씨 크기 등을 화면에 표시한다 (그림 6 참고).



그림 6 객체 탐지 결과

### 다. 관심 영역 내 장애물 탐지

본 절에서는 차선인식 기술을 기반으로 관심 영역(ROI: Region of Interest)을 설정하고 관심 영역 내부의 장애물(예: 자동차, 동물 등) 탐지 기술을 제안한다. 보다 구체적으로, 특정 포인트(point)가 관심 영역 내부에 있는지 확인하기 위해 Point in Polygon 알고리즘을 함수로 만들어 사용했다. 이 알고리즘은 2차원 공간에 있는 한 점이 다각형 내부에 있는지 확인하는 방법인데, 점에서 반직선을 그어 다각형과 겹치는 점이 홀수 개면 다각형의 내부에 있는 것이고, 짝수 개면 다각형의 외부에 있는 것으로 판별할 수 있다 (그림 7 참고).

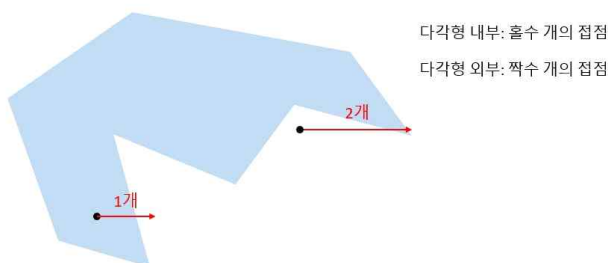


그림 7 Point in Polygon 알고리즘

차선을 먼저 인식하고 인식한 결과로 관심 영역을 식별한다. 그리고, 특정 포인트가 관심 영역 내부에 있으면 참인 값을 가지는 범위를 찾아내 객체를 인식하도록 하였다. 그림 8은 결과를 보여주며, 빨간색이 차선 인식, 파란색 사각형이 관심 영역 내 검출된 장애물(객체)이다.



그림 8 관심 영역 내 장애물 탐지

### III. 결론

본 연구에서는 자율주행 차량의 안전한 운행을 위해 필수적인 차선 인식과 객체 탐지 기술을 구현했고, 차선 인식 기반의 관심 영역 설정 방법과 관심 영역 내부의 객체(장애물)를 탐지하는 방법을 Point in Polygon 알고리즘을 통해 제안하고 구현을 통해 동작 여부를 확인했다. 향후 박스의 크기와 차선의 길이를 이용한 거리 예측과 움직이는 이미지(동영상)의 관심 영역 내부의 객체(장애물)를 탐지하는 연구를 진행할 예정이다.

### 참 고 문 헌

- [1] 도로교통공단 TAAS 교통사고 분석 시스템, 2021 가해운전자 연령층 별 교통사고
- [2] 박창규, et al. “멀티센서를 활용한 자율주행 자동차의 장애물 인식 및 예측 방법에 대한 연구”. 대한전자공학회 추계학술대회 (2022): p.01
- [3] 박현진, et al. “자율운행 선박을 위한 카메라 기반 장애물 인식 및 경고 시스템”. 한국컴퓨터종합학술대회 (2021): p.01
- [4] 차선 인식 관련 참고 문헌, Matt Hardwick, Simple Lane Detection with OpenCV, <https://medium.com/@mrhwick/simple-lane-detection-with-opencv-bfeb6ae54ec0>
- [5] Yolo 관련 참고 문헌, Sergio Canu, Yolo object detection using Opencv with Python, <https://pysource.com/2019/06/27/yolo-object-detection-using-opencv-with-python/>