

도심로 주행을 위한 딥러닝 기반 객체 검출 및 거리 추정 알고리즘 적용

Application of Deep Learning-based Object Detection and Distance Estimation Algorithms for Driving to Urban Area

서주영* · 박만복**

* 주저자 : 한국교통대학교 전자공학과 석사과정

** 교신저자 : 한국교통대학교 전자공학과 교수

Juyeong Seo* · Manbok Park**

* Dept. of Electronics, Korea National Univ. of Transportation

** Dept. of Electronics, Korea National Univ. of Transportation

† Corresponding author : Manbok Park, ohnnuri@ut.ac.kr

Vol. 21 No.3(2022)
June, 2022
pp.83~95

pISSN 1738-0774
eISSN 2384-1729
<https://doi.org/10.12815/kits.2022.21.3.83>

Received 17 December 2021
Revised 28 December 2021
Accepted 13 June 2022

© 2022. The Korea Institute of
Intelligent Transport Systems. All
rights reserved.

요약

본 논문은 자율주행 차량 적용을 위한 객체 검출과 거리 추정을 수행하는 시스템을 제안한다. 객체 검출은 최근 활발하게 사용되는 딥러닝 모델 YOLOv4의 특성을 이용해서 입력 이미지 비율에 맞춰 분할 grid를 조정하고 자체 데이터셋으로 전이학습된 네트워크로 수행한다. 검출된 객체까지의 거리는 bounding box와 homography를 이용해 추정한다. 실험 결과 제안하는 방법에서 전반적인 검출 성능 향상과 실시간에 가까운 처리 속도를 보였다. 기존 YOLOv4 대비 전체 mAP는 4.03% 증가했다. 도심로 주행시 빈출하는 보행자, 차량 및 공사장 고깔(cone), PE드럼(drum) 등의 객체 인식 정확도가 향상되었다. 처리 속도는 약 55 FPS이다. 거리 추정 오차는 X 좌표 평균 약 5.25m, Y 좌표 평균 0.97m으로 나타났다.

핵심어 : 자율주행, 딥러닝, 객체 검출, 거리 추정, 카메라

ABSTRACT

This paper proposes a system that performs object detection and distance estimation for application to autonomous vehicles. Object detection is performed by a network that adjusts the split grid to the input image ratio using the characteristics of the recently actively used deep learning model YOLOv4, and is trained to a custom dataset. The distance to the detected object is estimated using a bounding box and homography. As a result of the experiment, the proposed method improved in overall detection performance and processing speed close to real-time. Compared to the existing YOLOv4, the total mAP of the proposed method increased by 4.03%. The accuracy of object recognition such as pedestrians, vehicles, construction sites, and PE drums, which frequently occur when driving to the city center, has been improved. The processing speed is approximately 55 FPS. The average of the distance estimation error was 5.25m in the X coordinate and 0.97m in the Y coordinate.

Key words : Self-driving, Deep-learning, Object-detection, Distance-estimation, Camera

I. 서 론

1. 개요

자율주행 자동차 기술은 현재 주목받는 4차 산업 신기술을 넘어 상용화를 목표로 활발한 연구가 이뤄지고 있다. 그에 맞추어 현재 자율주행 기술인 고속도로지원시스템(HDS, Highway Driving Assist system) 기준 자율주행 2단계 수준인 운전자 지원 시스템(ADAS)이라는 이름으로 자동차 회사 및 부품 회사로부터 양산되고 있다.

완전한 자율주행의 상용화를 위해선 다양한 환경에서 인식 성능이 확보되어야 한다. 특히 복잡도가 더 높은 도심로에서도 자율주행 차량은 정확히 주변을 인식해야 한다. 도심로 자율주행에 대한 연구에 비해 비교적 주행 조건이 단순한 고속도로 주행 환경에서의 연구가 주로 이뤄져 왔다. 도심로 주행 상황까지 포함하는 인식 성능을 위한 연구가 더 활발히 진행될 필요가 있다.

자율주행 시스템은 인식, 판단 및 제어 총 3단계로 구성되어 있다. 인식 단계에서 자율주행 차량에 장착된 여러 센서가 각각 특성에 맞게 주변 환경에 대한 정보를 수집한다. 수집된 정보는 그후 판단 및 제어 단계에서 분석하고 다음 행동을 판단 및 계획하는 데에 사용된다. 인식 단계에서 사용되는 센서로는 카메라, 라이다, 레이다, GPS 등이 있다. 그중 합리적인 비용으로 비교적 많은 정보를 얻을 수 있는 카메라 센서는 차선 및 주변 객체를 감지한다. 그러한 데이터를 기반으로 주행 가능 영역과 식별된 객체 정보, 객체까지의 거리 등의 정보를 얻을 수 있다. 본 연구는 차량에 장착된 단안 카메라로부터 영상을 입력받아 실시간으로 딥러닝 기반 객체 검출을 실행한 결과를 기반으로 거리를 예측하는 시스템을 구축 및 실차 적용하는 것을 목표로 한다.

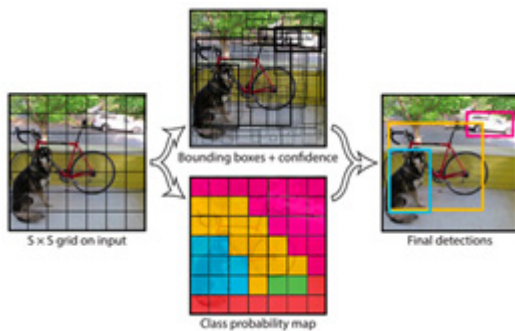
자율주행의 객체 검출은 안전 문제와 연관되어있기 때문에 실시간으로 처리할 수 있을 뿐만 아니라 100%에 가까운 정확도를 가져야 양산될 수 있다. 따라서 빠르면서 정확한 모델을 연구해야 할 필요가 있다. 빠른 처리 속도를 확보하기 위하여 딥러닝 객체 검출 모델 중 One-stage detector에 속하는 YOLOv4를 객체 검출 모델로 사용했다(Redmon et al., 2016). 도심로 주행 상황이라는 문제에 맞게 모델을 자체 데이터셋으로 전이 학습 하였다. 본 논문은 모델의 성능 개선을 위해 YOLOv4의 네트워크를 조정하는 방법을 선택했다. 입력 영상을 가로와 세로를 같은 개수로 나눈 S*S개의 grid로 분할한 뒤, 각 grid cell에서 예측과 분류를 수행하는 YOLO 모델의 특성을 이용하여 grid 분할 비율을 조정함으로써 검출 정확도를 높였다. 정확도를 측정하는 지표로는 mAP(mean Average Precision)을 사용했다.

또한 본 논문이 고안한 시스템은 실차 적용을 위해 인식한 정보를 바탕으로 판단하기 위해 객체 검출 결과에서 객체까지의 거리를 계산하는 알고리즘을 포함한다. 단안 카메라를 이용한 거리 추정에 관한 여러 연구가 진행되어왔다(Elzayat et al., 2018). 싱글 카메라는 스테레오 카메라에 비해 감지 불가 영역이 없다는 것과 두 카메라의 까다로운 calibration으로부터 자유롭다는 것, 그리고 더 저렴한 가격을 장점으로 가지고 있다. 객체 검출 결과인 bounding box의 좌표를 이용하여 거리를 추정하여 판단에 사용하도록 하였다. 카메라를 통해 추정한 거리는 종방향 및 횡방향 거리 정확도가 더 높은 라이다 센서를 통해 교차 검증하였다.

II. 관련연구

1. YOLOv4

YOLO는 딥러닝 프레임워크인 Darknet에서 제공하는 CNN(Convolutional Neural Network) 기반 객체 인식 네트워크이며 최초로 One-stage detection 방식을 고안해 실시간으로 객체 검출이 가능하도록 한 모델이다 (Redmon et al., 2016). 다양한 전처리 모델과 인공 신경망을 결합해서 사용하는 구조의 기존 객체 검출 모델과 다르게 YOLO는 위 과정이 통합되어 있어 구조가 간단하고 검출 속도가 빠른 것이 특징이다. ‘YOLO’라는 이름은 ‘You Only Look Once’의 약자로, 이미지 전체를 한 번만 본다는 YOLO의 또 다른 특징을 나타낸다. 이는 class에 대한 맥락적인 이해를 높이는 효과가 있다.



<Fig. 1> Process model of the YOLO object detection

Type	Filters	Size	Output
Convolutional	32	3 × 3	256 × 256
Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1
	Convolutional	64	3 × 3
Residual			128 × 128
Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1
	Convolutional	128	3 × 3
Residual			64 × 64
Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1
	Convolutional	256	3 × 3
Residual			32 × 32
Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1
	Convolutional	512	3 × 3
Residual			16 × 16
Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1
	Convolutional	1024	3 × 3
Residual			8 × 8
Avgpool		Global	
Connected		1000	
Softmax			

<Fig. 2> Architecture of Darknet-53

YOLO의 검출 방식은 다음과 같다. 입력 이미지를 S*S 비율의 grid로 분할한 뒤 각 grid cell에서 bounding box와 이 box들의 confidence와 class 확률을 예측한다. <Fig. 1>은 이 과정을 간단히 나타낸 이미지이다. YOLOv4에 사용되는 YOLOv3은 YOLO에서 보완된 방법으로 box가 아닌 grid cell마다 해당 연산을 하는 방식으로, 각 cell은 객체가 있을 확률, anchor box 정보, class 별 확률을 계산한다. Anchor box는 cell마다 갖는 사전에 지정된 비율의 box이다.

YOLO는 여러 버전을 거쳐 정확도와 속도를 높여왔다(Redmon and Farhadi, 2018). YOLOv4는 당시 최신 딥러닝 기법을 활용하여 모델을 설계하고 학습시켜 정확도 및 속도를 끌어올리는 연구를 진행하였고, 가장 성능이 높은 최적의 딥러닝 기법 조합을 소개했다(Bochkovskiy et al., 2020). 결과적으로 YOLOv4 네트워크는

Backbone 네트워크인 CSPDarknet53, Neck 네트워크인 SPP&PAN, 그리고 Head 네트워크인 YOLOv3으로 구성되어 있다. YOLO의 고질적 문제인 작은 객체 검출 시 성능 저하되는 문제를 해결하기 위해 입력 해상도를 높였고(608*608), receptive field가 높아진 해상도를 커버하기 위해 layer를 증가시켰다. 그리고 크기가 다른 여러 개의 객체를 검출하는 모델 용량을 늘리기 위해 더 많은 매개 변수를 사용하였다. 그에 따른 연산량의 증가는 더 향상된 backbone 네트워크를 통해 보완하였다. CSPDarknet53은 CSPNet으로부터 아이디어를 가져와 기존 YOLOv3의 Backbone 네트워크인 Darknet53과 결합한 네트워크이다. CSPDarknet에 적용된 CSPNet의 핵심 아이디어는 input feature map을 둘로 나누어 한 부분만 연산을 거치고 나머지 한 부분은 연산을 거치지 않도록 한 뒤 나중에 다시 합쳐지도록 하여 여러 cost를 줄이는 것이다. <Fig. 2>는 YOLOv3과 v4의 backbone 이 되는 네트워크인 Darknet-53의 구조이다.

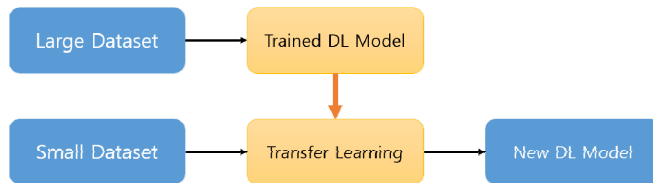
YOLOv4 이후에도 객체 검출 성능을 높이기 위한 연구가 계속 진행되어 YOLOv5, PP-YOLO(Long et al., 2020) 등의 네트워크가 공개되었다. 이중 가장 대중적으로 사용되는 모델은 YOLOv5와 YOLOv4이다. 검출 속도는 YOLOv5가 비교적 더 낮지만 YOLOv4는 FPS 성능을 최적화할 수 있고, YOLOv5와 달리 논문과 함께 배포되어 여러 최신 기법을 통한 실험 결과를 보여주며 아키텍처에 대한 이해가 수월하도록 돕는다는 점에서 두 모델의 차이가 있다. YOLOv5는 YOLO의 이름을 이어받기는 했으나 공개될 당시 v4에 비해 기존 YOLO가 가진 특징에서 벗어난 경향이 있다는 평가를 받았다(Zhang et al., 2021). YOLOv4가 YOLO 알고리즘을 응용하는 측면에서 목적에 부합했기 때문에 본 논문은 YOLOv4를 이용하여 연구를 진행하였다.

2. 전이학습

1) 전이학습

일반적으로 CNN 기반 딥러닝 모델을 제대로 훈련하려면 아주 방대한 양의 데이터가 필요하다. 이러한 목적으로 만들어진 데이터셋으로는 COCO Dataset, KITTI Dataset 등이 있다. 이런 데이터셋을 만들기 위해서는 많은 비용과 시간이 드는데, 새로운 문제마다 마련하는 일은 현실적으로 쉽지 않다. 이런 어려움을 해결하기 위해 나온 학습법이 전이학습(Transfer Learning)이다(Pan and Yang, 2010).

전이학습이란 아주 큰 데이터셋으로 훈련된 모델의 가중치를 가지고 와서 해결하고자 하는 과제에 맞춰 재보정을 하여 사용하는 것이다. 따라서 적은 수의 데이터를 가지고 원하는 과제를 수행하는 딥러닝 모델을 훈련할 수 있게 된다. 과거에 문제들을 해결하는 인간의 학습법과 유사하다.



<Fig. 3> Transfer Learning

본 논문은 darknet53.conv.74 가중치를 이용하여 전이학습을 수행하였다. 방대한 데이터셋으로 사전학습된 가중치를 도심로에서 볼 수 있는 객체들을 검출하도록 자체 데이터셋을 이용하여 학습시킴을 통해 도심로 객체 인식이라는 목적에 맞는 가중치를 새롭게 만들어 냈다. <Fig. 3>은 전이학습의 프로세스를 나타낸 이미지이다.

Ⅲ. 본 론

1. 제안하는 방법

앞서 설명했듯이 YOLO의 객체 검출 방식은 입력 이미지를 $S \times S$ grid로 분할하여 각 grid의 cell마다 예측을 수행하는 것이다.

시험 차량에 장착된 카메라의 사이즈는 가로 1920픽셀, 세로 1080픽셀의 16:9 비율이다. YOLOv4의 네트워크는 해당 카메라로부터 입력받은 이미지를 가로 N , 세로 N 의 1:1 비율 grid로 분할한다. 한 grid cell의 모양은 가로로 긴 직사각형이 된다. 한 grid cell은 하나의 예측을 한다. 본 연구에서 사용되는 데이터셋의 검출 대상이 되는 class 중 cone, stick, drum, person은 가로 폭이 짧고 세로 폭이 긴 비율을 가진다. 같은 이미지 면적 안에서 가로로 긴 직사각형의 cell이 정사각형의 cell보다 이러한 object들을 더 많이 포함할 확률이 높다. 따라서 본 논문은 이러한 관점에서 YOLOv4의 입력단이 grid를 분할할 때 차량에 장착된 카메라의 비율과 같은 16:9로 분할하도록 했다. 기존 608*608 input에서 800*450 input으로 변경하였다.

<Fig. 4, 5>는 이미지가 입력되면 사진에 지정된 비율에 따라 이미지가 분할되어 각 비율의 grid cell 모양에 생기는 차이를 나타내기 위한 예시이다. <Fig. 4>는 기존의 YOLO 네트워크가 갖는 $S \times S$ 비율의 분할이고 <Fig. 5>은 제안하는 방법의 16:9 비율 분할이다.

좌측 아래에 drum을 포함하는 cell을 예시로 들면, 위 이미지에선 가로로 긴 직사각형 cell에 세 개의 작은 drum이 포함되는데 아래 이미지에선 정사각형의 cell로 나뉘며 두 개의 cell만 포함하도록 바뀐다. 따라서 가로 폭이 좁고 세로로 긴 물체들을 더 효과적으로 검출할 수 있다.



<Fig. 4> Image segmented by $S \times S$ grid



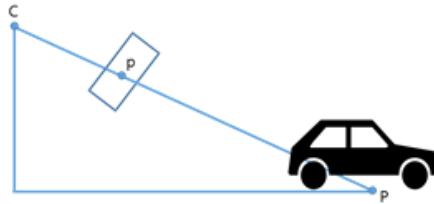
<Fig. 5> Image segmented by 16 x 9 grid

2. Bounding box 기반 거리 추정

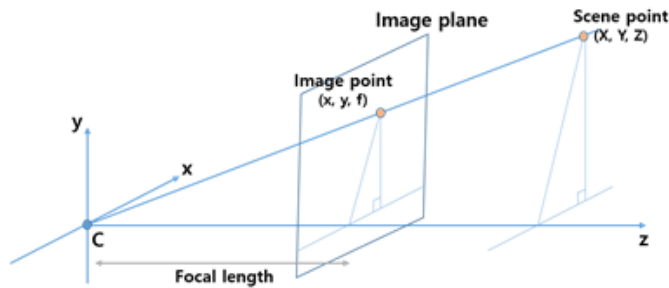
본 논문 제안하는 시스템은 실차 적용을 위해 객체 검출 결과로부터 거리 추정을 수행한다. 라이다, 레이더 등의 센서 없이 카메라를 이용한 딥러닝 기반 객체 추정 방법으로는 bounding box의 크기를 이용한 방법 (Elzayat et al., 2018)과 물체 검출과 거리 추정 각각 예측 모델을 사용하여 융합하는 방법(Cao et al., 2018)이 있다. 본 논문에서 사용된 방법은 bounding box를 이용한 방법으로, box의 밑변 중점을 object와 지면이 맞닿아 있는 지점으로 보고, 해당 지점과 카메라가 장착된 차량과의 상대적인 좌표를 계산하는 방식이다.

<Fig. 6>의 점 C는 카메라이고, 점 p는 이미지 평면에서의 object의 좌표이다. 점 P는 객체의 지면 좌표이다. 카메라의 월드 좌표는 C_w , 픽셀 좌표는 C_p 이고, 점 p의 월드 좌표는 p_w 이고, 픽셀 좌표는 p_p 이다. 거리

추정은 Homography matrix를 이용한 알고리즘을 사용하였다(Kriegman, 2007). Homography는 한 평면에서 다른 평면으로 점을 변환하는 평면 관계이다. Homography matrix는 평면의 2D 점을 나타내는 3차원 벡터를 변환하는 3x3 행렬이다. Homography는 직접적인 투영관계에 있는 두 평면 사이뿐 아니라 투영관계에 의해 직·간접적으로 연결되는 모든 평면들 사이에서도 일반적으로 성립한다. 이러한 벡터를 동종 좌표라고 한다. 아래 <Fig. 7>은 동종 좌표에 대해 설명한다.



<Fig. 6> The relationship between camera origin C and object position P

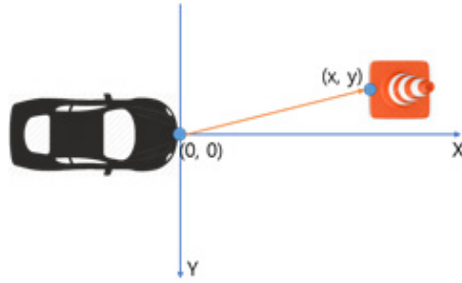


<Fig. 7> Homogeneous coordinates

동종 좌표는 컴퓨터 비전 내의 점을 나타내는 투영 좌표이다. 사진을 찍을 때 3D에서 2D로 변환되기 때문에 깊이 스케일이 손실된다. 그러나 좌표는 스케일을 통해서만 다르기 때문에 동일하다. 동종 좌표가 주어지면 모든 요소를 벡터의 마지막 요소(스케일 인자)로 나누면 데카르트 좌표는 마지막 요소를 제외한 모든 요소를 구성하는 벡터가 된다. <Fig. 7>은 이러한 내용을 도식화한 것이다. 이미지 평면의 삼각형이 실제로 더 멀고 더 클 수 있지만 여전히 동일한 이미지를 생성할 수 있다.

이러한 정보를 바탕으로 객체까지의 상대거리를 구하는 방법은 다음과 같다. 이미지 평면과 지면과의 변환 관계, 즉 두 관계에 대한 Homography matrix를 구하여 object의 지면에 맞닿은 지점의 이미지 픽셀 좌표(x, y)가 실제 월드좌표(X, Y, Z)의 Z=0 평면에서 대응되는 점을 구하는 방식이다. 차량 번호판 중앙 위치에서 지면으로 수선을 내려 만나는 지점을 (0, 0, 0)으로 두고 상대 x, y 좌표를 구하였다. 객체의 위치는 객체가 바닥에 맞닿은 지점으로 설정하였고, bounding box의 밑변 중앙 지점에 해당한다. 다음 <Fig. 8>은 차량을 기준으로 한 x, y 좌표축과 상대거리를 나타낸다.

객체의 위치에 해당하는 픽셀 좌표는 object detector로부터 bounding box 좌표를 입력받은 뒤 box의 밑변 중점에 사전에 계산된 Homography Matrix를 적용한다. 적용된 좌표는 이미지 평면과 실제 월드 좌표의 z=0 평면과의 관계가 적용된 실제 월드 좌표로 간주된다.



<Fig. 8> Distance from an object

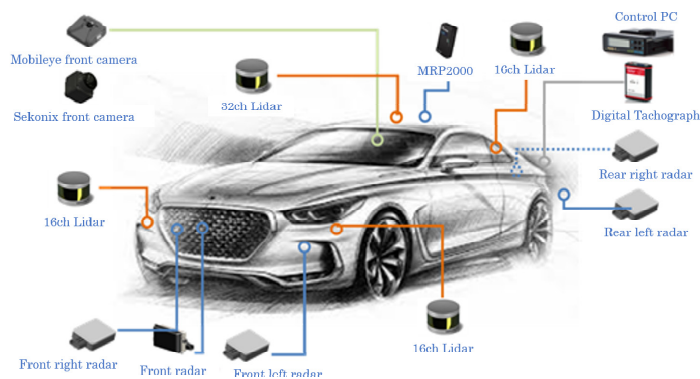
IV. 실험 결과 및 고찰

1. 시스템 구성

알고리즘의 성능 검증은 자율주행 플랫폼이 장착된 실차로 진행하였다. 차량은 현대 제네시스 G70로, PC와 전방카메라, 라이다, 레이더 등 각종 인식 센서들이 장착되어있다. 다음 <Fig. 9>은 실험 및 데이터 취득에 사용된 자율주행 차량의 외부 모습이다. <Fig. 10>는 <Fig. 9>의 차량에 장착된 센서의 구성을 나타낸 그림이다.



<Fig. 9> The exterior of the vehicle



<Fig. 10> The car with self-driving platform

2. 데이터셋

본 연구에서 고안된 딥러닝 모델은 일반적인 도심로 주행 상황에서 마주칠 수 있는 객체를 검출하고 회피하기 위해 고안되었다. 따라서 데이터셋의 검출 대상이 되는 총 15개의 class는 도심로 위 공사 구간에서 볼 수 있는 라바콘(cone), PE드럼(drum)과 차선 규제 시선 유도봉(stick), 정지선(stopline), 횡단보도(crossline), 신호등의 4가지 신호, 차량 4가지, 보행자(person)을 포함한다. 데이터셋의 class 목록을 한눈에 볼 수 있도록 <Table 1>로 정리하였다.

학습 및 검증을 위한 데이터셋 이미지는 충주 시내, 대구 알파시티, 화성 k-city 등 국내 도심로에서 시험 차량으로 주행하며 직접 취득한 가로 1920픽셀, 세로 1080픽셀의 jpg 이미지 28,570이다. 이미지들은 앞서 ‘시스템 구성’에서 소개한 플랫폼이 장착된 차량으로 국내 도심로 주행상황을 촬영한 영상으로부터 얻은 뒤 labeling tool을 사용하여 annotation을 생성했다. ‘도심로 주행’이라는 의도에 적합한 cone, stick, drum, 그리고 신호등 class를 모두 포함하는 조건을 가지며 추후 연구에 사용될 가능성을 고려하여 직접 데이터셋을 고안하기 위해 마련하였다.

데이터셋의 구성은 학습용 25,075장과 테스트용 3,500장으로 이루어져 있다. 아래의 <Fig. 11, 12>은 해당 데이터셋 이미지의 예시이다.

<Table 1> Classes

class ID	class
0	cone
1	stick
2	stopline
3	crossline
4	red_light
5	yellow_light
6	green_light
7	str_and_left
8	left_turn
9	drum
10	bus
11	bicycle
12	truck
13	car
14	person



<Fig. 11> An example of Dataset Images 1



<Fig. 12> An example of Dataset Images 2

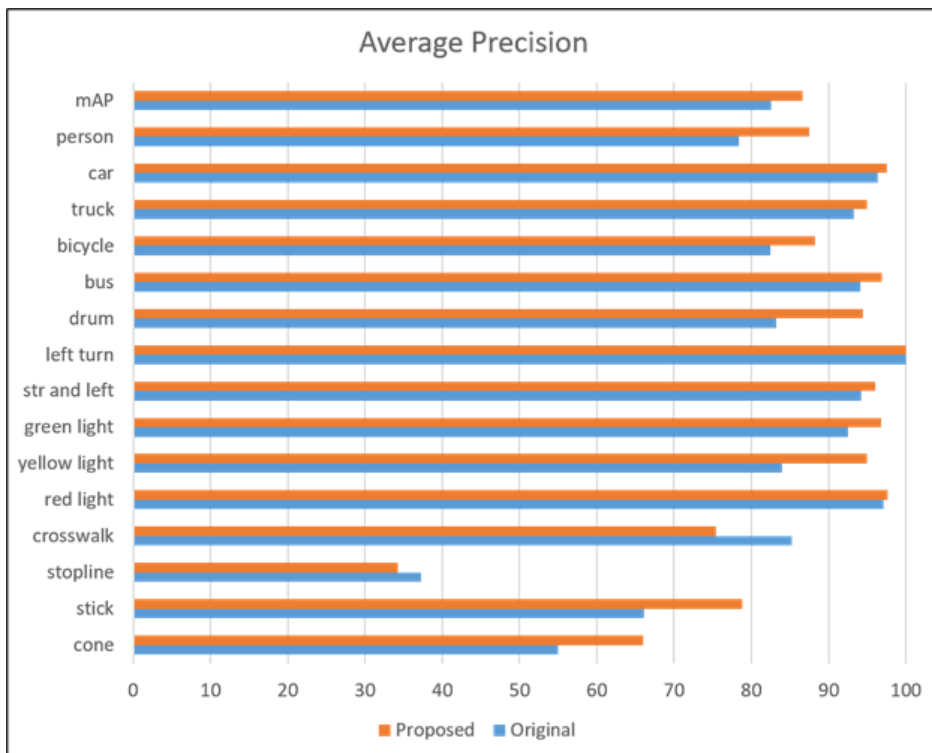
3. 객체 검출 모델 성능 실험

제안하는 객체 검출 및 거리 추정 시스템에서 객체 검출에 해당하는 detector의 검출 성능을 측정하기 위해 클래스별 AP(Average Precision)를 계산하였다. AP 계산에 사용된 데이터는 직접 취득한 데이터셋 이미지 3,500장이다.

<Fig. 13>은 모든 클래스의 mAP와 객체별 AP를 거시적으로 비교할 수 있도록 나타낸 그래프이다. ‘Original’의 데이터는 기존 YOLOv4 네트워크로 테스트한 결과를, ‘Proposed’의 데이터는 제안하는 네트워크로 테스트한 결과를 나타낸다.

<Table 2>는 객체별 AP 값을 나타낸 표이다. ‘Original’ 열의 데이터는 기존 YOLOv4의 테스트 결과를, ‘Proposed’ 열의 데이터는 제안하는 네트워크의 테스트 결과를 나타낸다. 동일한 테스트셋을 두 네트워크에 입력한 결과를 비교하였다. 전체 mAP가 4.03% 증가하였고, 신호등의 다섯 개 신호 각각 AP가 증가하였다. cone과 stick의 AP도 각각 증가하였다. 반면 stopline과 crosswalk에서는 AP가 감소하였다.

<Fig. 14-17>은 기존 YOLOv4 네트워크와 제안하는 네트워크를 같은 시나리오를 입력하여 객체 검출을 수행한 결과이다. <Fig. 14와 15>, <Fig. 16과 17>이 각각 같은 시나리오이다. <Fig. 14>에서 검출되지 않은 작은 작은 drum이 <Fig. 15>에서 검출되었다. 또한 <Fig. 16>에서 검출되지 않은 겹쳐진 부분의 cone, 그리고 중앙의 crossline이 <Fig. 17>에서 검출된 것을 확인할 수 있다. 검출 속도는 테스트셋 평균 장당 0.0175초로, 실시간에 적합한 속도를 보였다.



<Fig. 13> Average Precision comparison by object (above: Original YOLOv4, below: Proposed method)

<Table 2> Average precision by classes

unit	Original	Proposed	Variance
cone	54.98	65.99	11.01
stick	66.09	78.85	12.76
stopline	37.24	34.28	-2.96
crosswalk	85.19	75.39	-9.8
red light	97.12	97.59	0.47
yellow light	83.99	94.97	10.98
green light	92.54	96.74	4.2
str and left	94.23	96.07	1.84
left turn	100	100	0
drum	83.18	94.40	11.22
bus	94.07	96.94	2.87
bicycle	82.46	88.22	5.76
truck	93.25	94.99	1.74
car	96.33	97.56	1.23
person	78.38	87.49	9.11
mAP	82.60	86.63	4.03



<Fig. 14> The detection result of the original network 1



<Fig. 15> The detection result of the proposed network 1



<Fig. 16> The detection result of the original network 2



<Fig. 17> The detection result of the proposed network 2

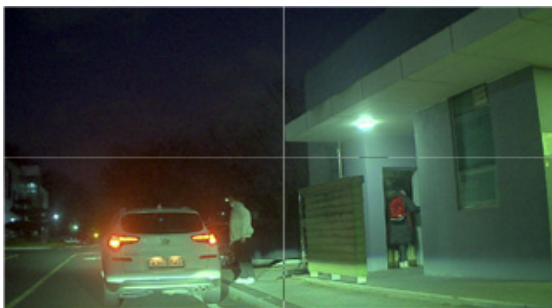
4. 거리 추정 알고리즘 성능 실험

거리 추정 알고리즘의 성능은 거리 정확도가 높은 것으로 알려진 라이다 센서로 교차 검증하였다. <Fig. 18, 19>은 각각 카메라로 본 검출 대상 차량과, 라이다로 전방의 검출 대상 차량을 감지한 장면이다. <Fig. 19>은 라이다에서 인식하는 차량의 주변 환경을 top-view로 바라본 모습이다. 그림의 오른쪽 방향이 차량 전방에 해당된다. 빨간 직사각형으로 표시된 부분은 라이다로 검출된 대상 차량의 모습이다.

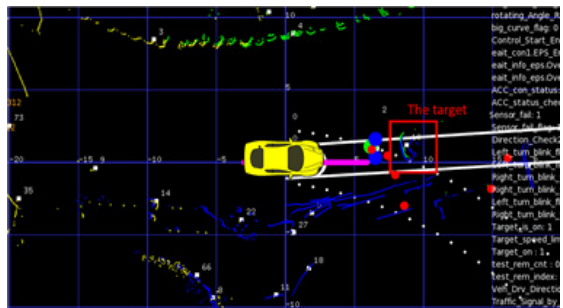
라이다를 통해 측정된 대상 차량까지의 거리를 기준으로 제안하는 알고리즘을 이용해 계산한 거리값을 비교한 뒤 오차를 측정하는 방식으로 거리 계산 알고리즘의 성능을 검증하였다. 10m부터 50m까지 10m씩 증가시킨 거리별로 5회 시험을 진행하여 평균 오차를 계산하였다. 거리별 X, Y 좌표 추정 거리 좌표값 오차의 절댓값을 아래 <Table 3>에 나타내었다. X 방향은 차량의 전방 방향을, Y 방향은 X 방향에 수직인 차량의 우측 방향을 나타낸다. 실험 결과 X 좌표 및 Y 좌표에서 각각 거리 추정 결과와 라이다로 측정한 결과의 평균 오차는 각각 5.25m, 0.97m이다. X, Y 좌표 모두 20m에서 가장 오차가 낮고 50m에서 가장 오차가 높았다. 전반적으로 오차는 Y 좌표보다 X 좌표에서 더 크게 나타났다.

<Table 3> Absolute Value of Estimated Distance Error by Distance

Distance (m)	X coordinate (m)	Y coordinate (m)
10	3.40	0.45
20	1.81	0.20
30	5.12	0.96
40	7.78	1.02
50	8.12	2.24
Mean Error (m)	5.25	0.97



<Fig. 18> The Target Car



<Fig. 19> Detection of the target car via Lidar

5. 고찰

제안하는 네트워크의 객체 검출 성능 실험과 거리 추정 성능 실험 결과는 다음과 같다. 객체 검출 결과는 mAP로 측정하였으며 기존 YOLOv4 알고리즘 대비 4.03% 증가하였다. 속도는 장당 평균 0.0175초로, 초당 약 57프레임을 처리하는 실시간성을 보였다. 거리 추정 결과는 거리별 X, Y 좌표 오차의 절댓값으로 측정하였

고 측정 거리는 10m부터 50m까지 10m씩 증가한다. X 좌표의 모든 거리 평균 오차 절댓값은 5.25m이고, Y 좌표는 0.97m이다.

객체 검출 부분에서 전반적인 성능 향상이 있었다. 제안하는 방법으로 가로 폭이 좁아진 grid cell은 cone, drum, stick, person 등의 가로 폭이 좁은 객체들의 인식 정확도를 높였고 그 외 신호등의 각 신호와 차량 등의 객체에서도 성능이 향상되었다. 또한 비슷한 색과 특징을 가지며 모양이 다른 cone, drum, stick을 각각 효과적으로 분류하였다. 동시에 기존의 모델 속도인 65 FPS에서 크게 저하시키지 않았다. 네트워크 조정 방법이 이미지 비율에 맞지 않아 적지만 손실되는 정보를 보완하는 것을 알 수 있다. 특히 작은 물체의 인식을 증가를 예시로 확인하였다.

거리 추정 성능은 Y 좌표에 비해 X 좌표의 정확도가 낮게 나타났다. 이것은 예측시 생성되는 bounding box의 위치가 정확하지 않은 것과 카메라 센서의 특징에 의해 종방향보다 횡방향 정보가 더 정확한 것을 원인으로 볼 수 있다. 단안 카메라에서 얻을 수 있는 정확도의 한계가 있음을 알 수 있다. 이렇게 추정된 거리 값은 다른 센서 측정 결과와 융합되어 더 높은 정확도의 객체 인식에 사용될 수 있도록 하거나 딥러닝을 사용한 거리 추정 정확도를 높이는 방법을 고안하여 활용할 수 있도록 연구되어야 할 것으로 보인다.

V. 결론 및 향후 연구 방향

본 논문은 딥러닝 네트워크를 사용하여 자율주행 차량의 객체 검출부터 거리 추정까지 수행하는 시스템을 구축하였다. 객체 검출 부분의 grid 분할 비율을 입력 이미지의 비율과 맞추어 YOLO 네트워크의 특성을 이용한 객체 인식 성능을 개선하는 방법을 제안하였다. 특히 세로 대비 가로 폭이 좁은 객체들의 정확도 향상이 있었다. 또한 검출 결과를 기반으로 거리 추정을 이루는 시스템으로 실차 적용 가능성을 살펴보았다.

이러한 결과를 바탕으로 향후 도로 주행상황에서 마주치는 객체들의 특징과 네트워크의 특성을 이용하여 네트워크를 변형하여 객체 검출 성능을 개선하는 방식을 더 세분화하여 해당 네트워크의 성능을 개선시키는 방법을 연구할 예정이다. 또한 검출 대상 객체가 다양한 만큼 각 객체별 특성을 이용하여 거리 추정 정확도를 높이는 방법을 찾고, 추정된 거리 정보는 추후 다른 센서와 융합하여 효과적으로 환경 인식에 사용될 수 있도록 할 예정이다.

ACKNOWLEDGEMENTS

본 논문은 산업통상자원부가 지원한 ‘자율주행기술개발혁신사업’의 지원을 받아 수행된 연구 결과입니다. [과제명: Lv.4 자율주행시스템의 FailOperational 기술개발 / 과제번호: 20018055]. 본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었음(과제번호 22AMDP-C160501-02). 또한 이 연구는 2022년도산업통상자원부가 지원한 ‘자율주행기술개발혁신사업’의 지원을 받아 수행된 연구 결과입니다. [과제명 : 수요응답형 자동발렛주차 및 서비스 기술 개발 / 과제번호 : 20018448]

REFERENCES

- Bochkovskiy, A., Wang, C. Y. and Liao, H. Y. M.(2020), *Yolov4: Optimal speed and accuracy of object detection*, arXiv preprint arXiv:2004.10934.
- Cao, Y., Wu, Z. and Shen, C.(2018), “Estimating Depth from Monocular Images as Classification Using Deep Fully Convolutional Residual Networks”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp.3174-3182.
- Elzayat, I. M., Ahmed, S. M., Mostafa, M. M., Mahmoud, H. R., El Munim, H. A., Ghoneima, M., Darweesh, M. S. and Mostafa, H.(2018), “Real-Time Car Detection-Based Depth Estimation Using Mono Camera”, *30th International Conference on Microelectronics(ICM)*, IEEE, pp.248-251.
- Kriegman, D.(2007), “Homography estimation”, *Lecture Computer Vision I, Computer Science Engineering(CSE)*, p.a 252.
- Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., Shen, H., Ren, J., Han, S., Ding, E. and Wen, S.(2020), *PP-YOLO: An effective and efficient implementation of object detector*, arXiv preprint arXiv:2007.12099.
- Pan, S. J. and Yang, Q.(2010), “A Survey on Transfer Learning”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp.1345-1359.
- Redmon, J. and Farhadi, A.(2018), *YOLOv3: An Incremental Improvement*, arXiv preprint arXiv:1804.02767.
- Redmon, J., Divvala, S., Girshick, R. and Farhadi, A.(2016), “You only look once: Unified, real-time object detection”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.779-788.
- Zhang, Y., Li, X., Wang, F., Wei, B. and Li, L.(2021), “A Comprehensive Review of One-stage Networks for Object Detection”, *2021 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, 1-6.