

딥러닝을 활용한 단안 카메라 기반 실시간 물체 검출 및 거리 추정

Monocular Camera based Real-Time Object Detection and Distance Estimation Using Deep Learning

김현우¹, 박상현[†]

Hyunwoo Kim¹, Sanghyun Park[†]

Abstract: This paper proposes a model and train method that can real-time detect objects and distances estimation based on a monocular camera by applying deep learning. It used YOLOv2 model which is applied to autonomous or robot due to the fast image processing speed. We have changed and learned the loss function so that the YOLOv2 model can detect objects and distances at the same time. The YOLOv2 loss function added a term for learning bounding box values x , y , w , h , and distance values z as classification losses. In addition, the learning was carried out by multiplying the distance term with parameters for the balance of learning. we trained the model location, recognition by camera and distance data measured by lidar so that we enable the model to estimate distance and objects from a monocular camera, even when the vehicle is going up or down hill. To evaluate the performance of object detection and distance estimation, MAP (Mean Average Precision) and Adjust R square were used and performance was compared with previous research papers. In addition, we compared the original YOLOv2 model FPS (Frame Per Second) for speed measurement with FPS of our model.

Keywords: Deep Learning, Object Detection, Distance Estimation, Monocular Camera

1. 서 론

카메라 센서는 가격이 저렴하고 데이터 처리가 편리하고 물체검출에 용이하다는 장점이 있기 때문에 자율주행 자동차와 로봇의 주변 환경 인지 연구에서 사용되는 대표적인 센서이다. 그 외에 사용되는 센서는 라이다, 레이더 등이 있다. 이 센서들을 활용한 기술로 자동차의 운전자 안전 시스템인 ADAS (Advanced Driving Assistant System)가 있다^[1]. 대표적인 ADAS 기능 중 차선 유지 시스템(Lane Keeping Assistant System), 적응형 순항 제어(Adaptive Cruise Control), 긴급 제동 보조 시스템(Autonomous

Emergency Braking) 등이 있다. 이중 차선 유지 시스템의 경우 단안 카메라만으로 기능 구현이 가능하지만, 적응형 순항 제어나 긴급 제동 보조 시스템의 경우 전방 물체의 거리값이 필요하여 단안 카메라만으로는 기능 구현이 어렵다. 이러한 이유로 라이다, 레이더와 같은 다양한 센서를 활용하고 있다^[1].

하지만, 라이다의 경우 가격이 고가이며 데이터를 다루는데 어려움이 있다. 또한 레이더의 경우도 센서 처리의 어려움이 있다. 그 때문에 여러 센서 사용 시 기술 개발의 어려움이 존재한다. 이러한 이유로 단안 카메라 기반의 물체검출과 거리 추정을 다음과 같은 방법으로 연구가 진행되었다. 첫 번째는 물체 검출 모델을 돌려서 나온 검출된 물체의 박스 크기를 통해 회귀 모델링 하여 거리 값을 추정^[2]하는 방법이다. 두 번째는 단안 카메라 기반 물체 검출과 단안 카메라 기반 disparity map 예측 모델^[3]을 각각 돌려 결과 값을 융합하는 형태로 사용하는 방법이다. 첫 번째 방법의 경우 물체별로 회귀 모델을 구해야 하며 정확도가 높지 않을 뿐 아니라 카메라의 진동에 많이 취약하다. 두 번째 방법의 경우 물체검출과 거리 추출 모델을 각각 돌려야 하므로 실시간 물체검출 및 거리 추출이 힘들다.

Received : Oct. 10. 2019; Revised : Oct. 30. 2019; Accepted : Nov. 13. 2019

※ This research was supported by the Research Grant from KETI through the Korea Agency for Infrastructure Technology Advancement funded by the Ministry of Land, Infrastructure and Transport of the Korean government (Project No.: 17TBIP-C126306-01).

1. Principal Researcher, Mobility Platform Research Center, Korea Electronics Technology Institute, Gyeonggi, Korea (kimhyunwoo@keti.re.kr)

† Associate Professor, Corresponding author: Mobility Platform Research Center, Korea Electronics Technology Institute, Gyeonggi, Korea (shpark@keti.re.kr)

이에 따라 실시간으로 단안 카메라 기반의 물체 검출 및 거리 추출을 하나의 모델에서 동시에 할 수 있으며 학습된 모든 클래스 별 물체의 거리 추출이 가능한 모델 및 학습 방법을 제안한다. 이 학습 방법은 기존 물체 검출의 모델에서 Backbone Network를 변경하지 않고 출력 네트워크와 비용 함수만을 변경하여 학습함으로써 기존 검출모델의 장점을 살리며 거리 추출을 기능을 추가할 수 있다. 또한 거리 추출의 정확도를 높이고 카메라가 오르막이나 내리막에서의 변화되는 상황에서도 거리를 올바르게 추출하기 위해 학습에 사용된 데이터에서의 거리 값은 라이다로 취득한 값으로 학습하고자 하였다. 때문에 학습 시 Annotation 파일에는 이미지에서 물체에 대한 BBOX (Bounding Box) 정보와 함께 라이다로 취득한 물체의 거리 값이 저장되어 있어야 한다. 이러한 조건에 부합한 데이터로 KITTI DATA^[4]의 3D OBJECT가 있다. KITTI DATA의 3D OBJECT 데이터의 경우 카메라 이미지 데이터와 Annotation 파일을 제공하는데 Annotation의 경우 물체의 BBOX와 라이다로부터 취득한 거리 값이 같이 저장되어 있으므로 본 논문에서는 KITTI DATA의 3D OBJECT 데이터를 사용한다.

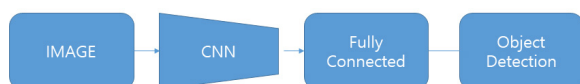
본 논문의 구성은 다음과 같다. 제 2장에서는 딥러닝 기반의 물체검출의 기본적인 원리를 다룬다. 3장에서는 물체검출 및 거리 추출을 동시에 할 수 있는 기초 모델 및 비용함수를 설명한다. 4장에서는 사용한 데이터와 실험 결과를 나타낸다. 5장에서는 본 연구의 결론과 향후 연구 방향을 제시한다.

2. 관련이론

딥러닝은 ILSVRC (imagenet large scale visual recognition competition)대회에서 AlexNet^[5]을 시작으로 VGGNet^[6], Google Net^[7]이 이 top-5 error를 6.7% 까지 떨어뜨렸고, ResNet^[8]이 top-5 error 3.5%로 사람의 top-5 error 5.1% 보다 적게 나왔다. 이에 따라 딥러닝은 물체 검출^[9-11], 영역 검출^[12,13] 등 다양한 곳에서 연구가 진행 되고 있다. 특히 물체 검출의 연구 결과는 현재 자율주행 자동차, 로봇 등에 적극 활용되고 있다.

2.1 딥러닝 물체 검출

딥러닝 기반의 물체 검출의 방식은 다음 [Fig. 1]과 같이 Convolution 레이어와 Fully Connected 레이어로 이루어져 있다.



[Fig. 1] Deep Learning Based Object Detection

CNN (Convolution Neural Network)을 이루고 있는 Convolution 레이어의 경우는 학습을 통해 검출이 잘 되도록 입력 이미지의 특징점을 찾는 역할을 하며 Fully Connected의 경우는 CNN 으로부터 전달받은 특징점으로 부터 검출을 잘하는 분류기가 되도록 학습되어 진다. 이러한 딥러닝 기반의 물체인식은 기존 컴퓨터 비전 기반의 물체 검출보다 높은 성능을 보이고 있다. 무엇보다 특정 환경이 아닌 다양한 환경에서도 높은 성능을 보여주고 있다. 이 외에도 영역검출, 자세 검출^[14] 등의 다양한 영상 인식 분야에서 두각을 나타내고 있어 연구 개발이 활발하게 이루어지고 있다. 대표적인 딥러닝 물체 검출 모델로는 SSD^[9], Faster R-CNN^[10], YOLOv2^[11] 등이 있다.

3. 딥러닝 기반 물체 검출 및 거리 추출 모델

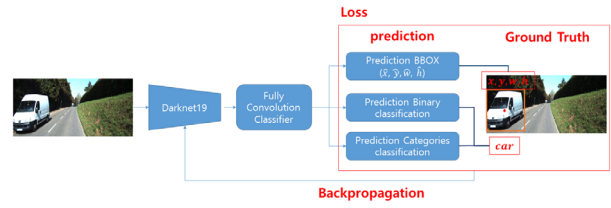
본 논문에서는 딥러닝을 활용하여 단안 카메라 기반 물체 검출 및 거리추출을 실시간으로 하기 위해 기존 딥러닝 물체 검출 모델의 출력 레이어와 비용 함수만을 변경하여 학습하는 것을 제안한다. 변경된 출력 레이어와 비용함수로 학습을 하여 물체의 위치 및 물체의 종류 뿐 아닌 검출된 물체의 거리 정보를 하나의 모델에서 같이 추출할 수 있다. 본 논문에서 사용된 기초 물체 검출 모델은 이미지 프레임당 처리 속도가 높은 YOLOv2 모델을 사용한다. 학습 시 KITTI 3D OBJECT DATA를 활용하여 학습한 후 성능을 검증한다.

3.1 딥러닝 물체 검출 모델 YOLOv2

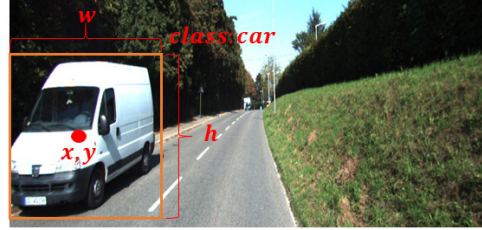
딥러닝 물체 검출의 경우 RPN (Region Proposal Network)^[10] 유무로 1-stage와 2-stage로 나뉜다. 2-stage는 RPN이 존재하는 물체검출 모델로 속도는 느리지만 정확도가 높고 1-stage는 속도가 빠르지만 정확도가 낮다. 자동차와 로봇과 같은 곳에서는 빠른 처리 속도를 요구하기 때문에 1-stage 모델이 많이 사용되며 본 논문에서도 1-stage의 대표모델인 YOLOv2를 기본 모델로 사용한다. YOLOv2의 모델을 살펴보면 10개의 Convolution 레이어와 5개의 Max Pooling 레이어를 통해 특징점을 추출하고 1x1 Convolution 레이어를 통해 Fully Connected 레이어를 대체하여 모델을 이루고 있다. 다음은 [Fig. 2]는 YOLOv2의 기반인 darknet-19 네트워크를 나타낸다^[11].

YOLOv2 기반의 물체 검출 학습에서는 이미지에서 위치를 물체가 존재하는 BBOX (Bounding BOX) 값(x, y, w, h)과 BBOX안에 어떤 물체인지를 분류하는 Binary classification, 어떤 물체인지를 결정하는 Categorical classification 비용 함수를 기반으로 학습 한다. 다음 [Fig. 3]은 대표적인 Block Diagram을 나타내고 있다.

Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

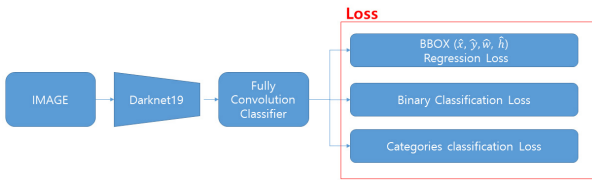


[Fig. 4] YOLOv2 Train Method Block Diagram

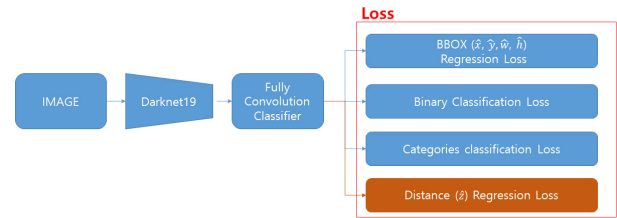


[Fig. 5] Annotation Information for YOLOv2

[Fig. 2] Network of darknet-19^[11]



[Fig. 3] YOLOv2 Loss function Block Diagram



[Fig. 6] Proposal Model Loss function Block Diagram

자세한 비용 함수의 식은 YOLOv1^[15]과 YOLOv2에 설명되어 있다. 학습을 위해서는 카메라 이미지 데이터와 물체별 BBOX (x, y, w, h)와 클래스 정보가 담긴 Annotation 파일을 기반으로 학습한다. 다음 그림 [Fig. 4]는 데이터에 따른 학습을 나타낸다.

위와 같은 출력 레이어와 비용함수 구조를 기반으로 카메라 이미지 데이터와 BBOX (x, y, w, h), 클래스 정보만 담긴 Annotation 파일로 학습하면 YOLOv2 모델은 이미지에서 물체의 위치와 종류만을 예측하게 된다. 여기서 Annotation에 담은 BBOX와 클래스 정보는 다음 [Fig. 5]에 나타내었다.

3.2 제안 하는 모델 및 비용 함수

제안하는 딥러닝 기반의 실시간 물체 검출 및 거리 추출 모델은 YOLOv2 모델에서 기존 카메라 이미지 데이터에서 물체가 존재하는 위치 값인 BBOX와 BBOX안의 물체가 어떤 물체 인지를 분류하는 형태에 추가로 거릿 값을 추출할 수 있도록 출력 레이어와 비용 함수를 변경하여 학습한다. 다음 [Fig. 6]은 제안하는 모델의 출력 레이어와 비용함수 Block Diagram을 나타내고 있다.

위 그림에서 추가된 Distance Regression Loss는 다음의 식 (1)과 같다.

$$\lambda_{zcoord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (z_i - \hat{z}_i)^2 \quad (1)$$

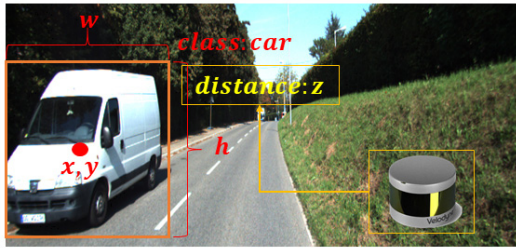
z_i : true distance

\hat{z}_i : prediction distance

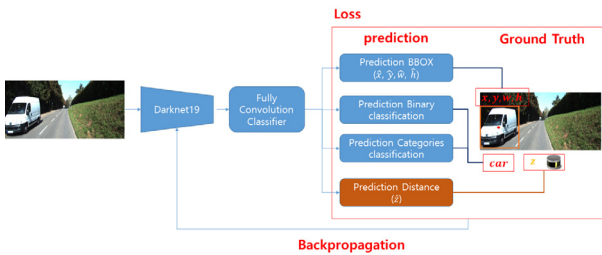
λ_{zcoord} : loss balance parameter

1_{ij}^{obj} : object presence

위 식에서 1_{ij}^{obj} 은 물체가 존재할 경우는 1 물체가 존재하지 않을 경우는 0으로 나타내는 것으로 모델이 물체가 없다고 판단했을 시 학습을 하지 않기 위한 파라미터이다. 또한 YOLOv2의 경우 앵커 박스를 설정하고 Grid Cell을 통해 예측 BBOX를 제한하여 학습하기 때문에 BBOX의 비용함수가 작게 발생한다. 즉, 제안하는 거리 추출 비용함수의 값이 BBOX보다 큰 값이 발생하는 경우가 많아 학습에 방해가 될 수 있다. 그러므로 각 비용함수의 균형을 맞추기 위한 균형 파라미터인 λ_{zcoord} 을 추가 하였다.



[Fig. 7] Annotation Information for Proposal Model



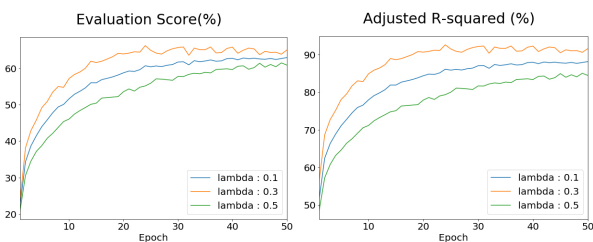
[Fig. 8] Proposal Model Train Method Block Diagram

3.3 모델 학습

3.2에서 설명한 바와 같이 기본 YOLOv2의 출력 레이어와 비용 함수를 변경하고 KITTI 3D OBJECT DATA를 활용하여 학습하였다. 다음 [Fig. 7]과 [Fig. 8]은 각각 사용된 데이터의 Annotation 정보와 모델과 데이터를 활용한 학습 구조를 나타낸 그림이다.

위와 같이 출력 레이어와 비용 함수를 변경하고 데이터를 구축한 후에 다음과 같은 학습 과정은 통해 학습한다.

1. 기본 YOLOv2 모델을 KITTI 3D OBJECT DATA의 이미지와 Annotation 파일에서 BBOX와 클래스 정보만 으로 학습 하여 pretrained model 완성.
2. 기존 YOLOv2 모델에서 비용 함수와 구조를 변경하고 만들어 놓은 pretrained model을 불러와 1에서 학습에 활용한 데이터의 Annotation에서 거리 값 정보를 추가로 불러와 fine tune 학습. 학습 과정에서 균형 파라미터인 λ_{zcoord} 은 다양하게 변형하며 학습을 진행하였으며 0.3일 때 학습이 안정적이었으며 성능이 가장 높았다. 다음의 [Fig. 9]는 균형 파라미터인 λ_{zcoord} 의 값에 따른 성능 지표를 나타낸 그림이다.



[Fig. 9] Performance Indicators by Parameter

3.4 모델 테스트

3.3 모델 학습과 같은 방법으로 학습한 후에 테스트할 때는 학습된 모델에 이미지 데이터만 입력으로 넣어주면 이미지에서의 물체의 위치, 종류, 거리 값을 같이 출력한다.

4. 실험결과

4.1 실험 환경

학습 및 테스트를 위해 사용한 프레임워크와 pc 환경은 다음 [Table 1]과 같다.

[Table 1] Experiment Environment

Deep learning Framework	Tensorflow 1.13.1
OS	Ubuntu 18.04
CPU	Intel core i7
Graphics Card	Nvidia GTX 1080 Ti (1 EA)

4.2 학습 데이터

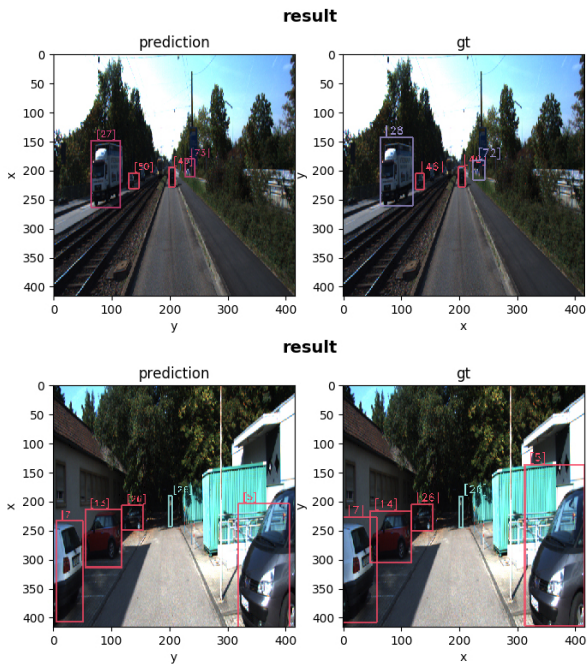
본 논문에서 제안하는 학습을 위해서는 이미지에 존재하는 물체의 BBOX (x, y, w, h)와 물체의 종류 외에 물체의 거리 값(z)도 필요하다. 또한 거리 값은 라이다로 취득한 값이어야 한다. KITTI 3D OBJECT DATA에서 제공하는 카메라의 이미지 데이터와 이미지 데이터 Annotation 파일은 본 논문에서 필요로 하는 이미지 객체에 대한 정보와 라이다로 취득한 객체의 거리 값이 포함되어 있다. 그 때문에 KITTI 3D OBJECT DATA를 본 논문에서 제안하는 모델의 학습 및 검증 데이터로 활용하였으며 총 7,848장의 데이터에서 7,000장은 학습데이터로, 1,000장은 검증데이터로 848장은 테스트 데이터로 사용하였다.

4.3 차량 검출 및 거리 추출

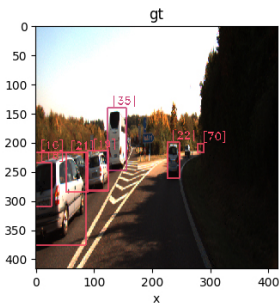
모델의 정확도를 확인하기 위해 차량 검출률 계산은 mAP (Mean Average Precision)을 사용하고 거리 정확도는 회귀의 정확도를 측정하는 Adjust R Square를 사용하였다. Adjust R Square 계산 시 미 검출된 물체는 포함하지 않았다. 다음 [Table 2]는 모델의 물체 검출과 거리 추출 정확도 및 속도를 나타낸다.

[Table 2] Test Result

Object Detection (mAP)	Distance Estimation (Adjust R Square)	FPS (Fram Per Second)
71.2%	92.3%	33 FPS



[Fig. 10] The Result Image about Prediction and Ground Truth



[Fig. 11] Kitti Image Data with Annotation Information

위 결과를 통해 기존 물체검출 결과의 가로 세로 길이를 통한 거리 예측 정확도인 80.4%^[2]보다 11.9% 높은 수치를 나타내고 있다. 추가로 RSME (Root Square Mean Error) 를 계산하면 값은 0.6123m를 확인 할 수 있었다. 속도 또한 기존 YOLOv2 모델의 35 FPS (Frame Per Second) 보다 2 FPS 낮은 것을 확인 하였다. 다음 [Fig. 10]은 물체검출 및 거리 추출을 나타낸 이미지의 일부이다.

5. 결 론

본 논문에서는 스테레오 카메라나 라이다, 레이더 등을 결합하여 물체검출과 거리 추출을 하던 방식을 대체할 단안 카메라 기반의 실시간 물체검출과 거리 추출을 하나의 모델에서 할 수 있는 방법을 제안한다. 또한 기존 물체 검출의 결과의 BBOX의 가로, 세로 값 기반으로 회귀 모델링하여 거리를 예측하는 방법보다 높은 성능을 내고자 물체검출과 함께 거리

추출을 같이 학습하고자 하였다. 무엇보다 기존의 방법은 클래스 별로 모델링을 모두 해야 하지만 제안하는 방법은 학습을 통해 클래스 별로 물체의 거리를 추출할 수 있다.

제안하는 모델을 자동차나 로봇에서 사용할 수 있도록 빠른 처리 속도를 가진 물체검출 1-stage의 대표 모델인 YOLOv2을 사용하였다. YOLOv2의 출력 레이어와 비용함수를 변경하여 학습하였다. 또한, 이미지의 물체에 따른 거리 값은 정확한 거리 값을 표기하고 있어야 한다. 특히 카메라가 오르막이나 내리막 같은 구간에서 변화된 구간에서도 정확한 거리 값을 취득해야 하므로 라이다를 통해 취득한 값이 거리 값으로 Annotation 파일에 표기되어 있어야 한다. KITTI 3D OBJECT DATA의 경우 이러한 조건을 만족하기 때문에 KITTI 3D OBJECT DATA를 사용하였다.

결과적으로 기존 딥러닝 물체검출 모델에서 출력 레이어와 비용함수를 변경하여 학습함으로써 물체 검출뿐 아니라 거리 추출이 가능하며 정확도 또한 기존 방법보다 높은 것을 확인 하였다. 그 뿐만 아니라 기존 모델의 속도인 FPS를 크게 저하시키지 않는다는 것을 확인 할 수 있다. 이러한 결과를 통해 기존 딥러닝 기반 물체 검출 모델에도 이러한 방법을 적용하면 물체 검출뿐 아닌 거리 추출이 가능할 것이다. 추가로 모델의 성능을 확인하기 위해 학습 데이터를 분석한 결과 다음 [Fig. 11]과 같이 하나의 물체에 여러 BBOX가 표기 되어 있으며 각 BBOX에 다른 거리 값이 표기되어 있는 것을 확인 할 수 있었다.

이러한 표기는 물체의 occlusion 영역을 나타낼 수 있으나 학습 시 동일 영역에서 같은 물체를 다른 거리 값으로 학습하기 때문에 모델의 성능을 감소시킬 수 있다. 추후 모델의 정확도를 높이하고자 occlusion 영역은 Annotation 파일에 포함하지 않은 데이터를 확보하여 학습하고자 한다. 또한, YOLOv2 뿐만 아닌, SSD, Faster R-CNN 등의 다른 모델에도 본 논문에서 제안하는 비용함수 및 학습 방법을 적용하여 모델별로 거리추출 성능을 비교 분석 후 AEB 시스템에 적용하여 실제 자동차에서 테스트 하고자 한다.

사 사

이 논문은 국토부의 재원으로 국토교통과학기술진흥원의 지원을 받아 수행된 연구사업 임(국토교통과학기술진흥원에서 부여한 과제번호 : 17TBIP-C126306-01).

References

[1] P. S. Heo, "A Study on the automotive ADAS market diffusion factors," *Symposium of the Korean Institute of communications and Information Sciences*, pp. 942-945, 2009.

- [2] I. M. Elzayat, M. A. Saad, M. M. Mostafa, R. M. Hassan, H. A. E. Munim, M. Ghoneima, M. S. Darweesh, and H. Mostafa, "Real-Time Car Detection-Based Depth Estimation Using Mono Camera," *2018 30th International Conference on Microelectronics (ICM)*, Sousse, Tunisia, 2018, DOI: 10.1109/ICM.2018.8704024.
- [3] Y. Cao, Z. Wu, and C. Shen, "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, 2018.
- [4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp. 3354-3361, 2012.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional," *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pp. 1097-1105, 2012.
- [6] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556 [cs.CV]*, 2015.
- [7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, DOI: 10.1109/CVPR.2015.7298594.
- [8] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, DOI: 10.1109/CVPR.2016.90.
- [9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *Computer Vision - ECCV 2016*, pp. 21-37, 2016.
- [10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pp. 91-99, 2015.
- [11] J. Redmon and A. Farhadi. "Yolo9000: Better, faster, stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 6517-6525, 2017.
- [12] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv: 1706.05587 [cs.CV]*, 2017.
- [13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, DOI: 10.1109/CVPR.2017.660.
- [14] A. Toshev and C. Szegedy. "Deeppose: Human pose estimation via deep neural networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp 1653-1660, 2014.
- [15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, DOI: 10.1109/CVPR.2016.91.



김현우

2015 홍익대학교 기계정보공학과(공학사)
 2017 한양대학교 미래자동차공학과(공학석사)
 2015~현재 전자부품연구원 연구원

관심분야: 인공지능, 컴퓨터 비전, 자율주행 자동차, 로봇틱스



박상현

2000 한국외대 컴퓨터공학과(공학사)
 2002 한국외대 컴퓨터공학과(공학석사)
 2017 한국외대 컴퓨터공학과(공학박사)
 2002~현재 전자부품연구원 책임연구원

관심분야: 컴퓨터비전, 임베디드 시스템, 자율주행자동차, 인공지능