Transactions of KSAE, Vol. 30, No. 8, pp.635-647 (August, 2022)

pISSN 1225-6382 / eISSN 2234-0149
DOI http://dx.doi.org/10.7467/KSAE.2022.30.8.635

자율 주행을 위한 딥러닝 기반 라이다 객체 인식 신경망 연구 분석

선 민 혁 · 백 동 희 · 공 승 현*

한국과학기술원 조천식모빌리티대학원

A Study on Deep Learning Based Lidar Object Detection Neural Networks for Autonomous Driving

Minhyeok Sun · Donghee Paek · Seung-Hyun Kong*

The Cho Chun Shik Graduate School of Mobility, Korea Advanced Institute of Science and Technology, Daejeon 34051, Korea
(Received 10 March 2022 / Revised 17 May 2022 / Accepted 19 May 2022)

Abstract : Object detection is one of the most crucial functions for autonomous driving because path planning, obstacle avoidance, and numerous other functions rely on the acquired information regarding the positions of objects on the road. To enable accurate object detection, numerous works utilize lidar as the primary sensor since it can accurately acquire 3D measurements and it is robust to adverse environmental conditions such as poor illumination. In this work, we aim to comprehensively review deep learning-based object detection using lidar, which has shown remarkable detection performance on various datasets. First, we explain the general concepts of deep learning-based lidar object detection along with the datasets and benchmarks that are commonly used in existing works. We then thoroughly discuss the latest state-of-the-art neural networks for lidar object detection. Finally, we provide suggestions on how to employ these networks in an autonomous driving system.

Key words : Autonomous driving(자율주행), Deep learning(딥러닝), Lidar(라이다), 3D Object detection(삼차원 객체 인식), Neural network(신경망)

1. 서 론

자율 주행 자동차는 크게 인지, 판단, 제어 3가지 핵심 기술로 구현된다. 그중 인지 기술은 사람의 감각기관과 같이 자동차 주변 여러 환경을 인식하는 기술로 인지 결과를 바탕으로 앞으로의 자동차 거동 판단과 이에 적절한 제어가 수행되므로 매우 중요한 기술이다. 인지 기술 중 객체 인식은 카메라, 라이다, 레이다 등 다양한 센서로부터 주변의 의미 있는 객체(차량, 보행자, 표지판 등)를 탐지하는 기술이다. 낮, 밤, 안개, 우천 등 자동차 주행에 있어 발생할 수 있는 다양한 상황과 관계없는 강인한 객체 인식 기술은 안전한 자율 주행 자동차 구현을 위해 필수적이다.

카메라 기반 객체 인식은 고해상도 이미지를 바탕으로 다양한 종류의 객체를 분류하고 탐지할 수 있지만, 빛이 적은 야간에는 검출성능이 현격히 떨어지는 등 주행 환경 변화에 대한 강인성이 부족하다. 또한 2차원 이미지의 형태로 주변 객체들을 감지함으로써 해당 객체의 정확한 3차원 공간적 정보를 수집하는 데 한계가 있다. 그에 반하여 라이다는 적외선 레이저 펄스 신호를 활용함으로 카메라와 달리 빛이 적은 야간에도 활용할 수 있고 정확한 3차원 정보를 검출할 수 있는 센서이다. 라이다는 송신기에서 적외선 레이저 펄스를 주변에 주사하고, 특정 지점에서 반사되어 수신기에 펄스가 도착하는 시간(Time of flight)을 측정하여 해당 장애물과의 거리를 측정한다. 자율 주행 자동차에 일반적으로 장착되는 라이다는 레이저 송신기와 수신기를 묶은 채널(Channel)을 수직 방향으로 다수 장착하여 상하 시야각을 구성하고 이를 회전하여 전 방향에 대해 장애물을 탐지할 수 있다. 라이다는 포인트 클라우드 형태로 데이터를 출력한다. 포인트 클라우드는 포인트들의 집합이며, 포인트는 레

*Corresponding author, E-mail: skong@kaist.ac.kr

이저 펄스가 반사되어 들어오는 한 지점에 대해 라이다 센서의 x, y, z축 기준 거리 정보를 담고 있다. 포인트 클라우드는 이미지 데이터의 픽셀과 달리 각각의 포인트에 대해 순서가 정해져 있지 않으며 규격화 되어 있지 않은 특징이 있다.

기존 연구들에서는 포인트 클라우드 데이터를 바탕으로 주변에 장애물이 어디에 위치하는지 파악하기 위해 여러 휴리스틱한 기술들을 활용하여 물체를 검출하였다. 예를 들면 유클리디언 군집화 기술은 무작위로 선택된 포인트에 대해서 특정 거리 내에 있는 모든 포인트들을 하나의 군집으로 편성한다.[2] K평균 군집화 기술은 미리 해당 장면에서 몇 개의 군집으로 분할할 것인지 지정하고, 사람이 직접 설계한 함수를 바탕으로 포인트들을 분할하여 군집을 편성한다.[3] 위 방식들은 직관적으로 포인트들을 군집화하여 장애물을 인식할 수 있지만, 사람이 직접 하이퍼 파라미터를 수정해야 하며, 이러한 파라미터들은 특정 상황마다 최고의 성능을 내기 위한 값이 변하므로 다양한 주변 환경에 강인하지 못한 단점이 있다. 최근 딥러닝 기술의 급격한 발전으로 딥러닝 기반 라이다 객체 인식 연구가 매우 활발히 진행되고 있다. 딥러닝 기반 라이다 객체 인식은 신경망 스스로 다양한 상황의 데이터를 통해 풍부한 특성을 추출하도록 학습하여 기존 휴리스틱한 기술들과 비교했을 때 강인성과 성능이 크게 앞서고 있다.

포인트 클라우드 데이터는 각 포인트 간 순서가 없고 규격화되지 않는 특징 때문에 이를 곧바로 YOLO,[4-6] R-CNN[7-10]와 같이 잘 알려진 이미지 기반 객체 인식 신경망에 입력하여 적용하는 것이 불가능하다. 따라서 포인트 클라우드를 다른 데이터 형태로 가공 후 입력하여 기존 이미지 기반 객체 인식 신경망의 입력으로 사용하는 방법, 포인트 클라우드를 입력으로 하는 새로운 특성 추출단을 구현하여 객체 인식을 위한 특성을 추출하는 방법 등 다양한 방법으로 라이다 객체 인식을 위한 연구가 진행되고 있다.

최근에는 라이다 객체 인식 연구가 매우 활발히 진행되어 객체 인식 신경망 구조의 다양성이 매우 증가하였으며, 이에 따라 발표된 신경망들에 대한 전체적인 구조, 흐름을 분석하는 연구도 함께 진행되고 있다. Arnold 등[11], Li 등[12], 고준호 등[13]은 각자 특정 기준을 바탕으로 라이다 객체 인식 신경망에 대한 분류와 분석을 제시하였다. Arnold 등[11]은 추가적으로 카메라를 활용한 3D 객체 인식 신경망을 다루고 있다. Li 등[12]은 3D 객체 인식을 위한 포인트 클라우드 특성 추출 방법에 대한 상세한 분석을 다루고 있다. 고준호 등[13]은 국내 논문 중 유일하게 라이다 객체 인식 신경망 분석을 다루고 있으며, 특히 라이다를 이미지 형태로 투영하는 신경망과 카메라, 라이다 두 센서를 융합하는 신경망에 대한 구체적 구조 설명으로 구성되어 있다. 기존 분석 연구에서는 당시 성능이 우수한 신경망에 대한 분석을 다루고 있어 최근 많은 성능 개선을 달성한 여러 새로운 신경망에 관한 내용이 존재하지 않는다. 또한 기존 분석 연구에서는 오로지 신경망 구조 설명에 대해서만 집중적으로 구성되어 있고 신경망들을 어떻게 응용하여 실제 자율주행에 적용할 수 있는지에 대해 다루고 있지 않는 한계점이 있다.

본 연구는 기존 연구 분석과 차별점을 두고자 인식 성능이 크게 발전한 최근 신경망들과 보다 자율 주행 적용에 적합한 신경망들을 집중적으로 선정하여 다루고 있으며, 추가적으로 신경망들을 자율주행 연구에 어떻게 활용하는지에 대한 제안과 방법을 다루고 있다. 본론에서는 딥러닝 기반 라이다 객체 인식 연구의 큰 흐름에 대해 다루고, 객체 인식 신경망들을 적절한 기준으로 분류하여 라이다 객체 인식 연구에 대한 개괄적 개념을 다룬다. 본 논문은 2장에서 우선 딥러닝 객체 인식 기술에서 자주 사용되는 용어를 정의하고, 3장에서 자율 주행 라이다 객체 인식 연구를 위한 데이터셋의 소개와 4, 5장에서 여러 방식의 라이다 객체 인식 신경망에 대한 구체적인 분석과 마지막 6장 결론에서 총정리로 구성되어 있다.

2. 용어 정의

구체적으로 라이다 객체 인식 신경망의 분석에 앞서, 본 장은 컴퓨터 비전 관련 딥러닝에서 전반적으로 사용하는 용어에 대해 정의하고자 한다. 딥러닝은 다양한 분야에서 연구가 활발히 진행 중이며 매년 수많은 논문이 발표되고 있다. 따라서 논문마다 개념을 표현하는 용어에 약간의 차이가 있으며 이를 정의하지 않고 읽으면 다른 의미로 전달될 우려가 있다. 따라서 본 장에서는 이후 여러 객체 인식 신경망의 구조를 설명하고, 각각의 핵심 아이디어를 보다 명료하게 전달하기 위해 컴퓨터 비전 관련 딥러닝에서 자주 사용되는 용어를 짚고 나아가고자 한다.

객체 인식(Object detection) 신경망은 Fig. 1의 (c)처럼 입력 데이터 안에 어떤 물체가 있는지 확인하여 해당 물체가 어떤 클래스 인지 분류(Classification)하고, 어느 위치에 있는지 검출(Localization)하는 인공지능 신경망이다. 객체 인식 신경망은 크게 특성 추출단, 객체 검출단, 검출 보정단으로 구성되어 있다. 특성 추출단(Feature extractor)은 데이터를 입력 받아 객체 검출을 위한 정보를 추출하는 단으로 인코더(Encoder) 또는 백본(Backbone)이라 부르며, 특성 추출단을 통해 전역적 특성(Global feature)

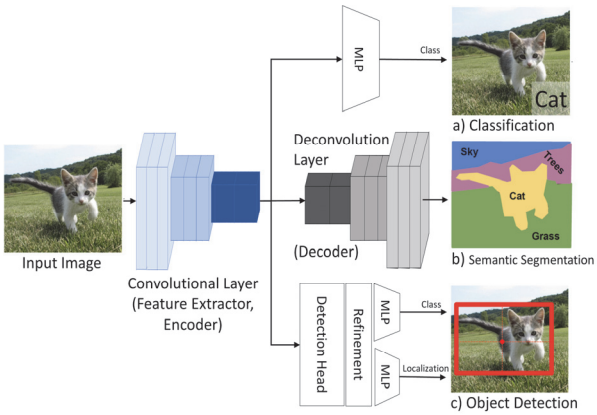


Fig. 1 Deep learning based computer vision examples

을 수집하는 과정을 인코딩이라 부른다. 특성 추출단은 보통 MLP(Multi Layer Perceptron) 또는 CNN[43](Convolutional Neural Network)으로 구성되어 있다. 추출되는 정보를 특성(Feature)이라고 부르며, 특성맵(Feature map)은 인코딩으로부터 출력된 특성들을 표현하는 데이터를 말한다. 객체 검출단(Detection head)은 딥러닝 신경망을 통해 수집한 특성을 바탕으로 클래스 분류와 경계 박스를 예측한다. 보다 정밀한 객체 분류와 경계 박스 검출을 위해 몇몇 신경망은 보정단(Refinement stage)을 활용하며 이러한 신경망을 다단계(2-stage)객체 인식 신경망, 보정단을 활용하지 않는 신경망을 단 단계(Single stage) 객체 인식 신경망이라 부른다.

의미 분할 신경망(Semantic segmentation)은 Fig. 1의 (b)처럼 객체 인식에서 더 나아가 각 데이터 구성 요소(이미지에서 픽셀, 포인트 클라우드에서 포인트)별로 각각이 어떤 의미를 가지는지 분류하는 신경망이다. 의미 분할 신경망은 객체 인식 신경망에서 인코딩하여 나온 특성맵에 디코딩 과정을 통해 결과를 출력하는데, 디코딩이란 인코딩 과정에서 추출된 전역적 특성(Global feature)을 원본 데이터 크기로 업스케일링하고, 전체 데이터의 특성에서 특정 지점이 어떤 의미 분할적 특성을 나타내는지 추론하는 과정이다. 의미 분할적 특성 또는 시멘틱 특성(Semantic feature)은 해당 요소의 특성이 전체 데이터에서 어느 부분에 구성되는지에 대한 정보가 포함되어 있어 객체 분류에 유용하게 활용할 수 있다.

3. 라이다 포인트 클라우드 데이터셋

다중 채널의 고해상도 라이다 포인트 클라우드 데이터(e.g. 64, 128 channels)는 입력 장면에 대한 포인트 정보가 풍부하여 객체 인식 성능에 유리하지만 매우 비싼 라이다 센서의 가격으로 인해 데이터 확보에 어려움이 있다. 또한 신경망을 학습하기 위해서는 수많은 입력 데이터에 정답 라벨을 부여하는 작업(Labeling)이 필요하며 특히 3D 공간상에서 정답 라벨을 작성하는 것은 매우 많은 시간과 노력이 필요한 작업이다.[14] KITTI,[15] nuScenes,[16] Waymo open dataset[17]은 연구자들이 위의 어려움을 극복하고 보다 범용적으로 라이다 객체 연구를 수행할 수 있도록 제작된 포인트 클라우드 포함 여러 센서 데이터를 제공한다. Table 1에서 보듯이 각 데이터셋들은 고해상도 라이다와 카메라, 레이다를 활용하여 다양한 실제 도로 주행 센서 데이터를 수집, 가공하고 이들을 무료로 공개함으로 활발한 라이다 객체 인식 연구에 크게 이바지를 하고 있다. 이미지 기반 객체 인식에서 PASCAL VOC,[18] COCO Challenge[19]와 같은 유명한 데이터 셋 기반 객체 인식 챌린지가 있는 것처럼 라이다 객체 인식 분야에서도 새로운 신경망에 대해 각 데이터셋마다 정해진 성능 평가를 진행하여, 해당 신경망이 기존 대비 얼마나 발전하였는지 다른 우수한 신경망들과 성능을 비교분석을 할 수 있다. 또한 최근에는 라이다를 활용하여 객체 인식에서 더 나아가 보다 다양한 임무(Task)를 수행하기 위한 데이터셋들이 공개되고 있으며, 이러한 다양한 데이터셋들은 연구자들에게 하여금 라이다 포인트 클라우드 데이터를 활용한 다양한 연구를 진행하는 데 큰 도움이 되고 있다.

Table 1 Lidar object detection dataset compression table

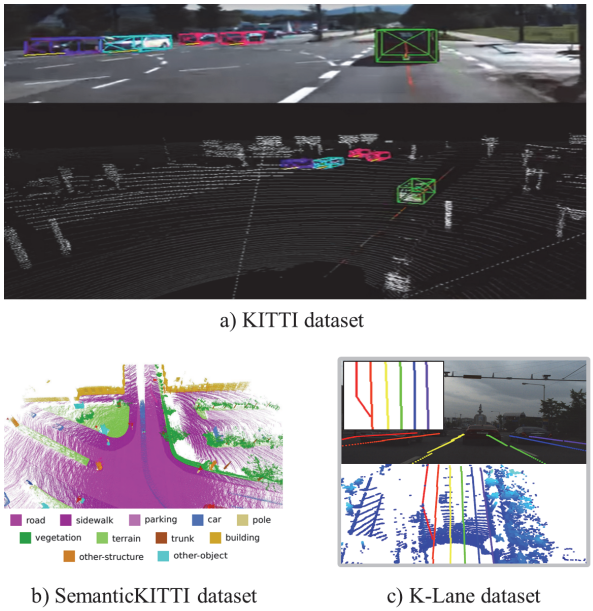
	KITTI	nuScenes	Waymo open dataset
Lidar sensor	1x Velodyne HDL-64 (64 Channels Lidar)	1x Velodyne HDL-32 (32-Channels Lidar)	1x 75m range, 4x HoneyComb 20m range
Additional sensors	2x Stereo Camera, GPS, IMU	6x Camera, 5x Radar, GPS, IMU	5x Camera
Annotated frames	15K	40K	230K
Scenes amount	22	1000	1150
Hours	1.5	5.5	6.4
Object class	8	23	4
3D Boxes amount	200K	1.4M	12M
Location	One City (Karlsruhe)	Two Cities (Boston, Singapore)	3 Regions (USA)
Scenes weather	Sunny, Cloudy	Various Weather	Sunny, Cloudy
Scenes time	Day	Day, Night	Day, Night, Dusk, Dawn
Published year	2012	2019	2019

3.1 KITTI

KITTI[15] 데이터셋은 가장 오래된 라이다 데이터셋 중 하나이며, 최근까지도 가장 범용적으로 사용되어 연구가 진행되고 있다. Karlsruhe Institution of Technology와 Toyota Technology Institute at Chicago에서 배포하고 있는 KITTI 데이터셋은 Fig. 2의 (a)와 같이 낮 시간 독일 Karlsruhe의 고속도로와 일반도로에서 64채널의 고해상도 라이다 센서(Velodyne HDL-64E)로 수집된 약 15,000 프레임의 데이터를 제공한다. 또한, 차량 전방 이미지를 라이다 데이터 프레임 레이트(10 fps)에 맞게 제공함으로 라이다, 카메라 센서 퓨전을 위한 연구에도 활용되고 있다. KITTI 데이터셋은 총 8가지 객체 종류에 대한 정보를 제공하고 있으며 자동차, 이륜차, 보행자 3가지 객체에 대해 인식 성능을 측정하고 있다. KITTI 데이터셋은 고해상도 포인트 클라우드 데이터를 활용할 수 있어 라이다 데이터셋 중 최근까지도 가장 많은 연구에 활용되고 있다. 다만 KITTI 데이터셋은 다른 데이터셋에 비해 데이터양이 적고, 오로지 맑은 날에만 수집되어 보다 다양한 주행 환경에 대한 데이터를 포함하지 않고 있다.

3.2 nuScenes

nuScenes[16] 데이터셋은 자율 주행 개발 적용을 목적으로 NuTonomy에서 구축하여 배포하고 있는 데이터셋이다. Boston과 Singapore의 서로 다른 4곳에서 데이터를 수집하였으며 32채널의 라이다에 추가적으로 5개의 레이더 센서, 6개의 카메라 센서로부터 다양한 데이터를 수집하여 센서 융합 연구에 활용할 수 있는 데이터를 제공한다. 약 40,000프레임에서 23가지 서로 다른 객체에 대한 정보를 제공한다. 또한 nuScenes 데이터셋은 맑은 날, 흐린 날, 비오는 날 등 다수의 기상조건하에 데이터를 수집하여 보다 다양한 주행 환경을 다룰 수 있다. 다만 nuScenes 데이터셋의 라이다는 KITTI 데이터셋 라이더의 절반인 32채널로 희박한 포인트 클라우드 데이터를 제공하고 있으며 이로 인해 정확한 객체 인식을 기대하는 신경망 연구에 다소 어려움이 있다. 아직까진 KITTI 데이터셋에 비해 활발한 연구가 진행되지 못하고 있지만 앞으로 주행 환경에 대해 강인한 신경망 개발을 위해 nuScenes 데이터셋이 더 적극적으로 활용될 것이라 전망된다.



a) KITTI dataset

b) SemanticKITTI dataset

c) K-Lane dataset

Fig. 2 KITTI, SemanticKITTI and K-Lane dataset examples

3.3 Waymo open dataset

Waymo open dataset[17]은 라이다를 활용한 자율 주행 자동차로 유명한 Google의 무인 자동차 개발 기업 Waymo에서 배포하고 있는 데이터셋이다. 위 데이터셋은 약 230,000 장으로 구성된 대용량 데이터셋으로 4개의 근거리용 라이다와 1개의 중거리용 라이다 총 5개의 라이다로부터 수집한 포인트 클라우드 데이터로 구성되어 있으며, 근거리용 라이다 데이터를 바탕으로 가까운 객체에 대한 보다 해상도 높은 정보를 제공한다. Waymo open dataset은 75 m 거리 안에 있는 객체에 대해 라벨을 제공하며 자동차, 사람, 이륜차, 표지판 총 4개에 대한 객체 정보를 제공한다. Waymo open dataset은 보다 다양한 신경망 학습을 위한 대규모의 데이터를 제공함으로 최근 이를 활용한 연구가 새롭게 진행되고 있지만 KITTI와 마찬가지로 맑은 날의 데이터만 존재하여 다양한 기상조건에 대한 학습은 불가능하다.

3.4 기타 라이다 포인트 클라우드 데이터셋

라이다 센서가 자율주행 자동차에서 점점 중용됨에 따라 라이다 포인트 클라우드 데이터를 바탕으로 객체 인식 외에 다른 목적을 위한 데이터 셋 또한 공개되고 있다. SemanticKITTI[52] 데이터셋은 Fig. 2의 (b)처럼 KITTI 데이터셋의 라이다 포인트 클라우드를 바탕으로 한 3D Point semantic segmentation 데이터를 제공한다. 위 데이터셋은 픽셀별로 라벨이 제공되는 기존 이미지 Segmentation 데이터셋처럼 라이다 포인트 클라우드마다 라벨을 적용하여 해당 포인트가 어느 Semantic에 포함되는지에 대한 정보가 담겨있다. 한국과학기술원에서 배포하고 있는 K-Lane[53] 데이터셋은 Fig. 2의 c와 같이 라이다 포인트 클라우드를 바탕으로 차선 인식을 위한 데이터셋으로써 최근 공개되었다. 기존 이미지 기반 차선 인식은 밤 또는

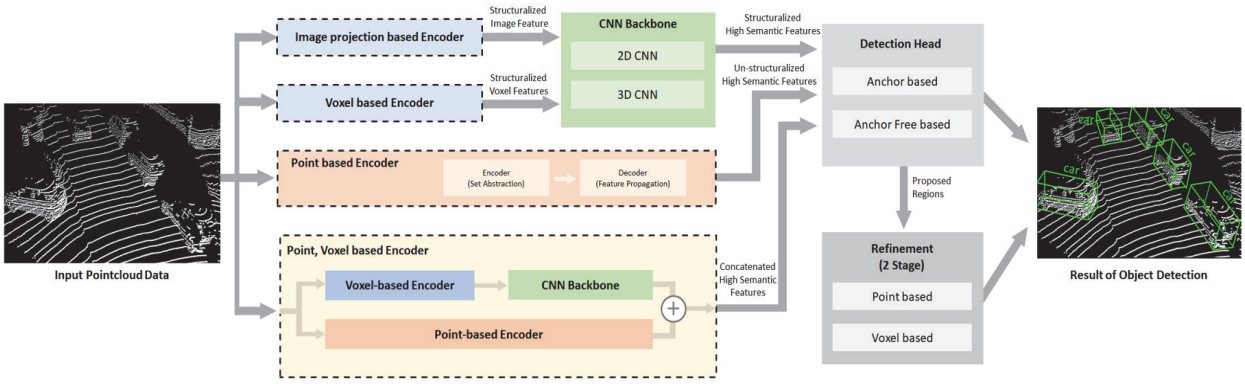


Fig. 3 Deep learning based lidar object detection neural network overall structure. As shown in the figure, the lidar object detection neural network extracts features in four major ways. After that, it can be divided into the single stage method that outputs a 3D bounding box directly through the detection head and the two stages method that additionally makes boxes more precisely with the refinement stage

악천후 환경 등 특정 주행 환경에서 성능이 열화되는 단점이 있다. 이와는 반대로 라이다는 포인트 클라우드는 카메라 이미지보다 비교적으로 주변 주행 환경 변화에 강인하다. 따라서 K-Lane 데이터셋 기반의 라이다 차선 인식 신경망은 기존 이미지 기반 차선 인식 신경망보다 다양한 주행 환경에서 강인한 성능을 가질 수 있다.

4. 라이다 객체 인식 신경망

4.1 라이다 객체 인식 신경망 전체 구조

라이다 객체 인식 신경망은 포인트 클라우드 특성 때문에 입력 데이터를 그대로 기존 이미지 객체 인식 신경망에 적용할 수 없다. 따라서 입력 데이터를 신경망에 대입하기 적절한 다른 형태의 데이터로 인코딩하는 과정이 필요하다. Fig. 3은 라이다 객체 인식 신경망의 일반적인 전체 구조 구성도이다. 라이다 객체 인식 신경망은 Fig. 3의 검은 점선 사각형과 같이 어떤 형태로 인코딩하는지에 따라 1) 이미지 투영 방식,[20-23] 2) 복셀 기반 방식,[26-30] 3) 포인트 기반 방식,[31-33] 4) 포인트-복셀 기반 방식[34-36] 4가지로 분류할 수 있다.

이미지 투영 방식은 포인트 클라우드를 이미지로 투영하여 CNN입력에 알맞은 3차원 텐서(폭, 길이, 채널) 형태로 변환한다. 복셀 기반 방식은 전체 포인트 클라우드 공간을 일정 크기의 정육면체로 분할하여 각 정육면체마다 대표 특성을 추출하여 4차원 텐서(폭, 길이, 높이, 채널)로 변환한다. 이미지 투영방식과 복셀 기반 방식으로 변환된 텐서 데이터의 채널에는 포인트 클라우드의 x, y, z축별 값과 수신 신호강도 등이 포함될 수 있다. 위 두 방식은 포인트 클라우드를 CNN에 적용할 수 있는 텐서로 규격화하는 방식으로 가공하고, 이를 CNN 백본에 입력하여 객체 인식을 위한 특성을 추출한다. 포인트 기반 방식은 포인트 클라우드를 따로 규격화하는 과정을 수행하지 않고 그 자체로부터 객체 인식을 위한 특성을 추출한다. 포인트로부터 직접 특성을 추출하기 위해서 위 방식은 PointNet++[25]을 주로 활용한다. 포인트-복셀 기반 방식은 포인트 기반 방식과 복셀 기반 방식 모두 동시에 활용한다. 이를 통해 포인트-복셀 기반 방식은 각각의 방식의 장점을 활용하고 서로의 단점을 보완하여 객체 인식을 위한 더 풍부한 특성을 확보할 수 있다.

위 4가지 방식으로 추출된 특성들은 객체 검출단으로 입력되어 미리 지정된 앵커 박스를 바탕으로 예측하는 앵커 방식(Anchor-based method), 또는 앵커 박스를 사용하지 않고 MLP를 활용한 앵커 미사용 방식(Anchor-free method)으로 3차원 경계 박스와 클래스 분류를 수행한다. 더욱 정확한 경계 박스 예측을 위해 일부 신경망은 객체 검출단 이후에 보정단을 추가하여 다단계 방식으로 구성한다. 다단계 방식 신경망은 일 단계에서 예측한 3차원 경계 박스를 관심 영역으로 지정하고, 그다음 관심 영역을 원본 포인트 클라우드에 대입하여 해당 영역에 포함된 포인트들을 따로 뽑아낸 뒤 추가적인 인코딩 과정을 통해 더욱 정밀한 경계 박스 예측을 위한 보정을 수행한다.

4.2 라이다 객체 인식 신경망 상세 구조

앞서 서술하였듯이, 라이다 객체 인식 신경망은 객체 인식을 위한 특성 추출을 위해 다양한 방식의 인코더를 활용한다. 본 장에서는 포인트 클라우드 데이터 인코딩 방식에 따라 각각의 상세 구조와 대표 신경망에 대해 다루고, 각 신경망들의 공통점과 차별점을 바탕으로 장단점을 분석하고자 한다.

4.2.1 이미지 투영 방식 (Image projection based method)

이미지 투영 방식은 포인트 클라우드의 여러 특성(원점으로 x, y, z축 기준 얼마나 떨어져 있는지, 신호 강도 등)을 바탕으로 Fig. 4처럼 다채널의 이미지로 가공 후 객체 인식을 수행하는 방법이다. FV(Front View)방식 투영은 Fig. 4의 (a), (b)와 같이 카메라와 동일하게 시야각을 지정하여 전방에 있는 포인트들을 평면에 투영하는 방식이다. BEV(Bird Eye View) 방식은 Fig. 4의 (c), (d)처럼 포인트 클라우드를 위에서 아래로 내려다보는 방향으로 투영하는 방식이다. 추가로 이미지 투영 방식에서는 라이다 센서 뿐만 아니라 카메라 센서의 이미지 데이터를 융합하여 성능을 높이고자 한 연구들이 여럿 있다. 이미지 투영 방식은 위와 같이 여러 방법으로 투영된 특성 이미지 데이터를 바탕으로 2D CNN을 활용한 기존 2D 객체 인식 신경망 구조를 활용하여 2차원 또는 3차원 경계 박스와 클래스 분류(Classification)를 수행한다.

MVCNN[20]는 3차원 입력 데이터를 마치 여러 카메라를 다양한 각도에서 찍은 것과 같이 여러 각도에서 이미지로 투영하여 복수의 FV 이미지를 생성한 후 각각의 이미지를 CNN층에 입력하여 클래스 분류를 수행하는 방법을 제시하였다. MV3D[21]는 포인트 클라우드 데이터를 x, y, z축 방향 거리와 신호 강도(Intensity)를 바탕으로 생성한 BEV 이미지와 FV이미지 그리고 추가적으로 전방 카메라 이미지 모두를 입력 받아 각각 독립적인 2D CNN을 통해 특성을 추출하였다. MV3D는 이후 출력된 각각의 특성맵들은 융합 객체 인식을 수행하는 층에 입력하는 방식으로 3D 객체 인식하는 신경망을 제시하였다. BirdNet[22]은 BEV 이미지를 2D 객체 인식 신경망인 Faster-RCNN[10]의 백본과 객체 검출단을 활용한 방법을 제시하였다. 후속 연구인 BirdNet+[23]는 Faster-RCNN의 내부 신경망에서 2D 이미지 인코딩 방법을 ResNet-50[44]으로 변경하고 추가로 FPN[45] 구조를 적용하는 방법을 제안하였다.

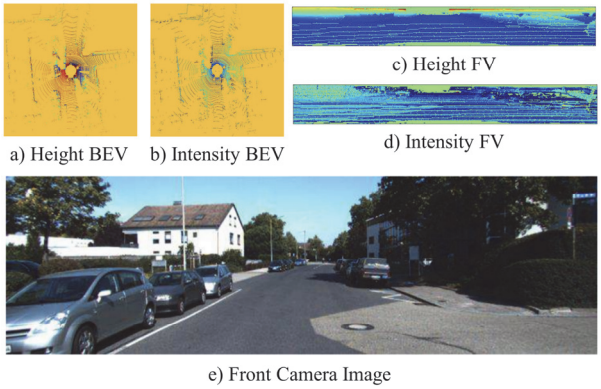


Fig. 4 Examples of image projection based method inputs

이미지 투영 방식의 신경망들은 모두 3차원 데이터를 2차원 이미지 형태로 투영하는데, 이에 따라 3차원 정보 중 적어도 한 차원에 대해서 정보의 손실이 발생하여 (BEV의 경우 높이 대한 정보, FV의 경우 깊이(라이다상 x축)에 대한 정보) 3차원 경계 박스 예측이 다소 부정확하다. 또한 위 방식은 BEV, FV, 전방 카메라 이미지에 대한 각각의 서로 다른 CNN 층이 필요하며 이후 융합을 위한 추가적인 층이 필요하다. 따라서 이미지 투영방식은 라이다 포인트 클라우드를 마치 기존 2D 이미지 객체 인식처럼 활용할 수 있는 방법을 제시함에 의의가 있지만 2D CNN을 주로 활용함으로 이후 다른 방식들에 비해 포인트 클라우드의 3차원적 특성을 온전히 다룰 수 없어 정확한 경계박스 예측이 어려운 단점이 있다.

4.2.2 복셀 기반 방식 (Voxel based method)

복셀 기반 방식에서는 입력 포인트 클라우드를 정해진 크기의 정육면체 집합으로 나눈 후, 각 정육면체 내부에 포함된 포인트들을 휴리스틱한 방법[26]이나 PointNet을 걸쳐 복셀별 대표 특성을 추출하고 묶어 4D 텐서(W, H, L, Channel)로 출력한다. 복셀 기반 방식은 해당 텐서들을 3D CNN 층에 입력하여 3차원 컨볼루션 필터가 x, y, z축 방향으로 이동하며 3차원 공간적 정보를 추출한다. 이후 추출된 특성들은 z방향으로 압축되며 이를 2D CNN 층에 입력하여 클래스 분류와 경계 박스 예측을 위한 추가적인 특성을 추출한 다음, 2D 객체 인식과 동일하게 객체검출단(Detection head)을 통해 객체 인식을 수행한다. 복셀 기반 방식은 3D CNN을 통해 z축 방향에 대한 충분한 정보를 추출하고 함축하여 기존 이미지 투영 방식에 비해 3D 경계 박스의 정확도가 대폭 상승하였다. 복셀 기반 방식은 3차원 정보를 직관적이며 규격화된 텐서로 표현하여 보다 풍부한 3D 공간적 정보를 추출할 수 있으며, 최근까지 여러 방식 중에서 가장 활발히 연구가 진행되고 있는 방식이다.

VoxelNet[27]은 복셀 기반 방식으로 포인트 클라우드 데이터를 바탕으로 사람이 따로 특성을 만들어 주지 않고 입력받은 데이터로부터 객체 인식 결과 출력까지 신경망 스스로 전 과정을 학습하는 최초의 End-to-end 라이다 객체 인식 신경망이다. 2017년 PointNet[24]의 발표로 포인트 클라우드에 추가적인 가공 없이 MLP와 대칭 함수(Symmetry function)를 활용하여 포인트 대표 특성을 추출하는 방법이 제시되었다. VoxelNet은 복셀별로 PointNet을 적용하여 복셀 대표 특성을 추출하고 이를 3D CNN을 통해 객체 인식을 하는 방법을 제시하였다. VoxelNet의 전체적 구조는 Fig. 5와 같이 구성되어 있다.

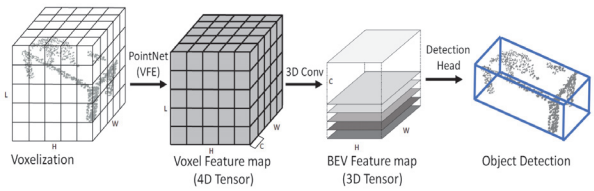


Fig. 5 Structure of voxel based object detection network

입력된 포인트 클라우드는 x, y, z축별로 0.4, 0.2, 0.2 m 단위로 복셀화 되며, 각 복셀별로 PointNet을 걸쳐 복셀별 대표 특성을 추출하는 VFE(Voxel Feature Encoding layer) 층을 통과한다. 이후 VFE층 출력 특성맵은 3D CNN 3개의 층을 걸쳐 전체 장면의 공간적 정보를 추출하고, 이를 z축 방향으로 압축하여 얻은 특성맵을 2D CNN 층을 통해 객체 인식을 예측한다. VoxelNet은 포인트 클라우드의 특성을 온전히 신경망 학습을 통해 수집하여 3D 객체 인식 성능을 크게 높였다. SECOND[28]는 VoxelNet과 거의 동일한 구조에 보다 빠른 연산 속도를 위해 데이터가 존재하지 않는 빈 공간의 복셀들을 CNN 연산에서 제외하고 실제 데이터 값이 존재하는 의미 있는 지역에 대해서만 연산을 수행하는 Sparse convolution[37,38]을 적용하였다.

3D CNN은 2D CNN과 비교했을 때 비교적 긴 수행시간이 필요하다. 따라서 기존 3D CNN 기반 특성 인코더에서는 연산 시간의 고려 때문에 다수의 출력 층 구성에 제약이 있으며, 이는 정확한 객체 인식을 위한 더 구체적인 특성 추출에 한계가 있는 단점이 있다. PointPillars[29]는 VoxelNet과 달리 연산이 비싼 3D CNN을 제외하고 바로 2D CNN을 적용하는 방법을 제시하였다. PointPillars는 입력 포인트 클라우드를 x, y 평면으로 그리드화한 후 해당 그리드에 포함하는 모든 포인트들, 즉 포인트 기둥에 대해 대표 특성을 추출하여 곧바로 BEV 이미지 형태로 출력한다. PointPillars의 BEV 이미지는 해당 기둥에 포함된 포인트들의 3D 공간상의 특성을 PointNet을 통해 충분히 추출하여 생성되었다는 점에서 기존 투영 방식의 BEV 이미지와 차이가 있다. 생성된 BEV 이미지는 2D 백본에 입력되어 출력 특성맵을 추출하고 SSD[42]구조의 객체 검출단을 통해 객체 인식을 수행한다. PointPillars는 오로지 2D CNN 만을 활용하여 실시간성이 현재까지 최고수준으로 뛰어나다.

Fast PointRCNN[30]은 다단계 신경망이며, 3D CNN과 2D CNN 모두 활용하여 수행속도와 객체 인식 성능의 균형을 적절히 조절한 신경망이다. Fast PointRCNN의 일단계는 SECOND의 Spatial convolution을 활용한 특성 추출단 구조를 사용한다. Fast PointRCNN은 특성 추출단에서 연산 량을 줄이기 위해 각 3D CNN 층마다 출력채널 수를 줄이지만, 대신 다수의 층으로 깊게 구성하여 다양한 스케일에 대한 공간적 특성을 추출할 수 있도록 구성하였다. 이후 2D CNN을 통해 추가적인 시멘틱 특성을 추출하여 이를 바탕으로 클래스 분류와 3차원 관심 영역(Region of interest, RoI)을 예측한다. 이 단계에서는 앞서 추측한 3차원 관심 영역을 포인트 클라우드와 2D CNN 특성맵에 적용하여 해당 영역 부분을 뽑아내고 이 두 특성을 합친 후 추가 신경망을 적용하여 경계 박스 보정하여 성능을 높였다.

4.2.3 포인트 기반 방식 (Point based method)

포인트 기반 방식은 이미지 투영이나 복셀화와 같은 데이터 전처리하지 않고 PointNet++[25]를 활용하여 바로 포인트로부터 객체 인식을 위한 특성을 추출하는 방식이다. 기존 복셀 방식과 달리 포인트를 인위적인 정육면체로 나누지 않고 포인트 그 자체로 특성을 추출하기 때문에 더 세밀하고 구체적인 각 포인트별 특성 추출이 가능하다.

PointNet++는 SA(Set Abstraction)층과 FP(Feature Propagation)층을 통해 일정 개수의 대표 포인트 특성과 전체 포인트들의 시멘틱 특성을 학습한다. Fig. 6에서 보듯이 SA층은 FPS(Farthest Point Sampling) 알고리즘을 통해 포인트 클라우드 데이터 안에 존재하는 모든 포인트를 정해진 개수의 대표 포인트로 샘플링한 후, 해당 대표 포인트 기준 일정 거리 안에 포함된 모든 포인트를 묶어 PointNet을 적용함으로 대표 특성을 추출한다. 해당 과정을 통해 원본 포인트 클라우드는 일정 개수의 대표 포인트들로 요약되며 해당 포인트들은 각자 일정 거리(Receptive filed)를 대표하는 특성을 가지고 있다. Fig. 7에서 FP층은 SA층을 통해 수집한 대표 특성을 바탕으로 보간법을 활용하는 디코딩을 수행한다. FP층은 이처럼 샘플링되어 추출된 특성들을 원본 스케일 데이터로 복구하여 객체 인식을 위해 필요한 각 포인트별 시멘틱 정보를 수집한다.

PointRCNN[31]은 포인트 기반 방식으로 처음 3D 객체 인식 방법을 제시한 다단계 신경망이다. PointRCNN은 일 단계에서 SA층과 FP층을 활용하여 각 포인트마다 시멘틱 정보를 추출하고 이를 바탕으로 해당 포인트가 배경에 포함되어 있는지, 또는 어떠한 물체에 구성되어 있는지 분류한다. 물체에 구성된 포인트들에 한해 3D 경계박스를 대략적 추측하여 3차원 관심 영역이 출력된다. 이후 이 단계에서는 출력된 관심 영역을 원본 포인트 클라우드에 대입하고 해당 영역에 포함된 포인트들에 대해 추가 특성을 추출하고, 일 단계에서 획득한 포인트 별

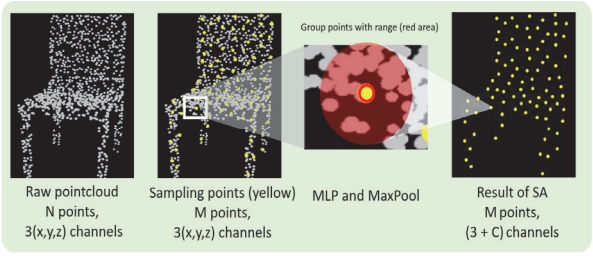


Fig. 6 Implementation of SA layers. Using the FPS algorithm, yellow points are selected as sampling points. After that, PointNet++ encoder extracts representative features of points within a specific range

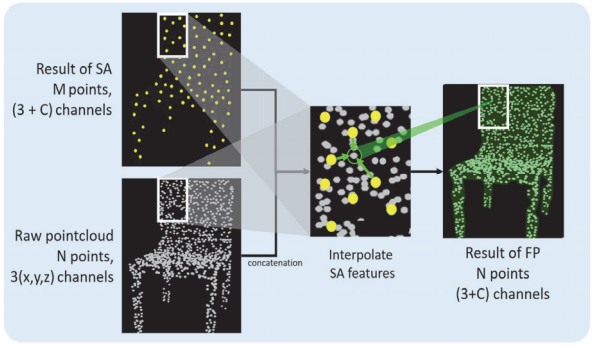


Fig. 7 Implementation of FA layers. FA layers concatenate SA layers output points and raw points and then interpolate the semantic features of raw points from 3 nearest SA layers key points

시멘틱 특성과 합친다. 마지막으로 합쳐진 두 특성을 활용하여 추가적인 인코딩 과정을 통해 정밀한 객체 인식을 수행한다.

STD[32]는 다단계 신경망으로 일 단계에서 모든 포인트에 대해 객체 또는 배경에 대한 정보가 담긴 시멘틱 특성을 추출한 다음 모든 포인트에 대해서 구형의 관심 영역을 지정한다. 관심 영역 안에 존재하는 포인트들은 앞서 추출한 시멘틱 특성을 포함하고 있는데, 해당 영역에 객체를 의미하는 포인트가 얼마나 포함하는지에 따라 NMS(Non Maximum Suppression) 알고리즘을 통해 일차적으로 후보 관심 영역 추린다. 그다음 각각의 후보 관심 영역에 PointNet 인코딩을 통해 3차원 박스 형태의 관심 영역이 예측되고 이를 다시 NMS를 걸쳐 일정 개수의 대표 관심 영역이 제시된다. 이 단계, 보정단에서는 선별된 대표 관심 영역을 원본 포인트 클라우드에 대입하여 해당 영역에 포함된 포인트들을 따로 추출한다. 추출된 포인트들을 복셀화 시킨 다음 VoxelNet의 인코딩 과정(VFE)을 통해 복셀별 추가 특성이 추출된다. 이후 모든 복셀별 특성맵을 한 줄로 길게 늘인 다음 해당 특성맵에 MLP를 적용하여 3차원 경계 박스와 객체의 종류를 예측하는 과정이 진행된다.

3DSSD[33]는 새롭게 대표 포인트를 선정하는 방법을 제시함으로써 기존 포인트 방식 신경망보다 정확도를 높이며 실행 속도를 또한 개선한 단 단계 신경망이다. 포인트의 개별 특성을 추출하기 위해 기존 SA층에서는 포인트별 유클리디언 거리를 바탕으로 한 D-FPS(Distance based FPS) 알고리즘을 활용하여 샘플링 포인트를 수집하였다. 포인트 클라우드에서 각 포인트들은 전체 공간 중 배경에 존재하는 비중이 높은데, D-FPS는 오로지 거리별로 샘플링 포인트를 결정하다 보니 비교적 의미 있는 물체에 대한 샘플 포인트는 적고 불필요한 배경에 대한 샘플 포인트들이 많은 문제가 발생한다. 3DSSD는 포인트별 각 특성들을 마치 거리 계산하는 듯이 각 특성에 대한 거리를 계산하여 샘플 포인트를 추출하는 F-FPS (Feature based FPS) 알고리즘을 제안하여 보다 의미 있는 물체에 대한 샘플 포인트를 늘리고자 하였다. 실제 객체 인식을 수행할 때 F-FPS와 D-FPS를 모두 활용하여 물체와 배경에 대한 모든 샘플링 포인트들을 통해 해당 장면에 대해 보다 다양한 특성을 추출하여 우수한 성능을 달성하였다.

4.2.4 포인트, 복셀 기반 방식 (Point, Voxel based method)

최근 연구에서는 위에 설명한 방식을 모두 활용하는 방향으로 성능 개선을 이루고 있다. 포인트 기반 방식의 신경망들은 포인트 클라우드를 가공하지 않고 그대로 활용함으로써 3D 공간적 정보를 비교적 정보 손실 없이 효과적으로 추출할 수 있다. 다만 특성을 추출하기 위해 샘플링 포인트를 정하는 FPS 알고리즘과 샘플링되어 추출된 정보를 다시 원본 포인트들에 할당하기 위한 FA층 연산에 추가적인 연산 시간이 소요된다. 또한, 보다 큰 스케일에 대한 대표 특성을 추출하기 위해서는 SA층 연산을 여러 번 수행해야 하며, 그 결과 샘플 포인트의 개수가 너무 적어지게 되어 다양한 스케일에 대한 충분 특성 추출에 한계가 있다. 반면 복셀 기반 방식의 신경망들은 CNN을 활용하여 규격화된 텐서 데이터에 서로 다른 크기의 컨볼루션 필터와 풀링 연산을 활용하여 다양한 스케일에 대한 특성들을 효과적으로 추출할 수 있다. 하지만 위 방식은 포인트 클라우드를 CNN에 적합한 입력 형태로 변형하기 위해 공간적 정보를 압축하면서 유용한 공간적 특성의 손실이 필연적으로 발생하게 된다.

PV-RCNN[34]은 위 두 가지 방식의 장점을 모두 활용하여 우수한 성능과 적절한 수행 속도를 갖추는 신경망을 제시하였다. 위 신경망은 PointNet++의 SA층을 활용하

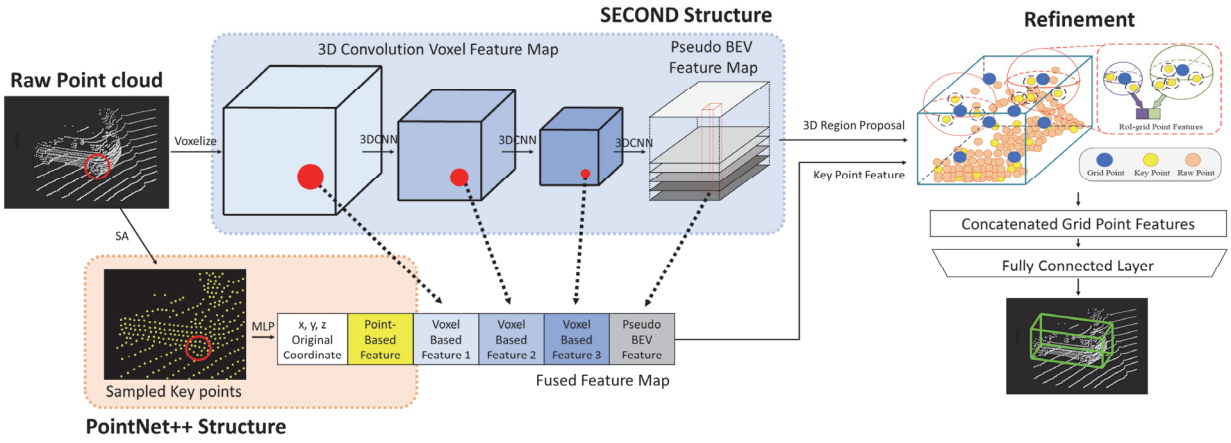


Fig. 8 PV-RCNN(Point, Voxel-based method) overall structure. Voxel representative features are generated with voxel based method, SECOND. In addition, key point representative features are extracted with point based method and concatenated into fused feature maps which are including various information within the receptive field. With fused feature maps, the refinement stage outputs precise 3D bounding boxes utilizing SA layers with fixed grid points which are located in the 3D region proposal

여 각 샘플 포인트(Fig. 8의 Key points)의 대표 특성을 추출하는 동시에(Fig. 8의 주황 점선 박스) 포인트 클라우드를 복셀화하여 SECOND 신경망의 3D CNN을 통해(Fig. 8의 파랑 점선 박스) 복셀 대표 특성을 추출한다. 그 다음 CNN의 중간 여러 층의 특성맵에서 앞서 구한 샘플링 포인트의 할당 범위만큼 뽑아(Fig. 8의 붉은색 원) 포인트 기반 특성맵과 합친다. 합쳐진 특성맵은 Key points의 특정 할당 공간에 대해 포인트 기반 특성 추출과 여러 스케일의 복셀 기반 특성 추출을 통해 수집한 다양한 특성들을 합친 융합 특성맵이다. 해당 특성맵을 통해 관심 영역이 예측되어 지고, 이후 검출 보정단에서는 관심 영역에서 균일하게 분포된 Grid point를 임의로 생성한 후, 해당 Grid point 주변에 존재하는 Key point들을 바탕으로 PointNet++을 추가로 적용하여 3차원 경계박스 보정을 수행한다. PV-RCNN은 포인트, 복셀을 통해 다양한 특성을 추출할 수 있어 3차원 경계 박스 정확도가 우수하다. PV-RCNN++[35]는 PV-RCNN에서 연산량을 줄이기 위해 모든 SA층에서 MLP를 활용하지 않고 여러 구역을 나눈 후 구역별 개별 특성을 평균값 풀링과 학습가능 한 가중치로 구성된 커널의 곱을 활용하여 추가 특성을 추출하였다.

Pyramid R-CNN[36]은 PV-RCNN와 마찬가지로 포인트 기반 특성과 복셀 기반 특성을 모두 활용하여 다양한 정보를 추출하는 다단계 객체 인식 신경망을 제안하였다. 기존 연구 신경망들은 보통 보정단에서 출력된 관심 영역 내부에 포함된 포인트들에 대해서만 추가 특성을 추출하여 경계 박스를 보정한다. Pyramid R-CNN의 보정단에서는 일 단계에서 출력된 관심 영역을 피라미드 구조와 같이 복수의 다단계 크기로 추가 구성하였다. 피라미드 구조의 다단계 크기 관심 영역은 센서로부터 멀리 떨어진 곳에 위치한 객체에 대해 기존 관심 영역 보다 큰 영역(피라미드 아래층)을 설정하여 보다 많은 주변 포인트들을 확보할 수 있고, 이를 바탕으로 추가 특성을 보다 많이 추출할 수 있게 하였다. 반면 센서에 가까이 위치하여 물체들이 밀집하게 존재하는 장면에 포함된 객체의 경우, 작은 영역(피라미드의 고층)이 활용되어 서로 다른 객체간의 분류를 효과적으로 수행할 수 있도록 하였다. 또한 Pyramid R-CNN은 RoI-Grid Attention층을 구성하여 Graph CNN,[39] Attention,[40] Point Transformer[41]을 활용한 학습 가능한 적절한 가중치를 특성에 곱하여 보정단의 객체 인식 성능을 높이고자 하였다.

5. 자율주행을 위한 신경망 활용

최근 연구들은 새로운 신경망 제시와 함께 소스 코드를 공개하여 다양한 연구자들에게 하여금 해당 신경망을 활용하여 더 발전된 연구를 도모하는 추세이다. 앞서 설명한 대부분의 신경망들 또한 논문과 함께 소스 코드를 공개하였다. 대표적으로 KITTI Benchmark[51]에서는 성능별 Table을 제공하며 오픈 소스 코드가 제공된 연구에 대해서 해당 링크를 공유한다. 연구자들은 해당 오픈 소스 코드를 통해 해당 신경망들을 각자 필요에 맞게 학습하여 다양한 방법으로 연구에 활용할 수 있다. 다만 연구자 개인이 직접 처음부터 학습하여 논문에 제시된 성능을 기대하기 위해서는 매우 고가의 컴퓨팅 장비(e.g., GPU)를 통해 오랜 시간학습을 진행해야 하는 번거로움

Table 2 A table of lidar object detection network evaluation on KITTI car 3D object detection

	KITTI Evaluation Car 3D AP			Time (s)	Stage	Year	Conference or Journal
	Easy	Moderate	Hard				
Image-Projection based							
MV3D	74.97	63.63	54.00	0.36	2	2017	CVPR
BirdNet	40.99	27.26	25.32	0.11	2	2018	ITSC
BirdNet+	70.14	51.85	50.03	0.1	2	2020	ITSC
Voxel-based							
VoxelNet	77.47	65.11	57.73	0.23	1	2018	CVPR
SECOND	83.13	73.66	66.20	0.05	1	2018	Sensors
PointPillars	82.53	74.31	68.99	0.016	1	2019	CVPR
Fast-PointRCNN	85.29	77.40	70.24	0.065	2	2019	ICCV
Point-based							
PointRCNN	86.96	75.64	70.70	0.1	2	2019	ICCV
STD	87.95	79.71	75.09	0.08	2	2019	ICCV
3DSSD	88.36	79.57	74.55	0.038	1	2020	CVPR
Point,Voxel-based							
PV-RCNN	90.25	81.43	76.82	0.08	2	2020	CVPR
PV-RCNN++	90.14	81.88	77.15	0.062	2	2021	-
Pyramid R-CNN	88.39	82.08	77.49	0.13	2	2021	ICCV

이 있다. 보다 빠르고 쉽게 신경망 활용을 위해 미리 학습이 완료된 사전 학습 모델(Pretrained model)을 사용하면 이러한 작업을 생략하고 곧바로 다른 연구에 응용할 수 있다. 본장에서는 사전 학습 신경망을 통해 자율 주행 상황에서 활용할 수 있는 방법과 오픈 소스의 Github 주소[46-50]를 참고하고자 한다.

도로 주행 중에는 차량 주변의 환경이 급격하게 변한다. 따라서 도로 자율 주행 시스템에서 활용되는 객체 인식 신경망은 이러한 다양한 변화에 즉각적인 탐지와 같은 높은 실시간성이 요구된다. 이처럼 실시간성을 중점으로 두는 시스템의 경우 신경망 복잡도가 낮아 실행속도가 빠른 SECOND, PointPillars, 3DSSD 등의 모델을 활용할 수 있다. Open-mmlab Github,[46] DV Lab Github,[49] Open-mmlab Github[50]에서는 위 신경망들을 KITTI 데이터셋으로 사전 학습된 모델을 공개하고 있으며, 이를 어떻게 사용하는지에 대한 상세한 설명을 제공하여 연구자들에게 하여금 곧바로 객체 인식 신경망을 응용할 수 있는 방법을 제공한다.

반면 자율 주차 시스템의 경우 높은 실시간성 보다는 제한된 거리 안의 차량 주변 여러 장애물에 대한 보다 정확한 위치와 섬세한 3차원 경계박스 검출이 중요하다. 위와 같은 경우에는 연산속도가 다소 떨어지지만 보다 정확하고 정교한 3차원 경계박스 예측이 가능한 PV-RCNN, PV-RCNN++ 신경망의 사전학습 모델을 활용할 수 있다. 해당 신경망을 발표한 저자는 Open-mmlab Github[46]에서 사전 학습된 모델을 배포하고 있다.

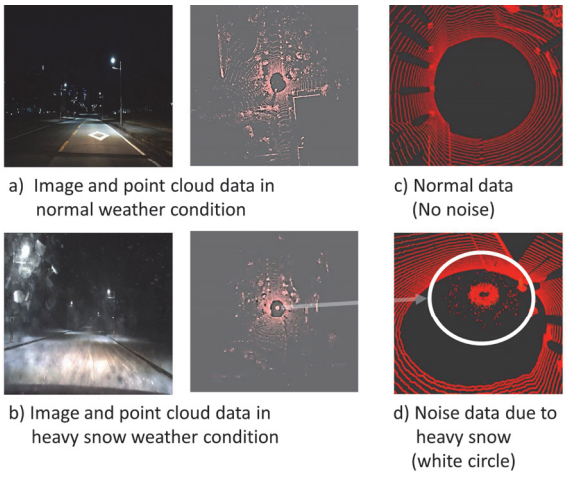


Fig. 9 Point cloud data comparison between normal weather conditions and heavy snow weather conditions. (b) Represent much fewer points at the far site than (a) due to the effect of snow. (d) Shows some noisy data near the lidar sensor compared with (c)

보다 다양한 기상조건을 고려하여 주행환경에 대해 강인한 시스템을 구현하고자 한다면, nuScenes 데이터셋을 통해 사전 학습된 신경망을 활용하는 것은 좋은 방법이다. KITTI 데이터셋은 64채널의 고해상도 라이다를 통해 데이터를 수집하여 현재까지 가장 활발히 해당 데이터셋을 활용하여 연구가 진행되고 있지만, 라이다 데이터 수집에 이상적인 기상 조건이 좋은 도로 상황의 장면

만 포함되고 있다. 해당 데이터셋으로 훈련된 신경망은 Fig. 9의 (b), (d)와 같이 악천후로 인해 센서로부터 먼 곳의 정확도가 낮고 센서 주변에 노이즈가 발생한 입력이 주어지면 검출 성능이 급격히 떨어지는 문제가 발생한다. 따라서 보다 다양한 환경에 강인한 시스템을 개발하기 위해서는 여러 기상조건에서 데이터를 수집한 nuScenes 데이터셋으로 사전 학습 모델을 활용하는 것은 효과적인 방법이다. Open-mmlab Github,[46] Yan Github[47]에서는 SECOND, PointPillars모델을 nuScenes 데이터셋을 바탕으로 학습한 Pre-trained 모델을 제공한다. 3DSSD 또한 DV Lab Github[49]에서 곧 nuScenes 데이터셋 바탕 pre-trained 모델을 제공할 것이라 공지하였다.

6. 결 론

본 논문은 라이다 객체 인식에 대한 개괄적 개념과 대표 신경망에 대해 알아보았다. 라이다 객체 인식은 카메라에서 수집할 수 없는 정확한 거리 정보를 수집할 수 있으며 낮에만 좋은 성능을 기대할 수 있는 카메라와 달리 주변 환경에 대해 강인한 장점이 있어 자율 주행 인지 기술 발전에 있어 매우 중요한 연구 분야 중 하나이다.

MVCNN, MV3D, BirdNet 신경망들은 라이다 포인트 클라우드를 FV, BEV 이미지로 투영하여 2D 객체 인식 신경망을 활용 객체 인식을 수행하였다. 2017년 PointNet을 통해 포인트 클라우드에서 직접 특성을 추출 하는 방법이 소개되었으며, 이후 연구에서는 이미지로 투영하는 방식 대신 직접 포인트 클라우드를 활용하여 특성을 추출하는 방향으로 연구가 진행되었다. VoxelNet, SECOND, PointPillars 신경망들은 포인트 클라우드를 복셀단위로 나눈 후 복셀별 대표 특성을 PointNet을 활용하여 추출하여 CNN에 적용하기 용이한 형태로 특성맵을 출력한 뒤 객체 인식을 수행하였다. PointRCNN, STD, 3DSSD 신경망은 PointNet++을 활용하여 샘플링 포인트에 대한 대표 특성과 모든 포인트에 대한 시멘틱 특성을 바탕으로 객체 인식을 수행하였다. 최근에는 PV-RCNN, Pyramid R-CNN와 같이 복셀 기반 방식, 포인트 기반 방식의 장점을 모두 융합하여 보다 좋은 성능을 달성한 신경망이 등장하였다.

Table 2에서 비교한 신경망 외에도 매우 다양한 신경망들이 존재하며, 앞서 분류한 방식처럼 포인트 클라우드를 어떻게 변환하여 특성을 추출하는지, 공간적 또는 시멘틱 정보를 어떤 형태의 인코더를 활용하여 추출하는지, 마지막 경계 박스 및 물체 검출을 위해 어떤 방식의 객체 검출단을 사용하고 보정단을 어떻게 구성하는지에 따라 매우 다양하고 융합된 방법들이 제시되고 있다. 매년 라이다 객체 인식 연구의 가장 좋은 성능을 달성하는 SOTA(State of the Art) 신경망이 지속적으로 업데이트 되고 있으며 이러한 연구 동력은 곧 자율 주행 자동차의 인지 성능 향상에 크게 이바지하고 있다. 자율 주행 자동차의 상용화가 빠르게 다가오고 있는 만큼[1] 3D 객체 인식 연구 분야는 앞으로도 계속해서 빠른 속도로 발전할 전망이다.

후 기

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(2021R1A2C3008370)을 받아 수행된 연구 결과임.

References

1) Y. Kim, T. Lee and B. Song, “Training and Performance Analysis of Vehicle Detection Neural Networks to Field Test and Simulation Datasets of Multi-Channel Lidar,” Transactions of KSAE, Vol.29, No.12, pp.1123-1132, 2021.
2) Z. Sun, Z. Li and Y. Liu, “An Improved Lidar Data Segmentation Algorithm Based on Euclidean Clustering,” Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC), pp.1119-1130, 2020.
3) K. P. Sinaga and M. S. Yang, “Unsupervised K-means Clustering Algorithm,” IEEE Access, Vol.8, pp.80716-80727, 2020.
4) J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.779-788, 2016.
5) J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.7263-7271, 2017.
6) J. Redmon and A. Farhadi, “Yolov3: An Incremental Improvement,” arXiv preprint arXiv:1804.02767, 2018.
7) R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.580-587, 2014.
8) R. Girshick, “Fast R-CNN,” In Proceedings of the IEEE International Conference on Computer Vision, pp.1440-1448, 2015.
9) K. He, G. Gkioxari, P. Dollár and R. Girshick, “Mask R-CNN,” In Proceedings of the IEEE

International Conference on Computer Vision(ICCV), pp.2961-2969, 2017.
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 Vol.28, pp.91-99, 2015.
11) E. Arnold, O. Y. A. Jarrah, M. Dianati, S. Fallah, D. Oxtoby and A. Mouzakitis, “A Survey on 3D Object Detection Methods for Autonomous Driving Applications,” IEEE Transactions on Intelligent Transportation Systems, Vol.20, No.10, pp.3782-3795, 2019.
12) Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao and J. Li, “Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review,” IEEE Transactions on Neural Networks and Learning Systems, Vol.32, No.8, pp.3412-3432, 2020.
13) J. H. Koh, J. K. Kim, Y. C. Kim and J. W. Choi, “Survey on Deep Learning-based 3D Object Detection Algorithms using LiDAR Data,” Communications of the Korean Institute of Information Scientists and Engineers, Vol.37, No.1, pp.61-71, 2019.
14) K. Cho, J. Im, M. Kim and S. Kang, “Feasibility Assessment of KODAS Through Autonomous Driving Recognition Challenge,” Transactions of KSAE, Vol.29, No.3, pp.233-241, 2021.
15) A. Geiger, P. Lenz and R. Urtasun, “Are We Ready for Autonomous Driving? The Kitti Vision Benchmark Suite,” IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.3354-3361, 2012.
16) H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan and O. Beijbom, “NuScenes: A Multimodal Dataset for Autonomous Driving,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp.11621-11631, 2020.
17) P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen and D. Anguelov, “Scalability in Perception for Autonomous Driving: Waymo Open Dataset,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp.2446-2454, 2020.
18) M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” International Journal of Computer Vision, Vol.88, No.2, pp.303-338, 2010.
19) T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” European Conference on Computer Vision(ECCV), pp.740-755, 2014.
20) H. Su, S. Maji, E. Kalogerakis and E. L. Miller, “Multi-View Convolutional Neural Networks for 3D Shape Recognition,” Proceedings of the IEEE International Conference on Computer Vision(ICCV), pp.945-953, 2015.
21) X. Chen, H. Ma, J. Wan, B. Li and T. Xia, “Multi-View 3D Object Detection Network for Autonomous Driving,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.1907-1915, 2017.
22) J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia and A. D. L. Escalera, “Birdnet: A 3D Object Detection Framework from Lidar Information,” 21st International Conference on Intelligent Transportation Systems(ITSC), pp.3517-3523, 2018.
23) A. Barrera, C. Guindel, J. Beltrán and F. García, “Birdnet+: End-to-End 3D Object Detection in LIDAR Bird’s Eye View,” IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp.1-6, 2020.
24) C. R. Qi, H. Su, K. Mo and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.652-660, 2017.
25) C. R. Qi, L. Yi, H. Su and L. J. Guibas, “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” Advances in Neural Information Processing Systems(NeurIPS), p.30, 2017.
26) D. Maturana and S. Scherer, “Voxnet: A 3D Convolutional Neural Network for Real-Time Object Recognition,” IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), pp.922-928, 2015.
27) Y. Zhou and O. Tuzel, “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.4490-4499, 2018.
28) Y. Yan, Y. Mao and B. Li, “Second: Sparsely Embedded Convolutional Detection,” Vol.18, No. 10, Paper No.3337, 2018.
29) A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom, “PointPillars: Fast Encoders for

Object Detection From Point Clouds," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp.12697-12705, 2019.
30) Y. Chen, S. Liu, X. Shen and J. Jia, "Fast Point R-CNN," Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV), pp.9775-9784, 2019.
31) S. Shi, X. Wang and H. Li, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.770-779, 2019.
32) Z. Yang, Y. Sun, S. Liu and J. Jia, "3DSSD: Point-Based 3D Single Stage Object Detector," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp.11040-11048, 2020.
33) Z. Yang, Y. Sun, S. Liu, X. Shen and J. Jia, "STD: Sparse-to-Dense 3D Object Detector for Point Cloud," Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV), pp.1951-1960, 2019.
34) S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang and H. Li, "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp.10529-10538, 2020.
35) S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang and H. Li, "PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection," arXiv preprint arXiv:2102.00463, 2021.
36) J. Mao, M. Niu, H. Bai, X. Liang, H. Xu and C. Xu, "Pyramid R-CNN: Towards Better Performance and Adaptability for 3D Object Detection," Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV), pp.2723-2732, 2021.
37) B. Liu, M. Wang, H. Foroosh, M. Tappen and M. Pensky, "Sparse Convolutional Neural Networks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.806-814, 2015.
38) G. Benjamin, "Spatially-sparse convolutional neural networks," arXiv preprint arXiv:1409.6070, 2014.
39) Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," Acm Transactions on Graphics, Vol.38, No.5, pp.1-12, 2019.
40) A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention Is All You Need," In Advances in Neural Information Processing Systems, pp.5998-6008, 2017.
41) H. Zhao, L. Jiang, J. Jia, P. H. Torr and V. Koltun, "Point Transformer," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp.16259-16268, 2021.
42) W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," European Conference on Computer Vision(ECCV), pp.21-37, 2016.
43) Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," Neural Computation, Vol.1, No.4, pp.541-551, 1989.
44) K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.770-778, 2016.
45) T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.2117-2125, 2017.
46) Open-mmlab Github, OpenPCDet, https://github.com/open-mmlab/OpenPCDet, 2022.
47) Y. Yan Github, Second.pytorch, https://github.com/traveller59/second.pytorch, 2019.
48) S. Shi Github, PointRCNN, https://github.com/sshaoshuai/PointRCNN, 2020.
49) DV Lab Github, 3DSSD, https://github.com/dvlab-research/3DSSD, 2020.
50) Open-mmlab Github, mmdetection3d, https://github.com/open-mmlab/mmdetection3d/blob/master/configs/3dssd, 2021.
51) KITTI 3D Object Detection Evaluation 2017, http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d, 2017.
52) J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of Lidar Sequences," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp.9297-9307, 2019.
53) D. H. Paek, S. H. Kong and K. T. Wijaya, "K-lane: Lidar Lane Dataset and Benchmark for Urban Roads and Highways," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR) Workshops, 2022.