Transactions of KSAE, Vol. 34, No. 5, pp.533-546 (May, 2026)

Copyright © 2026 KSAE / 246-03
pISSN 1225-6382 / eISSN 2234-0149
DOI http://dx.doi.org/10.7467/KSAE.2026.34.5.533

객체 검출 및 최소 깊이 추정을 위한 이미지-포인트 클라우드 융합 네트워크

신호수 · 이준웅*

전남대학교 산업공학과

Image-Point Cloud Fusion Network for Joint Object Detection and Minimum Depth Estimation

HoSu Shin · JoonWoong Lee*

Department of Industrial Engineering, Chonnam National University, Gwangju 61186, Korea

(Received 7 November 2025 / Revised 1 December 2025 / Accepted 2 December 2025)

Abstract : This study is aimed at proposing an integrated network to perform two tasks: object detection and depth estimation in autonomous driving and robotics. The approach combines a bird's eye view-based candidate generation module with an image-point cloud, cross-attention fusion structure to exploit complementary spatial and visual cues from both modalities. Moreover, an input-dependent query initialization module is employed to initiate detection in likely object regions, thereby reducing unnecessary candidates. To improve depth accuracy, Hungarian matching is applied, and performance is quantitatively evaluated by using the root mean square error. Experiments on the KITTI dataset demonstrated that the method achieved superior performance over existing approaches involving cars, pedestrians, and cyclists. These results indicate that the proposed network can provide robust and precise perception even in complex driving environments.

Key words : Cross-attention(교차 어텐션), Depth estimation(깊이 추정), Image-point cloud fusion(이미지-포인트 클라우드 융합), Input-dependent query initialization(입력 의존 쿼리 초기화), Object detection(객체 검출)

1. 서 론

최근 자율주행 기술은 빠른 발전으로 사회 전반에서 큰 관심을 받고 있다. 자율주행 차량이 안전하게 주행하려면 다양한 기술과 시스템이 유기적으로 연계되어야 하며, 그 중에서도 센서를 활용해 객체를 인식하는 시스템이 중요한 역할을 한다.[1] 카메라 영상이나 포인트 클라우드(Point cloud) 기반의 객체 검출 기술은 지속적으로 발전해 왔으나, 단일 센서만을 사용하는 모델은 복잡한 주행 환경에서 깊이 추정의 한계와 객체 검출 오류를 완전히 해결하기 어렵다. 이러한 문제를 극복하기 위해 최근에는 3D 포인트 클라우드와 2D 이미지를 결합하는 센서 융합(Sensor fusion)이 차세대 인지 시스템의 핵심 방향으로 주목받고 있다.[1-3]

단일 센서 기반 객체 인식 방법은 고유한 장점에도 불구하고 여러 한계를 지닌다. 우선, 단일 이미지 기반 방법은 저비용 대비 풍부한 영상 정보를 제공한다는 장점이 있으나, 깊이 정보가 부재하여 깊이 추정이 어렵다.[1,4] 단일 라이다 기반 방법은 깊이 정보를 제공하여 장면의 절대 깊이 추정에 강점이 있지만, 단일 객체에 대한 포인트들의 밀도가 낮고, 전체적으로 수량이 희소하여 원거리 객체나 복잡한 형상을 세밀하게 표현하기 어렵다. 아울러 재질, 색상, 텍스처와 같은 정보를 제공하지 않고, 센서 비용이 높다. 스테레오 카메라 기반 방법은 좌우 영상의 시차(Disparity)를 활용하여 깊이 정보를 추정할 수 있지만, 텍스처가 부족하거나 Ill-posed한 영역에서는 매칭 오류가 증가한다. 또한 긴 베이스라인 확보가 어려운 차량 환경에서는 원거리 깊이 추정에 대한 정밀도가 떨어진다. 따라서 최근에는 이종 센서들의 상호 보완적 특성을 결합하여 동종 단일 센서의 약점을 보완하고, 이종 센서들 간의 강점을 극대화하는 센서 융합 방식이 주목을 받고 있다.

*Corresponding author, E-mail: joonlee@chonnam.ac.kr

본 연구가 목표로 하는 카메라-라이다(Camera-LiDAR (Light Detection and Ranging)) 융합은 카메라가 제공하는 색상, 질감, 형태와 같은 시각 정보와 라이다가 제공하는 3D 포인트 클라우드를 결합함으로써, 동종 단일 센서의 단점을 보완할 수 있다. 즉, 이 두 센서를 함께 활용하면 객체의 재질, 색상, 텍스처와 깊이 정보를 확보할 수 있어서 다양한 환경 조건에서 객체의 인식 성능을 향상시킬 수 있다.

본 연구에서는 Sparse R-CNN[5]에 적용된 네트워크 구조를 확장하여 객체 검출과 객체까지의 최소 깊이 검출이 가능하도록 하였다. 여기에 라이다 포인트 클라우드로부터 구축한 Bird's Eye View(BEV)와 카메라 이미지의 특징을 교차 어텐션(Attention)[6]을 통해 결합하는 방식의 카메라-라이다 융합에 의해 객체 검출의 정확성을 높인다. 이때 적용된 교차 어텐션은 BEV 각 위치에서 자신과 밀접한 이미지 영역의 시각 정보를 집중적으로 포착한다. 이를 통해 BEV는 공간적 정밀도를 유지하면서 이미지가 제공하는 색상, 형태, 질감 정보를 효율적으로 통합할 수 있다.

기존의 센서 융합 연구에서는 단순히 특징 연결(Concatenation)이나 원소 단위 덧셈(Element-wise addition)을 활용하였는데, 이런 경우 서로 다른 입력의 특징이 무분별하게 합쳐져 불필요한 배경 잡음이 강화되거나 객체와 무관한 정보가 포함되는 문제가 있었다.[2] 반면, 본 연구에서 적용한 교차 어텐션 기반 융합 방식은 객체 인식에 필요한 정보를 선별적으로 활용하여 이러한 문제를 완화하고, 복잡한 환경에서도 정밀하고 강건한 객체 검출을 가능하게 한다.

기존의 2D 객체 검출 방법들은 초기 객체 후보를 앵커 박스나 사전 정의된 후보군에 의존하였다. 예를 들어, YOLO[7]와 Faster R-CNN[8]은 특징맵 위에 다양한 크기와 종횡 비율의 앵커 박스를 배치해 초기 객체 후보를 설정하였으며, Sparse R-CNN[5]은 학습 가능한 쿼리를 도입하였고, DiffusionDet[9]은 랜덤 후보군을 활용한다. 그러나 이러한 2D 객체 검출 방법들이나 기존의 깊이 추정 관련 방법들[10,11]은 앵커 기반 초기 객체 설정에 의존하기 때문에 앵커의 크기나 비율에 민감할 수 있다. 더 나아가 학습 가능한 쿼리나 랜덤 후보군 생성 방식은 입력 데이터와 무관하게 쿼리의 위치를 고정하거나 임의로 초기화하기 때문에 실제 객체 중심에서 떨어진 위치에서 예측을 시작할 수 있다. 이로 인해 네트워크는 여러 번의 보정 과정을 거쳐야 하며, 초기 단계에서 객체를 놓칠 가능성이 있다. 또한 랜덤 쿼리는 불필요한 배경 영역을 포함하여 오탐지를 증가시키는 문제를 유발한다.

이러한 한계를 극복하기 위해 본 연구에서의 초기 객체 생성은 BEV 특징을 기반으로 한 입력 의존 쿼리 초기화(Input-dependent query initialization)를 적용하였다. 이는 TransFusion[2]에서 제안된 방법으로, BEV 특징맵에서 클래스별 히트맵(Heat map)을 예측한 뒤 상위 k개의 위치를 객체들의 초기 중심으로 설정하여 실제 객체 중심에 가까운 곳에서 탐지를 시작한다. 이를 통해 디코딩 단계에서 과도한 정제 과정 없이 초기부터 객체 중심 근처에서 검출이 가능하다.

본 연구에서 제안한 컨볼루션 뉴럴 네트워크(Convolutional Neural Network, CNN)의 훈련과 평가에 단안 이미지, 포인트 클라우드, 객체의 2D 경계상자, 카메라에서 객체까지의 최소 깊이 정보가 필요하다. 그러나 이러한 정보를 갖춘 공개 데이터셋은 존재하지 않는다. 이에 본 연구에서는 KITTI 3D 객체 검출 데이터셋[12]에 Song and Lee[10]의 데이터 가공 방식을 적용하여 데이터셋을 구축하였다. 이 데이터셋으로 제안한 CNN을 기존 기법들과 비교 분석하였다.

이어지는 논문의 구성은 다음과 같다. 2장에서 관련 연구를 소개하고, 3장에서 본 연구에서 제안한 CNN을 설명한다. 4장에서는 실험 결과를 제시하고, 5장에서 논문을 마무리한다.

2. 관련 연구

이 장에서는 2D 객체 검출 및 깊이 추정과 관련된 연구 방법과 카메라-라이다 센서 융합 관련 연구 방법들을 소개한다.

2.1 2D 객체 검출 및 깊이 추정

객체를 2D 경계상자로 검출하고, 객체까지의 깊이를 예측하는 연구는 주로 단안 또는 스테레오 이미지를 활용한다. Dist-YOLO[11]는 YOLO[7]와 같은 객체 검출기를 기반으로, 각 객체의 예측 벡터에 절대 깊이 항을 추가하고 손실 함수에 깊이 오차 항을 포함시켜 단안 영상만으로 객체 검출과 깊이 추정을 동시에 수행하도록 확장한 방법이다. 별도의 깊이 추정 네트워크를 사용하지 않고, 기존 YOLO 백본과 검출 헤드를 그대로 공유하면서, 각 검출 박스에 대해 객체 클래스와 카메라에서 객체까지의 깊이를 함께 회귀하도록 설계되었다. Song and Lee[10]은 스테레오 영상으로부터 객체 검출과 깊이 추정 방법을 제안하였다. 그들은 좌우 이미지에서 추출한 특징맵을 이용해 비용 볼륨을 구축하고, 이를 기반으로 시차 회귀를 수행한다. 이후 회귀된 시차 특징을 백본과 디코더에서 생성된 이미지 특징맵과 융합하여 깊이를 예측한다.

이 방법들은 비교적 단순하고 효율적이라는 장점이 있으나, 객체들 간 경계가 불분명하고 객체들이 중첩되

는 상황에 취약하다. 원거리 객체의 경우 검출 정확도와 깊이 추정 성능이 저하되기도 한다.

2.2 카메라 - 라이다 센서 융합

카메라-라이다 융합 방식은 결합되는 단계에 따라 크게 결과 수준 융합, 제안 수준 융합, 그리고 포인트 수준 융합으로 분류된다.[2,3] 결과 수준 융합의 대표적 연구인 FPointNet[13]은 이미지 기반 2D 객체 검출기를 활용하여 3D 후보 영역을 형성하고, 해당 영역 내의 포인트 클라우드에 PointNet[14]을 적용하여 최종 객체 위치를 검출하였다. 이러한 방식은 구조가 단순하고 직관적이라는 장점이 있으나, 3D 객체 검출 결과가 2D 검출기의 성능에 크게 의존하기 때문에 2D 객체 검출기가 부정확한 경우 전체 성능이 떨어지는 단점이 있다.

AVOD(Aggregate View Object Detection)[15]는 제안 수준 융합 방법으로 라이다 BEV 특징맵에 앵커 기반 제안 생성기를 이용하여 3D ROI(Region of Interest)를 생성하고, 이를 동일한 ROI로 이미지와 BEV에 투영한 후 두 뷰의 대응 영역에서 RoIAlgin[16] 연산을 적용하여 특징을 추출한다. 이후 추출된 특징을 결합하여 최종 검출을 수행한다. 이러한 접근은 이미지와 라이다 정보를 동시에 활용할 수 있는 장점이 있으나, ROI 내부에 불필요한 배경 정보가 포함되어 최종 검출 정확도를 저해할 수 있다.

EPNet[17]은 포인트 수준 융합 방식으로 라이다 포인트를 이미지 평면에 투영한 뒤 해당 위치에서 추출한 이미지 특징을 각 포인트의 특징과 직접 결합한다. 이를 통해 시각 정보와 공간 정보를 포인트 단위에서 융합함으로써 객체 검출 성능을 향상시킨다. 이 방식은 포인트 클라우드의 희소성을 보완하는 동시에 각 포인트에 이미지에서 추출된 고차원 시각 특징을 결합하여 객체에 대한 의미 정보를 반영하는 장점이 있다. 그러나 카메라와 라이다 간의 캘리브레이션(Calibration)에 의존하며, 작은 정합 오차에도 성능이 저하될 수 있는 한계가 있다.

이러한 전통적인 세 가지 수준의 융합 방법들은 각각 장점을 가지면서도 정보 결합의 정밀도나 센서 정합 문제에서 한계를 보였다. 최근에는 이러한 문제를 극복하기 위해 교차 어텐션을 활용한 융합 연구가 활발히 진행되고 있다. 대표적으로 TransFusion[2]은 라이다 BEV에서 생성된 후보를 쿼리로 초기화한 뒤 이미지 특징과 교차 어텐션을 수행하여 BEV 위치와 관련성이 높은 시각 정보를 선택적으로 통합하였다. 이를 통해 ROI 수준에서 발생하던 불필요한 배경 잡음을 억제하고, 정확성과 효율성을 향상시켰다. 이러한 교차 어텐션 기반 융합은 센서 간 정합 오차에 덜 민감하며, 시각적·공간적 정보를 안정적으로 결합할 수 있다는 장점을 가진다. 결과적으로 교차 어텐션을 활용한 융합은 기존의 단순 결합 방식보다 더 세밀하고 효율적인 정보 교환을 가능하게 한다.

3. 제안 방법

이 장은 본 연구에서 제안된 객체 검출 및 깊이 추정 네트워크 구조와 손실 함수를 설명한다. 이 네트워크는 Fig. 1에 보인 바와 같이 단일 이미지와 3D 포인트 클라우드를 입력 받아 2D 객체 검출과 객체별 깊이를 종단 간(End-

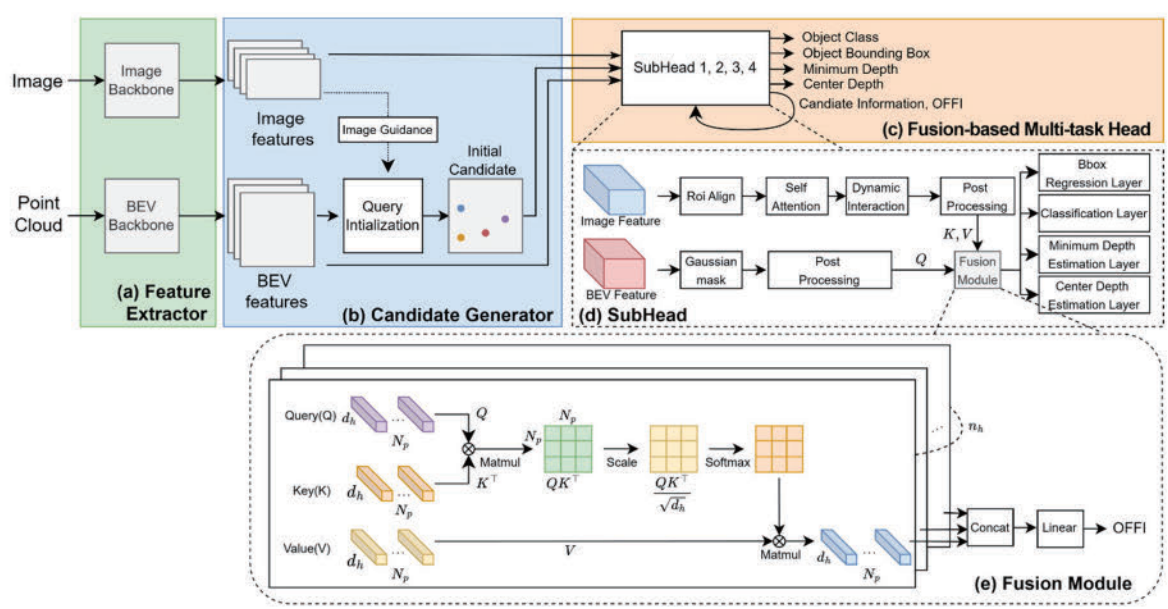


Fig. 1 Overall architecture of the proposed method

to-End)으로 예측한다. 네트워크는 크게 특징 추출기, 후보 생성기, 융합 기반 다중 작업 헤드로 구성된다.

3.1 특징 추출기

본 연구에서 제안된 특징 추출기는 카메라로 획득한 이미지와 라이다로 획득한 포인트 클라우드를 입력으로 받아 병렬로 처리한 뒤, 각각의 특징 맵을 생성한다.

3.1.1 이미지 백본

본 연구에서는 KITTI 3D 객체 검출 데이터 셋[12]의 공간 해상도 1242×375의 RGB 이미지를 ImageNet[18]에서처럼 채널별로 정규화를 거친 후, 이미지 백본 네트워크인 ResNet-50[19]에 입력한다. ResNet-50은 입력된 이미지로부터 여러 해상도의 특징맵을 단계적으로 추출한다. 각 단계의 출력은 입력 해상도에 대해 1/4, 1/8, 1/16, 1/32의 크기이며, 이들의 채널 수는 각각 256, 512, 1024, 2048이다. 이후 특징 피라미드 네트워크 (Feature Pyramid Network, FPN)[20]를 통해 이 특징맵들을 1×1 합성곱을 적용하여 256차원으로 변환한다. 그리고 이 변환된 특징맵과 업샘플링된 상위 단계의 특징맵을 같은 위치의 요소끼리 더한 후, 3×3 합성곱을 적용하여 해상도 1/4, 1/8, 1/16, 1/32에 해당하는 네 개의 256채널 특징맵 (P_2-P_5)을 생성한다. 이렇게 생성된 FPN 특징들은 바로 직전의 저해상도 맵으로부터 의미적 정보와 현 단계의 고해상도 맵으로부터 공간적 세부 정보를 결합한 것으로 다양한 크기의 객체들을 안정적으로 검출할 수 있게 한다. 이렇게 생성된 다중 스케일 이미지 특징은 이후에 BEV 특징과 결합되어 객체 검출 및 깊이 추정에 활용된다.

3.1.2 BEV 백본

본 연구에 적용한 BEV 백본은 VoxelNet[21]이다. 이 백본을 통해 라이다 포인트 클라우드로부터 본 연구의 목표인 객체 검출과 깊이 추정에 활용할 BEV 특징맵을 생성한다. 그 절차는 다음과 같다.

라이다 포인트 클라우드 데이터는 3차원 좌표인 x, y, z와 라이다에서 조사된 레이저의 반사강도로 구성되어 있는데, 희소하고 불균일하게 흩어져 있어서 CNN과 같은 네트워크에 적용하기 어렵다. 이를 해소하기 위해 포인트 클라우드를 복셀화(Voxelization) 과정을 거쳐 격자 단위로 나눈다. 먼저 포인트 클라우드를 이미지 데이터와 융합을 위해 라이다-카메라 캘리브레이션 데이터를 이용하여 카메라 좌표계로 변환한다. 그리고 VoxelNet에서 했던 것과 유사하게 카메라 전방으로 카메라 좌표계를 기준으로 X축 방향으로 $[X_{min}, X_{max}]$ = [-40, 40], Y축 방향으로 $[Y_{min}, Y_{max}]$= [-1, 3], Z축 방향으로 $[Z_{min}, Z_{max}]$ = [0, 100] (여기에서 단위는 m)의 범위를 관심 공간 (Space of Interest, SOI)으로 설정하고, 이 공간을 $(s_X, s_Y, s_Z) = (0.2, 0.2, 0.2)(m)$ 크기의 격자들로 분할한다. 이때 Z축 방향의 범위를 VoxelNet의 Z_{max}=70.4 m보다 더 멀리 설정한 것은 더 원거리에 있는 객체들도 검출하기 위함이다. SOI의 각 격자를 복셀(voxel)이라 한다. 따라서 복셀의 수는 400×20×500이 된다. 이후 각 복셀에 점들을 할당함으로써 포인트 클라우드의 복셀화가 완성된다.

각 복셀 V_v에 할당된 점들의 집합을 $\{p_{v,i}\}_{i=1}^{n_v}$라 한다. 여기에서 n_v는 V_v에 포함된 점의 개수이다. 계산 및 메모리 효율성 확보를 위해 VoxelNet처럼 복셀당 할당되는 점들의 수를 최대 32개로 제한한다. 할당된 점이 32개를 초과하면 먼저 할당된 32개만 유지하고, 점이 없는 복셀은 특징 추출과정에서 배제한다.

객체 검출과 깊이 추정을 위해 복셀화된 포인트 클라우드의 특징 추출이 필요하다. 이를 위해 각 복셀의 내부 점 정보를 VoxelNet에서 제안된 복셀 특징 인코딩(Voxel Feature Encoding, VFE) 모듈을 사용하여 고정 차원의 벡터로 임베딩(Embedding)한다.

VFE 모듈은 복셀에 할당된 점들의 좌표 x, y, z와 반사강도를 입력 받아, 점 단위로 다층 퍼셉트론(Multi-Layer Perceptron, MLP)을 통과시킨다. 이 MLP는 완전연결 층 - ReLU 함수 - 완전연결 층 - ReLU 함수로 구성되어 있으며, 4차원 입력을 32차원 벡터 $h_{v,i} \in \mathbb{R}^{32}$로 변환한다. 이 임베딩은 점의 공간적 위치와 반사 특성을 고차원 특징으로 변환한 것으로, 복셀 내부에서 점들이 가지는 상대적 위치 패턴을 표현하도록 학습된다. 이후 복셀 내의 모든 점들의 임베딩 $\{h_{v,i}\}_{i=1}^{n_v}$에 특징 차원별로 최대값을 선택하는 맥스 풀링(Max pooling)을 적용하여 1차 복셀 대표 벡터 $v_v^{(1)} \in \mathbb{R}^{32}$를 얻는다. 즉, 각 특징 차원마다 n_v개의 점들 중 가장 큰 값을 선택함으로써, 복셀 내부 점들의 특징 분포 중 가장 강한 특징을 추출한다. 이 벡터는 다시 각 점의 임베딩과 결합되어 $[h_{v,i}; v_v^{(1)}] \in \mathbb{R}^{64}$의 벡터를 형성한다. 이 결합 벡터는 앞서 언급한 MLP와 동일한 구성의 MLP를 거쳐 2차 점 단위 임베딩 $h'_{v,i} \in \mathbb{R}^{128}$으로 변환되고, 다시 동일한 방식으로 특징 차원별 맥스 풀링을 수행하여 2차 복셀 대표 벡터 $v_v^{(2)} \in \mathbb{R}^{128}$을 생성한다. 마지막으로 $[h'_{v,i}; v_v^{(2)}] \in \mathbb{R}^{256}$을 또 한 번의 MLP와 맥스 풀링 연산에 통과시켜 최종 복셀 특징 $v_v^{final} \in \mathbb{R}^{128}$을 생성한다. 이 벡터는 점 수준의 세밀한 위치 정보와 복셀 수준의 전역적 구조 정보를 모두 통합한 고정 차원 표현으로, 이후 3차원 합성곱 연산의 입력으로 사용된다.

이렇게 얻은 복셀별 특징 v_v^{final}들은 차원별로 각 복셀이 대응하는 3차원 공간상의 위치에 따라 격자 형태로 배열되어 $F_{4D} \in \mathbb{R}^{W \times D \times H \times C}$의 4차원 특징맵을 형성한다. 여

기서 C는 복셀별 특징의 차원, W, D, H는 각각 수평(X), 수직(Y), 전방(Z) 방향 복셀의 수량으로서 W는 400, D는 20, H는 500, C는 128이다. 이러한 4차원 특징은 두 개의 3D 합성곱 블록에 입력되어 인접 복셀 간의 공간적 상관관계를 학습한다. 첫 블록은 커널 크기 3×3×3, 스트라이드(Stride) 1로 지정되었고, 입력 128채널과 동일한 128채널로 변환한다. 이때 컨벌루션 결과에 배치 정규화와 ReLU 함수를 적용하여 비선형성을 부여한다. 두 번째 블록은 첫 블록과 동일한 크기의 커널과 스트라이드를 유지하되, 출력 채널 수를 128에서 256으로 늘려 보다 풍부한 공간 표현을 학습하도록 하였고. 두 번째 블록 역시 배치 정규화와 ReLU를 적용한다. 두 블록을 통과한 출력 특징은 $F'_{4D} \in \mathbb{R}^{W\times D\times H\times 256}$이다.

이후 F'_{4D}는 식(1)에 의해 수직(Y) 방향으로 평균을 취하여 XZ 평면상의 특징 $F_{BEV} \in \mathbb{R}^{W_B\times H_B\times C_B}$로 변환된다. 즉 $\mathbb{R}^{W\times D\times H}$가 $\mathbb{R}^{W_B\times H_B}$로 변환된 것이고, 이 $\mathbb{R}^{W_B\times H_B}$ 차원의 특징이 C_B개 있는 것이다. 여기서 $W_B = W = 400$, $H_B = H = 500$, C_B는 256이다.

$$F_{BEV}(X,Z,C) = \frac{1}{D}\sum_{Y=1}^{D} F'_{4D}(X,Y,Z,C) \quad (1)$$

우리는 이와 같이 XZ평면으로 투영된 특징 F_{BEV}를 BEV특징이라 한다. 이렇게 얻어진 BEV 맵은 포인트 클라우드의 3차원 구조 정보를 보존하면서 위에서 내려다본 것과 같은 2차원 표현을 제공한다. 이렇게 생성된 BEV 특징은 이후 단계에서 이미지 특징과 결합되어 객체 검출과 깊이 추정에 활용된다.

3.2 후보 생성기

본 연구에서 초기 객체 후보 생성은 TransFusion[2]에서 제안된 영상 기반 쿼리 초기화 방법을 이용한다. 이 방법은 BEV 특징만 활용한 객체 후보 생성[22]이 라이다 포인트가 희소한 영역에서 객체 탐지 누락을 발생시키는 문제의 보완을 위해 이미지 특징과 BEV 특징의 융합을 통해 보다 강건한 초기 후보 생성을 가능하게 한다. 이 두 특징의 융합은 교차 어텐션을 이용한다.

교차 어텐션에 적용할 이미지 특징은 FPN이 생성한 특징맵들 가운데 해상도가 가장 높은 P_2를 사용한다. 3D 맵인 P_2를 세로축으로 맥스풀링하여 2D 특징 $F_P \in \mathbb{R}^{W_I\times C_I}$로 변환한다. 여기서 C_I는 채널 수이고 W_I는 너비이다. 이어서 F_P의 열 벡터를 병렬로 서로 다른 완전 연결층을 통과시켜 차원 E=256의 임베딩 벡터를 두 개를 얻고, 하나는 BEV 특징과의 교차 어텐션의 키(K)로, 다른 하나는 밸류(V)로 이용한다.

3D 맵인 F_{BEV}는 F_P와 차원을 맞추기 위해 F_{BEV}의 채널마다 $W_B \times H_B$ 평면을 평탄화(Flatten)하여 W_BH_B차원 벡터로 변환하면 $F_L \in \mathbb{R}^{(W_BH_B)\times C_B}$이 된다. 이후 이미지 특징의 처리와 유사하게 F_L의 열 벡터를 완전 연결 층을 통과시켜 E차원의 벡터를 얻고, 이를 교차 어텐션의 쿼리(Q)로 사용한다. 즉, Q, K, V 각각은 Fig. 2에 보인 멀티헤드 교차 어텐션 메커니즘[6]에 헤드수 $n_h = 8$에 맞게 동일한 길이 $d_h = E/n_h$로 분할된 후 입력된다.

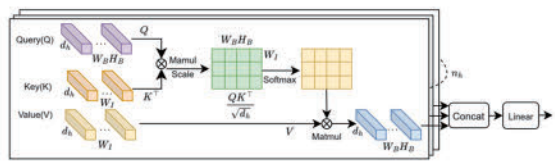


Fig. 2 Structure of the multi-head cross-attention

Fig. 2에 표현된 Matmul은 행렬곱을 나타낸다. 헤드별로 교차 어텐션 수행 후, 모든 헤드의 출력을 연결(Fig. 2의 Concat)하여 다시 E차원의 벡터를 얻고, 이를 완전 연결층(Fig. 2의 Linear)에 통과시킨다. 이 일련의 과정을 F_P와 F_L의 모든 열벡터에 대해 수행하면 2차원 맵 $F_{fused} \in \mathbb{R}^{(W_BH_B)\times C_B}$가 생성된다. F_{fused}를 F_{BEV}의 공간 해상도 (W_B, H_B)에 맞게 재구성하여 라이다-카메라 융합 특징맵 $F_{LC} \in \mathbb{R}^{W_B\times H_B\times C_B}$을 얻는다.

융합 특징인 F_{LC}만으로 초기 객체 후보 탐색이 가능하지만, 본 연구에서는 F_{BEV}도 이 작업에 활용한다. 이를 위해 식 (2)에 기술한 것처럼 F_{BEV}에 3개의 3×3 합성곱 블록(합성곱, 배치 정규화, ReLU로 구성)과 1×1 합성곱층을 거쳐 검출 대상 클래스 수 N_c와 동일한 채널을 갖는 특징을 얻고, 시그모이드 함수를 적용하여 히트맵 $H^{F_{BEV}} \in \mathbb{R}^{W_B\times H_B\times N_c}$를 생성한다.

$$H^{F_{BEV}} = \sigma(Conv_{1\times1}(f_{conv3}(f_{conv2}(f_{conv1}(F_{BEV}))))) \quad (2)$$

여기에서 $f_{conv}(\cdot)$는 3×3 합성곱 블록 내에서 수행되는 연산을 의미하며, $Conv_{1\times1}$는 1×1 합성곱을 나타내고, $\sigma(\cdot)$는 시그모이드 함수이다. $H^{F_{BEV}}$의 각 셀의 값은 그 셀의 위치가 객체 중심이 될 확률을 나타낸다.

F_{LC}에도 F_{BEV}로부터 $H^{F_{BEV}}$을 얻는 동일한 과정을 적용하여 히트맵 $H^{F_{LC}}$를 예측한다. 그리고 식 (3)과 같이 이들의 평균을 취해 최종 히트맵을 얻는다.

$$H^{Final} = \frac{1}{2}(H^{F_{BEV}} + H^{F_{LC}}) \quad (3)$$

우리는 H^{Final}에서 클래스에 관계없이 확률값이 높은 상위 N_p개의 픽셀을 선택하여 초기 객체 후보 집합을 구성한다. 본 연구에서는 실험적으로 $N_p = 200$으로 정하였다.

히트맵에서 선택된 후보 픽셀 좌표 (i, j)는 격자 해상도 s_X, s_Z와 복셀화 과정에서 정의한 SOI의 X, Z축 방향의 범위 $[X_{min}, X_{max}]$, $[Z_{min}, Z_{max}]$을 이용하여 식 (4)에 의해 3차원 점 $(\hat{X}, \hat{Y}, \hat{Z})$으로 변환된다.

$$(\hat{X}, \hat{Y}, \hat{Z}) = (X_{min} + (i + 0.5)s_X, \bar{y}_{cls}, Z_{min} + (j + 0.5)s_Z) \quad (4)$$

여기서 $\bar{y}_{cls}$는 KITTI 데이터셋에서 제공된 객체 클래스별 3D 경계상자들의 수직 중심인 Y좌표 평균이다. 이는 객체 클래스별 중심 높이 차이를 반영하기 위한 것으로 차량, 보행자, 사이클리스트 클래스 각각에 대해 $\bar{y}_{car}$=1.530, $\bar{y}_{pedstrian}$=1.768, $\bar{y}_{cyclsit}$=1.723으로 계산되었다. 이렇게 얻은 3차원 점은 카메라 내부파라미터 행렬 A를 이용하여 식 (5)에 의해 이미지 평면으로 투영된다.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} \hat{X}/\hat{Z} \\ \hat{Y}/\hat{Z} \\ 1 \end{bmatrix} \quad (5)$$

여기서 (u, v)는 이미지 평면의 좌표다. 이렇게 이미지 평면에 투영된 위치를 이미지상의 후보 중심점(c_x^{img}, c_y^{img})으로 정하고, 이 점을 중심으로 2D 경계상자를 설정한다. 이 경계상자의 높이(h)와 폭(w)은 DiffusionDet[9]의 절차를 참조하여 평균 0.5, 표준편차 0.25의 가우시안 분포에서 샘플링한 값으로 정한다. 이때 샘플링된 값이 0보다 작으면 0으로, 1보다 크면 1로 클리핑(clipping)하였다. 또한, 초기 객체 후보의 최소 깊이(D_{min})와 중심 깊이(D_{center}) 모두 식 (4)에서 계산된 $\hat{Z}$을 사용한다. 즉, 객체 후보의 정보는 클래스, 이미지 평면상의 중심 좌표, 경계상자 크기, 두 가지 깊이 정보 (class, c_x^{img}, c_y^{img}, w, h, D_{min}, D_{center})$\in \mathbb{R}^7$로 구성된다. 이렇게 생성된 초기 후보들은 융합 기반 다중 헤드 네트워크에 입력되어 객체 분류, 경계상자 회귀, 깊이 추정에 활용된다.

3.3 융합 기반 다중 작업 헤드

본 연구에서 제안된 다중 작업 헤드는 Fig. 1(c)에 보인 것처럼 네 단계의 서브헤드로 구성된 반복 정제 구조를 갖는다. 모두 동일한 구조를 갖는 서브헤드는 Fig. 1(d)에 보인 바와 같이 이미지와 포인트 클라우드로부터 생성된 특징맵들에서 객체 후보 영역에 해당되는 특징을 취해 객체의 클래스, 2D 경계상자, 최소 깊이 및 중심점 깊이를 예측한다. 이미지를 처리하는 헤드의 기본 구조는 Sparse R-CNN[5]의 헤드를 기반으로 하고, 이미지 특징과 BEV 특징을 융합하는 모듈이 추가된다. 각 단계에서 예측된 클래스, 경계상자, 깊이 정보는 순차적으로 다음 단계의 서브헤드에 입력되며, 이를 통해 객체의 위치, 크기, 깊이 값들을 점진적으로 보정한다. Fig. 1(e)에 묘사된 융합 모듈에 사용된 Q, K, V는 후보 생성기에 적용된 어텐션 모듈(Fig. 2)과 같이 각각 쿼리, 키, 밸류 벡터이며, 교차 어텐션에 의한 BEV특징과 이미지 특징의 융합에 사용된다.

3.3.1 객체 후보별 이미지 특징 추출

객체 후보의 객체 여부 판별을 위해 NVD[23]에서 했던 것처럼 FPN이 생성한 특징맵에서 객체의 경계상자 영역을 RoIAlign[16]으로 추출하여 7×7×256의 텐서로 변환한다. 여기서 256은 FPN 출력맵의 채널 수이다. RoIAlign은 객체 후보의 크기에 합당한 해상도의 특징맵에서 수행되는데, 이 특징맵의 선택은 식 (6)을 이용한다. 3.1.1절에서 설명했듯이 FPN은 4단계의 특징맵을 생성하므로 이들 가운데 객체 후보의 경계상자의 (w, h)에 따라 적절한 레벨 l을 결정하여 이 레벨에 해당하는 특징맵 P_ℓ을 선택한다.

$$\ell = \left\lfloor \ell_0 + \log_2\left(\frac{\sqrt{wh}}{224}\right) \right\rfloor, \ell_0 = 4 \quad (6)$$

이 절차를 통해 영상에 작게 투영된 객체는 고해상도 특징맵, 크게 투영된 객체는 저해상도 특징맵에서 RoIAlign이 수행되어 후보 크기에 적합한 특징을 추출할 수 있다. 이렇게 얻어진 후보별 특징을 영역 특징 정보(Region Feature Information, RFI)라 한다.

본 연구는 이미지에서 RFI뿐만 아니라 RFI의 비효율적인 특징을 걸러내기 위해 객체 이미지 특징 정보(Object-Image Feature Information, OIFI)도 필요로 한다. 그런데 다중 작업 헤드의 첫 서브헤드는 이전 단계가 없으므로 NVD[23]에서 했던 것처럼 RFI를 채널별로 평균 풀링을 하여 256차원의 벡터로 만들고 이를 OIFI로 삼는다. OIFI는 후보 영역 전체의 시각적 특징을 단일 벡터로 변환한 것으로 후보의 전역적인 특징을 가지고 있는 256차원의 벡터이고, 전체 후보 수 N_p 각각에 대해 OIFI가 생성된다. 두 번째 서브헤드부터는 이전 단계의 서브헤드에서 생성된 OIFI가 전달되므로 RFI를 풀링할 필요가 없다. 서브헤드에서 RFI와 OIFI가 생성된 이후 이들의 처리 과정 중에 있는 셀프 어텐션이나 동적 인스턴스 상호작용은 Sparse R-CNN[5]이나 NVD[23]에서 수행된 방법을 따랐다. 이러한 처리 과정을 거쳐 정제된 OIFI는 완전연결 층을 통과한 후, 드롭아웃(Dropout)과 레이어 정규화로 이루어진 후처리 과정을 거쳐 이미지 특징과 BEV 특징 융합 모듈에 입력되어 교차 어텐션의 키와 밸류로 사용된다.

3.3.2 객체 후보별 BEV 특징 추출

BEV 평면(X-Z평면)상의 객체 후보의 중심 좌표는 이전 단계 서브헤드에서 예측된 이미지상의 객체 중심점 (c_x^{img}, c_y^{img})로부터 얻어진다. 먼저 (c_x^{img}, c_y^{img})를 식 (7)에 의해 카메라 좌표계로 역투영한다.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = D_{center} \cdot A^{-1} \begin{bmatrix} c_x^{img} \\ c_y^{img} \\ 1 \end{bmatrix} \tag{7}$$

여기서 A는 식 (5)에서 사용된 카메라 내부 행렬이고, D_{center}는 서브헤드에서 예측된 객체 중심점의 깊이다. (X_c, Y_c, Z_c)는 식 (8)에 의해 BEV 평면으로 변환된다.

$$c_x^{BEV} = \frac{X_c - X_{min}}{s_X}, c_z^{BEV} = \frac{Z_c - Z_{min}}{s_Z} \tag{8}$$

여기에 사용된 파라미터들은 식 (4)에 사용된 것들과 동일하다. 이렇게 얻어진 (c_x^{BEV}, c_z^{BEV})는 BEV 특징맵 $F_{BEV} \in \mathbb{R}^{W_B \times H_B \times C_B}$상에서 각 후보의 중심 위치이며, 이후 가우시안 마스크의 중심점으로 사용된다. 단 첫 서브헤드에서는 H^{Final}에서 선정된 초기후보 중심점이 (c_x^{BEV}, c_z^{BEV})로 사용된다.

BEV 특징맵에서는 객체 후보마다 BEV 평면상에서 그 후보의 중심 좌표를 중심으로 가우시안 마스크를 적용하여 로컬 특징을 통합한다. 단순히 해당 중심 좌표의 특징만 취하면 캘리브레이션 파라미터 오차로 인해 실제 객체의 BEV상 위치와 불일치할 수 있기 때문에 단일 픽셀의 특징을 사용하는 대신 가우시안 마스크 기반 가중합 방식을 적용한다. 이는 후보 중심으로부터 거리에 따라 가중치를 차등 부여함으로써 중심부 특징을 강조하고 주변의 잡음을 억제하는 효과를 가진다. 후보 p에 대한 가우시안 마스크 $M_p(i,j)$는 식 (9)와 같이 정의된다.

$$M_p(i,j) = exp\left(-\frac{(i-c_x^{BEV})^2 + (j-c_z^{BEV})^2}{\sigma r^2}\right) \tag{9}$$

여기서 (i,j)는 마스크의 픽셀 좌표이고, σ는 가우시안 분포의 퍼짐을 조절하는 하이퍼 파라미터이며, r은 반지름을 나타낸다. 계산의 효율성을 고려하여 마스크는 BEV 맵 전체 영역이 아닌 $(2r+1) \times (2r+1)$ 크기의 윈도우 내에서만 계산한다. σ와 r은 CenterPoint[22]의 방법을 따라 차량의 경우 $r = 3, \sigma = 1$를, 보행자와 사이클리스트의 경우는 $r = 1, \sigma = 0.33$를 사용하였다. 또한 클래스별로 r과 σ 값이 고정되어 있으므로 식 (9)에 의해 정의되는 가우시안 마스크를 사전에 계산하여 룩업(Lookup) 테이블화함으로써, 매번 마스크를 생성하는 연산 비용을 없앴다.

이 가중치 마스크를 이용하여 BEV 특징 맵 $F_{BEV} \in \mathbb{R}^{W_B \times H_B \times C_B}$에서 후보 p에 대한 로컬 피처 $f_{p,c}$를 다음 식 (10), (11)에 의해 얻는다.

$$\tilde{f}_{p,c} = \sum_{i=c_x^{BEV}-r}^{c_x^{BEV}+r} \sum_{j=c_z^{BEV}-r}^{c_z^{BEV}+r} M_p(i,j) \cdot [F_{bev}]_{i,j,c} \tag{10}$$

$$f_{p,c} = \frac{\tilde{f}_{p,c}}{\sum_{i,j} M_p(i,j) + \varepsilon} \tag{11}$$

여기서 $c \in [0, C_B - 1]$는 채널 인덱스이고, $[F_{BEV}]_{i,j,c}$는 채널 c에서 위치 (i, j)에 해당하는 BEV 특징값이다. ε는 분모가 0이 되는 것을 방지하기 위한 상수로서 매우 작은 값을 부여한다. 즉, 식 (10)은 후보 중심 (c_x^{BEV}, c_z^{BEV})을 기준으로 주변 픽셀들의 BEV 특징의 가중합을 구하고, 식 (11)은 이 가중합을 전체 마스크의 합으로 나눈다. 이를 통해 후보 중심 주변의 국소 특징이 안정적으로 표현되며, 마스크의 크기나 범위에 따른 스케일 차이를 보정한다. 이러한 방식은 객체 중심에서 멀어질수록 특징의 기여도를 줄여 안정적인 로컬 표현을 형성하며, 결과적으로 후보 중심의 공간적 맥락 정보를 보존하면서도 주변의 특징을 반영할 수 있게 된다.

객체 후보 p에 대해 F_{BEV}의 채널마다 로컬 특징 $f_{p,c}$를 얻으면 벡터 $f_p \in \mathbb{R}^{C_B(=256)}$가 생성된다. 본 연구에서 이 벡터를 객체의 BEV 특징 정보(Object BEV Feature Information, OBFI)라 한다. 즉, OBFI는 객체 후보의 공간 내의 로컬 정보를 256차원으로 임베딩한 벡터로서 완전 연결 층을 통과한 후, 드롭아웃과 레이어 정규화의 후처리 과정을 거친다. 이렇게 정제된 OBFI는 이미지 특징과 BEV 특징의 융합 모듈에서 쿼리로 사용된다.

3.3.3 이미지-BEV 특징 융합

BEV 특징맵과 이미지 특징맵에서 추출된 OBFI와 OIFI를 교차 어텐션을 통해 융합한다. 이를 통해 두 특징을 결합하여 객체 검출과 깊이 추정 성능을 향상시킨다. 이 어텐션에서 OBFI는 쿼리(Q)로 OIFI는 키(K)와 밸류(V)로 사용된다. 이 세 벡터 $Q \in \mathbb{R}^{N_p \times 256}, K \in \mathbb{R}^{N_p \times 256}, V \in \mathbb{R}^{N_p \times 256}$는 Fig. 1(e)에 보인 융합 모듈에 입력되어 어텐션을 수행한다. 이 과정을 통해 OBFI와 OIFI간의 상호 연관성이 학습되며, BEV 특징과 의미적으로 관련된 시각 정보가 선택적으로 융합된다. 교차 어텐션을 거쳐 생성된 객체 융합특징(Object Fusion Feature Information, OFFI)은 후속 단계인 객체의 분류, 경계상자 회귀, 최소 깊이 추정, 중심 깊이 추정 레이어들에 입력된다.

이 네 개의 레이어는 병렬로 구성되어 있으며, 모두 동일한 256차원의 OFFI를 입력으로 받지만, 서로 다른 반

복 구조와 가중치를 가지는 독립적인 MLP로 설계되어 있다. 이때 MLP는 완전 연결층과 레이어 정규화, ReLU로 구성된 블록이다.

분류 레이어는 OFFI를 입력으로 받아 MLP 블록을 한 번 통과시키고 완전 연결층과 소프트맥스 함수를 통과시켜 클래스 확률을 예측한다. 경계상자 회귀 레이어는 입력된 OFFI에 3번의 MLP 블록을 통과시켜 공간 정보를 점진적으로 정제한 뒤, 마지막으로 완전 연결 층을 거쳐 객체의 위치와 크기를 보정하기 위한 4차원 좌표 오프셋 $(\Delta x,\ \Delta y, \Delta w, \Delta h)$을 산출한다. 이 오프셋을 통해 이미지 상에서 객체 후보의 경계상자 위치와 크기를 예측한다. 깊이 회귀 레이어는 OFFI를 3번의 MLP 블록을 통과시킨 뒤 완전 연결 층을 통과시켜 단일 스칼라 값을 출력한다. 깊이 회귀 레이어는 두 개의 독립적인 브랜치로 구성되는데, 그중 하나는 객체 중심의 깊이를 나머지 하나는 카메라로부터의 객체까지 최소 깊이를 추정한다. 두 브랜치는 동일한 구조지만, 서로 독립적으로 학습된다.

3.4 손실 함수

본 연구에서는 집합 예측 손실(Set prediction loss)[24]을 적용한다. 집합 예측 손실은 클래스 예측과 경계상자 예측을 동시에 고려하며, 이를 위해 예측값과 정답(Ground truth, GT) 간의 매칭이 필요하다. 매칭 비용 계산은 식 (12)와 같이 정의된다.

$$\mathcal{C} = \lambda_{cls}\mathcal{C}_{cls}+\lambda_{L1}\mathcal{C}_{L1} + \lambda_{giou}\mathcal{C}_{giou} \tag{12}$$

여기서 $\mathcal{C}_{cls}$ 는 예측과 정답 클래스 간의 초점 손실(Focal loss)[25]이며, $\mathcal{C}_{L1}, \mathcal{C}_{giou}$ 는 각각 L1 손실과 GIoU(Generalized Interaction over Union)[26] 손실을 의미한다. 이때 L1 손실은 예측된 경계상자와 정답 경계상자 간의 좌표차이를 절댓값으로 계산한 것이다. $\lambda_{cls}, \lambda_{L1}, \lambda_{giou}$은 각 비용 성분의 가중치로, 각각 2, 5, 2로 설정하였다. 이렇게 계산된 매칭 비용을 바탕으로 최적 수송 기법(Optimal transport approach)[27]을 사용하여 각 정답과 예측을 매칭한다. 구체적으로 각 정답에 대해 매칭 비용이 가장 낮은 상위 m개 예측을 양성(Positive) 샘플로 선택하고, 나머지는 음성(Negative) 샘플로 간주한다. 이렇게 선택된 샘플을 이용하여 최종 손실 함수는 식 (13)과 같이 정의된다.

$$\mathcal{L} = \lambda_{cls}\mathcal{L}_{cls}+\lambda_{L1}\mathcal{L}_{L1} + \lambda_{giou}\mathcal{L}_{giou} + \lambda_{\min}\mathcal{L}_{\min} + \lambda_{center}\mathcal{L}_{center} \tag{13}$$

여기서 $\mathcal{L}_{cls}$는 예측과 정답 클래스 간의 초점 손실이며, $\mathcal{L}_{L1}$, $\mathcal{L}_{giou}$는 각각 L1 손실과 GIoU 손실을 의미한다. 이 세 항목의 가중치 $\lambda_{cls}, \lambda_{L1}, \lambda_{giou}$는 매칭 비용계산에서 사용된 값과 동일하게 설정하였다. 또한, $\mathcal{L}_{\min}$과 $\mathcal{L}_{center}$는 각각 객체의 최소 깊이와 중심 깊이 추정에 대한 손실을 의미하며, 두 항목 모두 스무스(Smooth) L1 손실을 적용하였다. 깊이 항목의 가중치인 $\lambda_{\min}$과 λ_{center}는 두 손실의 중요도를 조절하며, 본 연구에서는 두 값을 모두 1로 설정하였다.

4. 실험 결과

이 장에서는 제안된 네트워크의 성능을 정량적, 정성적으로 평가한다. 먼저 공개 데이터셋을 활용하여 기존 기법과의 비교를 통해 제안된 네트워크의 객체 검출 및 깊이 추정 성능을 검증한다. 이어서 후보 생성 방식에 따른 성능 변화를 실험하여 초기 객체 후보군 설정이 결과에 미치는 영향을 분석한다. 또한 이미지 특징과 BEV 특징의 융합 방식이 객체 탐지 정확도와 깊이 추정 정밀도에 어떤 영향을 미치는지 평가한다. 마지막으로 정량적 지표뿐 아니라 시각화 결과를 통해 제안된 네트워크가 가려진 객체나 원거리 객체에 대해 어떠한 강점과 한계를 보이는지도 논의한다. 이를 통해 제안 방법의 실질적 유효성과 개선 가능성을 종합적으로 고찰한다.

4.1 데이터셋

본 연구는 제안된 네트워크 학습과 평가를 위해 이미지, 포인트 클라우드, 객체의 2D 경계상자, 카메라에서 각 객체까지의 최소 깊이가 포함된 데이터셋이 필요하다. 그러나 이러한 정보를 모두 갖춘 공개 데이터셋 확보가 쉽지 않아 Song and Lee[10]가 제안한 방법을 활용하여 객체의 2D 경계상자와 깊이에 대한 GT 데이터를 확보하였다. 이 GT 확보에 사용한 KITTI 3D 객체 검출 데이터셋[12]은 스테레오 이미지, 3D 라이다 스캐너로 얻은 포인트 클라우드, 객체의 3D 경계상자, 사용된 센서들 사이의 캘리브레이션 정보를 갖고 있다. KITTI 데이터셋의 3D 경계상자는 라이다의 포인트 클라우드와 영상의 정합을 통해 레이블링된 정보를 기반으로 객체와 객체의 공간적 위치를 합리적으로 포착한 것으로 볼 수 있다. 본 연구에서는 Song and Lee가 했던 대로 3D 경계상자를 영상에 투영하여 2D 경계상자를 얻고, 3D 경계상자의 8개 꼭지점 중 Z좌표값이 가장 작은 값을 객체까지의 최소 깊이에 대한 GT로 정하였다. Table 1은 본 연구에서 학습과 평가에 사용한 데이터셋의 구성을 제시한 것이며, 검출 대상이 자동차(Car), 보행자(Pedestrian), 사이클리스트이다.

Table 1 Dataset composition for training and evaluation

	# of Images	# of Car	# of Pedestrian	# of Cyclist
Training	3,712	14,357	2,207	734
Evaluation	3,769	14,385	2,280	893

4.2 학습 전략

제안된 네트워크의 초기 가중치는 모듈별로 구분하여 설정하였다. 이미지 특징 검출 모듈은 DiffusionDet[9]의 COCO 데이터집합에 의한 학습 가중치를 초기값으로 사용하였다. 반면, 포인트 클라우드로부터의 히트맵 예측 모듈은 TransFusion[2]에서 제안된 방법을 따라 KITTI 집합에 있는 라이다 포인트 클라우드를 이용해 사전 학습하였다. 이 과정에서 BEV 백본과 히트맵 생성 모듈을 함께 학습하여, BEV 공간에서 객체의 중심을 효과적으로 표현하도록 하였다. 사전 학습이 완료된 후, 학습된 두 모듈의 가중치는 전체 네트워크 학습 시 초기값으로 사용된다. 이때 포인트 클라우드에 의한 히트맵 생성 모듈은 전체 네트워크 학습 과정에서 동결(Freeze)하여 가중치가 갱신되지 않도록 하였으며, BEV 백본만 전체 네트워크 학습 동안 지속적으로 갱신되도록 하였다. 이는 포인트 클라우드에 의한 히트맵 생성 모듈이 전체 네트워크 학습 시 해당 모듈의 가중치가 분류 손실, 경계상자 손실, 깊이 회귀 손실 등에 의해 변동되면 오히려 BEV 히트맵의 일관성이 무너질 위험이 있기 때문이다. 반면 BEV 백본은 전체 학습 동안 미세 조정(Fine-tuning)이 되도록 가중치 갱신을 허용하였다.

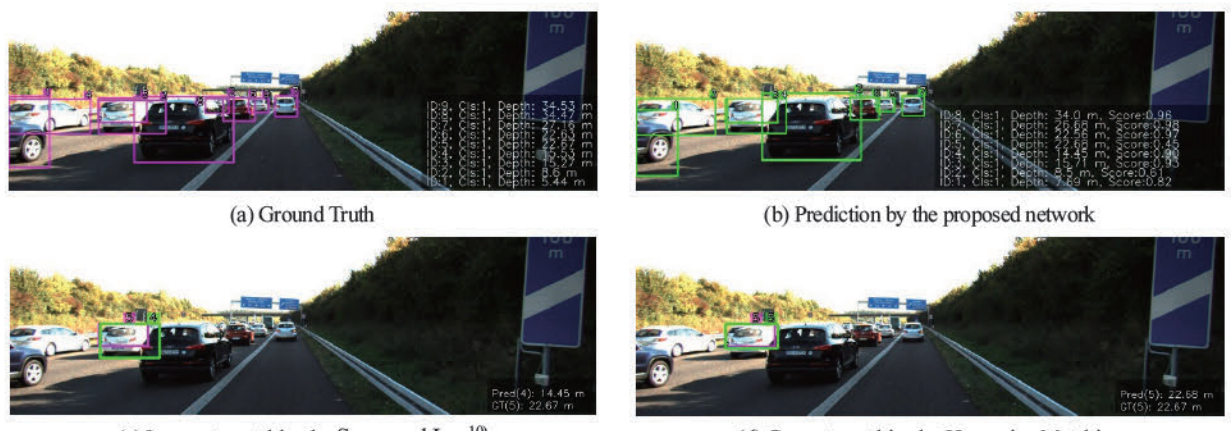
제안된 네트워크는 Pytorch 라이브러리를 통해 구현되었고, 모든 학습은 4개의 RTX 2080Ti GPU에서 수행되었다. 최적화 기법으로는 AdamW[28]를 사용하였고, 하이퍼 파라미터 β_1=0.9, β_2=0.999, 가중치 감쇠(Weight decay)= 0.0001을 사용하였다. 데이터 증강 기법으로는 랜덤 수평 반전을 적용하였다. 이때 반전 변환은 영상뿐만 아니라 라이다 포인트 클라우드에도 동일하게 적용했으며, 이를 통해 두 센서 간의 투영 대응 관계를 유지하였다. 네트워크 학습은 배치 크기 16으로 총 14,000회 반복하였고, 매 1,000회 반복마다 가중치를 저장하였다. 이후 저장된 모든 가중치 중에서 성능이 가장 우수한 것을 최종 결과로 선택하였다. 초기 학습률은 0.0001로 설정하였고, 전체 반복 횟수의 71.5%와 85.7% 지점에서 학습률을 0.1배로 감소시키는 전략을 적용하였다. 해당 학습률 조정 시점은 사전 실험을 통해 결정한 것이다.

4.3 성능 평가 방법

본 연구에서 객체 검출 성능은 COCO mAP로 평가하고, 깊이 추정 성능은 예측 결과와 정답을 1:1로 정확하게 매칭하기 위해 헝가리안 매칭[29]을 적용한 후, 매칭된 쌍에 대해서 RMSE(Root Mean Square Error)를 계산하여 평가하였다.

Fig. 3은 예측 결과와 정답 간의 매칭 방법이 성능에 미치는 영향을 보인 사례로서, Song and Lee[10]의 매칭 방법과 본 연구에서 적용한 헝가리안 매칭 결과를 비교한 것이다. Fig. 3(a)는 실제 정답 경계상자와 해당 객체의 정답 깊이를 나타낸다. Fig. 3(b)는 제안한 네트워크가 예측한 경계상자를 표현한 것으로, 그림의 우측에 각 상자별로 예측된 깊이와 신뢰 점수가 표시되어 있다.

Song and Lee의 방식은 정답과 IoU가 0.5 이상인 예측들 중 가장 신뢰도가 높은 예측을 선택해 정답과 매칭한다. Fig. 3(c)는 이 방식을 토대로 수행된 매칭 사례를 제시한 것이다. 이 사례에서는 정답 5번과 예측 4, 5번의 IoU가 0.5 이상이었고, 이 예측들 중 4번의 신뢰도가 더 높아 정답 5번에 매칭된 예측은 5번이 아니라 4번이었다. 그러나 이 매칭은 잘못된 것이다. Fig. 3(b)에 보인 예측 결과



(a) Ground Truth

(b) Prediction by the proposed network

(c) Incorrect matching by Song and Lee[10]

(d) Correct matching by Hungarian Matching

Fig. 3 Comparison of matching strategies for depth evaluation

를 보면 예측 4번, 5번 모두 깊이 검출은 매우 정확했지만, 잘못된 매칭에 의해 깊이 검출 오류는 커진다. 반면, Fig. 3(d)는 헝가리안 매칭을 적용하여 동일한 장면에서 정답과 예측이 올바르게 매칭된 결과를 보여준다. 이와 같이 헝가리안 매칭은 복잡한 장면에서도 중복 매칭을 방지하고, 정확한 깊이 성능 평가를 가능하게 한다.

본 연구는 2D 경계상자 기반 객체 검출과 깊이 추정을 동시에 수행하는 네트워크의 성능을 평가하기 위해, nuScenes[30]에서 제안된 NDS (nuScenes Detection Score)의 개념을 고려한 2D 깊이용 통합 성능 지표를 정의하였다. 기존 NDS는 3차원 객체의 위치, 크기, 방향, 속도 등을 포함한 다양한 요소를 종합적으로 평가하지만, 2D 경계상자와 최소 깊이를 예측하는 본 연구에 직접적으로 적용하기에는 한계가 있다. 따라서 본 연구에서는 2D 객체 검출과 깊이 추정 성능을 동시에 반영할 수 있도록 단순화한 통합 평가 점수 NDS_{2D}를 식 (14)와 같이 정의한다.

$$NDS_{2D} = \frac{\sum_{i=1}^{N_c} N_i \cdot \left[\lambda \cdot mAP^{(i)} + (1-\lambda)\cdot\left(1 - min\left(1, \frac{RMSE^{(i)}}{\alpha}\right)\right)\right]}{\sum_{i=1}^{N_c} N_i} \quad (14)$$

여기서 $mAP^{(i)}$는 i번째 클래스에 대한 2D 경계상자 검출 성능이고, $RMSE^{(i)}$는 해당 클래스에 대한 객체 최소 깊이의 RMSE이다. N_i는 i번째 클래스의 평가 샘플 수이며, 이를 통해 클래스별 데이터 분포의 불균형을 반영한 가중 평균을 수행한다. λ는 객체 검출 성능과 깊이 추정 성능 간의 상대적 중요도를 조절하는 가중치 계수이고, α는 깊이 오차를 정규화하기 위한 상수이다. 우리는 λ는 0.5, α는 5로 설정하였다.

4.4 성능 평가

제안된 방법을 기존의 객체 검출 및 깊이 추정 기법들과 비교하는 실험을 수행하였다. 기존 방법은 Song and Lee,[10] Dist-YOLO,[11] AVOD,[15] EPNet[17]이며, 평가 대상 클래스는 차량, 보행자, 사이클리스트 세 가지다.

4.4.1 객체 검출 및 깊이 추정 정량적 결과

Table 2는 제안된 방법과 기존 방법들의 객체 검출 성능을 mAP(%) 기준으로 비교한 결과를 제시한 것이다. Table 2를 보면 제안된 방법은 모든 클래스에 대해 기존 방법들보다 mAP 수치가 높았다. 기존 방법들은 차량과 같이 크기가 큰 객체에서는 비교적 높은 성능을 보였으나, 상대적으로 크기가 작고 학습 데이터가 적은 보행자와 사이클리스트의 검출 성능은 현저히 낮았다. 반면, 제안된 방법은 보행자와 사이클리스트에 대해서도 기존 방법 대비 높은 성능을 보였다. 이는 BEV 특징과 이미지 특징을 효과적으로 융합하여 작은 객체나 희소하게 표현된 객체까지 안정적으로 검출한 결과로 해석할 수 있다.

Table 2 Comparison of object detection performance with other methods

Methods	Car	Ped.	Cyc.
Song and Lee[10]	31.34	14.33	8.41
Dist-YOLO[11]	29.12	13.47	4.32
AVOD[15]	54.02	21.02	20.22
EPNet[17]	59.12	27.11	21.13
Ours	**62.45**	**28.28**	**25.58**

Table 3 Comparison of depth estimation results with other methods

Methods	Car	Pedestrian	Cyclist
Song and Lee[10]	1.38	1.29	1.14
Dist-YOLO[11]	2.09	1.18	1.71
AVOD[15]	1.10	0.75	0.78
EPNet[17]	1.08	0.74	0.72
Ours	**1.04**	**0.68**	**0.70**

Table 3은 제안된 방법과 기존 기법들의 객체별 깊이 추정 성능을 RMSE(m) 기준으로 비교한 결과를 제시한 것이다. Table 3을 보면 제안된 방법은 모든 클래스에 대해 기존 방법보다 RMSE 수치가 낮았다. 즉 제안된 방법이 객체 종류에 관계없이 정밀한 깊이 추정 성능을 보였다.

Table 4 Comparison of unified detection and depth estimation metrics with other methods

Methods	NDS_{2D}	Params (M)	Runtime (ms)
Song and Lee[10]	0.455	12.33	29
Dist-YOLO[11]	0.436	5.13	15
AVOD[15]	0.552	66.73	100
EPNet[17]	0.581	16.23	39
Ours	**0.596**	96.77	129

Table 4는 제안한 방법과 기존 방법들의 종합적인 성능을 NDS_{2D}, 파라미터 수, 그리고 추론 시간을 기준으로 비교한 결과를 제시한 것이다. 제안한 방법은 NDS_{2D} 값이 0.596으로 모든 비교 대상 중 가장 높은 성능을 보였으며, 특히 AVOD와 EPNet같은 기존 센서 융합 기반 기법들보다 정확도가 더 우수하였다. 반면에 우리가 제안한 방법은 파라미터 수가 상대적으로 많고 추론 시간도 길었다.

그러나 정확도가 높기 때문에 앞으로 추론 속도 개선이 이루어지면 실제 자율주행 환경에서도 적용 가능성이 높을 것으로 판단된다.

4.4.2 객체 검출 및 깊이 추정 정성적 결과

Fig. 4는 제안된 네트워크와 비교 대상 방법들이 검출한 2D 경계상자와 객체까지의 최소 깊이를 이미지에 표현하여 비교한 결과이다. Fig. 4(a)는 정답을 나타내며, Fig. 4(b)~(e)는 각각 Song and Lee,[10] Dist-YOLO,[11] AVOD,[15] EPNet[17]의 결과를 Fig. 4(f)는 제안된 네트워크의 결과를 나타낸 것이다. 실험 결과, Fig. 4(b)와 Fig. 4(c) 모두 바로 앞 우측 차량 검출에 실패했으며, Fig. 4(b)와 Fig. 4(e)는 가려진 차량(Occluded vehicle)을, Fig. 4(c)와 Fig. 4(d)는 보행자를 탐지하지 못하는 한계를 보였다.

반면, 제안된 네트워크는 모든 객체를 정확하게 검출하였으며, 깊이 추정 결과에서도 제안된 네트워크가 보다 정확하게 깊이 추정을 수행함을 확인하였다.

4.5 객체 후보군 설정 분석

본 절에서는 제안된 네트워크에서 객체 후보 수와 후보 생성 방식이 객체 검출 및 깊이 추정 성능에 미치는 영향을 분석한다. 후보군은 네트워크가 탐지를 시작하는 초기 위치 집합으로, 수량과 배치 방식에 따라 성능과 연산 효율성이 달라질 수 있다.

후보가 너무 적으면 실제 객체가 포함되지 않아 탐지 누락이 증가하고, 반대로 후보가 지나치게 많으면 불필요한 후보가 늘어나 연산량이 증가한다. 이를 확인하기 위해 후보의 개수를 100, 200, 300으로 설정하여 객체 검출 성능과 깊이 추정의 정확도를 비교하였다. Table 5는 제안된 네트워크에서 후보 수량을 변화시켰을 때 객체 검출 성능과 깊이 예측 정확도의 변화를 비교한 결과이다.

Table 5 Comparison of object detection and depth estimation performance by candidate number

Number	mAP			RMSE		
	Car	Ped.	Cyc.	Car	Ped.	Cyc.
100	59.85	26.28	23.67	1.11	0.83	0.87
200	62.45	28.28	24.58	1.04	0.68	0.70
300	62.47	28.37	24.78	1.02	0.68	0.69

Table 5를 보면 후보 개수가 100개일 때보다 200개로 늘렸을 때 세 클래스 모두 mAP와 RMSE가 개선됨을 알 수 있다. 특히 보행자의 경우 mAP가 26.28에서 28.28로 향상되었고, RMSE도 0.83 m에서 0.68 m로 크게 개선되었다. 후보 개수를 300개로 늘린 경우에도 성능은 향상되었지만, 개선 폭은 크지 않았다. 이는 후보 개수가 일정 수준 이상 증가하면 추가적인 성능 향상은 미미함을 의미하며, 처리 시간을 고려할 때 200개의 후보가 적절한 선택임을 시사한다.

Table 6은 후보군 생성 방식에 따른 객체 검출 성능과

(a) Ground Truth (b) Results by Song and Lee (c) Results by Dist-YOLO (d) Results by AVOD (e) Results by EPNet (f) Our results

Fig. 4 Qualitative comparison with other methods

깊이 예측 정확도의 차이를 비교한 결과이다. 비교 대상은 두 가지로, 무작위 초기화(Random Initialization, RI) 후보 생성 방식과 입력 의존 쿼리 초기화(Input-Dependent Query Initialization, IDQI) 방식이다. 무작위 후보 생성은 BEV 맵 상의 위치를 임의로 선택하여 후보를 배치하는 방식이며, 두 방식 모두 후보의 개수는 200개로 동일하다.

Table 6 Object detection and depth estimation performance according to the initial candidate generation methods

Approach	mAP			RMSE		
	Car	Ped.	Cyc.	Car	Ped.	Cyc.
RI	61.32	27.22	24.03	1.07	0.72	**0.69**
IDQI	**62.45**	**28.28**	**24.58**	**1.04**	**0.68**	0.70

Table 6을 보면, 랜덤 방식은 차량 mAP 61.32, 보행자 mAP 27.22, 사이클리스트 mAP 24.03이었으며, RMSE는 각각 1.07 m, 0.72 m, 0.68 m였다. 반면, 입력 의존 쿼리 초기화 방식은 차량 mAP 62.45, 보행자 mAP 28.28, 사이클리스트 mAP 24.58로 모든 클래스에 대해 검출 성능이 향상되었다. 깊이 예측 정확도는 두 방법이 엇비슷했다. 이는 입력 의존 쿼리 초기화 방식이 무작위 방식 대비 효과적임을 입증한다.

4.6 이미지 특징과 BEV 특징의 융합 방법 분석

Table 7은 제안된 네트워크에서 이미지와 BEV 특징 융합 방법에 따른 객체 검출 및 깊이 예측 성능을 비교한 결과이다. 비교 대상으로는 단순 연결(Concatenation, Concat), 요소별 합(Elementwise-Sum, ES), 교차 어텐션(Cross-Attention, CA) 등 세 가지 방식이다. 단순 연결은 이미지 특징벡터와 BEV 특징벡터를 채널 차원으로 이어 붙여 하나의 특징 벡터로 만드는 방식이며, 요소별 합은 두 특징 벡터의 동일한 위치와 채널에 대응하는 값을 더하는 방식이다.

Table 7 Comparison of object detection and depth estimation performance according to feature fusion methods

Approach	mAP			RMSE		
	Car	Ped.	Cyc.	Car	Ped.	Cyc.
Concat	61.33	25.22	23.77	1.09	0.72	0.78
ES	60.59	25.22	23.03	1.23	0.77	0.86
CA	**62.45**	**28.28**	**24.58**	**1.04**	**0.68**	**0.70**

Table 7을 보면 교차 어텐션 방식이 세 클래스 모두 가장 높은 객체 검출 성능과 낮은 깊이 예측 오차를 기록하였다. 특히 보행자의 경우 교차 어텐션 방식이 다른 방법에 비해 성능이 높음을 확인할 수 있다. 반면, 단순 연결이나 요소별 합 방식은 검출 성능이 낮고, 깊이 추정 오차가 큰 경향을 보였다. 이는 단순한 특징 결합 방식보다 서로 다른 특징 간 상호작용을 학습하는 교차 어텐션 방식이 효과적임을 시사한다.

5. 결 론

본 연구는 이미지 특징과 포인트 클라우드 특징의 융합을 토대로 객체 검출과 깊이 예측을 동시에 수행하는 네트워크를 제안하였다. 제안된 방법은 융합 특징을 입력으로 한 초기 후보군 생성 모듈을 도입하여 초기부터 객체가 존재할 가능성이 큰 위치를 선택함으로써 부정확한 후보들을 줄이고 검출 효율을 향상시켰다. 또한, 헝가리안 매칭 기반의 깊이 평가 방법을 적용하여 정확한 정답-예측 매칭을 보장하고, RMSE를 이용해 깊이 예측 성능을 정량적으로 평가하였다.

KITTI 데이터 집합에 의한 실험 결과, 제안된 방법은 기존 방법 대비 차량, 보행자, 사이클리스트 모두에서 높은 객체 검출 성능과 정밀한 깊이 추정 성능을 보였다. 특히, 융합 특징 기반 초기 후보군 생성 방식은 랜덤 후보군에 의한 방식보다 객체 검출 및 깊이 예측 성능에서 우수한 결과를 보였다.

향후 연구에서는 제안한 방법을 다양한 센서 융합 환경으로 확장하고, 경량화 기법을 도입하여 연산 효율성을 개선하며, 실시간 자율주행 시나리오에 적용 가능성을 모색할 계획이다.

후 기

이 연구는 2023년도 산업통상자원부 및 산업기술평가관리원(KEIT) 연구비 지원에 의한 연구임(“20025038”).

References

1) J. Y. Seo and M. B. Park, “A Study on Deep Learning Fusion Techniques for Enhanced Object Detection Performance Using Camera and Radar Data,” Transactions of KSAE, Vol.32, No.7, pp.583-589, 2024.
2) X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu and C. L. Tai, “TransFusion: Robust Lidar-Camera Fusion for 3D Object Detection with Transformers,” IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.1090-1099, 2022.
3) Y. Li, H. Hu, Z. Liu, X. Xu, X. Huang and D. Zhao, “Influence of Camera-Lidar Configuration on 3D Object Detection for Autonomous Driving,” IEEE International Conference on Robotics and Automation, pp.9018-9025,

2024.

4) S. M. Kim, Y. S. Kim, H. S. Jeon, D. S. Kum and K. B. Lee, "Autonomous Driving Technology Trend and Future Outlook: Powered by Artificial Intelligence," Transactions of KSAE, Vol.30, No.10, pp.819-830, 2022.
5) P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, H. Tomizuka and J. Shi, "Sparse R-CNN: End-to-End Object Detection with Learnable Proposals," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.14454-14463, 2021.
6) A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez and I. Polosukhin, "Attention Is All You Need," Advances in Neural Information Processing Systems, Vol.30, pp.1-11, 2017.
7) J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
8) S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.39, No.6, pp.1137-1149, 2016.
9) S. Chen, P. Sun, Y. Song and P. Luo, "DiffusionDet: Diffusion Model for Object Detection," IEEE/CVF International Conference on Computer Vision, pp.19830-19843, 2023.
10) J. G. Song and J. W. Lee, "CNN-Based Object Detection and Distance Prediction for Autonomous Driving Using Stereo Images," Int. J. Automotive Technology, Vol.24, No.3, pp.773-786, 2023.
11) M. Vajgl, P. Hurtik and T. Nejezchleba, "Dist-YOLO: Fast Object Detection with Distance Estimation," Applied Sciences, Vol.12, No.3, Paper No.1354, 2022.
12) A. Geiger, P. Lenz and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.3354-3361, 2012.
13) C. R. Qi, W. Liu, C. Wu, H. Su and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.918-927, 2018.
14) C. R. Qi, H. Su, K. Mo and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.652-660, 2017.
15) J. Ku, M. Mozifian, J. Lee, A. Harakeh and S. L. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1-8, 2018.
16) K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask R-CNN," IEEE/CVF International Conference on Computer Vision, pp.2961-2969, 2017.
17) T. Huang, Z. Liu, X. Chen and X. Bai, "EPNet: Enhancing Point Features with Image Semantics for 3D Object Detection," European Conference on Computer Vision, pp.35-52, 2020.
18) J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. Fei Fei, "ImageNet: A Large-Scale Hierarchical Image Database," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.248-255, 2009.
19) K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.770-778, 2016.
20) T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.2117-2125, 2017.
21) Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.4490-4499, 2018.
22) T. Yin, X. Zhou and P. Krahenbuhl, "Center-Based 3D Object Detection and Tracking," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.11784-11793, 2021.
23) J. G. Song, J. M. Park and J. W. Lee, "CNN Combined with a Prior Knowledge-Based Candidate Search and Diffusion Method for Nighttime Vehicle Detection," International Journal of Control, Automation and Systems, Vol.22, No.3, pp.963-975, 2024.
24) N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov and S. Zagoruyko, "End-to-End Object Detection with Transformers," European Conference on Computer Vision, pp.213-229, 2020.
25) T. Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, "Focal Loss for Dense Object Detection," IEEE/CVF International Conference on Computer Vision, pp.2980-2988, 2017.
26) H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid and S. Savarese, "Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.658-666, 2019.
27) Z. Ge, S. Liu, Z. Li, O. Yoshie and J. Sun, "OTA: Optimal Transport Assignment for Object Detection," IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.303-312, 2021.

28) I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” arXiv preprint arXiv:1711.05101, 2017.
29) H. W. Kuhn, “The Hungarian Method for the Assignment Problem,” Naval Research Logistics Quarterly, Vol.2, No.1-2, pp.83-97, 1955.
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,” IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.11621-11631, 2020.