모바일 로봇을 위한 엣지 컴퓨팅에서의 실시간 2D/3D 객체인식

김재영 [1] 문형필 2
[1,2] 성균관대학교 기계공학부

bdori2005@g.skku.edu

Real time 2D/3D Object Detection on Edge Computing for Mobile Robot

Jae-Young Kim[1], Hyungpil Moon[2]
[1,2]Mechanical Engineering, Sungkyunkwan University

요 약

모바일 로봇의 자율주행을 위하여 인터넷이 제약된 환경에서도 가능한 Edge computing 에서의 Object Detection 이 필수적이다. 본 논문에서는 이를 위해 Orin 보드에서 YOLOv7 과 Complex_YOLOv4 를 구현하였다. 직접 취득한 데이터를 통해 YOLOv7 을 구현한 결과 0.56 의 mAP 로 프레임당 133ms 가 소요되었다. Kitti Dataset 을 통해 Complex_YOLOv4 를 구현한 결과 0.88 의 mAP 로 프레임당 236ms 가 소요되었다. Comple_YOLOv4 가 YOLOv7 보다 더 많은 데이터를 예측하기에 시간은 더 소요되지만 높은 정확성을 가지는 것을 확인할 수 있었다.

1. 서론

안전한 모바일 로봇 구동을 위해서는 실시간으로 주위 장애물을 탐지할 수 있는 실시간 객체인식이 필요하다. 또한 로봇에 탑재하여 처리하도록 설계된 Edge Computing 을 통해 네트워크의 불안정성에도 덜 민감하게 된다. 이는 로봇이 네트워크와의 연결이 보장되지 않는 도전적인 환경에서도 더욱 안정적으로 작동할 수 있게 한다. 이를 위하여 Edge Computing 에서 Image Data 와 LiDAR Data 를 통한 실시간 2D/3D Object Detection 을 구현하였다.

2. 본론

2.1 2D Object Detection

2D Object Detection을 효과적으로 수행하기 위해, YOLO(You Look Only Once)의 최신 버전인 YOLOv7을 Orin보드에서 구현하였다. YOLO는 2D이미지를 입력으로 받아, 해당 이미지 내의 객체들을 Bounding Box(BBOX)로 표시하며, 동시에 각 객체의 Class정보도 제공하는 객체 인식 알고리즘이다[1]. YOLO의 이름은 이미지 전체를 단 한 번만 본다는 특징에서 유래되었다. 이로 인해 처리속도가 빠르고 False Positives가 낮다는 장점이 있다. 그중 YOLOv7의 특징으로는 E-ELAN과 bag-of-freebies를 사용하여 학습능력과 네트워크 성능을 향상시켰다는 점이 있다. YOLOv7을 구현하고자 경기도 성남시 금토천에서 Scout Mini로봇에 ZED 2 카메라를 부착하여 취득한 데이터에서 COCO Dataset에 학습된 객체들을 인식하였다. Stereo Camera를 사용하는 것의 이점을 살리고자 BBox의 중심점의 Depth도 출력하도록 하였고, Ros Packaging을 통해 모바일로봇에 포팅하였다. mean Average Precision(mAP)와 소요 시간을 측정하여 분석하였다. 결과는 Figure1과 같고, 프레임당 평균 소요시간은 133ms, mAP는 0.56의 YOLOv7을 구현할 수 있었다.

Figure 1. Result of YOLOv7 at Mobile Robot

2.2 3D Object Detection

3D Object Detection 을 위하여 Orin 보드에 Complex-YOLOv4 를 구현하였다. Complex-YOLO 는 LiDAR 센서를 기준으로 전방 40m, 좌우로 40m 씩, 위로 1.25m 밑으로 2m 를 사용하고 이를 $\mathcal{P}_\Omega$(1)라고 한다.

$$\mathcal{P}_\Omega = \{\mathcal{P} = [x,y,z]^T \mid x \in [0,40m], y \in [-40m,40m], z \in [-2m,1.25m]\} \quad (1)$$

그러므로 수집된 Point Cloud 중에서 Inference 에 사용할 부분을 Crop 한다. Crop 한 Point Cloud 를 Bird Eye View(BEV)의 RGB-Map 으로 변환한다. 해당 RGB-Map 을 (2), (3)를 통해 8cm 해상도의 1024*512*3 의 Grid Map 으로 변환한 후, CNN 과 E-RPN 을 통해 Inference 를 진행한다.

$$\mathcal{P}_{\Omega i \to j} = \{\mathcal{P}_{\Omega i} = [x,y,z]^T \mid S_j = f_{PS}(\mathcal{P}_{\Omega i}, g)\} \quad (2)$$

$$z_g(S_j) = \max\left(P_{\Omega i \to j} \cdot [0,0,1]^T\right)$$
$$z_b(S_j) = \max\left(I(P_{\Omega i \to j})\right) \quad (3)$$
$$z_r(S_j) = \min\left(\frac{1.0, \log(N+1)}{64}\right) \qquad N = |P_{\Omega i \to j}|$$

이를 통해 해당 RGB-Map 내의 객체들을 BBOX 로 표시하며, 동시에 각 객체의 Class 정보와 'Heading' 을 예측한다[2]. BBox 의 Z 축을 제외한 X, Y 축에 대한 정보만을 Figure 2 와 같이 예측한다. Complex-YOLO 는 'Car', 'Person', 'Cyclist' 세가지 Class 로 구성된 weight 파일을 제공하는데, 각 Class 별로 사전에 height 정한다. Z 축에 대한 예측을 진행하지 않고, 기존에 정한 height 를 사용하므로 물체의 Z 축에 대한 정보는 부정확하다는 한계가 있다.

Kitti Bagfile 로 진행한 결과는 Figure3 와 같고, 프레임당 평균 236ms, mAP 는 0.88 임을 확인할 수 있었다. 객체의 Heading 과, Z 축 정보도 다루기 때문에, YOLOv7 에 비해 시간은 100ms 정도 추가로 요구되지만, 높은 mAP 를 가지는 것을 확인할 수 있다.

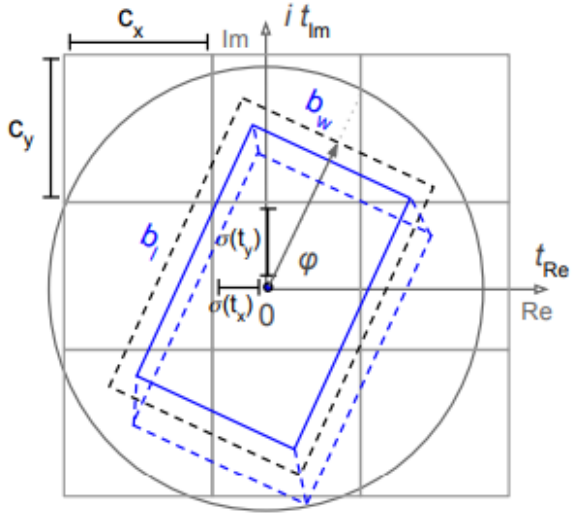

Figure 2. 3D Bounding box Regression

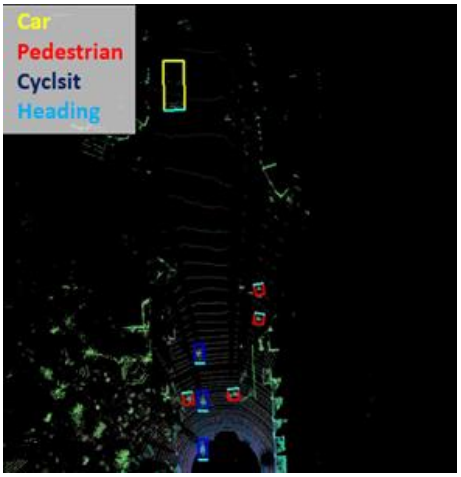

Figure 3. Result of Complex_YOLOv4 with Kitti Dataset

3. 결론

본 연구에서는 제약된 환경에서도 Mobile Robot 의 자율주행을 위한 Edge computing 에서의 Real time Object Detection 을 진행하였다. YOLOv7 을 통한 2D Object Detection 을, Complex_YOLOv4 를 통한 3D Object Detection 을 구현하였다. YOLOv7 은 약 7.5FPS, 0.56 mAP 를 가지며, Complex_YOLOv4 는 4FPS, 0.88 mAP 를 가진다. 추후 모델 경량화를 진행하여 더욱 신속하고 정확한 Object Detection 을 구현할 예정이다.

ACKNOWLEDGMENT

이 논문은 정부(교육부-산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임 (P0022098, 2023 년 미래형자동차 기술융합 혁신인재양성사업)

참고문헌

[1] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao; *YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7464-7475, 2023

[2] *Martin Simony, Stefan Milzy, Karl Amendey, Horst-Michael Gross; Complex-YOLO: An Euler-Region-Proposal for Real-time 3D Object Detection on Point Clouds*, Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pp. 0-0, 2018