

배달 로봇 응용을 위한 LiDAR 센서 기반 객체 분류 시스템

LiDAR Sensor based Object Classification System for Delivery Robot Applications

박 우 진*, 이 정 규*, 박 채 운*, 정 윤 호**

Woo-Jin Park*, Jeong-Gyu Lee*, Chae-woon Park*, Yunho Jung**

Abstract

In this paper, we propose a lightweight object classification system using a LiDAR sensor for delivery service robots. The 3D point cloud data is encoded into a 2D pseudo image using a Pillar Feature Network (PFN), and then passed through a lightweight classification network designed based on Depthwise Separable Convolutional Neural Networks (DS-CNN). The implementation results show that the designed classification network has 9.08K parameters and 3.49M Multiply-Accumulate (MAC) operations, while supporting a classification accuracy of 94.94%.

요 약

본 논문에서는 배달 서비스 로봇 응용을 위한 LiDAR 센서 기반 경량화된 객체 분류 시스템을 제안한다. 3차원 포인트 클라우드 데이터를 Pillar Feature Network (PFN)을 사용하여 2차원 pseudo image로 인코딩한 후, Depthwise Separable Convolution Neural Network (DS-CNN)에 기반하여 설계된 네트워크를 통해 객체 분류를 수행하는 경량화된 시스템을 설계하였다. 구현 결과, 설계한 분류 네트워크의 파라미터 수와 Multiply-Accumulate (MAC) 연산 수는 각각 9.08K 및 3.49M이며, 94.94%의 분류 정확도를 지원 가능함을 확인하였다.

Key words : LiDAR, Point Cloud Data, PointPillars, DS-CNN, Delivery Robot

1. 서론

현대 사회에서는 소비자의 편의성과 효율성을 높이기 위해 배달 서비스의 수요가 증가하고 있다. 특히 최근 몇 년간, 자율 주행 기술의 발전과 함께 배달 서비스 로봇의 응용이 주목받고 있다. 현재 상용화된 배달 서비스 로봇들은 주로 카메라, RADAR(Radio Detection And Ranging), LiDAR(Light Detection And Ranging) 센

서 등 다양한 센서를 활용하여 주변 환경을 인식하며 자율 주행을 수행한다. 이 중 LiDAR 센서는 직진성이 강한 레이저의 반사를 이용하여 물체의 포인트 클라우드 데이터를 생성하므로, 수동 광학 및 능동 RADAR/마이크로파 기기와 비교할 때, 자연광에 의존하지 않고 훨씬 더 높은 공간 해상도로 정밀한 데이터를 생성하며 주변 환경을 인식할 수 있는 장점이 있다[1].

포인트 클라우드 데이터는 X, Y, Z 좌표와 reflectivity

* School of Electronics and Information Engineering, Korea Aerospace University

★ Corresponding author

E-mail : yjung@kau.ac.kr, Tel: +82-2-300-0133

※ Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by Korean government (MSIT) (No. 2022-0-00960), and the CAD tools were supported by IDEC. Manuscript received Sep. 3, 2024; revised Sep. 14, 2024; accepted Sep. 25, 2024.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

와 같은 추가적인 속성도 포함하는 3D 좌표 데이터이다. 포인트 클라우드 데이터를 voxelization을 통해 feature를 추출 후, 3D convolution 연산을 수행하기 위해서는 많은 연산량과 파라미터가 요구된다. 이는 전력 소모와 수행 시간을 증가시키므로 실시간으로 classification을 수행해야 하는 배달 서비스 로봇 응용에는 비효율적이다.

이에, 본 논문에서는 PFN(Pillar Feature Network)과 DS-CNN(Depthwise Separable Convolutional Neural Network) 기반으로 최적화된 분류 네트워크 구조를 사용하여, 배달 서비스 로봇을 위한 경량화된 분류 시스템을 제안한다.

PFN은 포인트 클라우드 데이터를 효율적으로 처리하기 위해 개발된 네트워크 구조로, 포인트 클라우드 데이터를 pseudo image로 인코딩하여 3D 연산이 아닌 2D 연산으로 처리하므로 연산량과 메모리 사용량이 적어[2] 경량화된 분류 시스템에 적합하다. 또한, DS-CNN 기반의 경량화된 딥러닝 아키텍처는 제한된 컴퓨팅 자원에서도 높은 성능을 발휘할 수 있어 경량화된 분류 시스템에 유용하다[3].

분류 시스템은 위의 딥러닝 네트워크를 기반으로 LiDAR 센서를 저가형 SBC(Single-Board Computer)인 Raspberry Pi 4와 연결하여 구현하며, 포인트 클라우드 데이터를 전처리 후, 분류한 결과를 실시간으로 제공한다.

본 논문의 구성은 다음과 같다. II장 본론의 1절에서 제안한 임베디드 시스템의 구조를 설명하고 2절에서는 8개의 class의 객체와 분류기의 학습에 사용한 dataset에 대해 정의한다. 3절에서는 PointPillars를 활용한 인코딩 방식인 PFN에 대해 설명하고, 4절에서 pseudo image를 통해 class를 분류하는 DS-CNN 기반의 분류

네트워크를 소개한다. 5절에서 임베디드 시스템의 알고리즘을 block diagram으로 나타낸다. III장 실험에서 제안한 분류 시스템을 통해 실제 환경에서 추론한 결과를 제시한다. IV장 결론에서 본 논문에서 제안한 시스템의 한계점과 향후 연구 방향으로 논문을 마무리 짓는다.

II. 본론

1. 시스템 구조

시스템의 구조는 그림 1과 같다. 자율 주행 배달 서비스 로봇이 실외에서 마주하는 객체에 대한 한 프레임의 포인트 클라우드 데이터를 LiDAR 센서가 이더넷 통신을 통해 Raspberry Pi 4로 전송한다. Raspberry Pi 4는 한 프레임의 포인트 클라우드 데이터를 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)을 통해 clustering을 수행한 후, PFN과 DS-CNN 기반 분류 네트워크를 거쳐 classification을 수행한다.

Table 1. OS1 LiDAR Specification.

표 1. OS1 LiDAR 상세 사양

Item	Specification
Range at 10%	90m
Max Range	200m
Vertical Field of View	45 °
Channels of Resolution	128
Max points per second	5.2M
Max Frame Rate	20Hz
Operating Temperature	-40 ~ 60 °C
Ingress protection	IP68 & IP69K

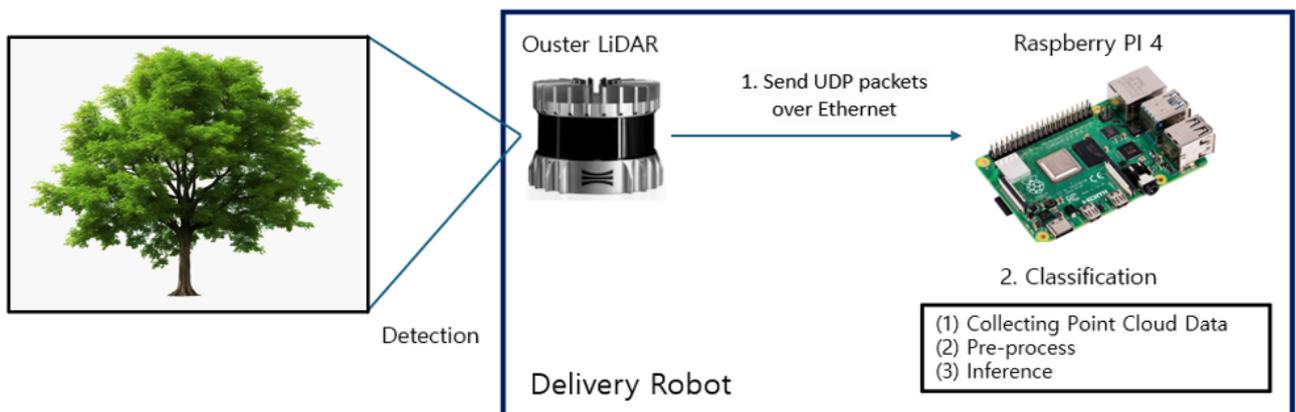


Fig. 1. Configuration of the proposed system.

그림 1. 제안된 시스템 구성

LiDAR 센서는 레이저 펄스를 쏘고 반사되어 돌아오는 시간을 측정하여 반사체의 위치 좌표를 측정하며, 위 시스템에서 사용한 LiDAR 센서는 Ouster 사의 OS1 모델이다[4]. 실험에서 사용된 LiDAR 센서의 사양은 표 1에서 확인할 수 있다.

2. Dataset 정의

시스템이 분류하고자 하는 객체를 총 8가지로 구분하여 표 2에 나타내었다. Dataset은 로봇이 야외에서 배달을 수행할 때 마주칠 수 있는 객체들로 구성하였다.

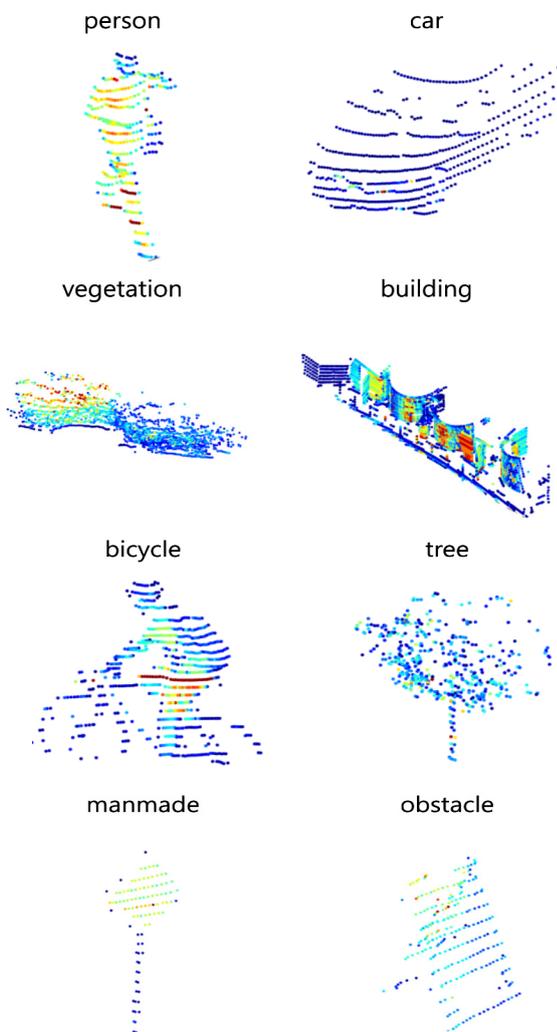


Fig. 2. Examples of collected point cloud data.
그림 2. 수집한 포인트 클라우드 데이터의 예시

학습과 추론에 사용할 데이터는 open dataset인 Nuscenes dataset에서 추출하거나, 직접 LiDAR 센서로 취득한 frame에서 필요한 데이터를 전처리하여 수집하였다. 그림 2는 수집한 8가지 class의 포인트 클라우드 데이터 예시이다. 전체 dataset은 각 class당 400개

Table 2. Class of dataset.

표 2. 데이터셋의 클래스

Class	Definition
Bicycle	Standing bicycle and bicycle on which a person rides
Building	Building
Car	Car that is driving or parked on the road
Manmade	Fixed outdoor facilities such as bus stops and signs
Obstacle	Movable objects, such as labacons
Person	Pedestrian
Tree	Tall tree
Vegetation	Vegetation and short tree

로 약 3,200개이며, 90%인 약 2,800개를 train data로, 10%인 약 400개를 test data로 사용하였다.

3. PointPillars를 활용한 인코딩(PFN)

LiDAR 센서를 통해 얻은 포인트 클라우드 데이터는 X, Y, Z의 3차원 위치 좌표 및 reflectivity와 같은 추가적인 속성을 가진 점들의 집합이므로 raw data를 인공 지능 분류기의 입력으로 사용하는 것은 일반적으로 3D convolution을 진행하는 network model이 요구된다. 3D convolution은 2D convolution 대비 높은 연산량과 메모리 사용량을 요구하므로 경량화된 시스템에 적합하지 않다.

따라서 본 논문에서는 포인트 클라우드 데이터의 feature를 추출하기 위해 PointPillars를 사용하는 방법을 채택하였으며, 포인트 클라우드 데이터를 PointPillars를 사용해 pseudo image로 인코딩하여 2D convolution을 수행하였다. 이는 연산량과 메모리 사용량을 크게 줄여 분류 속도를 향상시킬 수 있다.

4. 분류 알고리즘(DS-CNN)

DS-CNN이란 Depthwise Separable Convolutional Neural Network로, CNN의 변형된 구조이다. 이는 CNN에 비해 모델의 복잡도와 연산량을 크게 줄이면서 성능을 유지할 수 있는 장점이 있다[3].

DS-CNN은 depthwise convolution과 pointwise convolution을 순차적으로 거쳐 convolution 연산을 수행한다. Depthwise convolution은 입력 채널 각각에 대해 독립적인 convolution을 수행한다. 다음 단계인 pointwise convolution은 1×1 convolution을 수행한다. 기존 CNN 대비 DS-CNN의 파라미터와 연산량

Table 3. Performance evaluation results for various models.

표 3. 다양한 모델에 대한 성능 평가 결과

Model	Version	Pseudo image size	Conv1	Conv2	Conv3	Conv4	Fcl1	Fcl2	Fcl3	Fcl4	Accuracy (%)	Parameters (K)	MAC (M)
ResNet	Version1	(24×24×64)	64	128	256	512	512	8	-	-	86.98	1210	114
	Version2	(62×62×64)	64	128	256	512	512	8	-	-	88.33	1210	690
LeNet	Version1	(24×24×64)	32	64	128	-	2688	8	-	-	94.08	133	22
	Version2	(24×24×64)	32	64	-	-	1334	1334	8	-	94.44	1930	15
Customized DS-CNN	Version1	(24×24×64)	64	64	64	-	64	128	128	8	93.23	58	9
	Version2	(24×24×64)	128	256	-	-	256	128	128	8	93.4	112	26
	Version3	(24×24×64)	32	64	-	-	64	64	64	8	92.0	10	3
	Version4	(24×24×64)	32	64	-	-	32	32	8	-	94.94	9	3

은 모두 식 (1)과 같은 형태로 나타낼 수 있다[5]. Dk 는 convolution 필터의 크기, N 은 convolution 필터의 수를 의미한다. 이처럼 적은 파라미터 수와 연산량을 가지며 성능을 유지할 수 있는 딥러닝 모델은 경량화된 시스템에 적합하다.

$$\frac{DSCNN}{Conventional CNN} = \frac{1}{N} + \frac{1}{Dk^2} \quad (1)$$

가. 분류 네트워크 성능 평가

성능이 가장 좋은 분류 네트워크를 선정하기 위해 ResNet[6], LeNet[7], DS-CNN을 변형하여 함께 분류 성능을 평가하여 표 3에 정리하였다. 네트워크를 학습할 때, loss 함수로 NLL(Negative Log-Likelihood)를 사용하였고, 최적화를 위해 Adam(Adaptive Moment Estimation) optimizer[8]를 사용하였다.

하이퍼 파라미터인 epoch는 200, batch size는 24, learning rate는 0.001, pseudo image의 크기는 (24×24×64)로 설정하여 실험을 진행하였다. Pseudo image의 크기는 ResNet 기반의 model에서 pseudo image의 크기에 따른 실험을 진행하여 설정한 값이다. 표 3의 ResNet version1은 pseudo image의 크기를 (24×24×64)로 설정한 모델이며, ResNet version2는 (62×62×64)로 설정한 모델이다. 파라미터 수와 연산량 및 정확도의 관계를 고려하여 pseudo image의 크기를 (24×24×64)로 설정하였고, LeNet과 DS-CNN 모델의 레이어 구성을 변경하며 다양한 실험을 진행하였다. 그 결과 파라미터와 연산량 대비 정확도가 가장 높은 DS-CNN version4라는 경량화된 분류 네트워크를 설계할 수 있었다.

5. 분류 알고리즘

그림 3는 Raspberry Pi 4 개발환경에서 동작하는 분류 알고리즘의 순서도이다. Ouster LiDAR와 Raspberry Pi 4를 이더넷으로 연결한 임베디드 시스템은 분류 네트워크를 loading한 후, 프레임 단위로 그림 3 순서도의 collecting data, pre-process, inference 과정을 반복하여 LiDAR 센서가 감지한 물체에 대한 classification을 수행한다.

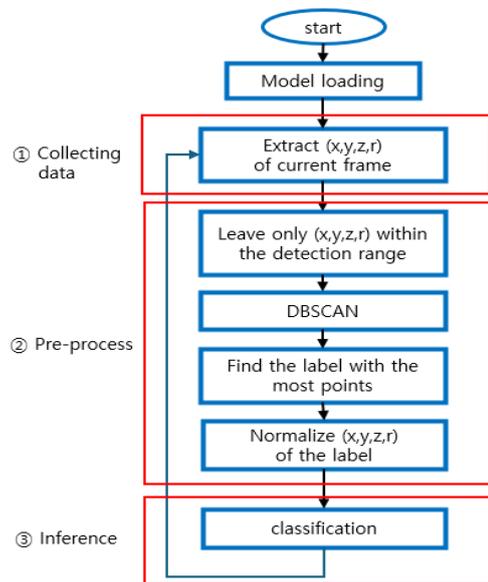


Fig. 3. Flow chart of classification algorithms.

그림 3. 분류 알고리즘 순서도

가. Collecting Data

Raspberry Pi 4 개발 환경에서 Ouster 사가 제공하는 Ouster-SDK[9]를 사용하여 LiDAR 센서가 감지한 포인트 클라우드 데이터를 얻을 수 있다. 위 시스템에서는 프레임 단위로 r(Reflectivity) 정보를 streaming하

는 명령어 내부에 한 프레임의 (x, y, z)를 제공하는 명령어를 삽입하여 Raspberry Pi 4에서 LiDAR 센서가 감지한 (x, y, z, r) 정보를 프레임 단위로 연속해서 얻을 수 있도록 하였다.

나. Pre-process

(1) 감지할 범위 제한

배달 서비스 로봇의 classification은 물체에 따른 배달 서비스 로봇의 대응 변화가 목적이므로, 감지 범위 외의 물체의 classification은 중요도가 낮다. 따라서 위 시스템에서는 감지 범위를 설정하여 감지 범위 내의 포인트 클라우드 데이터만을 남기는 1차 전처리를 수행하였다. 감지 범위 내의 포인트 클라우드 데이터에 대해서만 2차 전처리로써 DBSCAN을 수행하여 전처리 측면에서 경량화를 이룰 수 있었다. 위 시스템에서 감지 범위는 정면 5m, 좌우 양쪽 3m씩으로 설정하였다.

(2) DBSCAN

DBSCAN[10]은 밀도 기반 군집화 알고리즘으로, 입실론(ϵ) 반경 내에 최소 포인트 수가 존재하면, 해당 포인트를 클러스터에 포함하는 과정을 반복하면서, point의 밀도를 기반으로 clustering한다. 따라서 하이퍼 파라미터인 입실론과 최소 포인트 수의 설정이 중요하다. 위 시스템에서 사용된 입실론과 최소 포인트 수는 각각 0.5와 10으로, LiDAR 센서로부터 취득한 포인트 클라우드 데이터를 적절히 clustering하는 파라미터를 실험적으로 찾아내었다.

(3) 가장 많은 포인트를 갖는 label 찾기, 해당 label의 (x, y, z, r) 정규화

배달 서비스 로봇의 감지 범위 내에는 여러 물체가 존

재할 수 있다. classification을 수행하기 위한 한 물체를 선택해야 하는데, 가장 많은 포인트를 가진 물체로 선택하는 것이 합리적이다. LiDAR 센서의 동작 원리상, 가장 많은 포인트를 가진 물체가 배달 서비스 로봇에 가장 가까운 물체일 확률이 높으며, 이는 잠재적으로 위험이 될 수 있기 때문이다. classification을 수행할 물체를 선정한 후에는 해당 물체의 포인트 클라우드 데이터를 정규화해야 한다. 정규화 과정은 normalization과 translation으로 구성되며, 다양한 물체에 대한 일관된 스케일링을 제공한다. 즉, 다양한 크기의 물체를 동일한 기준으로 비교하고 처리할 수 있게 된다.

다. Inference-PFN, DS-CNN

Classification을 수행할 물체의 포인트 클라우드 데이터를 앞서 설명한 PFN의 입력으로 사용하여, pseudo image를 생성한다. 이 pseudo image를 앞서 최적화한 DS-CNN의 입력으로 사용하여 물체의 classification을 수행한다.

III. 실험

제안된 임베디드 시스템을 구성한 후, Raspberry Pi 4 개발 환경에서 실시간 분류 프로그램을 동작시켰다. 프로그램을 실행한 후, Raspberry Pi 4는 현재 프레임의 특정 물체에 대한 Classification 결과를 frame id와 함께 지속적으로 출력한다. 시각적인 확인을 위해 classification 할 물체의 포인트 클라우드 데이터를 csv파일 형태로 저장 후, 그림 4와 같이 제목을 “Predicted Class: 예측 결과” 형태로 plot하였다. 표 4는 DS-CNN의 class별 정확도이다. 표 5는 1000 frame 동안 classification을

Table 4. Confusion matrix for classification accuracy.

표 4. 클래스별 정확도에 대한 혼동 행렬

Object		Predicted							
		Bicycle	Building	Car	Manmade	Obstacle	Person	Tree	Vegetation
True	Bicycle	0.97	0	0	0	0	0	0	0.03
	Building	0	1	0	0	0	0	0	0
	Car	0	0.02	0.98	0	0	0	0	0
	Manmade	0	0.02	0.04	0.82	0.07	0	0.02	0.02
	Obstacle	0	0.03	0	0	0.93	0.03	0	0
	Person	0	0	0	0	0	1	0	0
	Tree	0	0	0.05	0	0	0.02	0.88	0.05
	Vegetation	0	0	0	0	0	0	0	1

수행하여 계산한 frame의 profiling 결과와 fps(frame per second)이다.

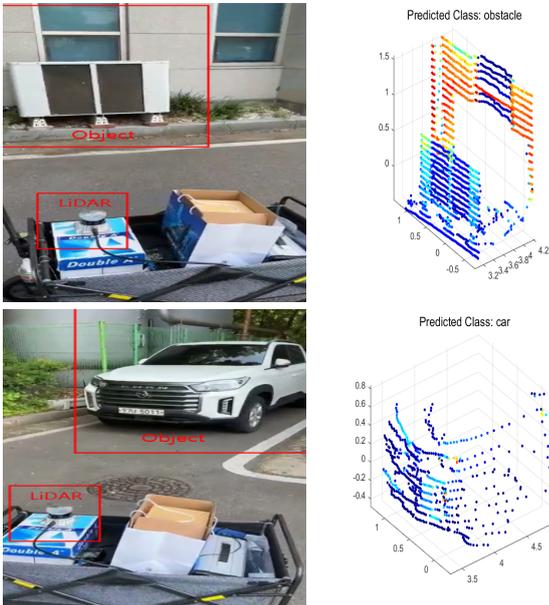


Fig. 4. Experimental environment and point cloud data.

그림 4. 실제 환경 및 포인트 클라우드 데이터

Table 5. Profiling Results.

표 5. 프로파일링 결과

Profiling		Time
Collecting Data / Pre-Process		0.0387s
Inference	Pillar layer	0.0275s
	Pillar encoder	0.0055s
	DS-CNN	0.0082s
Total Time		0.0764s
Performance		13.08fps

IV. 결론

본 논문은 LiDAR 센서와 Raspberry Pi를 통해 배달 서비스 로봇을 위한 객체 분류 시스템을 구성하고, PointPillars와 DS-CNN을 사용하여 시스템을 경량화하였다. 그 결과, 약 13fps으로 classification을 수행하였으며, DS-CNN의 Class 별 정확도는 표 4와 같이 나타났다. 전체 정확도는 약 94.94%를 달성하였다.

표 5의 실시간 분류 알고리즘의 profiling 결과, pre-process와 PFN의 Pillar 생성에 소요된 시간이 크다는 것을 알 수 있었다. 또한 분류 네트워크 분석 결과, custom DS-CNN의 pointwise convolution이 전체 연산량의 약 60%를 차지하는 것을 알 수 있었다. 따라

서, pre-process와 Pillar 생성 알고리즘을 최적화하고, Lowering을 통한 GEMM(General Matrix Multiplication) 하드웨어 설계를 통해 pointwise convolution을 가속화하여 fps를 향상시키는 방향으로 연구를 진행할 계획이다.

References

[1] F. Amzajerjian, "Role of lidar technology in future nasa space missions," *MRS Spring Meeting*, 2008, pp.1076-K04-01.

[2] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom, "PointPillars: fast encoders for object detection from point clouds," *Conference on Computer Vision and Pattern Recognition*, 2019, pp.12689-12697. DOI: 10.48550/arXiv.1812.05784

[3] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, et al., *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, 2017.

[4] Ouster, High-resolution OS1 lidar sensor: robotics, trucking, mapping, security | Ouster, <https://ouster.com/products/hardware/os1-lidar-sensor>

[5] Shang J-J, Phipps N, Wey I-C, Teo TH. "A-DSCNN: Depthwise Separable Convolutional Neural Network Inference Chip Design Using an Approximate Multiplier," *Chips*, vol.2, no.3, pp. 159-172. 2023. DOI: 10.3390/chips2030010

[6] Kaiming He et al., "Deep residual learning for image recognition," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.770-778, 2016. DOI: 10.1109/CVPR.2016.90

[7] LeCun, Yann, et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol.86, no.11, pp.2278-2324, 1998. DOI: 10.1109/5.726791

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv: 1412.6980, 2014. DOI: 10.48550/arXiv.1412.6980

[9] Ouster, Ouster SDK - Ouster Sensor SDK 0.12.0 documentation, <https://static.ouster.dev/sdk-docs/index.html>

[10] Ester, Martin, et al, "A density-based algorithm for discovering clusters in large spatial databases with noise," *kdd*, vol.96, no. 34, pp.226-231, 1996.

BIOGRAPHY

Woo-Jin Park (Member)



2019~present : BS degree course in School of Electronics and Information Engineering, Korea Aerospace University.

Jeong-Gyu Lee (Member)



2019~present : BS degree course in School of Electronics and Information Engineering, Korea Aerospace University.

Chae-Woon Park (Member)



2023 : BS degree course in School of Electronics and Information Engineering, Korea Aerospace University.
2023~present : MS degree course in School of Electronics and Information Engineering, Korea Aerospace University.

Yunho Jung (Member)



1998 : BS degree in Department of Electrical and Electronic Engineering, Yonsei University.

2000 : MS degree in Department of Electrical and Electronic Engineering, Yonsei University

2005 : Ph.D degree in Department of Electrical and Electronic Engineering, Yonsei University.

2005~2007 : Senior Engineer, Samsung Electronics.

2007~2008 : Research professor, Institute of Information Engineering, Yonsei University.

2008~present : Professor, School of Electronics and Information Engineering, Korea Aerospace University