Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 10, No. 2, May, Pp. 253-260

Optimizing autonomous navigation: advances in LiDAR-based object recognition with modified Voxel-RCNN

Firman[*1], Arief Suryadi Satyawan[2], Helfy Susilawati[1], Mokh. Mirza Etnisa Haqiqi[1], Khaulyca Arva Artemeysia[1], Sani Moch. Sopian[1], Beni Wijaya[1], M. Ikbal Shamie[1]
Universitas Garut, Indonesia[1]
Research Center for Telecommunication, National Research and Innovation Agency (BRIN), Indonesia[2]

Article Info

Keywords:
Autonomous Vehicles, LiDAR, Voxel-RCNN, Object Recognition, Point Cloud Dataset, 3D Detection

Article history:
Received: December 22, 2024
Accepted: March 06, 2025
Published: May 31, 2025

Cite:
Firman, “Optimizing Autonomous Navigation: Advances in LiDAR-based Object Recognition with Modified Voxel-RCNN”, *KINETIK*, vol. 10, no. 2, May 2025. https://doi.org/10.22219/kinetik.v10i2.2199

*Corresponding author.
Firman
E-mail address:
24052121044@fteknik.uniga.ac.id

Abstract

This study aimed to enhance the object recognition capabilities of autonomous vehicles in constrained and dynamic environments. By integrating Light Detection and Ranging (LiDAR) technology with a modified Voxel-RCNN framework, the system detected and classified six object classes: human, wall, car, cyclist, tree, and cart. This integration improved the safety and reliability of autonomous navigation. The methodology included the preparation of a point cloud dataset, conversion into the KITTI format for compatibility with the Voxel-RCNN pipeline, and comprehensive model training. The framework was evaluated using metrics such as precision, recall, F1-score, and mean average precision (mAP). Modifications to the Voxel-RCNN framework were introduced to improve classification accuracy, addressing challenges encountered in complex navigation scenarios. Experimental results demonstrated the robustness of the proposed modifications. Modification 2 consistently outperformed the baseline, with 3D detection scores for the car class in hard scenarios increasing from 4.39 to 10.31. Modification 3 achieved the lowest training loss of 1.68 after 600 epochs, indicating significant improvements in model optimization. However, variability in the real-world performance of Modification 3 highlighted the need for balancing optimized training with practical applicability. Overall, the study found that the training loss decreased up to 29.1% and achieved substantial improvements in detection accuracy under challenging conditions. These findings underscored the potential of the proposed system to advance the safety and intelligence of autonomous vehicles, providing a solid foundation for future research in autonomous navigation and object recognition.

1. Introduction

The rapid advancement of autonomous vehicle technology has transformed various industries, including logistics, agriculture, and warehousing, where autonomous systems are increasingly employed in dynamic yet confined environments [1][2][3][4][5]. A critical component of autonomous systems is their ability to detect and recognize objects within their surroundings accurately [6]. Object recognition not only enhances situational awareness but also plays a pivotal role in enabling autonomous vehicles to anticipate the behavior of objects, make informed decisions, and avoid potential collisions. This capability is vital for ensuring the safety and reliability of autonomous navigation systems, particularly in restricted and cluttered environments where operational parameters are highly constrained [7][8].

Despite significant progress in object detection technologies, many existing methods primarily focus on object detection without delving into advanced object recognition. This limitation reduces the autonomous system's ability to interpret the intent or movement patterns of objects, potentially leading to unsafe interactions [9]. The challenge becomes more pronounced in confined spaces where autonomous vehicles must maneuver carefully and predict object behavior accurately. Traditional approaches, which often rely on 2D image-based sensors or basic LiDAR implementations, fall short in providing the nuanced recognition needed for complex, real-world scenarios. Addressing this gap is crucial to advancing autonomous vehicle technologies, particularly for applications that demand both precision and adaptability in object recognition [10][11].

This research proposes a novel approach to enhance autonomous vehicle navigation through an innovative application of Light Detection and Ranging (LiDAR) technology combined with a modified Voxel-RCNN framework [12]. LiDAR technology offers high-resolution point cloud data that is instrumental in detecting and recognizing objects in 3D space. The proposed method aims to transform raw LiDAR data into actionable insights by classifying objects with high accuracy across six predefined object classes. Enhancements to the Voxel-RCNN model are introduced to improve classification performance, addressing the shortcomings of traditional methods. By benchmarking this approach against existing methodologies, the study seeks to demonstrate not only the effectiveness but also the novelty of the proposed solution, ultimately contributing to safer and more intelligent autonomous vehicle systems in restricted environments.

Through this study, a significant contribution to the field of autonomous navigation is anticipated by bridging the gap between basic object detection and advanced object recognition. The findings could potentially set a new benchmark for safety and efficiency in environments where autonomous vehicles operate, offering a robust solution to current industry challenges.

2. Research Method

This section presents a structured and systematic approach to developing a robust 3D object detection system by leveraging the Voxel-RCNN architecture, which is well-suited for processing three-dimensional spatial data. The pipeline spans several interconnected stages, from raw data preparation to advanced visualization techniques, ensuring accuracy, scalability, and applicability to real-world scenarios.

The process begins with dataset creation, where raw point cloud data is collected, preprocessed, labeled into six classification categories, and converted into KITTI format (.pkl) for compatibility with the Voxel-RCNN model. The dataset is then divided into three subsets: training, validation, and testing. The training dataset is used to develop the Voxel-RCNN model through an optimization process that minimizes the loss function, while validation data is utilized during training to ensure accuracy and prevent overfitting. Finally, the testing dataset is reserved for evaluating the model's performance on unseen data. The evaluation process involves performance metrics such as Bird's Eye View (BEV) and 3D visualization, providing a comprehensive assessment of the trained model. Figure 1 illustrates the complete workflow of the 3D object detection system, highlighting each stage from dataset preprocessing to model evaluation.

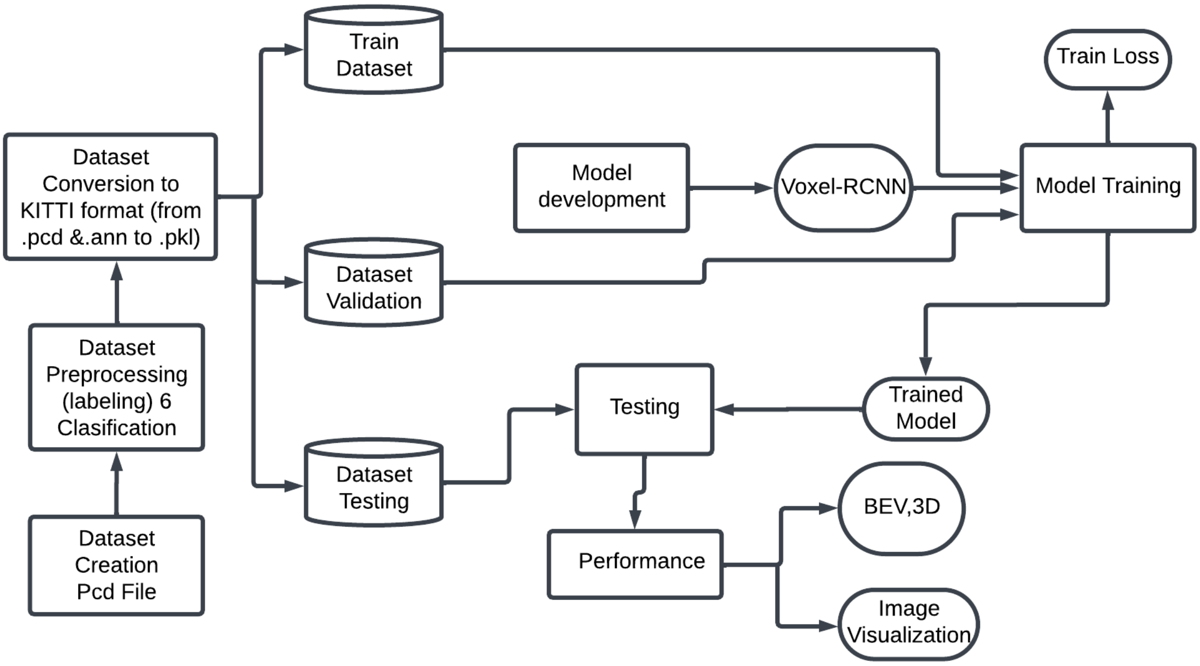

Figure 1. Process Architecture

2.1 Dataset Creation and Preprocessing

The process begins with the creation of a dataset in Point Cloud Data (PCD) format [13]. PCD is a commonly used file format for representing three-dimensional spatial information, where each point is defined by its (x,y,z)(x, y, z)(x,y,z) coordinates, along with optional attributes such as intensity or reflectance [14]. For instance, a LiDAR sensor generates point clouds that capture an object's physical dimensions and spatial location in a scene [15]. Suppose the dataset comprises N point clouds, each containing M_i points (where i = 1, 2, ………., N), the total number of points in the dataset is computed in Equation 1.

$$Total\ Points = \sum_{i=1}^{N} \mathrm{M}i \quad (1)$$

These raw point clouds are processed and labeled into six predefined object categories, such as vehicles, pedestrians, cyclists, and background [16]. Labeling assigns a unique class CCC to each point, forming the ground truth for training [17]. This classification step ensures that the model learns to distinguish between different object classes effectively.

2.2 Dataset Conversion to KITTI Format

After labeling, the dataset undergoes conversion into KITTI format, a widely accepted standard for benchmarking 3D object detection models [18]. KITTI format comprises structured files containing both 3D point clouds and their corresponding annotations [19]. Each object is represented by a bounding box, defined by its spatial extents (xmin, ymin, zmin, xmax, ymax, zMax) These bounding boxes encapsulate the dimensions and positions of objects, enabling the model to learn object localization in three-dimensional space. The conversion also involves serializing the data into compact .pkl files for efficient storage and loading. The memory requirement for annotations can be estimated in Equation 2.

$$Annotation\ Size\ (bytes) = K \times 6 \times sizeof(float) \tag{2}$$

Where *K* is the total number of bounding boxes, and sizeof(float) (typically 4 bytes) represents the memory allocated to store each coordinate.

2.3 Dataset Splitting for Validation and Testing

To train and evaluate the model, a dataset comprising a total of 2,467 samples was utilized [20]. This dataset was carefully collected in a controlled environment within the National Research and Innovation Agency (BRIN), specifically at the Samaun Samadikun area in Bandung. The controlled setting ensures high-quality data with consistent environmental parameters, making it a reliable foundation for developing and testing object detection systems.

The dataset is divided into three subsets: training, validation, and testing, ensuring a robust evaluation framework. Specifically, the training dataset consists of 1,904 samples, accounting for approximately 77% of the total data. These samples are utilized to optimize the model parameters, allowing it to effectively learn and generalize patterns in the data. The validation dataset includes 368 samples, representing approximately 15% of the total data. These samples serve to fine-tune hyperparameters and monitor the model's performance during training, thereby preventing overfitting. Finally, the testing dataset comprises 195 samples, equivalent to 8% of the total data. These are reserved exclusively for evaluating the model's performance and generalization capabilities after training [21].

This division follows a well-balanced ratio, ensuring sufficient data for training while retaining adequate samples for validation and testing. The consistent sampling method further enhances the dataset's reliability, particularly given its origin in a controlled BRIN environment. During validation and testing, the Intersection over Union (IoU) metric is employed to quantify the alignment between predicted and ground truth bounding boxes [22]. IoU is defined mathematically in Equation 3.

$$IoU = \frac{Area\ Union}{Area\ Overlap} \tag{3}$$

A higher IoU indicates better alignment, and thresholds (e.g., IoU ≥ 0.5) are used to classify detections as true positives or false positives.

2.4 Model Development Using Voxel-RCNN

The Voxel-RCNN model serves as the core of this workflow. Voxelization is the initial step in processing point clouds, dividing the 3D space into a uniform grid of small cubes called voxels [23]. Each voxel aggregates features from the points within it, simplifying processing through neural networks [24]. The number of voxels in each dimension is determined by the voxel size and spatial range in Equation 4.

$$\boldsymbol{Nx} = \left\lceil \frac{xmax - xmin}{s} \right\rceil, \quad \boldsymbol{Ny} = \left\lceil \frac{ymax - ymin}{s} \right\rceil, \quad \boldsymbol{Nz} = \left\lceil \frac{Zmax - Zmin}{s} \right\rceil \tag{4}$$

The voxelized data is then fed into the Voxel-RCNN network, which extracts features using convolutional layers and predicts bounding boxes and object classes with high accuracy.

2.5 Model Training

The training process aims to minimize a loss function that combines classification and regression components. The classification loss (e.g., cross-entropy) evaluates how well the model predicts object classes, while the regression loss (e.g., Smooth L1 Loss) measures the accuracy of predicted bounding box coordinates. The total loss function is formulated in Equation 5.

$$\boldsymbol{Ltotal} = Lclassification + \lambda \cdot Lregression \tag{5}$$

Here, λ is a hyperparameter balancing the two components [25]. Model parameters are iteratively optimized using algorithms such as Stochastic Gradient Descent (SGD) or Adam.

2.6 Model Testing and Performance Evaluation

After training, the model is tested on the testing dataset to assess its generalization performance [26]. Key evaluation metrics include:

Precision: The proportion of correctly predicted objects among all predictions in Equation 6.

$$\boldsymbol{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{6}$$

Recall: The proportion of correctly predicted objects among all ground truth objects in Equation 7.

$$\boldsymbol{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negative} \tag{7}$$

F1-Score: The harmonic mean of Precision and Recall in Equation 8.

$$\boldsymbol{F1} = 2 \cdot \frac{Precision \cdot Recal}{precision \cdot Recal} \tag{8}$$

Mean Average Precision (mAP) [27]: The average precision across all object classes and IoU thresholds, providing an overall performance metric in Equation 9.

$$\boldsymbol{mAP} = \frac{1}{n}\sum_{i=0}^{n} \mathrm{APi} \tag{9}$$

2.7 Visualization in BEV and 3D

The final stage involves visualizing model predictions in Bird’s Eye View (BEV) and 3D space. BEV projects the 3D point cloud onto a 2D plane, providing a top-down perspective that is particularly useful in applications like autonomous driving, where spatial awareness is critical. Conversely, 3D visualization overlays predicted bounding boxes onto the original point cloud, enabling qualitative evaluation of prediction accuracy and alignment. The computational complexity of visualization depends on the resolution of the point cloud and the number of detected objects. Effective visualization enhances interpretability and identifies areas for model improvement.

3. Results and Discussion

Voxel-RCNN is a widely used model for 3D object detection, and modifications to its architecture and hyperparameters can significantly influence the performance. This study evaluates three modifications applied to the Voxel-RCNN model using a custom LiDAR dataset. The analysis considers the loss trends, BEV (Bird’s Eye View) accuracy, and 3D detection accuracy under different difficulty levels (Easy, Moderate, Hard). The discussion also includes comparisons with prior research to highlight improvements and contributions to LiDAR-based object recognition. Compared to previous studies that focus primarily on structured environments, this research investigates the model’s adaptability to more complex real-world LiDAR datasets.

3.1 Training Loss Analysis

The loss graphs indicate the stability and convergence rate of each model variation. The default model exhibits relatively stable loss reduction, whereas Modifications 1 and 2 show slight fluctuations, suggesting sensitivity to architectural or hyperparameter changes. Modification 3 achieves a smoother convergence, indicating improved model stability. However, the final loss values remain similar across modifications, implying that improvements in detection performance might arise from other factors, such as feature extraction efficiency. This trend is clearly illustrated in Figure 2, which presents the loss graph for each model variation throughout the training process.

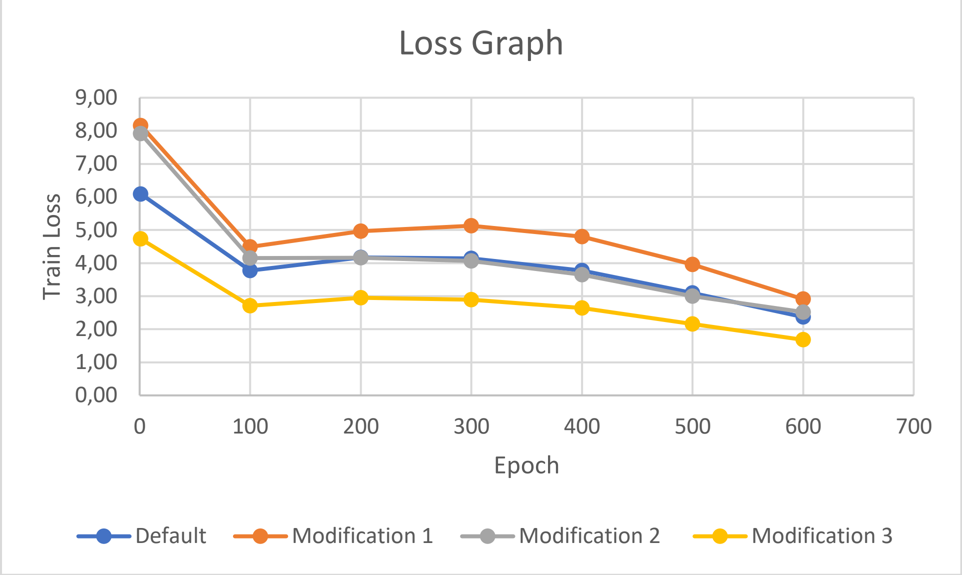

Figure 2. Loss Graph

3.2 Detection Metrics Evaluation

The evaluation of Bird's Eye View (BEV) and 3D object detection metrics focused on six object classes Human, Wall, Car, Cyclist, Tree, and Cart under three levels of difficulty: Easy, Moderate, and Hard. The findings are summarized below in Table 1, Table 2, Table 3, and Table 4.

Table 1. Evaluation Metrics BEV and 3D Model Default (Custom Dataset)

Clasification	BEV			3D		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Human	54,3302	29,5195	6,2965	50,783	11,3822	4,5455
Wall	26,5057	14,9042	3,917	20,2877	11,4263	3,0303
Car	68,1377	57,7764	32,2209	55,834	36,4733	4,3867
Cyclist	31,7906	23,127	10,1849	22,2535	10,3853	9,0909
Tree	21,1267	10,2238	1,5579	14,8123	6,2982	0,1535
Cart	36,1061	29,6834	5,0671	31,2497	7,7124	1,0101

Table 2. Evaluation Metrics BEV and 3D Modification 1 (Custom Dataset)

Clasification	BEV			3D		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Human	52,5102	24,1813	2,9589	42,3793	8,4948	1,2987
Wall	20,0590	10,4190	2,9131	17,7451	5,7487	2,2062
Car	70,6275	60,7917	37,3534	59,3902	35,7529	4,7910
Cyclist	27,9477	23,0143	10,7455	18,5866	10,6727	9,0909
Tree	20,7057	9,8964	1,1453	14,9544	4,4833	0,3636
Cart	32,6655	30,0914	11,3268	32,0197	17,1054	1,5152

Table 3. Evaluation Metrics BEV and 3D Modification 2 (Custom Dataset)

Clasification	BEV			3D		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Human	51,2788	28,6414	3,8841	47,6951	15,9204	2,2727
Wall	24,8640	11,8884	3,9601	17,4174	9,1753	3,0303
Car	69,7595	58,6979	37,1323	56,3269	35,5246	10,3147

Cite: Firman, "Optimizing Autonomous Navigation: Advances in LiDAR-based Object Recognition with Modified Voxel-RCNN", KINETIK, vol. 10, no. 2, May 2025. https://doi.org/10.22219/kinetik.v10i2.2199

Cyclist	35,4734	21,0508	9,9243	24,4987	10,1612	9,0909
Tree	26,0298	15,3060	2,2727	19,7907	11,3935	0,2045
Cart	29,8904	28,4611	5,9867	29,0122	8,0001	2,5735

Table 4. Evaluation Metrics BEV and 3D Modification 3 (Custom Dataset)

Clasification	BEV			3D		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Human	55,2619	25,8747	6,1640	51,1075	10,5138	4,5455
Wall	23,0354	10,9941	4,2712	16,6164	7,5547	2,2727
Car	67,8333	56,0881	35,5241	51,7047	31,8638	3,7728
Cyclist	31,6481	21,8701	4,2546	22,0561	5,7668	3,0303
Tree	25,5878	14,8133	1,0101	19,4566	10,7731	1,0100
Cart	29,3512	27,5013	4,6807	28,7830	7,9860	0,8686

Human: The default model achieves BEV scores of (54.33, 29.52, 6.30) and 3D scores of (50.78, 11.38, 4.54). Modification 3 slightly enhances 3D accuracy (51.10, 10.51, 4.54), while Modifications 1 and 2 show performance declines. The results suggest that modifications do not significantly enhance human detection compared to the default model.

Wall: The default model performs poorly in detecting walls, with BEV scores of (26.50, 14.90, 3.91) and 3D scores of (20.28, 11.42, 3.03). Modification 2 offers slight improvements, while Modification 1 performs the worst. The results indicate that modifications do not drastically improve wall detection, likely due to the planar nature of wall structures in point cloud data.

Car: The default model achieves BEV scores of (68.14, 57.77, 32.22) and 3D scores of (55.83, 36.47, 4.38). Modification 2 enhances 3D accuracy in hard conditions (10.31%), demonstrating better adaptability in complex driving scenarios.

Cyclist: The default model achieves moderate performance, with BEV scores of (31.79, 23.12, 10.18) and 3D scores of (22.25, 10.38, 9.09). Modification 3 introduces a significant performance drop, suggesting that modifications might have inadvertently reduced feature discrimination for small and narrow objects.

Tree: The default model achieves low accuracy, with BEV scores of (21.12, 10.22, 1.55) and 3D scores of (14.81, 6.29, 0.15). Modification 2 shows slight improvements in 3D accuracy (19.79, 11.39, 0.20), suggesting better adaptability to irregularly shaped tree structures.

Cart: The default model achieves BEV scores of (36.10, 29.68, 5.06) and 3D scores of (31.24, 7.71, 1.01). Modification 1 enhances BEV accuracy, while Modification 2 slightly improves 3D detection.

Overall, while modifications provide minor improvements for certain object classes, the default model remains competitive, particularly in car and cyclist detection. Modifications mainly enhance performance in complex scenarios but require further tuning to ensure consistent accuracy gains across all object types.

3.3 Comparison with Previous Research

Prior research on LiDAR-based object recognition has focused predominantly on vehicle detection in structured environments, such as highways and urban roads. The default model, trained on the KITTI dataset, achieves superior car detection but struggles with other object classes. This study extends prior work by optimizing Voxel-RCNN for a more diverse dataset that includes infrastructure elements and smaller objects. Compared to previous works that employ highly curated datasets, this study evaluates model performance under more challenging real-world conditions, providing insights into robustness and generalizability. The performance of the default model on the KITTI dataset is summarized in Table 5, showcasing its strengths and limitations in a controlled environment.

The primary contribution of this research lies in its comparative analysis of different Voxel-RCNN modifications tailored for real-world applications. Unlike previous studies that primarily fine-tune models for specific datasets, this study explores architectural modifications to improve detection across varied object types. The findings highlight the importance of dataset adaptability and modifications that balance performance across all object categories rather than prioritizing single-class optimization. Table 6 presents the evaluation metrics of the default model applied to the custom dataset, offering a baseline for assessing the impact of the proposed modifications.

Table 5. Evaluation Metrics BEV and 3D Voxel RCNN Model Default (Kitti Dataset)[12]

Clasification	BEV			3D		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Car	95.52	91.25	88.99	92.38	85.29	82.86

Table 6. Evaluation Metrics BEV and 3D model Default (Custom Dataset)

Clasification	BEV			3D		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Human	54,3302	29,5195	6,2965	50,783	11,3822	4,5455
Wall	26,5057	14,9042	3,917	20,2877	11,4263	3,0303
Car	68,1377	57,7764	32,2209	55,834	36,4733	4,3867
Cyclist	31,7906	23,127	10,1849	22,2535	10,3853	9,0909
Tree	21,1267	10,2238	1,5579	14,8123	6,2982	0,1535
Cart	36,1061	29,6834	5,0671	31,2497	7,7124	1,0101

3.4 LiDAR-based Object Recognition and Scenario Analysis

LiDAR technology is essential for 3D object recognition, particularly in autonomous navigation and smart infrastructure applications. This study evaluates the influence of LiDAR penetration, point density, and occlusions on detection accuracy. Experimental scenarios include varying distances, occlusion levels, and object orientations to analyze detection robustness under real-world conditions.

The findings emphasize the importance of LiDAR feature extraction techniques to maintain detection consistency across different object categories. While the default Voxel-RCNN model performs well in structured settings, modifications enhance detection under more complex scenarios by improving feature encoding for occluded and irregularly shaped objects.

4. Conclusion

Considering both BEV and 3D results, Modification 2 emerges as the most balanced model, improving 3D detection in complex scenarios while maintaining BEV accuracy. Modification 1 shows inconsistent improvements, whereas Modification 3 demonstrates stability in BEV but suffers from 3D detection losses. The default model remains a strong baseline, but the modifications provide valuable insights into tuning detection performance for specific object classes.

This study highlights the trade-offs associated with different modifications to Voxel-RCNN. Modification 2 presents the best overall performance, especially in 3D object detection, while Modification 3 provides stability in BEV performance. Future work should focus on hybrid approaches that leverage the strengths of each modification while mitigating their respective weaknesses. Additional validation using real-world data and simulation environments will be crucial for further model improvements.

Notation

x = Global vechile x-position
y = Global vechile y-position
z = Global vechile z-position
N = horizon Valeu
K = total number of bounding box

Acknowledgement

This research was supported by funding from the Indonesia Endowment Fund for Education (Lembaga Pengelola Dana Pendidikan) through the National Research and Innovation Agency's Research and Innovation for Advanced Indonesia (RIIM) program (BRIN). The collaboration between the Faculty of Engineering, Universitas Garut, and BRIN has been invaluable in facilitating this research, and we would like to express our sincere gratitude for their continued support and cooperation.

References

[1] M. Nadeem Hangar, Q. Ahmed, F. Khan, and M. Hafeez, “A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges,” *Sensors*, vol. 21, p. 706, 2021. https://doi.org/10.3390/s21030706

[2] R. Keith and H. La, “Review of Autonomous Mobile Robots for the Warehouse Environment,” 2024. https://doi.org/10.48550/arXiv.2406.08333

[3] A. Roshanianfard, N. Noguchi, H. Okamoto, and K. Ishii, “A review of autonomous agricultural vehicles (The experience of Hokkaido University),” *J. Terramechanics*, vol. 91, pp. 155–183, 2020. https://doi.org/10.1016/j.jterra.2020.06.006

[4] M. Ibiyemi and D. Olutimehin, “Revolutionizing logistics: The impact of autonomous vehicles on supply chain efficiency,” *Int. J. Sci. Res. Updat.*, vol. 8, pp. 9–26, 2024. https://doi.org/10.53430/ijsru.2024.8.1.0042

[5] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A Survey of Autonomous Driving: Common Practices and Emerging Technologies,” *IEEE Access*, vol. PP, p. 1, 2020. https://doi.org/10.1109/ACCESS.2020.2983149

[6] R. Qian, X. Lai, and X. Li, *3D Object Detection for Autonomous Driving: A Survey*. 2021. https://doi.org/10.48550/arXiv.2106.10823

[7] R. Qian, X. Lai, and X. Li, “3D Object Detection for Autonomous Driving: A Survey,” *Pattern Recognit.*, vol. 130, 2022. https://doi.org/10.1016/j.patcog.2022.108796

[8] F. Liu, Z. Lu, and X. Lin, “Vision-based environmental perception for autonomous driving,” *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.*, 2023. https://doi.org/10.1177/09544070231203059

[9] L. Peng, H. Wang, and J. Li, “Uncertainty Evaluation of Object Detection Algorithms for Autonomous Vehicles,” *Automot. Innov.*, vol. 4, 2021. https://doi.org/10.1007/s42154-021-00154-0

[10] L. Lidar, L. Bai, S. Member, Y. Zhao, X. Huang, and S. Member, “Enabling 3D Object Detection with a,” vol. 14, no. 8, pp. 2–5, 2015.

[11] Y. Wang, W. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-LiDAR from Visual Depth Estimation : Bridging the Gap in 3D Object Detection for Autonomous Driving”.

[12] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, “Voxel R-CNN : Towards High Performance Voxel-based 3D Object Detection,” 2020.

[13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017. https://doi.org/10.1109/TPAMI.2016.2577031

[14] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[15] N. L. W. Keijsers, “Neural Networks,” *Encycl. Mov. Disord. Three-Volume Set*, pp. V2-257-V2-259, 2010. https://doi.org/10.1016/B978-0-12-374105-9.00493-7

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020. https://doi.org/10.1109/TPAMI.2018.2844175

[17] J. Shin, J. Kim, K. Lee, H. Cho, and W. Rhee, “Diversified and Realistic 3D Augmentation via Iterative Construction, Random Placement, and HPR Occlusion,” *Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023*, vol. 37, pp. 2282–2291, 2023. https://doi.org/10.1609/aaai.v37i2.25323

[18] A. Dosovitskiy *et al.*, “an Image Is Worth 16X16 Words: Transformers for Image Recognition At Scale,” *ICLR 2021 - 9th Int. Conf. Learn. Represent.*, 2021.

[19] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[20] Y. Bengio, *Learning deep architectures for AI*, vol. 2, no. 1. 2009. https://doi.org/10.1561/2200000006

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.

[22] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[23] Z. Chao, F. Pu, Y. Yin, B. Han, and X. Chen, “Research on real-time local rainfall prediction based on MEMS sensors,” *J. Sensors*, vol. 2018, pp. 1–9, 2018. https://doi.org/10.1155/2018/6184713

[24] G. Cohen and R. Giryes, “Generative Adversarial Networks,” *Mach. Learn. Data Sci. Handb. Data Min. Knowl. Discov. Handbook, Third Ed.*, pp. 375–400, 2023. https://doi.org/10.1007/978-3-031-24628-9_17

[25] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 9626–9635, 2019. https://doi.org/10.1109/ICCV.2019.00972

[26] Q. Zhong and X.-F. Han, “Point Cloud Learning with Transformer,” 2021. https://doi.org/10.48550/arXiv.2104.13636

[27] A. Howard *et al.*, “Searching for MobileNetV3,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324. https://doi.org/10.1109/ICCV.2019.00140