A deep learning solution for compressed semantic segmentation of LiDAR point cloud maps

Bui Thi Thanh Tam[1*], Cao Van Toan[1], Phan Huy Anh[1],
Pham Van Quoc[2], Pham Dang Duong[2]

[1]Institute of Information Technology and Electronics, Academy of Military Science and Technology, 17 Hoang Sam, Nghia Do, Hanoi, Vietnam;
[2]VNU University of Science, 182 Luong The Vinh, Thanh Xuan, Hanoi, Vietnam.
[*]Corresponding author: thanhtambui85@gmail.com
Received 4 Aug. 2025; Revised 26 Sep. 2025; Accepted 10 Oct. 2025; Published 30 Oct. 2025.
DOI: https://doi.org/10.54939/1859-1043.j.mst.IITE.2025.131-138

ABSTRACT

Navigating Unmanned Aerial Vehicles (UAVs) in Global Navigation Satellite System (GNSS)-denied environments often relies on pre-built Light Detection and Ranging (LiDAR) maps. However, the large memory footprint and high computational cost of these point cloud maps pose significant challenges for resource-constrained UAVs. This paper proposes a deep learning solution using a lightweight, modified RandLA-Net architecture to efficiently compress and semantically segment these maps. Our results demonstrate a significant reduction in model size and memory usage while maintaining competitive segmentation accuracy, presenting a viable solution for real-time, on-board processing on embedded systems.

Keywords: Deep learning, Localization and navigation; Point cloud; LiDAR; Semantic segmentation.

1. INTRODUCTION

The autonomy of Unmanned Aerial Vehicles (UAVs) in complex environments is a cornerstone of modern robotics, with applications ranging from search and rescue to military reconnaissance and infrastructure inspection. A primary challenge, especially in indoor or confined spaces like dense urban canyons, is the unreliability of GNSS signals [1]. To overcome this, UAVs often rely on Simultaneous Localization and Mapping (SLAM) techniques that utilize pre-built maps for robust localization. LiDAR sensors are frequently used to build these maps, generating detailed 3D point clouds [2].

While these maps are information-rich, their size presents a significant bottleneck. Large-scale point clouds can easily reach gigabytes in size, demanding substantial storage and computational power for tasks like localization and path planning. This is often beyond the capacity of the lightweight, power-efficient embedded computers typically found on small UAVs [1].

Deep learning has emerged as a powerful paradigm for 3D point cloud processing [2-5]. Two prominent architectures in semantic segmentation are KPConv [3] and RandLA-Net [4]. KPConv uses kernel points to define flexible and deformable convolutions, achieving state-of-the-art accuracy, but often at the cost of high computational complexity and large model sizes. In contrast, RandLA-Net is an efficient and lightweight architecture that leverages random point sampling to process large-scale point clouds directly, making it a strong candidate for real-time applications on constrained hardware.

This paper investigates the feasibility of using a deep learning approach to simultaneously compress and segment LiDAR maps for UAV navigation. We propose a lightweight variant of the RandLA-Net architecture tailored for resource-constrained systems.

This paper's primary contributions are threefold. First, we propose and evaluate a lightweight RandLA-Net model, benchmarking it against its full version and the state-of-the-art KPConv network. Second, we provide a comprehensive analysis of the critical trade-offs for UAV

deployment, specifically between segmentation accuracy (mean Intersection-over-Union (mIoU), overall Accuracy (Acc)), model size, and data storage. Finally, we quantify the impact of key parameters, such as input point density and preprocessing choices, on the proposed model's overall performance and efficiency.

This paper is structured as follows: Related work section 3 provides the principles of the RandLA-Net architecture. Section 4 describes our experimental setup and a detailed analysis of the results. Section 5 concludes the paper and suggests directions for future research.

2. RELATED WORK

Deep learning on 3D point clouds has evolved through several distinct approaches, each with its own strengths and weaknesses. These can be broadly categorized as follows:

Projection-based and Voxel-based Networks: To leverage the success of 2D Convolutional Neural Networks (CNNs), early methods projected 3D point clouds onto 2D images from multiple viewpoints or flattened them into top-down representations [5]. While effective, this process can lead to the loss of geometric details due to occlusion and discretization. Voxel-based methods convert point clouds into 3D grids and apply 3D CNNs [2]. Although they achieve strong results, their memory and computational costs increase cubically with resolution, making them less suitable for large-scale, high-density scenes.

Point-based Networks: This family of methods processes raw point clouds directly, respecting their irregular structure. The pioneering work, PointNet [6], used shared Multi-Layer Perceptrons (MLPs) to learn per-point features, followed by a symmetric pooling function (e.g., max-pooling) to achieve permutation invariance. However, PointNet processed each point independently, failing to capture local geometric context. PointNet++ addressed this by introducing a hierarchical architecture that applies PointNet recursively on nested partitions of the point cloud, capturing local features at multiple scales [7]. A key component of PointNet++ is its sampling strategy, which often relies on Farthest Point Sampling (FPS). While FPS provides good coverage of the point set, its quadratic computational complexity $\left(\mathcal{O}\left(N^2\right)\right)$ makes it a significant bottleneck for large-scale point clouds with millions of points.

Graph-based Networks: These methods treat a point cloud as a graph, where points are nodes and spatial relationships define edges. For instance, Superpoint Graphs (SPG) first partitions a large point cloud into a set of simple, geometrically homogeneous superpoints [8]. These superpoints then serve as nodes for a graph, which is processed by a graph neural network. This approach can effectively manage large scenes, but the expensive pre-processing steps of graph construction and partitioning make it less ideal for end-to-end, real-time systems.

Kernel-based Networks: This category aims to define a true convolution operator directly on points. KPConv (Kernel Point Convolution) is a leading example [3]. It defines a set of convolution weights that are spatially located by a small set of kernel points. KPConv is highly flexible, and its deformable version can learn to adapt its kernel shape to the local geometry, leading to state-of-the-art accuracy. However, this high descriptive power comes with increased model complexity and computational cost, which can be challenging for resource-constrained devices.

Approaches to creating lightweight models for point cloud processing typically follow two main paths: (1) post-training optimization, such as pruning or quantization of large, complex models like KPConv; and (2) efficient architecture design from the ground up. The first path can reduce model size, but the process is often complex and may not be optimal for specialized embedded hardware. In contrast, the second path focuses on using building blocks that are inherently computationally efficient. RandLA-Net is a prime example of this approach [4]. Instead of relying on expensive operations like Farthest Point Sampling $\left(\mathcal{O}\left(N^2\right)\right)$ in PointNet++ or the

complex kernels of KPConv, RandLA-Net's entire architecture is built upon two extremely efficient core components: random sampling, which has $\left(\mathcal{O}(1)\right)$ complexity, and shared MLPs. This inherent simplicity and efficiency make RandLA-Net not only fast but also an ideal foundation for systematic parameter reduction and customization. Our work focuses on RandLA-Net [4] because its design philosophy directly targets efficiency and scalability on large point clouds, making it the most promising foundation for a lightweight model suitable for UAVs.

3. METHODOLOGY

RandLA-Net's efficiency stems from its combination of extremely fast random sampling and a powerful yet local feature aggregator designed to compensate for the stochastic nature of the sampling.

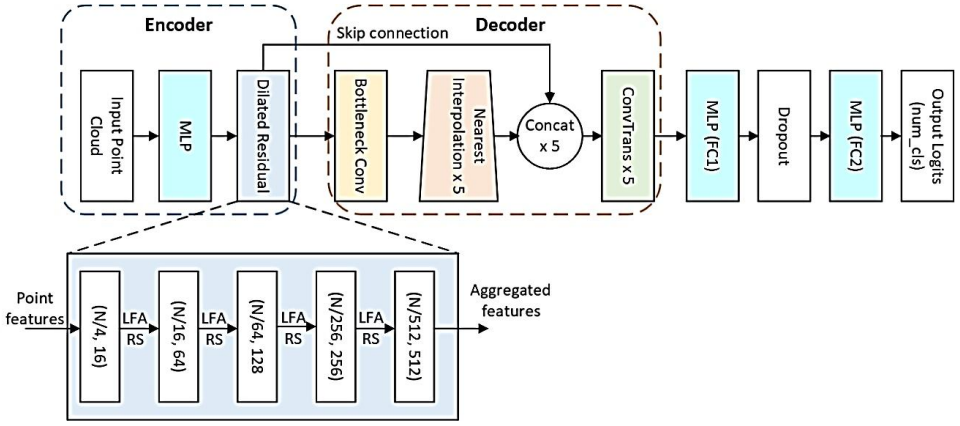

Figure 1. *The structure of RandLA-Net.*

The structure of RandLA-Net is shown in figure 1. The architecture follows a robust encoder-decoder structure with skip connections to ensure both efficiency and high performance. The network takes an entire point cloud with N points, where N represents the number of points sampled from the larger environment map (e.g., a room or a segment of a building) that are processed simultaneously by the network. The encoder progressively downsamples the point cloud through five consecutive layers, called a Dilated Residual Block. Each of these blocks utilizes a Local Feature Aggregation (LFA) module, which is crucial for preserving complex geometric details by progressively increasing the receptive field for each point. Following the LFA module, Random Sampling (RS) is used to efficiently and significantly reduce the number of points, enabling the processing of massive datasets. The decoder mirrors the encoder's structure, using five layers of upsampling and feature interpolation to restore the original point cloud resolution. High-resolution feature maps from the encoder are passed to the decoder via skip connections, which help in retaining fine-grained details lost during downsampling. The LFA consists of two main units: Local Spatial Encoding (LocSE) and Attentive Pooling.

(1) Local spatial encoding (LocSE): This unit explicitly encodes the precise relative positions of neighboring points to augment their features, allowing the network to learn local geometric patterns. For a given center point p_i, the process is as follows:

- Find neighbors: The K nearest neighbors of p_i, denoted as $\left\{p_i^1,\ldots,p_i^K\right\}$, are identified using a K - Nearest Neighbor (KNN) algorithm based on Euclidean distances.
- Encode relative position: For each neighboring point p_i^k, a rich positional encoding r_i^k is computed by concatenating the coordinates of the center point, the neighbor point, their difference, and the Euclidean distance between them. This is then passed through a shared MLP:

$$r_i^k = \text{MLP}\left(p_i \oplus p_i^k \oplus \left(p_i - p_i^k\right) \oplus \left\|p_i - p_i^k\right\|\right) \quad (1)$$

where $\oplus$ denotes the concatenation operation. This explicit encoding of relative spatial information is crucial for the network to understand local geometry.

- Feature augmentation: The original feature vector of the neighbor, f_i^k, is concatenated with its computed positional encoding r_i^k to produce an augmented feature vector, $\hat{f}_i^k$.

(2) Attentive pooling: Instead of using information-lossy pooling methods like max or average pooling, RandLA-Net uses an attention mechanism to intelligently aggregate the augmented features from all neighbors.

- Compute attention scores: A shared function $g(\bullet)$, composed of an MLP and a softmax operator, is used to learn a unique attention score s_i^k for each augmented neighboring feature $\hat{f}_i^k$. The learnable weights of the MLP are denoted by W_{att}.

$$s_i^k = g\left(\hat{f}_i^k, W_{att}\right) = \text{Softmax}\left(\text{MLP}\left(\hat{f}_i^k, W_{att}\right)\right) \quad (2)$$

  These scores act as a soft mask, automatically emphasizing the most important features in the local neighborhood.

- Weighted summation: The final, informative feature vector for the center point p_i, denoted as $\tilde{f}_i$, is computed as the weighted sum of all its neighboring features:

$$\tilde{f}_i = \sum_{k=1}^{K}\left(\hat{f}_i^k . s_i^k\right) \quad (3)$$

The entire RandLA-Net architecture is constructed from computationally efficient building blocks: random sampling and shared MLPs. Unlike other networks that rely on complex or expensive operations (e.g., FPS in PointNet++, graph construction in SPG), RandLA-Net's performance is derived from the repeated application of the simple, lightweight LFA module. This inherent efficiency and modularity make it an ideal candidate for network slimming and parameter reduction.

The representational power of RandLA-Net is rooted in its Local Feature Aggregation (LFA) module, particularly its ability to explicitly encode relative spatial information (Local Spatial Encoding) and aggregate features using an attention mechanism (Attentive Pooling). This mechanism allows the network to focus on the most critical geometric features within a neighborhood, rather than relying solely on the complexity of the feature space dimension.

We hypothesize that because the LFA module is so effective at distilling geometric information, the architecture possesses a "representational buffer" that allows for a reduction in the feature dimensions (*d_out)* without a catastrophic loss in performance. Reducing d_out directly decreases the number of parameters in the MLP weight matrices (Equations (1) and (2)), leading to a significant reduction in the overall model size. This is an architectural slimming method that is more direct, efficient, and easier to implement than complex pruning techniques.

The size of the RandLA-Net model is primarily determined by the number of learnable parameters in the shared MLPs within the LFA modules and the final fully-connected layers. The number of parameters in an MLP layer is a function of its input and output feature dimensions. In the standard RandLA-Net encoder, the per-point feature dimension (*d_out*) progressively increases.

We propose our lightweight model by systematically reducing these output feature dimensions at each layer of the network. For instance, by changing the feature progression to a slower-growing or smaller set of dimensions, we can significantly decrease the number of weights in the MLP matrices in equations (1) and (2). This directly leads to a substantial reduction in the overall model size.

We hypothesize that the LFA module, with its explicit spatial encoding and attentive pooling, is powerful enough to learn discriminative features even in a lower-dimensional space. Therefore, we can achieve a much smaller model footprint, suitable for embedded systems, without a catastrophic loss in segmentation accuracy. The experiments presented in the following section are designed to validate this hypothesis and quantify the resulting trade-offs.

4. RESULTS AND DISCUSSION

We evaluate our models on the S3DIS (Stanford Large-Scale 3D Indoor Spaces) dataset, one of the most common benchmarks for indoor point cloud semantic segmentation. The dataset, collected with a Matterport sensor, consists of dense 3D scans from 6 indoor areas covering over 6,000 m², encompassing 271 rooms. Each point in the cloud is labeled with one of 13 semantic classes (e.g., ceiling, floor, wall, chair). To ensure consistency and direct comparability with prior work, we follow the standard evaluation protocol of focusing our analysis on Area 1. Despite being an indoor environment, the diversity of objects and scene clutter in S3DIS presents a significant challenge, making it well-suited for evaluating the ability of our proposed model to distinguish fine-grained geometric features. We use two primary metrics for segmentation performance: mean Intersection-over-Union (mIoU) and overall Accuracy (Acc). To evaluate suitability for embedded systems, we also measure model size and dataset size. All models were trained and evaluated using the Python programming language with deep learning frameworks.

Table 1. *Parameters of RandLA-Net.*

k_n	16	train_steps	500
num_layers	5	val_steps	100
num_points	*40960*	sub_sampling_ratio	[4, 4, 4, 4, 2]
num_classes	13	*d_out*	*[16, 64, 128, 256, 512]*
sub_grid_size	*0.04*	max_epoch	100
batch_size	6	learning_rate	1e-2

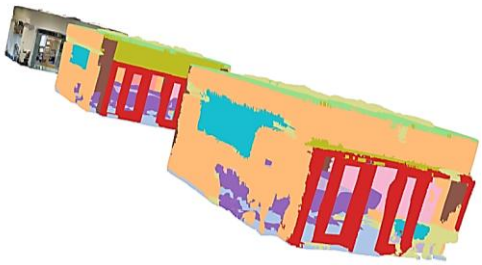

Figure 2. *3D views of the raw input, the ground truth, and the predicted room, shown from a wide to a close-up perspective .*

Main parameters of the RandLA-Net model [4] are listed in table 1. We first compare the performance of the RandLA-Net against two KPConv variants [3]: a rigid version (Light_KPFCNN) and a deformable version (Deform_KPFCNN). Table 2 presents a performance comparison between RandLA-Net and KPConv models. Figure 2 presents a wide to a close-up view of a room's point cloud from the S3DIS dataset, displaying the raw data, the ground truth data, and the corresponding prediction data from RandLA-Net.

Table 2. *Comparison of RandLA-Net and KPConv performance.*

	mIoU (%)	**Acc (%)**	**Model Size (MB)**	**Dataset Size (MB)**
***Light_KPFCNN* [3]**	**88.63**	**98.5**	187.05	687.92
***Deform_KPFCNN* [3]**	**89.13**	**99.5**	202.72	687.92
***RandLA-Net* [4]**	74.13	88.72	**61.4**	**83.9**

As expected, the KPConv models achieve superior segmentation accuracy, with mIoU scores approaching 90%. However, this comes at a very high cost: their model sizes are over 180 MB, and the required dataset size of Area 1 is nearly 700 MB. In contrast, the full RandLA-Net model is significantly more lightweight. Its model size is more than three times smaller, and its dataset size is more than eight times smaller than the KPConv variants. This result clearly establishes the efficiency of the RandLA-Net architecture, although with a noticeable gap in accuracy.

The primary focus of our work is to develop a model that is even more efficient than the standard RandLA-Net (full model). We created a "Light model" by adjusting the network's parameters with *d_out* = [16, 32, 64, 128, 256]. Table 3 compares the performance of this light model to the full model.

Table 3. *Comparison of the full and light model performance.*

	mIoU (%)	**Acc (%)**	**Model size (MB)**	**Dataset size (MB)**
Full model	74.13	**88.72**	61.4	83.9
Light model	***74.76***	86.82	***18.7***	83.9

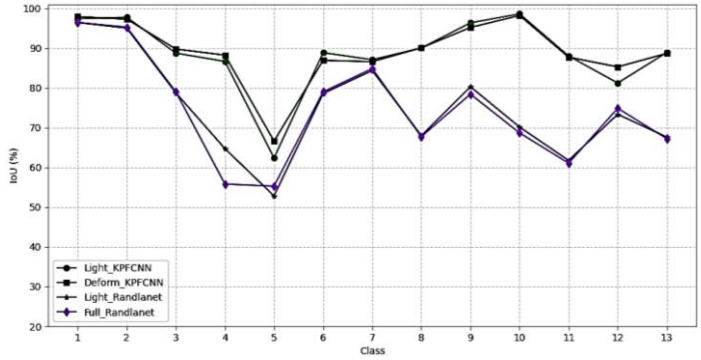

Figure 3. *Comparing the IoU values of the Light_KPFCNN, Deform_KPFCNN, and RandLA-Net networks.*

The results are compelling. Our lightweight RandLA-Net reduces the model size from 61.4 MB to just 18.7 MB - a reduction of nearly 70%. Surprisingly, this dramatic compression results in a negligible degradation of segmentation performance. In fact, the mIoU score is approximately 74%, while the overall accuracy sees only a minor drop of approximately 2%. This demonstrates that the RandLA-Net architecture can be significantly optimized for efficiency with negligible impact on its effectiveness, making the light model an excellent candidate for embedded deployment.

Figure 3 provides a detailed illustration of the per-class IoU performance. As expected, the two KPConv variants achieve superior results across most classes, which is consistent with their architectural design that uses complex kernels to capture fine geometric details. In contrast, RandLA-Net's performance, while lower, remains highly competitive and clearly demonstrates the trade-off between accuracy and computational efficiency - a core philosophy of its architecture. Notably, all models, including KPConv, struggle on certain classes (e.g., class 4, 5), suggesting these challenges are inherent to the data's complexity rather than a specific weakness of one model. Most importantly, the performance curves of our Light model (Light_Randlanet) and the Full model (Full_Randlanet) are nearly identical, providing strong visual evidence for our main conclusion: the RandLA-Net architecture can be significantly compressed with a negligible impact on its segmentation effectiveness, validating its potential for embedded deployment.

For UAVs, it may be necessary to process point clouds of varying densities, either due to sensor limitations or as a compression strategy. We evaluated the robustness of our light model by systematically reducing the number of input points from 40,960 down to 1,024.

Table 4. *Performance of the light RandLA-Net with a decreasing number of input points.*

num_points	***40960***	***8192***	***4096***	***1024***
mIoU (%)	74.77	69.79	64.93	49.07
Acc (%)	86.82	84.51	82.80	63.62

The results in table 4 show a clear trend: segmentation performance is highly dependent on the input point density. As the number of points decreases, both mIoU and accuracy degrade significantly. Dropping from approximately 41k points to approximately 8k points results in a 5% drop in mIoU. Further reduction to just approximately 1k points causes a drastic fall in mIoU to below 50%. This highlights that while the model is efficient, a sufficient point density is crucial for reliable semantic understanding.

Another way to compress the input data is to increase the *sub_grid_size* during preprocessing, which effectively downsamples the point cloud. We tested the effect of increasing this parameter on our light model, with a model size of 18.7 MB and 8,192 input points.

Table 5. *Performance of the light model with varying sub_grid_size.*

sub_grid_size	**mIoU (%)**	**Acc (%)**	**Dataset Size (GB)**
0.04	69.79	84.51	2.36
0.05	69.28	84.49	1.97
0.06	69.88	84.27	1.76

As shown in table 5, increasing the sub_grid_size from 0.04 to 0.06 reduces the input dataset size of S3DIS from 2.36 GB to 1.76 GB, a saving of over 25%. This efficiency gain comes at the cost of an approximately 2.7% drop in mIoU. This experiment presents a clear and practical trade-off for system designers: sacrificing a small amount of accuracy can yield significant savings in dataset size and, consequently, transmission bandwidth and processing load.

5. CONCLUSIONS

This research addressed the critical challenge of deploying semantic segmentation models for LiDAR point clouds on resource-constrained UAVs. We proposed and validated a lightweight deep learning solution based on a modified RandLA-Net architecture.

Our key finding is that the proposed lightweight RandLA-Net provides an exceptional balance between efficiency and performance. With a model size of only 18.7 MB, it is less than one-tenth the size of comparable KPConv models while achieving a respectable mIoU of 74.76%. This result is new and significant, as it demonstrates that substantial model compression can be achieved with minimal to no loss in segmentation accuracy, a crucial factor for real-world applications. The results of this work can be directly applied to the field of autonomous UAV navigation in GNSS-denied environments. By enabling on-board processing of large LiDAR maps, our solution facilitates more intelligent and reactive navigation behaviors.

We acknowledge that a limitation of this study is its evaluation solely on the indoor S3DIS dataset. While this dataset serves as a challenging and standard benchmark, validating the model's performance on large-scale outdoor datasets (e.g., Semantic3D, SemanticKITTI) is a necessary next step to confirm its generalizability for broader UAV applications. However, the results presented in this work successfully validate a critical principle: an exceptional balance between efficiency and accuracy can be achieved on resource-constrained systems by systematically tailoring an architecture, like RandLA-Net, that is already designed for efficiency. This provides a solid foundation for autonomous UAV navigation in structured environments.

Future work should focus on several areas. Further optimization through techniques like network quantization and pruning could reduce the model size and computational cost even more. Additionally, evaluating the model's performance on real-world flight data from a UAV platform

would be the next logical step to validate its practical benefits. Finally, extending the framework to handle dynamic objects and perform instance segmentation would further enhance the navigational capabilities of the UAV.

REFERENCES

[1]. Chang, Y., *"A Review of UAV Autonomous Navigation in GPS-Denied Environments,"* Robotics and Autonomous Systems, Vol. 170, p. 104533 (2023).
[2]. Meng, H.-Y., *"VV-Net: Voxel VAE Net with Group Convolutions,"* in IEEE/CVF International Conference on Computer Vision (ICCV) (2019).
[3]. Guibas, H. T., *"KPConv: Flexible and Deformable Convolution for Point Clouds,"* in IEEE/CVF International Conference on Computer Vision (ICCV), Seoul (2019).
[4]. Hu, Q., *"RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds,"* in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA (2020).
[5]. Yang, B., *"PIXOR: Real-Time 3D Object Detection from Point Clouds,"* in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018).
[6]. Qi, C. R., *"PointNet: Deep Learning on Point Sets for 3D Classification,"* in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017).
[7]. Qi, C. R., *"PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,"* in Advances in Neural Information Processing Systems (NeurIPS) (2017).
[8]. Landrieu, L., *"Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs,"* in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018).
[9]. Armeni, I., *"Joint 2D-3D-Semantic Data for Indoor Scene Understanding,"* in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017).

TÓM TẮT

Giải pháp ứng dụng mạng học sâu nén và phân đoạn ngữ nghĩa cho bản đồ đám mây điểm LiDAR

Việc điều hướng các thiết bị bay không người lái (UAV) trong môi trường không có tín hiệu GNSS thường dựa vào các bản đồ LiDAR được xây dựng sẵn. Tuy nhiên, dung lượng bộ nhớ lớn và độ phức tạp tính toán cao của các bản đồ đám mây điểm này đặt ra những thách thức đáng kể cho các UAV có tài nguyên hạn chế. Bài báo này đề xuất một giải pháp học sâu sử dụng kiến trúc RandLA-Net gọn nhẹ để nén và phân đoạn ngữ nghĩa hiệu quả các bản đồ này. Kết quả cho thấy giải pháp đề xuất giúp giảm về dung lượng mô hình và tập dữ liệu, trong khi vẫn duy trì độ chính xác phân đoạn tương đương với mô hình RandLA-Net cơ sở, qua đó đưa ra một giải pháp khả thi cho việc xử lý thời gian thực trên thiết bị trong các hệ thống nhúng.

Từ khoá: Học sâu, Định vị dẫn đường; Bản đồ đám mây điểm; LiDAR; Phân đoạn ngữ nghĩa.