

GreenSegNet: A Novel Deep Learning Architecture for Urban Vegetation Segmentation From MLS Data

Aditya Aditya^{ID}, *Graduate Student Member, IEEE*, Bharat Lohani^{ID}, Jagannath Aryal^{ID}, *Member, IEEE*, and Stephan Winter^{ID}, *Senior Member, IEEE*

Abstract—Deep learning (DL) models combined with mobile laser scanning (MLS) datasets have demonstrated immense potential for vegetation segmentation. However, restricted performance and inconsistent behavior across datasets by generic DL models offer notable concerns. Furthermore, to capture the characteristic distribution of vegetation points toward effective segregation, a dedicated model for vegetation segmentation is essential. In addition, with curated class-specific DL models being conceptualized, the same is indispensable for vegetation. To address this problem, we propose a novel DL architecture, green segmentation network (GreenSegNet), tailored for vegetation segmentation from MLS point cloud data. Toward a comprehensive assessment, GreenSegNet has been investigated on MLS datasets from three study sites, Chandigarh, Toronto3D, and Kerala. GreenSegNet has illustrated state of the art (SOTA) as well as consistent segmentation performance across all the datasets. GreenSegNet has achieved mean intersection over union (mIoU) as follows: Chandigarh 96.43%, Toronto3D 92.70%, and Kerala 90.16%. In addition, with less than one million parameters, the architecture is the most efficient with respect to the number of parameters among the representative DL models. The associated ablation studies conform to the effectiveness of GreenSegNet. Unlike other SOTA models, GreenSegNet is found robust across different datasets and terrains.

Index Terms—Deep learning (DL) architectures, green segmentation network (GreenSegNet), mobile laser scanning (MLS), point cloud semantic segmentation, urban forests, vegetation points segmentation, vegetation segmentation.

I. INTRODUCTION

VEGETATION is integral to the health and well-being of urban inhabitants. Accurate mapping of vegetation is critical for their upkeep, capacity building, and conservation arrangements. Recently, deep learning (DL) has demonstrated remarkable strides in the computer vision domain, enabling enhanced semantic segmentation capabilities [1], [2], [3], [4].

Manuscript received 5 June 2024; revised 28 August 2024; accepted 2 September 2024. Date of publication 5 September 2024; date of current version 19 September 2024. This work was supported by the Data Science (DS) Research of Frontier and Futuristic Technologies (FFT), Department of Science and Technology, Government of India. (*Corresponding author: Aditya Aditya*)

Aditya Aditya is with the Department of Infrastructure Engineering, University of Melbourne, Parkville, VIC 3010, Australia, and also with the Department of Civil Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India (e-mail: mraditya3646@gmail.com).

Bharat Lohani is with the Department of Civil Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India (e-mail: blohani@iitk.ac.in).

Jagannath Aryal and Stephan Winter are with the Department of Infrastructure Engineering, University of Melbourne, Parkville, VIC 3010, Australia (e-mail: jagannath.aryal@unimelb.edu.au; winter@unimelb.edu.au).

Digital Object Identifier 10.1109/TGRS.2024.3454990

Furthermore, dense point cloud supplied by mobile laser scanning (MLS) systems in urban scenes facilitates holistic vegetation mapping, including individual trees and structural parameters [5]. Accurate vegetation segmentation from MLS dataset is challenging considering the complex structures and configurations exhibited by vegetation points. There is a significant variability within vegetation considering genus and structural composition. Moreover, vegetation points interlaced with other objects, such as buildings, wires, and poles, combined with interclass entanglement, pose notable challenges. Scattered occurrences, and occlusion by other objects, further increase the complexity of the segmentation task.

Heuristic methods [5], [6] or machine learning techniques with handcrafted features [7], [8] have limited applicability toward vegetation segmentation. These methods fail to generalize and involve substantial manual intervention for feature identification. DL models can learn relevant features necessary for segmentation. Accordingly, in a benchmarking study for vegetation segmentation involving representative DL models and multiple MLS datasets, Aditya et al. [9] have recommended specific DL model considering the data characteristics. With regards to leading accuracies, there is a different DL model suggested for each dataset. Considering the inconsistent behavior and restricted accuracies of DL models, the study highlighted the need for a novel DL architecture specifically for vegetation segmentation. In addition, vegetation exhibits a characteristic arrangement of laser points inside canopy, which can be leveraged toward its segregation [9]. This point distribution is fundamental to vegetation and can be exploited by a dedicated DL architecture.

In recent years, there is a surge in the development of curated class-specific DL models [10], [11], [12], [13]. These DL models with binary segmentation can learn relevant contextual features pertaining to a class, leading to its effective segregation from the rest of the data. On similar lines, a vegetation-specific DL model can offer distinct advantages. The model can learn features, such as leaf patterns and distribution, branch structures and orientation, and canopy densities, contributing to a more precise segmentation. Moreover, a dedicated model can easily adapt to structural variations in vegetation, including trees, bushes, and shrubs, offering the possibility of a simpler and lightweight architecture while operating with less computational resources. Therefore, a novel DL model specifically for vegetation segmentation is essential.

In this article, we present a novel DL architecture, green segmentation network (GreenSegNet), tailored for vegetation segmentation from MLS point cloud datasets. GreenSegNet delves into the intrinsic characteristics and nuances of vegetation and extracts discriminative features relevant to vegetation, resulting in enhanced segmentation. Furthermore, the model has illustrated consistent performance across datasets while capturing the fundamental semantics of vegetation toward segmentation. For a comprehensive evaluation, the model has been implemented on MLS datasets from three geographies specifically, Chandigarh [14], Toronto3D [15], and Kerala [14]. Besides exhibiting varying scene complexity and per-point features, the datasets are characterized by multifaceted vegetation along with different proportions of vegetation points. The contributions of our research can be summarized as follows:

- 1) implementation of the PointVector DL model to ascertain the efficacy of vectorized feature representation (VFR) toward capturing the characteristic distribution of vegetation points for segmentation;
- 2) design of a novel DL architecture, GreenSegNet, for vegetation points segmentation from MLS datasets;
- 3) demonstration of the state-of-the-art (SOTA) performance of GreenSegNet on varied datasets;
- 4) finally, demonstration of the most efficient performance with respect to the number of parameters among the representative DL models while exhibiting consistency.

II. RELATED WORKS

Point cloud semantic segmentation is a fundamental task in computer vision particularly in the context of analyzing and understanding 3-D scenes. Multiple DL models have been conceptualized to infer the semantic label for each point. These models can be broadly categorized into projection-based [16], voxel-based [17], and point-based methods [1], [2], [3], [4]. With information loss in projection-based methods and computational complexity observed in voxel-based methods, point-based methods directly operating on raw point cloud and inferring per-point semantics are preferred. Point-based methods can be further categorized based on the employed feature extraction mechanism.

A. Pointwise Learning Based on MLP or Convolution or a Combination of Both

These methods use multilayer perceptron (MLP) or convolution or a combination of both for mapping features to a high-dimensional embedding space. The first point-based network was pioneered by PointNet [18]. It uses shared MLP to infer per-point semantics. To leverage the local structure of point cloud, PointNet++ [19] was proposed. To process large-scale point cloud in a single pass, RandLANet [1] has been proposed with random sampling and local feature aggregation module. Furthermore, SCFNet [2] was proposed to handle the orientation-sensitive features. PointNeXt [20] adopted an inverted residual MLP block combined with improved training and scaling strategies. Inspired by the image-based convolution, KPConv [3] directly applies convolution operation on the

point cloud. It utilizes a variable set of kernel points, located uniformly in space within a spherical domain, for carrying the convolutional weights. To extend the convolution operation on unordered point clouds, PointCNN [4] was proposed. It employs an X-Conv operator to simultaneously weigh and permute the input features associated with the points. X-Conv uses both MLP and convolution operations for feature learning.

B. Pointwise Learning Based on Graph Convolution

These methods employ graph convolutional networks (GCNs) [21] to effectively leverage the geometric correlations between points. To enrich the topological representation, EdgeConv [22] was proposed. It learns the edge embeddings by constructing local graph and describes the point–neighbor relationship. Du et al. [23] proposed a local–global graph convolutional method to extract relevant contextual features. To capture the diverse relationship between points, Wei et al. [24] proposed AGConv. It generates adaptive kernels to establish relationships between point pairs, thus improving the flexibility of point cloud convolutions. PointMetaBase [25] derived critical insights on the building blocks of DL models and conceptualized a PointMetaBase block by integrating relevant components, including neighbor update, neighbor aggregation, point update, and position embedding.

C. Pointwise Learning Based on Attention Mechanism

These methods utilize attention mechanism [26] or transformers for extracting global contextual information by modeling long-term dependencies. Point transformer [27] uses vector self-attention and position information in a localized manner, enabling scalability to large scenes with enhanced segmentation. SPoTr [28] has been proposed to reduce the computational complexity associated with transformers. It utilizes a set of adaptively located self-positioning points for global cross attention. On similar lines, PointNAT [29] employs some learned key points for modeling global features. In addition, a hybrid block favoring localization or globalization has been proposed.

With significant advancement in generic DL models, recent years also saw the onset of class-specific or scene-specific DL models. Jiang et al. [10] proposed RailSeg for railway point cloud semantic segmentation. Zhang et al. [11] introduced UnrollingNet to conduct a multilabel segmentation from 3-D point clouds of tunnels including seepage. Li et al. [12] developed PlantNet for plant species detection, while Yu et al. [13] proposed PowerLine-Net for semantic segmentation of extra high-voltage transmission lines. Similarly, a class-specific DL model is required for improved accuracy and robustness across varied datasets for vegetation segmentation.

There are several studies on vegetation detection in urban landscapes. Some of the studies have employed handcrafted features. You et al. [6] used planer projection and point density to filter objects, before utilizing Euclidean cluster extraction algorithm for vegetation extraction. Fan et al. [7] extracted individual trees from MLS using a combination of clustering and random forest technique. Other studies have used machine

learning techniques with restricted feature learning capabilities. Schmohl et al. [30] proposed a deep single-shot object detection network to extract trees from aerial laser scanner (ALS) data. The network utilizes a convolutional backbone for feature extraction. A supervoxels-based method [31] has been proposed to identify street trees while automatically extracting and classifying pole-like objects from MLS point cloud. Singh and Yadav [32] introduced the 3D-MFDNN method for vegetation segmentation from ALS data utilizing 3-D multifeature descriptors. Li et al. [8] have designed a two-step strategy for tree segmentation from MLS using statistical features and discrete AdaBoost algorithm. These studies lack the necessary ingredients to capture complexity with limited robustness and adaptability.

In recent years, there are some DL-based studies as well. However, they utilize generic DL networks as part of segmentation procedure. Wang et al. [33] developed a method for extracting street trees from MLS based on morphological analysis. They have used RandLANet [1] for segmentation with vegetation as one of the classes. Yang et al. [34] performed tree points extraction using point transformer [27] on multispectral LiDAR. Wang et al. [35] extract road-side trees from vehicle-mounted MLS point clouds by employing a graph-based semantic segmentation network. Zhang et al. [36] performed urban vegetation volume analysis by using PointCNN [4]. Luo et al. [37] have used MS-RRFSegNet for segmenting tree points in the MLS dataset. The existing studies are unable to learn contextual features with restricted accuracies, thus reinforcing the need of a dedicated DL model for vegetation segmentation.

III. PROPOSED METHOD

GreenSegNet learns effective features leading to enhanced segmentation of vegetation points. GreenSegNet utilizes a hierarchical structure [20] to capture both local and global information. The architecture consists of an encoder and a decoder. The encoder consists of four stages for segmentation [1, 2, 3, 2]. The first stage describes the stem MLP, while the rest are composed of one SetAbstraction (SA) block appended by GreenBlock (GB) as indicated. The encoder facilitates variable receptive fields at different layers for better neighbor representation. The decoder part hierarchically propagates features from subsampled points to original points. It is inspired from [20] and employs inverse distance weighted average incorporating k nearest neighbors. Furthermore, skip links are used after each level before using fully connected MLPs for updating each point's feature. The architecture of GreenSegNet is illustrated in Fig. 1.

A. SetAbstraction

The SA module downsamples the incoming points using farthest point sampling (FPS) while aggregating the neighboring features using ball query [20]. A fixed region mandated by ball query allows it to generalize across stages, thus positioning it better than k nearest neighbor (kNN). It employs shared MLPs, grouping layer, convolutional layer, and a reduction layer. The

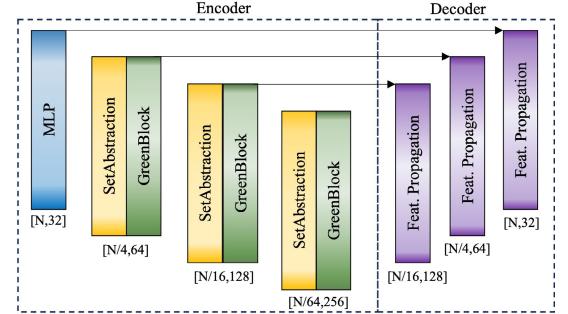


Fig. 1. GreenSegNet architecture. SA is appended by GB at each stage.

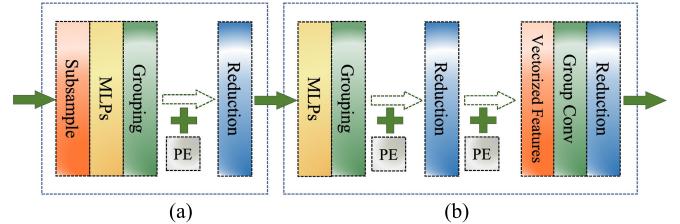


Fig. 2. Components of (a) SA and (b) GB. PE stands for position embedding.

components of the SA block (Fig. 2), motivated by Qian et al. [20] and Lin et al. [25], can be described as follows:

$$f_i^{m^1} = \text{MLP}(f_i) \quad (1)$$

$$p_{N(i)}, f_{N(i)} = \text{Group}\left(\left(p_j - p_i, f_i^{m^1}\right) | j \in N(i)\right) \quad (2)$$

where $p_i \in \mathbb{R}^3$ and $f_i \in \mathbb{R}^d$ represent the coordinates and features of point i , respectively, and d indicates the feature length. Let $f_i^{m^1}$, $N(i)$, and p_j denote the derived features, the neighbors of point i , and coordinates of each neighbor of point i , respectively. $p_{N(i)}$ and $f_{N(i)}$ represent the grouped positions and grouped features for the neighbors of point i , respectively. Furthermore, $\text{pe}_{N(i)}$ represents the positional embeddings for the neighbors of point i , while C_1 and R_1 indicate the convolution and max-pooling reduction operations, respectively,

$$\text{pe}_{N(i)} = C_1(p_{N(i)}) \quad (3)$$

$$f_i^{m^2} = R_1(\text{pe}_{N(i)} + f_{N(i)}). \quad (4)$$

SA module encodes the input features to a higher dimension using shared MLPs (1) followed by grouping using ball query and 32 neighbors (2) [20]. Positional embeddings are calculated using the grouped positions (3). Next, the intermediate features are determined by concatenating the position embeddings with grouped features, thus avoiding the dimensional mismatch between positions and features. Finally, the features are aggregated using max-pooling reduction layer (4).

B. GreenBlock

GB extracts the discriminating contextual features by exploiting the unique semantic characteristic of vegetation points. It enables appropriate feature learning capabilities by integrating positional embeddings, grouped features, and vectorized representation of features. Vectors, combined with

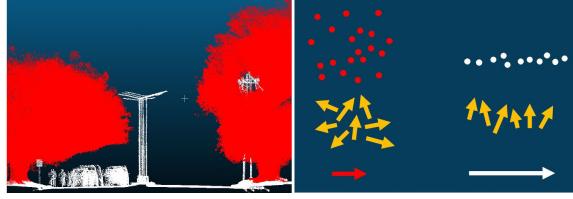


Fig. 3. Aggregation of vectors for capturing the characteristic distribution of vegetation points. Red and white points represent vegetation and nonvegetation classes, respectively. Aggregated vector of points representing vegetation will be comparatively smaller than that of nonvegetation points.

magnitude and direction, are more expressive than scalars toward exhibiting anisotropy. The block utilizes 3-D vectors [38], [39] with 3 degrees of freedom enabling the same number of independent variations in the parameters defining the features. Furthermore, vector representation combined with sum aggregation can extract rich contextual information in relation to the characteristic arrangement of vegetation points. The resulting vector for vegetation points will be comparatively smaller than that of nonvegetation points, thus enabling better segmentation. The components of the GB (Fig. 2) can be formulated as follows:

$$f_i^{m^3} = \text{MLP}(f_i^{m^2}) \quad (5)$$

$$p'_{N(i)}, f'_{N(i)} = \text{Group}\left(\left(p_j, f_j^{m^3}\right) | j \in N(i)\right) \quad (6)$$

$$f_i^{m^4} = R_1(p_{N(i)} + f'_{N(i)}) \quad (7)$$

$$f_i^{m^5} = f_i^{m^4} + p'_{N(i)} \quad (8)$$

$$\theta_1, \theta_2, l = \text{Conv}\left(f_i^{m^5}\right) \quad (9)$$

$$\text{lin} = \text{Cat}(l.\sin\theta_1.\sin\theta_2, l.\cos\theta_1.\sin\theta_2, l.\cos\theta_2) \quad (10)$$

$$f_i^{m^6} = R_2(\text{lin}) \quad (11)$$

$$f_i^{m^7} = \text{GroupConv}\left(f_i^{m^6}\right). \quad (12)$$

In GB, incoming features from SA block are taken to a higher dimension before grouping (5). The features are aggregated after combining position embeddings [27] and grouped features (7) [25]. This ascertains appropriate features realization. Furthermore, to enhance the representation of spatial information in the learned features, the aggregated features are again concatenated with position embeddings (8) [27]. This ensures that only effective features from the pool of position and intermediate features are going forward. Transformation variables, such as the angles including θ_1 and θ_2 , and the linear l , are learned by employing convolutions (9) [39]. The transformation is applied on the intermediate features to obtain the vectorized features (10). Subsequently, the aggregation is performed using the sum operation for all points (11). This enables the model to learn effective features by capturing the characteristic distribution of vegetation points (Fig. 3). In the next step, the vectors are again converted into scalars using group convolutions operating simultaneously on all dimensions (12). Finally, the features are concatenated with incoming features.

IV. EXPERIMENTS AND RESULTS

GreenSegNet has been investigated on MLS datasets from three sites, namely, Chandigarh, Toronto3D [15], and Kerala

(Fig. 4). These sites are chosen considering the geospatial variability and data diversity, so results are conclusive across datasets. The datasets vary in terms of geographical locations, scene complexity, point density, number of attributes, vegetative diversity, and class composition. Chandigarh and Toronto3D datasets have relatively less scene complexity in comparison with the Kerala dataset, owing to a systematic and coordinated urban environment. Toronto3D dataset has additional color information compared with Chandigarh and Kerala datasets. Kerala dataset has the highest point density and vegetation composition, followed by Chandigarh and Toronto3D datasets, in order. Toronto3D dataset has significantly low proportion of vegetation points, critical to accommodate class imbalance conditions. Chandigarh and Kerala datasets from northern and southern parts of India, respectively, and Toronto3D dataset from Canada present necessary topographical and vegetative diversity.

A. Datasets

1) *Chandigarh*: Chandigarh dataset has been captured from the city of Chandigarh, India, with Riegl VQ-450 by Geokno India Private Ltd. The original data have been manually labeled into two hierarchical levels, with 67 classes in level 1 and varying subclasses in level 2 [14]. For this study, the dataset has been employed for around 1 km with 195 million points. Each point has five attributes, including X, Y, Z, intensity (I), and class label. These attributes are also referred to as original per-point features in this article. The dataset exhibits low level of scene complexity owing to a formally planned urban environment combined with homogeneous vegetation. There is limited vegetative diversity, both at structural as well as spatial levels. The dataset describes a systematic environment with vegetation occurring almost uniformly on both sides of the roads. There are street lights and electric poles with powerlines along the road. Vegetation points account for around 37.32% of the total points.

2) *Toronto3D*: Toronto3D [15] is a benchmark point cloud dataset acquired from the city of Toronto, Canada. The data have been captured for around 1 km by vehicle-mounted MLS system (Teledyne Optech Maverick). It consists of around 78 million points and comes labeled into eight classes, such as road, building, natural, road marking, car, fence, utility line, and unclassified points. Combined with additional color information, each point has eight attributes, including X, Y, Z, I, R, G, B, and class label. The dataset illustrates a developed and coordinated urban road environment packed up with buildings and intermittent trees. However, with low percentage and scattered appearance of vegetation, the dataset is quite challenging for segmentation tasks. Vegetation points constitute around 11.73% of the total points.

3) *Kerala*: Kerala dataset has been acquired from Nellanaid village in the Thiruvananthapuram district of Kerala, India, with Riegl VQ-450 mounted on an SUV by Geokno India Private Ltd. The original data come manually labeled into 67 classes along with object instances [14]. For this study, the employed data stretch for around 1 km with nearly 222 million points. Each point comes with five associated attributes, including X, Y, Z, I, and class label. The dataset

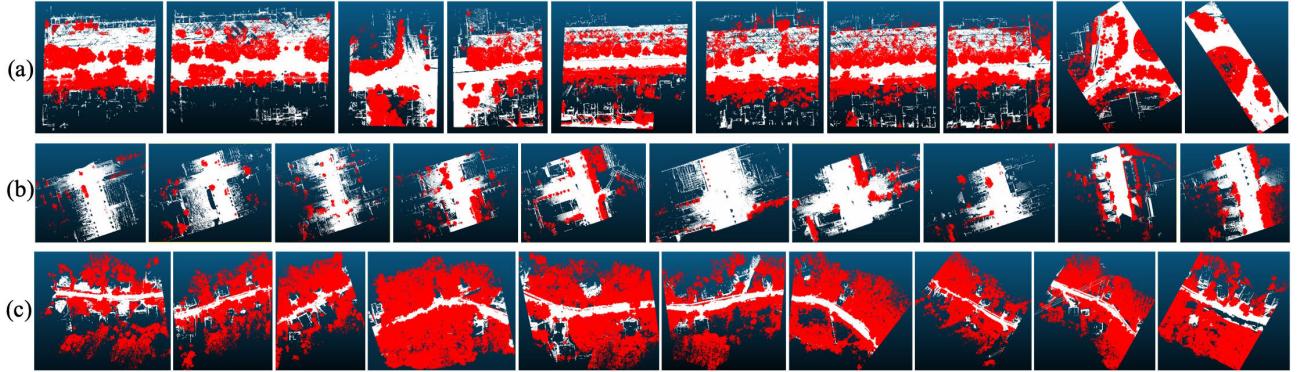


Fig. 4. Ten tiles of different datasets. (a) Chandigarh, (b) Toronto3D, and (c) Kerala. Red and white points represent vegetation and nonvegetation classes, respectively.

represents an environment dominated by dense vegetation with numerous object categories. The dataset exhibits high scene complexity owing to a large number of object categories combined with high variability within these object categories as well. Furthermore, it describes multifaceted vegetation both in terms of genus and anatomical characteristics, which are intricately blended. There are multiple instances of electric poles covered with vegetation and electric wires passing through the vegetation. Vegetation points constitute around 60.75% of the total points.

B. Evaluation Metrics

Overall accuracy (OA) (see 13), intersection over union (IoU) (see 14), and mean IoU (mIoU) (see 15) are used to assess the performance of models, as shown below. To account for class imbalance and compensate for incorrect segmentation, IoU has been used [9]

$$OA = \frac{TP + TN}{TP + FP + TN + FN} \quad (13)$$

$$IoU_i = \frac{TP_i}{TP_i + FP_i + FN_i} \quad (14)$$

$$mIoU = \sum_{i=1}^n \left(\frac{TP_i}{TP_i + FP_i + FN_i} \right) / n \quad (15)$$

where TP, TN, FP, and FN indicate the true positive, true negative, false positive, and false negative, respectively. i represents a particular class, while n denotes the number of classes, which is two in this case, i.e., vegetation and nonvegetation.

C. Methodology and Implementation Details

The experimentation is in three steps. First, the datasets are relabeled to represent only two classes, namely, vegetation and nonvegetation. Next, the datasets are split into ten tiles for training and testing purposes. In a single experiment, one tile is used for testing, while the rest nine tiles are employed for training and validation. Finally, GreenSegNet has been tested in a tenfold cross-validation mode to enhance the reliability of results. The model has been run ten times on every dataset, with each tile as testing data and the remaining ones as training data (Fig. 5). The reported result is an average of these ten experimental outcomes. The experiments have been

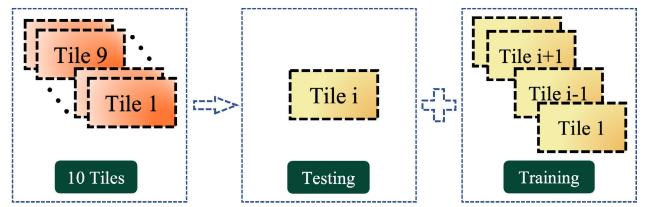


Fig. 5. Tenfold cross-validation mode of experimentation with ten tiles. Between every model and dataset, there are ten experiments. In an experiment, one tile acts as the testing data, while the remaining ones are taken as training data. i can assume integral values from 1 to 10.

performed on PyTorch framework utilizing GPU NVIDIA Tesla V100 (32 GB) and processor Intel Xeon Gold 5218 2.30 GHz \times 32. Other relevant implementation details are the following: epochs 100, batch size 4, optimizer AdamW, and a learning rate 0.01 with a weight decay of 0.0001. With regards to ablation studies, PointVector [39] has also been implemented in an identical manner.

D. Semantic Segmentation Results by Datasets

This section describes the dataset-wise segmentation results (Table I) of GreenSegNet. The section further outlines the dataset portions contributing to enhanced performance of GreenSegNet over the recommended DL models [9]. Fig. 6 shows the experimental setwise results of GreenSegNet. GreenSegNet has been compared with models from each category of point-based methods, including those utilizing just MLPs, convolutions, or a combination of both. Data characteristics, such as the proportion of vegetation points, scene complexity, object categories, and vegetation diversity, have been considered to assess the behavior of GreenSegNet. Overall, scene complexity appears to be one of the most important factors dictating the behavior of model. It is largely described by the number of different object categories and the associated interwovenness. Furthermore, it accounts for the variation within an object category as well. Chandigarh and Toronto3D datasets exhibiting relatively less complicated scenes demonstrate better mIoUs than those of the Kerala dataset.

1) *Chandigarh*: GreenSegNet has demonstrated the finest accuracies with an mIoU of 96.43%. The improvement in mIoU over the best performing model [9], PointMetaBase [25],

TABLE I

BINARY SEGMENTATION RESULTS [9]: AVERAGED TENFOLD CROSS-VALIDATED VALUES. VALUES ARE IN PERCENTAGE.
 () VALUES ARE STANDARD DEVIATIONS
 ASSOCIATED WITH GREENSEGNET

	Models	mIoU	OA	IoU Veg	IoU N-Veg
Chandigarh	PointCNN	93.32	96.69	91.70	94.93
	KPConv	89.16	94.28	87.03	91.28
	RandLANet	92.08	96.03	90.37	93.79
	SCFNet	89.47	94.52	87.33	91.63
	PointNeXt	80.80	89.29	77.45	84.14
	SPoTr	78.88	88.89	73.94	83.82
	PointMetaBase	95.24	97.78	93.89	96.59
	PointVector	96.13	98.19	95.08	97.18
	GreenSegNet (Std. Dev)	96.43 (3.37)	98.30 (1.69)	95.52 (4.09)	97.34 (2.72)
Toronto3D	PointCNN	89.22	97.93	80.87	97.56
	KPConv	91.26	97.94	84.99	97.52
	RandLANet	88.98	97.78	80.54	97.43
	SCFNet	85.61	96.14	75.59	95.62
	PointNeXt	87.05	96.95	77.58	96.52
	SPoTr	60.32	91.17	29.76	90.89
	PointMetaBase	88.69	97.37	80.54	96.85
	PointVector	89.99	97.73	82.70	97.27
	GreenSegNet (Std. Dev)	92.70 (3.47)	98.67 (1.05)	86.96 (6.70)	98.43 (1.29)
Kerala	PointCNN	85.68	92.66	88.59	82.77
	KPConv	81.14	90.18	85.60	76.68
	RandLANet	78.64	88.88	84.13	73.15
	SCFNet	74.25	86.21	81.32	67.17
	PointNeXt	69.38	83.51	78.21	60.54
	SPoTr	82.71	91.11	86.34	79.09
	PointMetaBase	84.60	92.05	87.57	81.63
	PointVector	86.62	93.31	89.72	83.53
	GreenSegNet (Std. Dev)	90.16 (3.26)	95.15 (1.60)	92.34 (2.44)	87.98 (4.60)

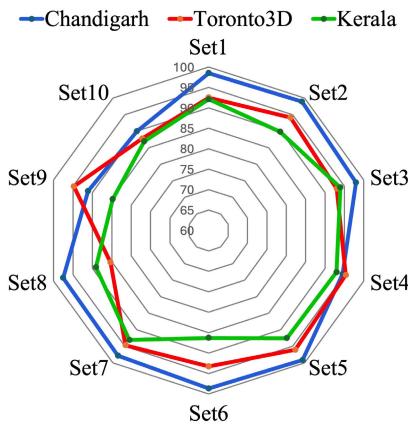


Fig. 6. Experimental setwise mIoUs of GreenSegNet.

for the dataset is 1.19%. The improvement is coming from areas near settlements, electric poles, and at the interface of tree trunks and ground. As indicated in the rightmost zoomed-in portion of Fig. 7, GreenSegNet has been able to differentiate low vegetation and ground. Even in regions located far from the sensor, the model is showing impressive results. Best performing SOTA model, PointMetaBase, is showing deviations in these regions. This can be attributed to the relevant contextual information extracted by ingesting

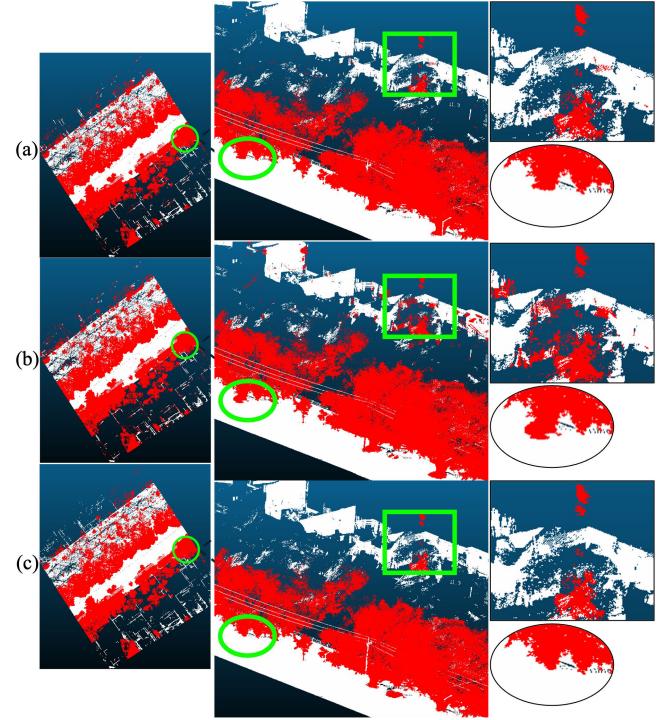


Fig. 7. Segmentation results on a single tile from Chandigarh dataset. Red and white points represent vegetation and nonvegetation classes, respectively. (a) GreenSegNet, (b) PointMetaBase, and (c) ground truth. Three levels of zoom (highlighted in green) from left to right to compare the model results.

positional embeddings at various stages and sum aggregation of VFR. With the classes, low vegetation, and ground sharing almost similar point distribution, vectorized features have limited contribution owing to minimal canopy penetration. However, position embeddings at different scales seem to capture relevant semantics for segmentation. However, there is some aberration in detecting small patches of vegetation points lying at a distance.

2) *Toronto3D*: With 92.70% mIoU, GreenSegNet has produced the leading accuracy on the dataset [15]. The improvement over the recommended model [9], KPConv (omni-supervised) [40], for the dataset is 1.44%. The improvement is mostly observed at the interface of vegetation points. As indicated in the leftmost zoomed-in portion of Fig. 8, GreenSegNet has been able to detect vegetation effectively, while the KPConv is showing significant aberration compared with reference data. This can be attributed to the vectorization of only effective features from the pool of position and grouped features, leading to enhanced segmentation. Yet, there are portions where the model has been unable to detect low vegetation.

3) *Kerala*: GreenSegNet has attained an mIoU of 90.16% on the Kerala dataset outperforming representative DL models by a significant margin. The increment in mIoU over the recommended DL model [9] PointCNN [4] is 4.48%. The key to the improved performance of GreenSegNet over PointCNN is its ability to segregate low vegetation from the ground points. With these portions lying close to the sensor characterizing high point density, they have a considerable impact on the accuracy. A zoomed-in portion of Fig. 9 shows the efficacy of GreenSegNet over PointCNN. With PointCNN,



Fig. 8. Segmentation results on a single tile from Toronto3D dataset. Red and white points represent vegetation and nonvegetation classes, respectively. (a) GreenSegNet, (b) KPConv, and (c) ground truth. Three levels of zoom from right to left (highlighted in green) to compare the model results.

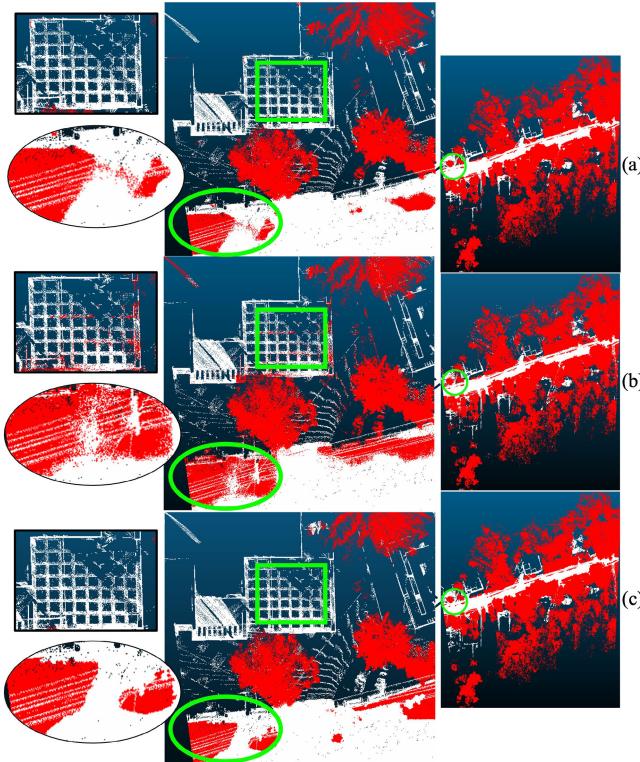


Fig. 9. Segmentation results on a single tile from Kerala dataset. Red and white points represent vegetation and nonvegetation classes, respectively. (a) GreenSegNet, (b) PointCNN, and (c) ground truth. Three levels of zoom from right to left (highlighted in green) to compare the model results.

red patches can be observed in nonvegetation points, while GreenSegNet is close to the reference data. This can be attributed to the dense vegetation contributing to the efficacy

of vector representation combined with variable receptive fields. Dense point cloud enables significant variation in feature vector magnitudes for the two classes along with fine-grained features learning, leading to effective vegetation segregation. Still, portions, such as the lower part of tree trunks with similar diameters as poles, powerlines passing through dense vegetation, and poles covered partly with vegetation, pose significant challenges to the model.

V. DISCUSSION

GreenSegNet has achieved leading accuracies across datasets (Table I). With datasets characterized by a wide range of scenarios with varying scene complexity and vegetation diversity, changing proportion of vegetation points and available attributes while representing different geographical interfaces and sensor characteristics, GreenSegNet has translated itself as the best-suited DL architecture for vegetation segmentation from MLS data. By leveraging spatial information and vectorizing the rich contextual features, GreenSegNet has been able to capture relevant distinguishing features. Feature vectors of vegetation are aligned in all directions as opposed to other dominant classes, such as ground, pavements, and buildings, where the vectors are aligned in a particular direction (Fig. 3). The resulting vector of vegetation has a smaller magnitude than that of nonvegetation. This is also evident from the qualitative results. The model's ability to segregate low vegetation and ground on the Kerala dataset, and demarcation of clear boundaries between vegetation and non-vegetation in case of Toronto3D dataset, is a testimony of rich feature learning describing the semantic relationship of points.

A. Challenging Regions

Although GreenSegNet has achieved notable and improved segmentation performance in comparison with other SOTA DL models, there are several challenging portions (Fig. 10). In Chandigarh dataset with a typical urban environment, tree trunks have been segmented as nonvegetation. These inconsistencies have been observed particularly in areas away from the sensor with low point density [Fig. 10(a) and (b)]. This can be attributed to multiple instances of pole-like objects in the dataset sharing similar geometric properties as tree trunks. In addition, low point density restricts the extraction of fine-grained features, contributing to ambiguity. From an architectural perspective, tree trunks are classified as vegetation through their association with the canopy. Tree trunks themselves do not take any advantage of the vector representation of points. In case of Toronto3D dataset, segmentation discrepancies have been observed on points located far from the sensor. Vegetation points have been segmented as nonvegetation points [Fig. 10(c)]. This can be associated with higher proportion of nonvegetation points. Sparse points and isolated patches present limited information with less meaningful features to be considered as vegetation. For Kerala dataset characterizing dense vegetation, irregularities have occurred in detecting pole-like objects in the thick of vegetation [Fig. 10(d)]. Poles have been segmented as vegetation. With these poles exhibiting similar geometrical and spatial

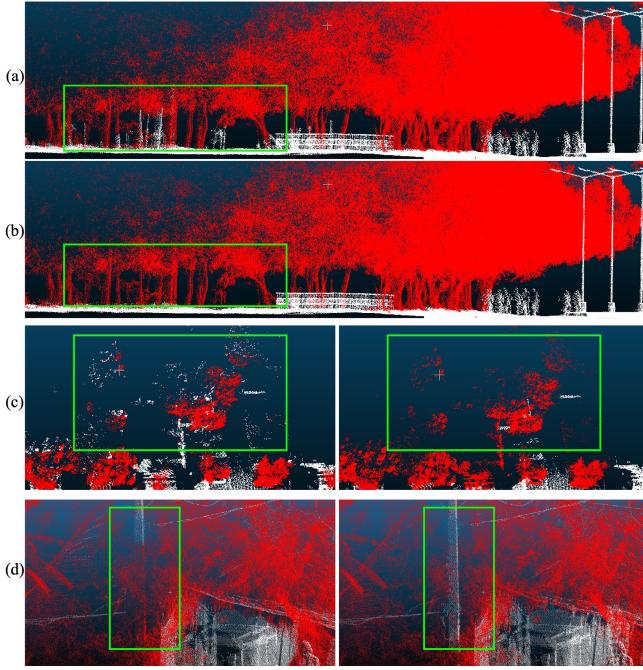


Fig. 10. Segmentation errors. (a) Chandigarh (GreenSegNet), (b) Chandigarh (ground truth), (c) Toronto3D (showing top view, left: GreenSegNet and right: ground truth), and (d) Kerala (left: GreenSegNet and right: ground truth).

characteristics as trunks or branches, boundaries between poles and surrounding foliage have become blurred. Overlapping objects, proximity, and occlusion further contribute to such inaccuracies. Again from an architectural perspective, with the canopy above the poles, the poles are being segmented as vegetation, i.e., specifically as tree trunks.

B. Parameters and FLOPs

GreenSegNet is the most efficient among the representative DL models [9] with respect to the number of parameters (Table II). The associated parameters and billion floating point operations (GFLOPs) are 0.91 million and 5.01, respectively. Earlier, RandLANet [1] was the most efficient one with 1.2 million parameters. GreenSegNet is second to PointMetaBase in terms of number of FLOPs and throughput.

Parameter count relates to the number of trainable parameters in a DL model. Fewer parameters correspond to less memory usage and faster training. This enables GreenSegNet to be deployed in resource-constrained environments and embedded systems. Moreover, the model processing translates into low operational costs, small runtime, and larger spatial coverage in terms of scalability. However, with the employment of group convolutions resulting in longer inference time, GreenSegNet may not be preferable for real-time applications in its current form.

C. Ablation Studies

1) *Comparison With PointVector*: With GreenSegNet employing VFR, a comparison with PointVector [39] is important. GreenSegNet has outperformed PointVector on all the datasets (Table I). PointVector has been partly successful in capturing the differentiating pattern of vegetation points,

TABLE II

COMPUTATIONAL EFFICIENCY OF ARCHITECTURES. THE VALUES ARE COMPUTED FROM KERALA DATA WITH A BATCH SIZE OF 16 EXCEPT *

Models	Parameters (m)	GFLOPs	Throughput (ins./s)
RandLANet *	1.20	5.8	-
PointMetaBase-L	2.74	1.97	207.03
PointNeXt-L	7.13	15.23	179.68
SPoTr	66.38	232.86	46.32
PointVector-L	4.22	10.74	167.30
GreenSegNet	0.91	5.01	202.87

TABLE III

ABLATION STUDIES: VALUES ARE IN PERCENTAGE AND FROM A SINGLE EXPERIMENT. GB: GREENBLOCK, PE: POSITION EMBEDDINGS, LA: LOCAL AGGREGATION, AND VFR: VECTORIZED FEATURE REPRESENTATION

	Models	mIoU	IoU Veg	IoU N-Veg
Chandigarh	PE in GB	96.12	95.11	97.13
	W/O PE in LA	96.53	95.62	97.43
	W/O PE in GB	92.06	90.21	93.9
	W/O VFR in GB	95.62	94.49	96.75
	GreenSegNet	97.55	96.90	98.20
Toronto3D	PE in GB	87.10	79.21	94.99
	W/O PE in LA	83.94	74.68	93.19
	W/O PE in GB	79.05	67.87	90.23
	W/O VFR in GB	78.88	67.23	90.52
	GreenSegNet	87.77	80.12	95.43
Kerala	PE in GB	91.74	94.39	89.09
	W/O PE in LA	92.04	94.44	89.64
	W/O PE in GB	90.67	93.56	87.79
	W/O VFR in GB	90.28	93.32	87.24
	GreenSegNet	92.58	94.85	90.31
Toronto3D	W/O color info	93.65	89.04	98.25
	With color info	94.11	89.81	98.41

outperforming the recommended models on Chandigarh and Kerala datasets, while attaining the second best results after KPConv on the Toronto3D dataset. The improvement is significant on the Kerala dataset considering the data complexity and associated augmentation headroom.

2) *Comparison With Architectural Components*: GreenSegNet has been conceptualized after careful assimilation of different components (Table III). First, case of comparison is where the position embeddings are determined in the GB and concatenated with neighboring features (PE in GB). Second case of comparison is without using position embeddings for initial feature aggregation in GB (W/O PE in LA). Third case of comparison is without using position embeddings at all in the GB (W/O PE in GB). Final case of comparison is without using the VFR in the GB (W/O VFR in GB). The results clearly indicate the improved performance with the adopted configuration of GreenSegNet architecture. As evident, position embeddings and VFR are critical components of GB and significantly augment the accuracy.

3) *Variation With Color Attributes*: To demonstrate the impact of additional color information on the performance, GreenSegNet has been implemented on the Toronto3D dataset without color attributes (Table III). The results are indicative of superior performance with additional color information.

Furthermore, the results illustrate the capability of GreenSegNet to capture the additional color information toward enhanced segmentation.

VI. CONCLUSION

Accurate vegetation mapping is fundamental for proper management strategies of urban forests. MLS datasets combined with DL techniques offer necessary framework for urban vegetation segmentation. However, accurate mapping of vegetation is challenging due to the inherent complexity and variability of vegetation, dynamic nature, and interwovenness with man-made structures. Furthermore, varying point cloud density, presence of noise, isolated patches, and occlusions introduce additional issues. Restricted accuracies and inconsistent behavior illustrated by generic DL models necessitate a dedicated DL model for vegetation segmentation. Furthermore, a dedicated model can leverage the unique spatial distribution of points inside the canopy toward segmentation. Moreover, the development of curated class-specific DL models further reinforces the need for such a dedicated model.

To address this need, we proposed a novel DL architecture, GreenSegNet, for vegetation segmentation from MLS point cloud data. GreenSegNet has demonstrated SOTA performance across multiple MLS datasets, surpassing recommended models by a significant margin. The model has shown robust as well as consistent results across datasets. With careful assimilation of necessary ingredients, such as VFR, positional embeddings, and aggregation techniques, GreenSegNet has been effective in leveraging the discriminative features. The model has been tested on MLS datasets from three geographies for a comprehensive evaluation. The improvements in mIoU over the best performing models are 1.19%, 1.44%, and 4.48% for Chandigarh, Toronto3D, and Kerala datasets, respectively.

In addition to exhibiting notable accuracies and robustness, GreenSegNet has also been very fast and lightweight with just 0.91 million parameters. The model extracts necessary contextual features by recognizing the complicated structures and appearances of vegetation while underpinning its relationship with surrounding objects, thus translating enhanced segmentation performance. Furthermore, it captures the characteristic distribution of vegetation points toward segmentation.

GreenSegNet has a wide range of applications. In the context of urban green space, GreenSegNet enables precise vegetation mapping. This is vital for managing urban forests, assessing tree health, and supporting sustainable development initiatives. In infrastructure monitoring and management, GreenSegNet assists in the maintenance of utilities and transportation networks across large areas. Vegetation spread on powerlines, roads, and railways can be effectively detected, enabling timely interventions and avoiding operational hazards. GreenSegNet facilitates various downstream and derived studies, such as individual tree segmentation, carbon sequestration, urban heat islands, and air quality. These studies have been mandated by various international conventions as well. The model aids in the accurate representation of trees in digital twins and digital urban inventories. Overall, GreenSegNet enables more informed decision-making in complex urban landscapes. GreenSegNet will be an essential tool for urban

planners, utility service providers, conservationists, governing bodies, and the local community.

Among all notable capabilities, GreenSegNet has some limitations. The model has a higher inference time on account of group convolutions, rendering it less preferable for real-time applications. Some studies can focus on alternatives of group convolutions. It may be noted that GreenSegNet is conceptualized for enhanced vegetation segmentation from MLS datasets. It will just segregate vegetation points from the rest of the data, without any species-level distinction. The model may be used for forestry surveys only to obtain vegetation points. Methods for individual tree segmentation and morphological parameters will follow. Future research can also focus on the extension of GreenSegNet to aerial and terrestrial LiDAR data. With class-specific segmentation getting traction, the study provides the necessary framework for exploiting class-specific characteristics and appropriate feature representation resulting in effective segmentation.

ACKNOWLEDGMENT

The authors acknowledge Geokno India Private Ltd. and Chandigarh Administration for data from LiDAVerse.com.

REFERENCES

- [1] Q. Hu et al., "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11105–11114.
- [2] S. Fan, Q. Dong, F. Zhu, Y. Lv, P. Ye, and F.-Y. Wang, "SCF-Net: Learning spatial contextual features for large-scale point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14499–14508.
- [3] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPCConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6410–6419.
- [4] Y. Li et al., "PointCNN: Convolution on X-transformed points," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
- [5] Y. Zhao, Q. Hu, H. Li, S. Wang, and M. Ai, "Evaluating carbon sequestration and PM_{2.5} removal of urban street trees using mobile laser scanning data," *Remote Sens.*, vol. 10, no. 11, p. 1759, Nov. 2018.
- [6] H. You, S. Li, Y. Xu, Z. He, and D. Wang, "Tree extraction from airborne laser scanning data in urban areas," *Remote Sens.*, vol. 13, no. 17, p. 3428, Aug. 2021.
- [7] W. Fan, B. Yang, F. Liang, and Z. Dong, "Using mobile laser scanning point clouds to extract urban roadside trees for ecological benefits estimation," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 211–216, Aug. 2020.
- [8] Q. Li, P. Yuan, X. Liu, and H. Zhou, "Street tree segmentation from mobile laser scanning data," *Int. J. Remote Sens.*, vol. 41, no. 18, pp. 7145–7162, Sep. 2020.
- [9] A. Aditya, B. Lohani, J. Aryal, and S. Winter, "Benchmarking deep learning architectures for urban vegetation point cloud semantic segmentation from MLS," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–14, 2024, Art. no. 4405214, doi: [10.1109/TGRS.2024.3381976](https://doi.org/10.1109/TGRS.2024.3381976).
- [10] T. Jiang et al., "RailSeg: Learning local-global feature aggregation with contextual information for railway point cloud semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–29, 2023, Art. no. 5704929, doi: [10.1109/TGRS.2023.3319950](https://doi.org/10.1109/TGRS.2023.3319950).
- [11] Z. Zhang, A. Ji, K. Wang, and L. Zhang, "UnrollingNet: An attention-based deep learning approach for the segmentation of large-scale point clouds of tunnels," *Autom. Construct.*, vol. 142, Oct. 2022, Art. no. 104456.
- [12] D. Li et al., "PlantNet: A dual-function point cloud segmentation network for multiple plant species," *ISPRS J. Photogramm. Remote Sens.*, vol. 184, pp. 243–263, Feb. 2022.
- [13] H. Yu et al., "Deep-learning-based semantic segmentation approach for point clouds of extra-high-voltage transmission lines," *Remote Sens.*, vol. 15, no. 9, p. 2371, Apr. 2023.

- [14] B. Lohani, V. Kumar, P. Khan, A. Aditya, and M. Kumar. (2023). *Lidaverse—A Repository of Labelled LiDAR Point Cloud Files and Instance Objects for India*. [Online]. Available: <https://www.lidaverse.com>
- [15] W. Tan et al., “Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 797–806.
- [16] A. Boulch, J. Guerry, B. Le Saux, and N. Audebert, “SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks,” *Comput. Graph.*, vol. 71, pp. 189–198, Apr. 2018.
- [17] B. Kumar, G. Pandey, B. Lohani, and S. C. Misra, “A multi-faceted CNN architecture for automatic classification of mobile LiDAR data and an algorithm to reproduce point cloud samples for enhanced training,” *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 80–89, Jan. 2019.
- [18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.
- [19] C. R. Qi et al., “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Proc. Adv. NIPS*, vol. 30, 2017, pp. 1–24.
- [20] G. Qian et al., “PointNeXt: Revisiting PointNet++ with improved training and scaling strategies,” in *Proc. Adv. NIPS*, vol. 35, Dec. 2022, pp. 23192–23204.
- [21] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, *arXiv:1609.02907*.
- [22] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph CNN for learning on point clouds,” *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
- [23] Z. Du, H. Ye, and F. Cao, “A novel local-global graph convolutional method for point cloud semantic segmentation,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 1, no. 1, pp. 1–15, Jun. 2022.
- [24] M. Wei et al., “AGConv: Adaptive graph convolution on 3D point clouds,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9374–9392, Aug. 2023.
- [25] H. Lin et al., “Meta architecture for point cloud analysis,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 17682–17691.
- [26] A. Vaswani, “Attention is all you need,” in *Proc. Adv. NIPS*, 2017, pp. 1–23.
- [27] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, “Point transformer,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16259–16268.
- [28] J. Park, S. Lee, S. Kim, Y. Xiong, and H. J. Kim, “Self-positioning point-based transformer for point cloud understanding,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vancouver, BC, Canada, Jun. 2023, pp. 21814–21823.
- [29] Z. Zeng, H. Qiu, J. Zhou, Z. Dong, J. Xiao, and B. Li, “PointNAT: Large-scale point cloud semantic segmentation via neighbor aggregation with transformer,” *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–18, 2024, Art. no. 5704618, doi: [10.1109/TGRS.2024.3407761](https://doi.org/10.1109/TGRS.2024.3407761).
- [30] S. Schmohl, A. Narváez Vallejo, and U. Soergel, “Individual tree detection in urban ALS point clouds with 3D convolutional networks,” *Remote Sens.*, vol. 14, no. 6, p. 1317, Mar. 2022.
- [31] J. Li and X. Cheng, “Supervoxel-based extraction and classification of pole-like objects from MLS point cloud data,” *Opt. Laser Technol.*, vol. 146, Feb. 2022, Art. no. 107562.
- [32] D. P. Singh and M. Yadav, “3D-MFDNN: Three-dimensional multi-feature descriptors combined deep neural network for vegetation segmentation from airborne laser scanning data,” *Measurement*, vol. 221, Nov. 2023, Art. no. 113465.
- [33] W. Wang, Y. Fan, Y. Li, X. Li, and S. Tang, “An individual tree segmentation method from mobile mapping point clouds based on improved 3-D morphological analysis,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 2777–2790, 2023.
- [34] J. Yang, R. Gan, B. Luo, A. Wang, S. Shi, and L. Du, “An improved method for individual tree segmentation in complex urban scenes based on using multispectral LiDAR by deep learning,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 6561–6576, 2024.
- [35] P. Wang et al., “Road-side individual tree segmentation from urban MLS point clouds using metric learning,” *Remote Sens.*, vol. 15, no. 8, p. 1992, Apr. 2023.
- [36] W. Zhang et al., “Voxel-based urban vegetation volume analysis with LiDAR point cloud,” in *Proc. Fabos Conf. Landscape Greenway Planning*, vol. 7, Aug. 2022, pp. 1–12.
- [37] H. Luo, K. Khoshelham, C. Chen, and H. He, “Individual tree extraction from urban mobile laser scanning point clouds using deep pointwise direction embedding,” *ISPRS J. Photogramm. Remote Sens.*, vol. 175, pp. 326–339, May 2021.
- [38] C. Deng, O. Litany, Y. Duan, A. Tagliasacchi, and L. Guibas, “Vector neurons: A general framework for SO(3)-equivariant networks,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12180–12189, doi: [10.1109/ICCV48922.2021.01198](https://doi.org/10.1109/ICCV48922.2021.01198).
- [39] X. Deng, W. Zhang, Q. Ding, and X. Zhang, “PointVector: A vector representation in point cloud analysis,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 9455–9465.
- [40] J. Gong et al., “Omni-supervised point cloud segmentation via gradual receptive field component reasoning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 11668–11677.



Aditya Aditya (Graduate Student Member, IEEE) received the B.Tech. degree in mining engineering from the Indian Institute of Technology (ISM) Dhanbad, Dhanbad, India, in 2016. He is currently pursuing the Ph.D. degree with the Department of Infrastructure Engineering, University of Melbourne, Parkville, VIC, Australia, and the Department of Civil Engineering, Indian Institute of Technology Kanpur, Kanpur, India.

His research interests include deep learning for point cloud semantic segmentation and urban forests.



Bharat Lohani received the Ph.D. degree from the University of Reading, Reading, U.K., in 1999.

He is currently a Professor at the Department of Civil Engineering, Indian Institute of Technology Kanpur, Kanpur, India. His research interests include LiDAR data classification using deep learning techniques, LiDAR application in forest and water conservation, solar insolation estimation, and especially LiDAR simulation for autonomous systems.



Jagannath Aryal (Member, IEEE) received the Ph.D. degree in optimization and systems modeling from C-fACS, Christchurch, New Zealand, in 2010.

He is currently an Associate Professor with the Department of Infrastructure Engineering, University of Melbourne, Parkville, VIC, Australia. His research interests include optimal utilization of Earth observation, geoinformation, and geostatistics to develop new methods for object recognition in urban green space and disaster environment.



Stephan Winter (Senior Member, IEEE) received the Ph.D. degree from the University of Bonn, Bonn, Germany, in 1997, and the Habilitation degree from Technical University Vienna, Vienna, Austria, in 2001.

He is currently a Professor of spatial information science at the Department of Infrastructure Engineering, University of Melbourne, Parkville, VIC, Australia. His research interest in intelligent mobility, especially for urban sustainability, motivates data analytics research in urban green spaces.