Detection of individual trees in UAV LiDAR point clouds using a deep learning framework based on multi-channel representation

Zhipeng Luo, *Graduate Student Member, IEEE,* Ziyue Zhang, Wen Li, *Student Member, IEEE,* Yiping Chen, *Senior Member, IEEE,* Cheng Wang, *Senior Member, IEEE,* Abdul Nurunnabib, and Jonathan Li, *Senior Member, IEEE*

***Abstract*—Individual tree detection is critical for forest investigation and monitoring. Several existing methods have difficulties to detect trees in complex forest environment due to insufficiently mining descriptive features. This study proposes a deep learning framework based on a designed multi-channel information complementarity representation for detecting trees in complex forest using UAV laser scanning point clouds. The proposed method consists of two main stages: ground filtering and tree detection. In first stage, a modified graph convolution network with a local topological information layer is designed to separate the ground points. Unlike most existing parametric methods, our ground filtering method avoids the optimal parameters selection to adapt to different kinds of environments. For tree detection, a top-down slice (TDS) module is firstly designed to mine the vertical structure information in a top-down way. Then, a special multi-channel representation (MCR) is developed to perserve different distribution patterns of points from complementary perspectives. Finally, a multi-branch network (MBNet) is proposed for individual tree detection by fusing multi-channel features, which can provide discriminative information for MBNet to detect trees more accurately. MBNet was evaluated on seven forest areas (UAV LiDAR data with the mean size of 14,000 m^2 and point density of 250 points/m^2). Experimental results showed that the proposed framework achieves excellent performance. Our method obtains promising performance with mean recall of 89.23% and mean F1-score of 87.04%.**

***Index Terms*—Tree detection, Ground filtering, UAV LiDAR, Deep learning, Multi-channel representation.**

I. Introduction

FOREST plays an imperative role in the earth's ecosystem. It has significant impact on maintaining environmental conditions, such as habitat protection, air quality and water cycle [1]. These required for inventorying forest correctly and efficiently. Traditionally, forest inventory needs expensive labor cost, high time consumption [2] and several constraints, such as the weather and field survey conditions [3, 4]. With the progress of LiDAR (Light Detection and Ranging), UAV laser scanning system is widely used in forest inventory. Due to the ability of capturing 3D forest structures effectively and accuracy [5], UAV point clouds provide available solutions for numerous forest management tasks, such as estimation of tree species, height, wood volume and crown size, biomass, and so on [6, 7]. Detection of individual trees is important in forest inventory for subsequent estimation of necessary parameters, such as the tree species, height, diameter, crown size and location.

Generally, tree detection from UAV LiDAR point clouds includes an important preprocessing stage: ground filtering. A suitable filtering algorithm can obtain clean point clouds of non-surface objects, thus providing good initial data for tree extraction. Several researches have been undertaken to study the problem, such as [8–10]. Although these methods achieve good quality results under relatively flat environments, the challenge remains in processing non-flat areas. In this work, we seek the possibility to use a modified graph convolutional networks (GCNs) to improve filtering performance.

Several methods have been proposed to detect individual trees from UAV LiDAR point clouds. There are three main classes: raster based methods, point clouds based methods, and multi-source data fusion based methods. Rasterization is one of the most common ideas. These methods firstly generate canopy height model (CHM) by calculating the the digital elevation model (DEM) of the earth¡¯s surface and canopy surface height. Then the local maximum is used to determine the potential position of treetop crown, following by the pouring algorithm or the region growing (RG) algorithm, to delineate the tree crowns [11, 12]. Some improved CHM-based methods include [13–15]. These raster-based methods have achieved significant performance on tree structural attributes determination in urban simple forest. Since CHM is generated by gridding and interpolation, it inevitably leads to information loss, such as covering some important height information. The determination of tree crowns using region growing or pouring algorithms consumes expensive time. In addition, the rasterized image generated by CHM may not be the most discriminative representation to preserve the spatial relationship among 3D points, which would have negative impact on the

This work was supported by the National Natural Science Foundation of China under grants of 41871380, and the Natural Sciences and Engineering Research Council of Canada under a grant of 50503-10284. (Corresponding author: Yiping Chen and Jonathan Li)

Z. Luo, W. Li, Y. Chen, and C. Wang are with Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen 361005, China (zpluo@stu.xmu.edu.cn; liwen777@stu.edu.cn; chenyiping@xmu.edu.cn; cwang@xmu.edu.cn).

Z. Zhang is with the School of Computer Science, University of Nottingham Ningbo China, Ningbo, ZJ 315100, China (scyzz1@nottingham.edu.cn).

J. Li is with the Department of Geography and Environmental Management, University of Waterloo, Waterloo ON N2L 3G1, Canada (junli@uwaterloo.ca).

Nurunnabib, A. is with the Department of Geodesy and Geospatial Engineering, Institute of Civil and Environmental Engineering, University of Luxembourg, L-1359 Luxembourg (abdul.nurunnabi@uni.lu).

Z. Luo and Z. Zhang contribute equally to this article.

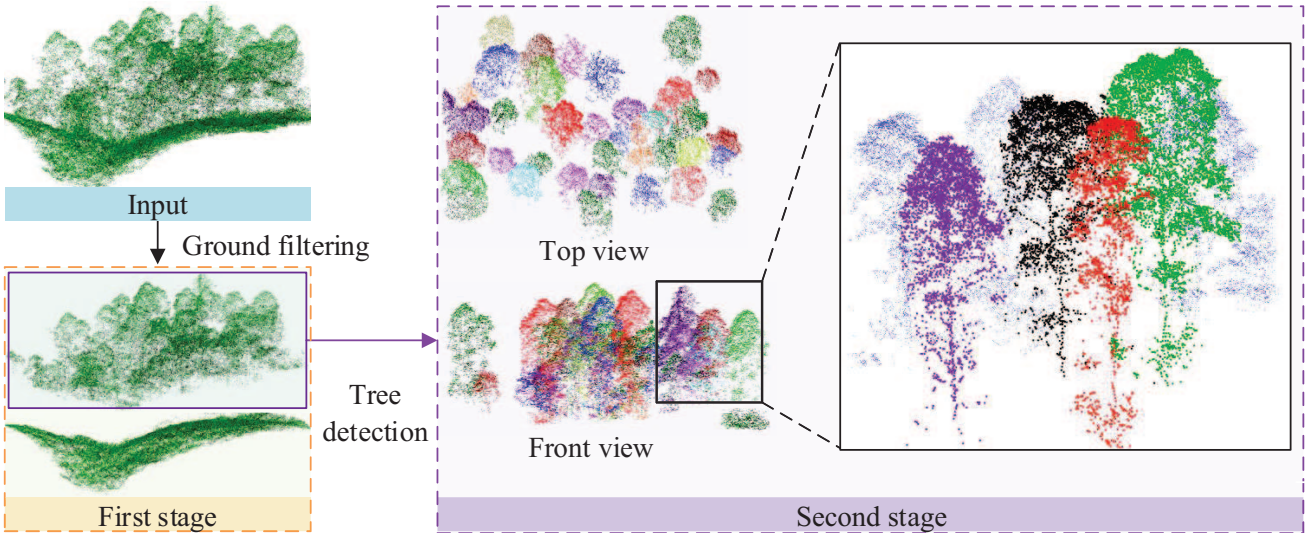

Fig. 1: Two main stages of our framework: ground filtering and tree detection. The ground filtering stage divides the input into off-ground and ground points. Tree detection module takes the off-ground points (purple box in first stage) as input data and outputs the individual trees.

extraction of tree features. All these shortcomings make raster based methods difficult to extract trees in complex forests.

Contrary to CHM based methods, point based methods analyze directly the 3D point clouds. The avoidance of generating CHM makes these methods have the main advantage on low information loss as well as mining the spatial structure of forest. In [16], a distance based segmentation method was presented to generate tree crown region in a mixed conifer forest. In [17], a voxel-based method is developed to produce a 3D solid model of a tree for accurate estimation of the volume of the woody material. In [18], the authors developed a bottom-to-top method based on the intensity and 3D structure to segment trees in deciduous forests. And in [19], the authors introduced the PTrees, a multi-scale dynamic point cloud segmentation method to extract trees. Recently, [20] presented a supervoxel and local convexity based method to label trees in urban spaces. Although this method has achieved promising results, the computational performance and ability of processing in complex environments remain a challenge.

Recently, with the development of computer vision and artificial intelligence, deep learning (DL) [21] has been widely applied in image processing. Promising results on various image processing tasks, such as image classification [22], semantic segmentation [23, 24], and object detection [25, 26], have demonstrated its outstanding potential of feature extraction in point clouds. In this work, we attempt to apply the DL approach to improve the descriptiveness of features generated by UAV LiDAR point clouds in complex forests. Our proposed framework consists of two stages: ground filtering and individual tree detection. Specifically, a modified GCNs is developed for ground filtering; while a multi-channel representation is proposed to utilize fully the forest spatial vertical structure information. Then, a multi-branch network (MBNet) is designed to mine more discriminative features for the tree detection. There are three main contributions of this work.

- A modified GCNs model is developed for ground filtering. Different from most of existing parametric filter algorithms, our model is a nonparametric and data-driven method, so can be applied to various landforms, such as rough slopes, dense vegetation areas and discontinuous terrains.
- A top-down slice multi-channel representation (MCR) of forest is developed to reorganize the spatial structure. Compared with traditional representations, such as CHM and point based representation, MCR can provide more discriminative structure information as well as detail features to detect trees from complex forests.
- A multi-branch network (MBNet) is proposed to merge multi-level information by concatenating features generated from different branches. Compared with existing approaches, MBNet can mine the distribution pattern to obtain a more discriminative descriptor.

The rest is structured as follows: Section 2 details the GCN based filtering method and the MBNet. Experimental results are presented in Section 3. Section 4 concludes our work.

II. The Proposed Method

As shown in Figure 1, the proposed framework contains two stages: ground filtering and individual tree detection.

A. Ground filtering

As mentioned in Section 1, most of the existing ground filtering methods can achieve satisfactory results under relatively flat areas, such as the urban environments. However, it is still difficult to filter ground points in complex forests. To improve the performance in abrupt slope, a modified graph convolutional networks (GCNs) model is developed to learn deep features for ground filtering. As a variant of regular convolutional neural networks (CNNs), GCNs can capture the pairwise dependencies between variables and the information from graph. Recently, GCNs have achieved huge success in processing graph structured data, especially in 3D point clouds, such as FoldingNet [27] and DGCNN [28].

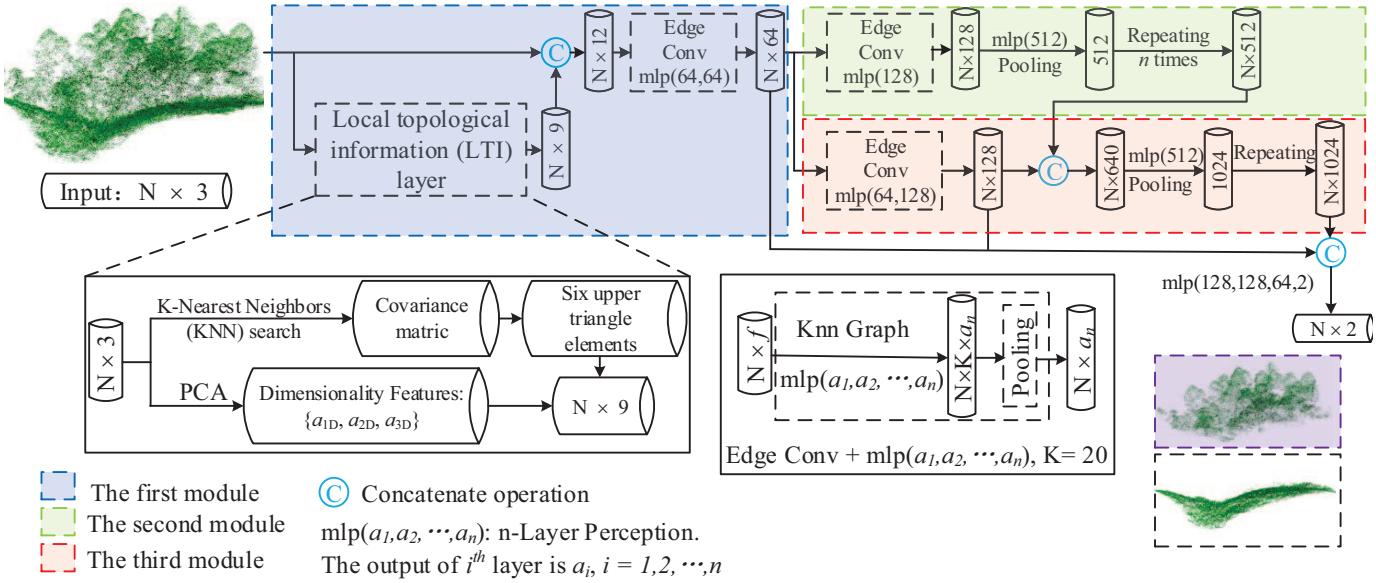

Fig. 2: Flowchart of ground filtering method. It is a Modified GCN framework with three modules. The first module is the local topological information (LTI) layer, which provides local features from neighbor points. The second and third modules are the GCN based networks that designed to describe the global and local features, respectively.

Due to the ability of extracting topological relationships between neighboring points, we introduce GCNs as the basic network to mine the local information to improve the ground filtering performance. More specifically, the ground filtering method consists of three modules (represented by different colors), as shown in Figure 2. The first module is the local topological information (LTI) layer, which provides a way to mine the local features from neighbor points. The second and third modules are the GCN based networks that designed to describe the global and local features, respectively.

The LTI layer is the core part, which takes the raw point clouds as input and outputs the shallow features with nine dimensions. More specifically, LTI computes the local covariance information as well as the dimensional features. For each point, 20 neighbor points are searched using K-Nearest Neighbors (KNN) search algorithm and the covariance matrix is calculated. Because the covariance matrix is a real symmetric matrix, the six upper triangle elements can be selected as covariance features. In addition, dimensionality features, (a_{1D}, a_{2D}, a_{3D}) [29, 30], are chosen as extra geometric information.

In summary, compared with the original GCN, our modified model enjoys several advantages. Firstly, LTI layer can provides more detail information. Local spatial relationship between neighbor points is fully utilized and geometric properties are preserved in the input data. Secondly, the combination of local and global features enhances the descriptiveness of the modified model. Different from the original GCN, extra global features in our modified model compensate for the deficiencies in the overall characterization. Thirdly, the use of focal loss down-weights the loss assigned to well-classified samples (in our work, for example, a point with a very high or low z value will be a well-classified sample) and pay more attention on the hard samples (e.g. the bushes). Therefore, under the guidance of this loss function, the model can be trained more efficiently in the direction of convergence.

B. Individual tree detection

Detecting objects from 3D point clouds has attracted huge attention. Several works have been developed to undertake this task, including voxel based methods [31, 32], view based methods [33–35], point based methods [36, 37] and multi-representation fusion based methods [38, 39]. The presence of noise and pseudo outliers caused by mutual occlusion and interpolation in complex forests makes it difficult to apply the voxel or point based methods. Possible remedies to avoid noise and outlier effects are to use of robust statistical approaches, but usually it takes more time than the classical approaches for point based processing [40]. Therefore, in this work, we consider the view based idea and propose a multi-branch network (MBNet). Firstly, different from the typical way of multi-view projection, considering the fact that most forests have rich vertical structure information, we design a top-down slice (TDS) module to obtain the multi-layer slice representation of forest in a top-down way. Secondly, to describe the distribution pattern of points contained in different slices, we propose a special multi-channel representation (MCR) module to represent these slices. Finally, based on MCR, a multi-branch network (MBN) module is designed for individual tree detection. Figure 3 presents the framework of MBNet, which consists of three modules: TDS, MCR and MBNet.

1) Top-down slice (TDS) module: After the first stage, we can obtain two separated data subsets: the ground and non-ground point subsets. We denote the non-ground point set as $P_{non} = \{p_1, p_2, ..., p_n\}$, where n is the number of non-ground points. Firstly, the ranges of X, Y, Z are normalized to [0,1]. Then the TDS operation divides P_{non} evenly into different subsets along the z-axis according to a given resolution. TDS is defined as a mapping, as follows:

$$TDS(P_{non}) = \{L_1, L_2, ..., L_K\}, \quad (1)$$

where $L_i = \{p | p \in P, \lceil z \times k \rceil = i\}, i = 1, 2, ..., K$, and K is the number of slices, z is the coordinates of point p along

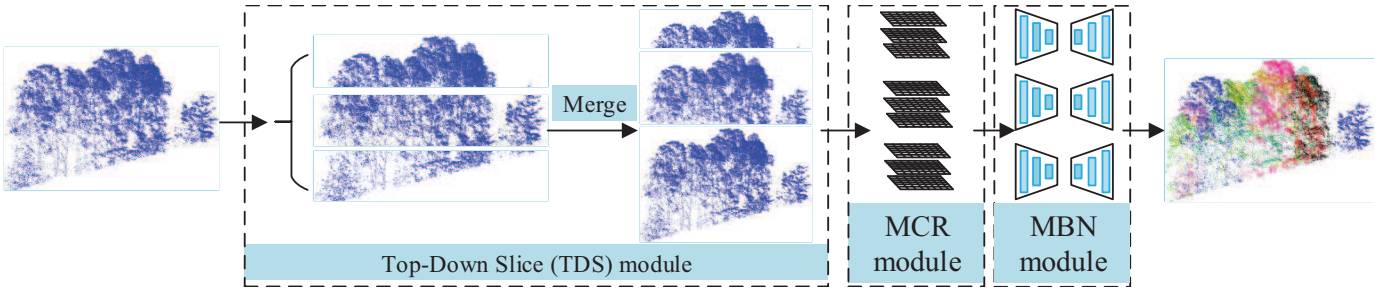

Fig. 3: Flowchart of MBNet. It consists of three modules: TDS, MCR and MBNet.

the Z axis, $\lceil\ \rceil$ denotes the ceiling function. Finally, as shown in Figure 3 , the merge operation integrates these slices into sequence segments:

$$\{S_1, S_2, ..., S_K\}, \tag{2}$$

$$S_i = S_{i-1} + L_{i-1}, S_1 = L_1, i = 2, ..., K. \tag{3}$$

Considering the fact that trees in a forest tend to exhibit three distinct vertical distributions: the spherical canopy in the top layer, rod-shaped trunk in the middle layer and scattered shrub in the bottom layer, we slice the forest into three segments, i.e., $k = 3$. Therefore, the spatial vertical structure of trees would be preserved in these slices.

2) Multi-channel representation (MCR) module: The MCR module is designed to obtain representation of slice generated in TDS module. Traditional forest representation methods tend to take only the height information. For example, the common CHM based methods generate CHM by using the height of each point. However, in addition to the height value, several other properties can also provide valuable information in processing complex forest.

Different from the traditional methods, in this work, we consider two additional properties: local height gradient and point density. The local height gradient captures the point distribution change of local position along vertical direction of the forest. A high gradient means that there is a high possibility of gaps between different trees in the corresponding local region. Therefore, local height gradient is of great importance for determining the canopy. Additionally, point density can help to extract tree top more accurately. This is because that, according to the plant growth process, there would be more points at the top of tree and its local area. Then, for each slice, we propose a three-channel representation, which consists of density, height and local height gradient information. More specifically, given arbitrary slice S_i, a mapping is designed to map S_i to a three-channel grid image I_i with given resolution r. In this paper, we set $r = 512$. Algorithm 1 shows how to generate the mapping.

As shown in Figure 4, each channel preserves distinctive distribution pattern of trees. Grid image with density channel presents the distribution of points, while height channel captures the location of tree-tops and describes the edge of canopy. Complementary features contained in these channels can be used to enhance the descriptiveness of the proposed method. Compared with the traditional CHM based methods, the proposed approach contains more distribution structures,

Algorithm 1: Mapping slice S_i into image I

Input: Slice: $S_i = \{p_j | p_j \in P, \lceil z \times k \rceil = i, j = 1, 2, !\text{-}, n_i\}$; Image resolution r
Output: I: Mapping image
$I = Zeros(r, r, 3)$; $C = Cell(r, r)$; ;
// STEP.1 : Generating the density channel ;
for $j = 1; j \leq n_i$ **do**
  $x = \lceil x_{p_j} \times r \rceil$; $y = \lceil y_{p_j} \times r \rceil$;
  $I(x, y, 1) \leftarrow I(x, y, 1) + 1$; // Density channel ;
  $C(x, y) \leftarrow p_j$; // Puting point p_j into $C(x, y)$;
// STEP.2 : Generating the Height channel ;
for $h = 1; h \leq r$ **do**
  for $t = 1; t \leq r$ **do**
    $I(h, t, 2) \leftarrow |max(z) - min(z)|$; // where z is the z-axis of point p, and $p = (x, y, z) \in C(h, t)$;
// STEP.3 : Generating the Height gradient channel ;
for $h = 1; h \leq r$ **do**
  for $t = 1; t \leq r$ **do**
    $I(h, t, 3) \leftarrow \sum_{i=-1}^{1} \sum_{j=-1}^{1} \Delta(i, j)$;//where $\Delta(i, j) = |I(h + i, t + j, 2) - I(h, t, 2)|$;
// Normalizing each channel of I to [0,255] ;
$I \leftarrow Norm_{[0,255]}(I)$;
return I;

which would contribution to increase detection performance. Besides, for the Algorithm 1, with three loops, its time complexity is $O(n_i)$ + $O(r^2)$ + $O(r^2)$, where n_i is the number of points in slice S_i, and r is the given resolution of mapping image. In this paper, the image resolution is set as $r = 512$. Therefore, the time complexity of Algorithm 1 can be considered as $O(n_i)$, i.e., linear in the number of input points. This is acceptable in practical application.

3) Multi-branch network (MBNet) module: Extracting trees from grid image belongs to the object detection task in image processing. However, since the grid image contains only one type of object (i.e., trees), it is feasible to treat tree extraction as an instance segmentation task. In this work, we use the encoder-decoder (ED) as the basic network. For an ED network, the encoder extracts feature from input data, and encodes these features into low dimensional representation, a feature vector, in a latent space, while the decoder uses

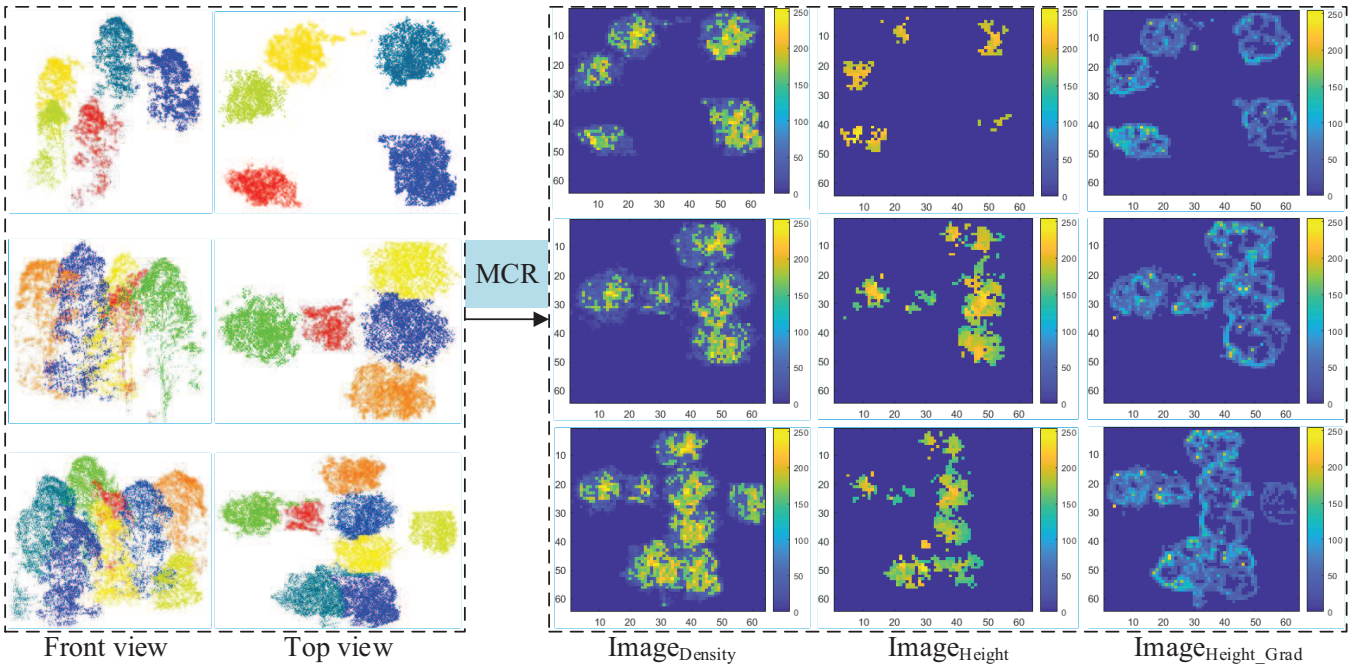

Fig. 4: (a) Raw data, (b) Representation generated by MCR module. Trees in (a) are denoted with different colors manually. Each kind of color represents one individual tree. In (b), $Image_{Density}$, $Image_{Height}$, and $Image_{Height\_Grad}$ denote the channel with density, height and height gradient information, respectively.

TABLE I
DETAILS OF ENET. THE INPUT SIZE IS 512 × 512. REGULAR(I) DENOTES A REGULAR CONVOLUTION WITH I × I FILTER. DILATED(I) DENOTES THE DILATED CONVOLUTION [41] WITH I × I FILTER, AND THE ASYMMETRIC(I) DENOTES AN ASYMMETRIC CONVOLUTION [42] WITH I × 1 FILTER AND 1 × I FILTER. FIGURE 5 SHOWS THE INITIAL BLOCK AND BOTTLENECK MODULE.

Name		Description	Output size		Name	Description	Output size
Initial block		Regular(3)	16×256×256	**Block4**	Bottleneck 4.0	Up sampling + Regular(3)	64×128×128
Block1	Bottleneck 1.0	Down sampling + Regular(3)	64×128×128		Bottleneck 4.1	Regular(3)	
	4×Bottleneck1 .x(x=1,2,3,4)	Regular(3)			Bottleneck 4.2	Regular(3)	
Block2	Bottleneck 2.0	Down sampling + Regular(3)	128×64×64	**Block5**	Bottleneck 5.0	Up sampling + Regular(3)	16×256×256
	Bottleneck 2.1	Regular(3)			Bottleneck 5.1	Regular(3)	
	Bottleneck 2.2	Dilated(1)		Full convolution			2×512×512
	Bottleneck 2.3	Asymmetric(5)					
	Bottleneck 2.4	Dilated(3)					
	Bottleneck 2.5	Regular(3)					
	Bottleneck 2.6	Dilated(7)					
	Bottleneck 2.7	Asymmetric(5)					
	Bottleneck 2.8	Dilated(15)					
Block3	Repeat Block2, without bottleneck 2.0						

the deconvolution operation to recover the input image, and predicts the label for each pixel. ED network has been applied successfully in semantic and instance segmentation, such as U-Net [43], U-Net++ [44] and ENet [45, 46]. The method in [46] is a modified ENet using a well-designed loss function for the binary instance segmentation and achieves promising performance. Therefore, in our work, we utilize ENet and the loss function in [46] as basic module and design a multi-branch network (MBNet) to learn the semantic features for instance segmentation from three segments generated in TDS module.

As shown in Figure 5 (a), the MBNet module contains three parallel modified ENets, which have the same network structure. Features generated by each ENet are fused by an adding operation, following by a softmax layer. In each ENet, the encoder has three blocks, while the decoder contains two blocks. An initial block is added between input and encoder. A full convolution layer is used to output the final feature maps. Table I presents details of each block, and Figures 5 (c) and (d) show the initial block and the bottleneck module, respectively.

It is necessary to point out that the training of MBNet module is performed in a two-step-training way. The first step is the pre-trained step. In this stage, each branch is trained individually. Then, we can obtain three pre-trained parameter sets. In the second stage, three branches are trained together and the outputs of these branches are fused in an element-wise addition way. The advantage of two-stage-training way is that the model can be trained more steadily and converge faster. This is because these three branches learn their own features from different segments. Training in one step may

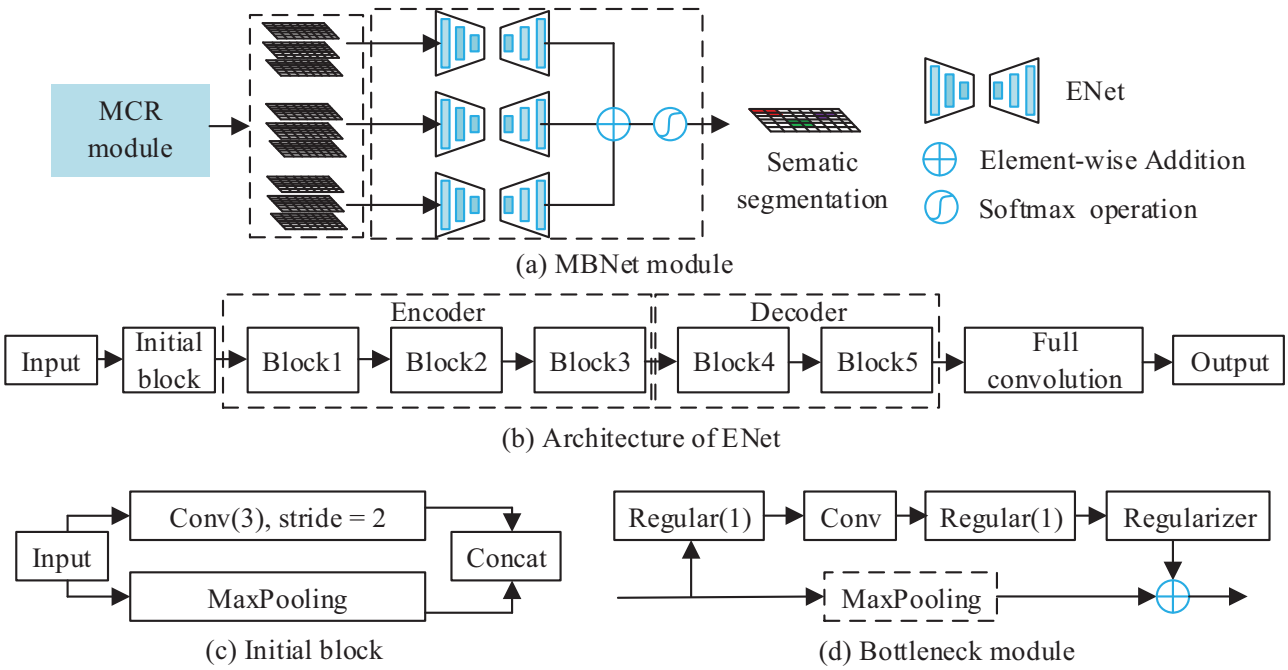

Fig. 5: (a) The flowchart of MBNet module,(b) the architecture of ENet, (c) Initial block, and (d) The bottleneck module in Table I. In initial block, the size of MaxPooling is 2 × 2 and the stride is 1, while the Conv(3) denotes a 3 × 3 regular convolution with 13 kernels. In bottleneck module, the main branch consists of three convolutional layers: a 1 × 1 regular convolution that reduces the dimensionality, a main convolutional layer and a 1 × 1 regular convolution that designs to expend the dimensionality. If there is down sampling in a bottleneck, a MaxPooling operation is added to the main branch. Details are presented in Table I.

TABLE II
MAIN PARAMETERS OF LASER SCANNER.

Parameters	Values
Model	Riegl UVX-1
Place of origin	Austria
Size /mm	277×180×125
Weight /kg	3.5
Survey-grade accuracy /mm	10
Laser emission frequency /kHz	550
Scan speed /scans.s^{-1}	200
Field of view /(°)	330

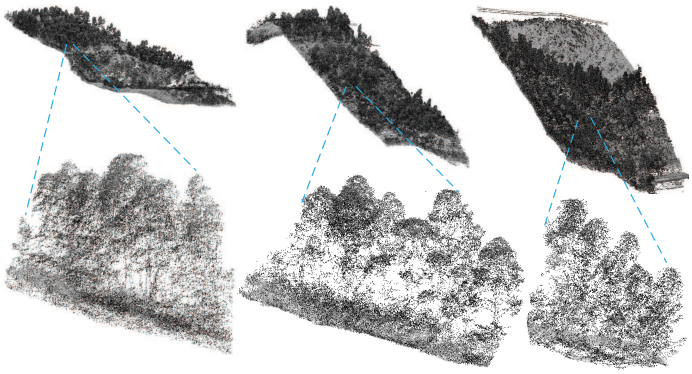

Fig. 6: Samples from study area. These forests are extremely noisy.

influence each other and may cause serious oscillations in the optimization process.

III. Experimental Results and Discussion

A. Study area

The study area is the Shaoguan $(24°42'N, 113°53'E)$, located in Guangdong, China. The dataset was acquired with a ALS system, which consists of an unmanned aerial vehicle (UAV) and a lightweight and compact laser scanner. The model of UAV is DJI M600 Pro (SZ DJI Technology Co. China). The flying altitude and speed of UAV were up to 150 m and 10 m/s, respectively. The main parameters of the laser scanner are provided in Table II.

Seven areas with complex forests were selected for our experiments. They are in different complex levels. The altitude gap ranges from 106m to 64m. Figure 6 shows three samples. Obviously, all these samples contain mixed noise, especially unordered outliers that bring huge difficulty for detecting individual trees. To further analyze these selected areas, several key statistical information is presented in Table III. From Table III, we can see that these areas are of high-density point clouds (approximately 250 $points/m^2$ in each area). For example, area 1 contains the largest number of points, which is close to 4,500,000. The number of points in area 7 is the lowest, but it is still more than 2,500,000. Besides, it can be inferred from the height information that all these areas are very rough. Especially, the largest altitude gap is 106 m in Area 1. In addition, Figure 7 presents the height histogram for each area, which shows the terrains are diverse and no-flat. Therefore, these selected areas are suitable for our study. It needs to point out that the ground truth for ground filtering algorithm and individual tree detection in this work are obtained by careful manual classification. Specially, they are independently labeled and verified by three people to make the annotation as accurate as possible.

Several experiments are conducted to evaluate the proposed method in the following subsections, including tree detection using MBNet, efficiency testing and model design analysis.

TABLE III
KEY STATISTICAL INFORMATION OF SEVEN STUDY AREAS. THEY ARE SORTED ACCORDING TO THE ALTITUDE GAP

	Area1	Area2	Area3	Area4	Area5	Area6	Area7
Size (m^2)	20,231	12,495	11,708	13,168	12,636	15,115	16,110
Point number	4,498,425	4,268,649	3,655,193	3,898,510	3,426,171	3,976,401	2,595,690
Max height (m)	212	212	210	208	201	181	159
Min height (m)	106	107	124	115	123	95	95
Altitude gap (m)	106	105	86	93	78	86	64

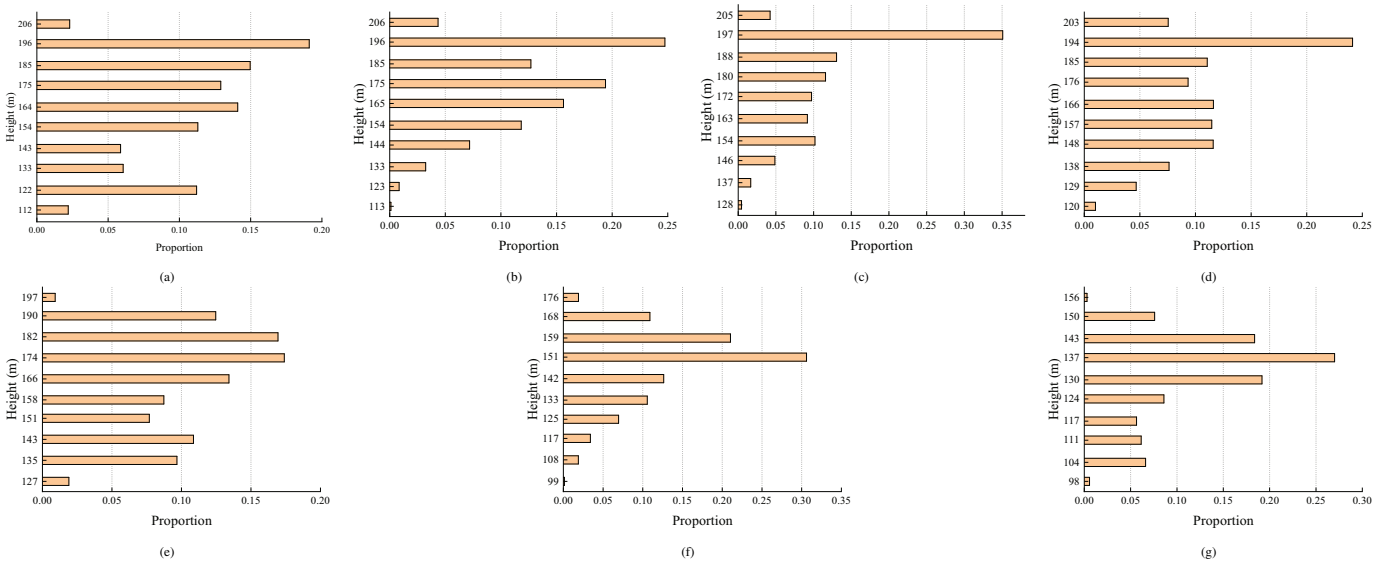

Fig. 7: Height histograms of seven areas.

TABLE IV
DEFINITIONS OF TYPE I ERROR, TYPE II ERROR, TOTAL ERROR , AND KAPPA COEFFICIENT k.

		Predicted results			
		Ground	Non-ground		
Ground-truth	Ground	a	b	a+b	f=(a+b)/e
	Non-ground	c	d	c+d	g=(c+d)/e
		a+c	b+d	e = a+b+c+d	
Type I error		b/(a+b)			
Type II error		c/(c+d)			
Total error		(b+c)/e			
k	[(a+d)/e - p]/ (1 - p), where p = f×(a+c)/e + g×(b+d)/e, a,b,c and d are the true positive, false negative, false positive and true negative, respectively.				

TABLE V
OPTIMAL PARAMETERS FOR CSF IN SEVEN AREAS.

	Area 1	Area 2	Area 3	Area 4	Area 5	Area 6	Area 7
Cloth resolution (m)	2.04	9.11	5.88	7.75	2.38	5.48	4.04
Max iterations	200	750	500	800	500	300	450
Classification threshold (m)	4.57	11.85	8.93	8.18	4.04	11.33	7.46

TABLE VI
COMPARING GROUND FILTERING RESULTS GENERATED BY DIFFERENT METHODS. I, II, T. AND K DENOTE TYPE I ERROR, TYPE II ERROR, TOTAL ERROR AND KAPPA COEFFICIENT, RESPECTIVELY.

Area	Metric	Area 1	Area 2	Area 3	Area 4	Area 5	Area 6	Area 7	Means	Max	Min
CSF [47]	I (%)	2.49	2.04	**2.01**	**4.57**	3.31	6.53	**3.54**	**3.49**	-	2.01
	II (%)	9.34	9.11	8.66	11.85	14.42	16.6	12.25	11.74	-	8.66
	T. (%)	6.00	5.88	5.03	8.93	8.45	12.28	8.53	7.87	-	5.03
	k	87.52	87.75	89.12	81.86	82.3	74.75	81.46	83.53	89.12	-
PointNet [36]	I (%)	4.28	2.58	4.32	9.18	2.97	8.98	8.83	5.88	-	2.58
	II (%)	**9.14**	6.22	4.06	8.51	15.41	11.26	5.95	**8.65**	-	4.06
	T. (%)	5.92	4.35	4.24	8.96	7.68	10.75	7.15	7.00	-	4.24
	k	87.21	90.76	90.32	81.64	82.48	77.73	84.29	84.91	90.76	-
DGCNN [28]	I (%)	4.12	2.00	3.20	7.68	2.82	8.97	8.20	5.28	-	**2.00**
	II (%)	9.62	5.92	4.02	**8.34**	**13.31**	**11.25**	**5.81**	8.32	-	4.02
	T. (%)	6.01	3.91	**3.43**	8.15	6.82	10.73	6.78	6.54	-	3.43
	k	86.99	91.69	**92.13**	83.30	84.49	77.75	85.09	85.92	92.13	-
Ours	I (%)	**2.39**	**1.85**	4.23	6.86	**2.09**	**5.66**	7.05	4.30	-	1.85
	II (%)	10.49	**5.85**	**3.71**	9.09	14.76	15.15	7.03	9.44	-	**5.85**
	T. (%)	**5.41**	**3.77**	4.00	**7.91**	**6.79**	**10.59**	**6.80**	**6.46**	-	**3.77**
	k	**88.14**	**91.98**	90.85	**83.65**	**84.47**	**77.84**	**85.13**	**86.00**	**91.98**	-

B. *Ground filtering*

In this work, we conducted experiments on the above datasets and compared our method with previous ground filtering algorithm, including the deep learning based methods, PointNet [36] and DGCNN [28], as well as the classic method, the cloth simulation filtering (CSF) proposed in [47]. CSF uses the cloth simulation algorithm to achieve state-of-the-art results and is available in the open-source CloudCompare (http://www.cloudcompare.org/).

We used the metrics provided in [48] and [49] to measure the performance. As shown in Table IV, [48] proposed three metrics for quantitative analysis. Type I error measures the rate of ground points mislabeled as non-ground points, while Type II error represents the percentage of non-ground points mislabeled as ground points. Total error shows the rate of all mislabeled points. Besides, the Cohen's kappa coefficient (k) [49] is widely used in most of filtering algorithms [50–52]. It measures the inter-ratio agreement more robustly than a percentage. Since the CSF is a parameter method, we evaluated the performance of CSF on each area using grid search method to obtain the optimal parameters. Table V shows the optimal parameters for seven areas. It needs to point out that, as ours is a supervised method, training data is required. Therefore, in our experiments, when an area is used as testing data, other six areas are considered as training data. In addition, our method was trained with TensorFlow on a NVIDIA Tesla P100 GPU. The batch size was set to 8 and the initial learning was 0.001. When training the model, we used adaptive moment estimation (Adam) with a momentum of 0.9. The number of epochs was 50.

Table VI presents the compared results. Obviously, our method achieves excellent performance. More specifically, our method obtains the best results on areas 2. Besides, in areas 1, 4, 5, 6 and 7, our method also achieves competitive results. Additionally, compared to CSF and PointNet, our method increases the performance by 2.47% and 1.08% in overall average k coefficient, respectively. These results mean that our method can filtering the ground points more precisely. On the overall average total error and Type II error, our method also has the obvious advantage, with a significant reduction comparing with other three methods. That means our method can reject object points more effectively and has a smaller proportion of all error points.

The reason for the better performance of our method is that the local graph structure using in the ground filtering framework can preserve the relationship among neighbor points. Compared with PointNet and DGCNN, the local features of ground points are captured by our method more effectively. Besides, compared with CSF, as a non-parametric method, our method does not need to find the optimal parameters

To further present the comparative results, we visualized the ground and non-ground points for some areas. As shown in Figure 8, our method achieves excellent performance in preserving ground points. Note that filtering ground points in steep areas is another challenge [10]. Our method also labels the ground points as non-ground points in some areas, just

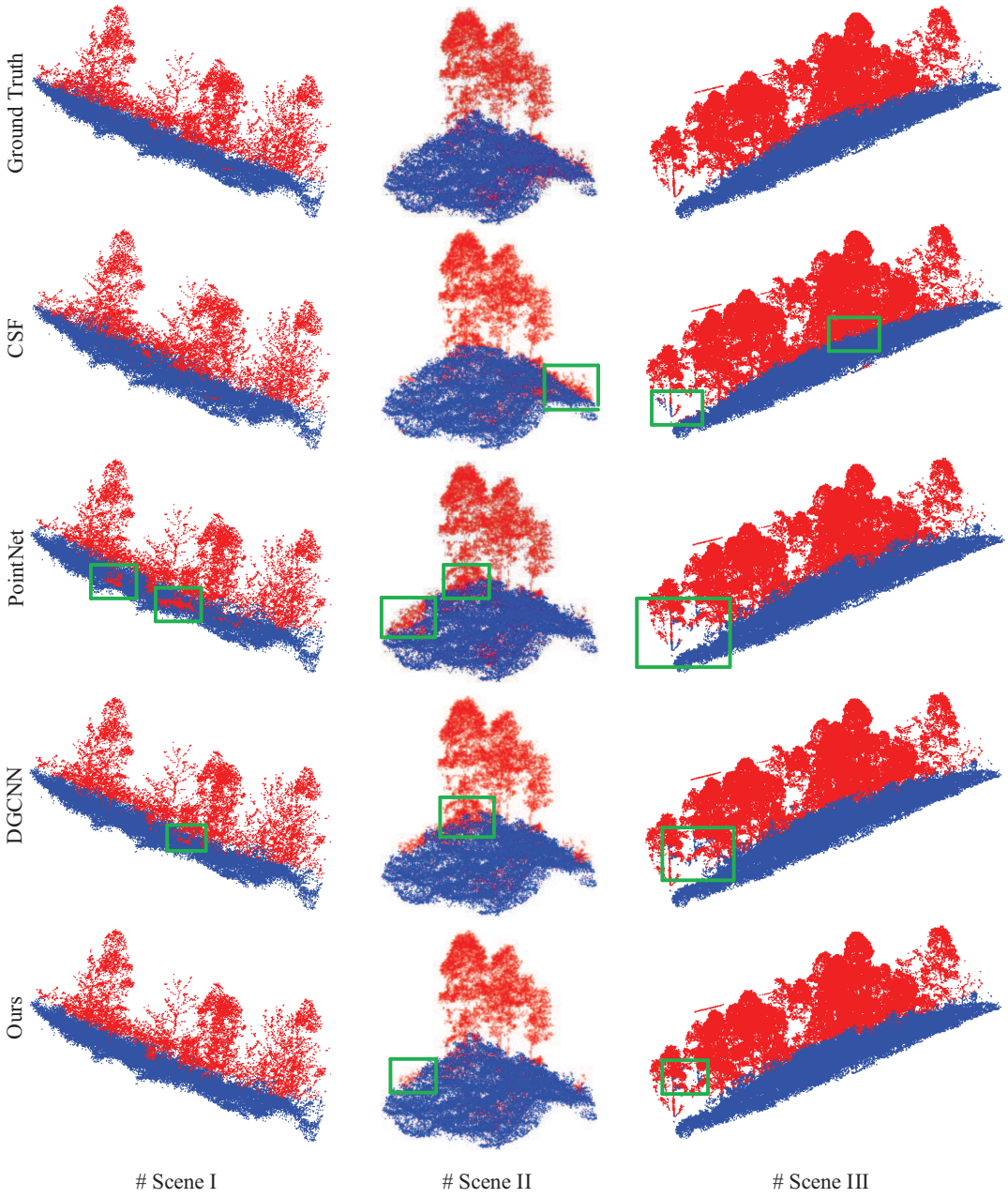

Fig. 8: Results of ground filtering generated by different methods on several scenes. The ground and non-ground points are marked by blue and red color, respectively. The areas marked by the green boxes are areas with more mislabeling error points.

as shown in the scenes II and III. However, the number of mislabeling points is very low and acceptable in practice.

C. Tree detection using MBNet

In our work , tree detection contains three steps, including building the synthetic dataset, training MBNet on synthetic dataset and testing MBNet on both synthetic and real datasets.

1) Synthetic dataset generation: Generating synthetic data is of great importance for the proposed method. It is known that supervised method based on deep learning requires a large number of labeled training samples. For example, in image processing field, ImageNet [53], a huge dataset, provides a variety of labeled images to improve the performance of image processing methods. Similarly, in our work, we designed a simple but effective method to generate large number of synthetic samples. Specifically, after ground filtering, individual trees were firstly extracted manually from the non-ground points. Then, part of these individual trees was selected randomly and put together to form a sample. Note that, to better simulate the real scene, the location of each tree is also random. Finally, trees in this sample are projected into a 2D grid image using Algorithm 1. A large number of training samples can be obtained by repeating this way of random sampling. Table VII describes the detail of synthetic dataset and Figure 9 shows several samples.

2) Training MBNet on synthetic dataset: The proposed model was trained on the above synthetic dataset. We trained

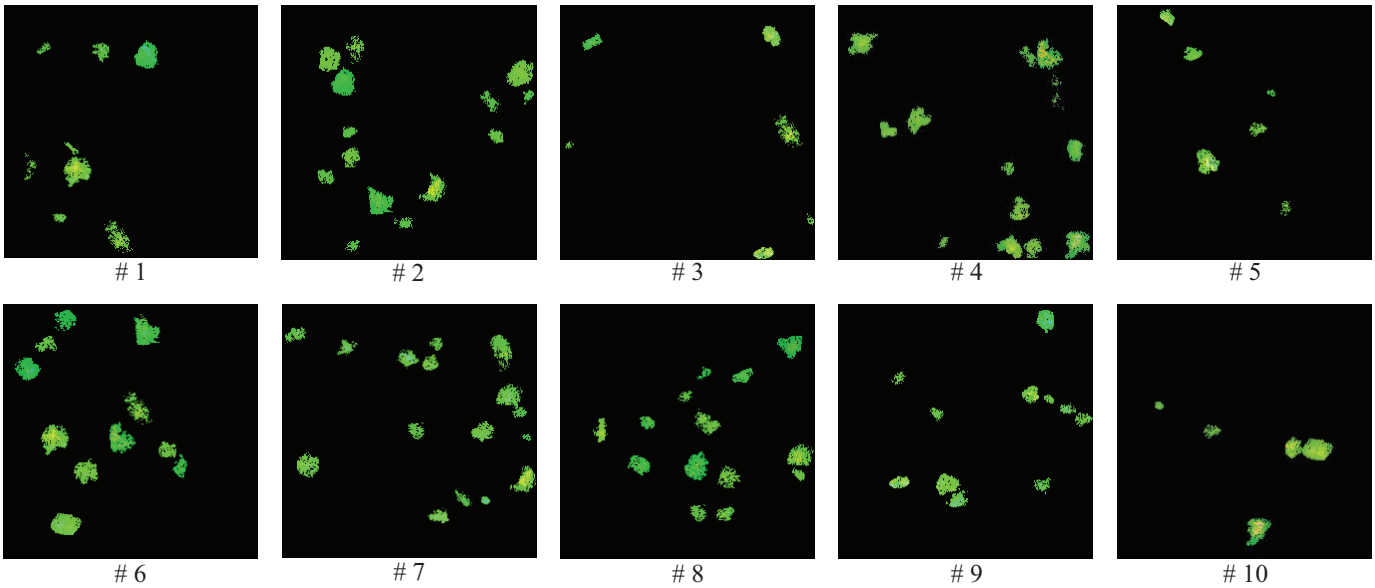

Fig. 9: Several samples from synthetic dataset.

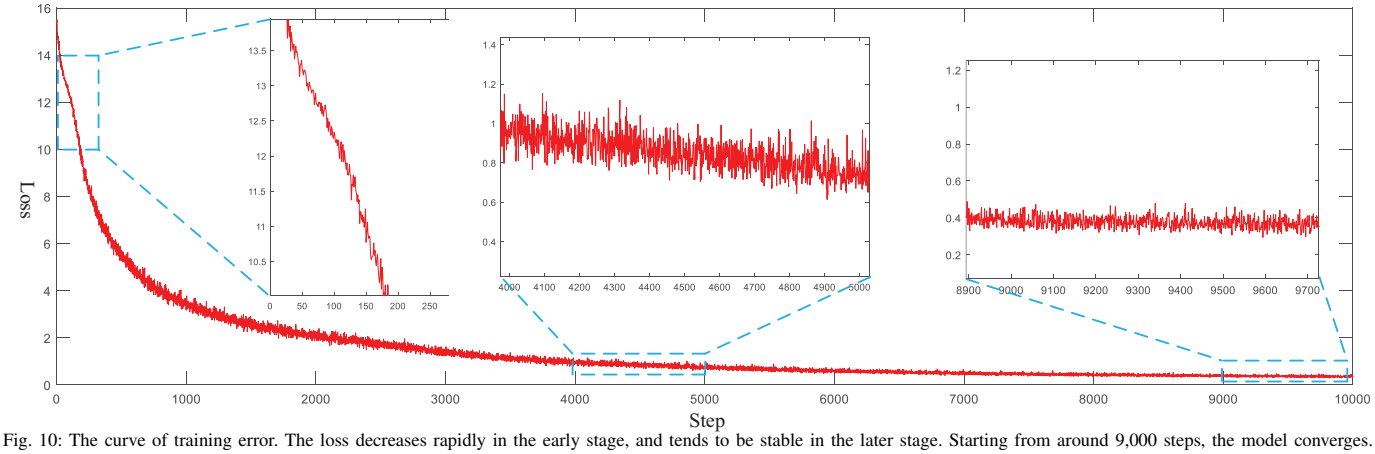

Fig. 10: The curve of training error. The loss decreases rapidly in the early stage, and tends to be stable in the later stage. Starting from around 9,000 steps, the model converges.

TABLE VII
DETAILS OF THE SYNTHETIC DATASET.

	Training sample number	Testing sample number	Mean point number	Mean tree number
Synthetic dataset	1,200	200	22,309/sample	12/sample

the model with Tensorflow on a NVIDIA Tesla P100 and the initial learning rate is set to 0.001 and decreased by half in every 20 epochs. The batch size is 32. The pre-trained stage has the same hyper-parameters. Figure 10 shows the curve of training error. It was found that in the first 1,000 steps, the error decreased significantly, which means that through the previous training, the model has obtained the optimal direction of descent. Then, from 1,000 to 7,000 steps, the deceleration of error rate gradually slows down. This means that the model is steadily moving towards the direction of convergence. Finally, starting from around 9,000 steps, the error tends to be stable, which means the model has converged. Therefore, considering that in this work, the number of training samples is 1200 and the batch size is 32, the number of epoch is set to 250.

3) Testing MBNet: We evaluated the proposed detection approach on real datasets, i.e. seven areas. The Mask-RCNN [54] and a baseline, classic CHM and RG based algorithm (denoted as RG-CHM), were used for comparing with the proposed method. Considering the complexity of forest, we studied the most important aspect of performance for each sample, the number of trees. Following three metrics are used to measure the performance:

$$Precision = \frac{TP}{TP + FP}, \tag{4}$$

$$Recall = \frac{TP}{TP + FN}, \tag{5}$$

$$F_1\ score = \frac{2 \times Precesion \times Recall}{Precesion + Recall}, \tag{6}$$

where TP, FP and FN are the number of true positives, false positives and false negatives, respectively.

TABLE VIII
COMPARATIVE RESULTS OF TREE DETECTION GENERATED BY DIFFERENT METHODS.

	RG-CHM			Mask-RCNN [54]			Our method		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
Area 1	**97.79**	26.25	41.38	77.93	45.69	57.60	94.44	**83.51**	**88.64**
Area 2	85.86	**86.31**	86.08	74.69	46.11	57.01	**97.14**	77.95	**86.49**
Area 3	77.36	58.82	66.83	73.06	53.86	62.00	**93.09**	**89.68**	**91.35**
Area 4	**77.78**	16.67	27.45	68.69	70.11	69.39	**77.78**	**95.83**	**85.87**
Area 5	75.09	8.33	15.00	68.69	56.52	62.01	**80.00**	**95.24**	**86.96**
Area 6	**82.36**	61.46	70.39	71.96	44.51	55.00	74.34	**91.96**	**82.22**
Area 7	70.51	80.95	75.37	75.24	66.10	70.37	**85.19**	**90.48**	**87.75**
Means	80.96	48.39	54.64	72.89	54.70	69.91	**85.99**	**89.23**	**87.04**

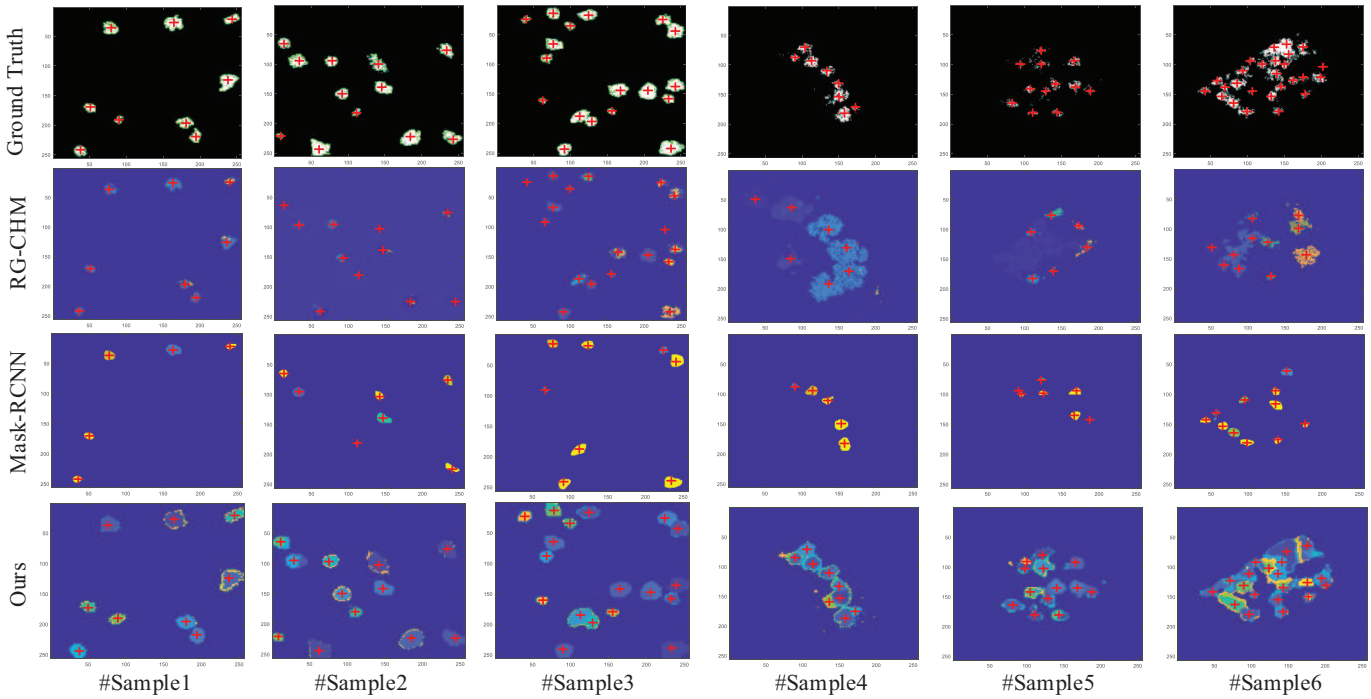

Fig. 11: Tree detection results generated by different methods on several samples. The first three columns are the results from synthetic dataset, while the last three columns are the results generated from the real dataset. "+" with red color denotes as individual tree.

Table VIII presents the performance of different methods. As shown in Table VIII, by achieving mean precision of 85.99%, mean recall of 89.23% and mean F1-score of 87.04%, our method outperforms the baseline and Mask-RCNN. Furthermore, in each area, the F1-score of our method is higher than that of other two methods.

The better results achieved by our method mainly because of two reasons. Firstly, the multi-branch representation of forests preserves the distribution patterns of tree points with different heights. Compared to the CHM representation used in RG-CHM method, the MCR contains rich hierarchical structure information that can improve the descriptiveness of our method significantly. Secondly, compared to Mask-RCNN, our designed multi-branch network first takes multi-channel representation as input, which would provide rich vertical distribution information at different heights. Then, the fusion module used in our network allows the information contained in each 2D grid images to be fused together to complement each other, which would enhance the ability of feature extraction.

Figure 11 shows several detection results from 2D grid image generated by different methods. Obviously, compared with RG-CHM and Mask-RCNN, our method has more significant advantage over them. Furthermore, our method has a better balance between the recall and precision. For each sample, the area of crown and the number of trees predicted by our method are more accurate than RG-CHM and Mask-RCNN.

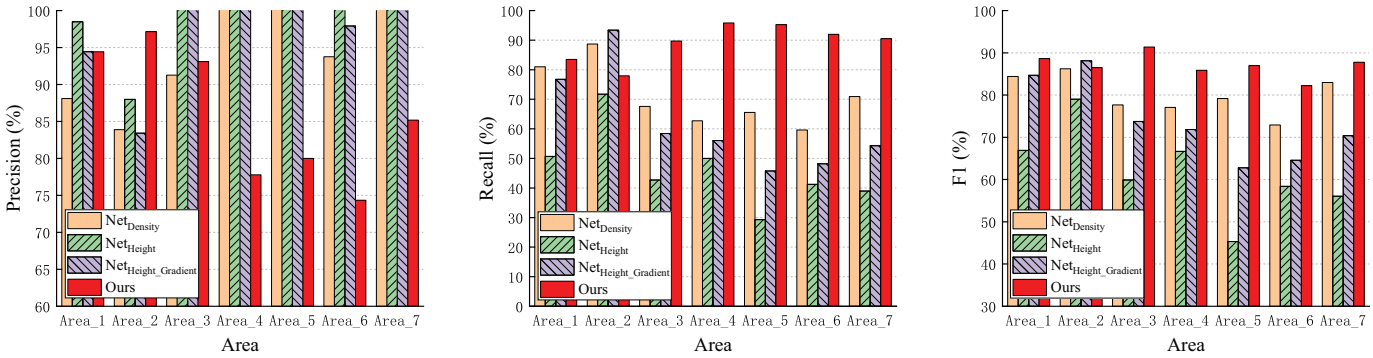

Fig. 12: Performance of different representations on seven areas. The metrics are (a) precision, (b) recall and (c) F1 score, respectively.

TABLE IX
INFERENCE TIME OF DIFFERENT METHODS ON REAL DATASET OF SEVEN AREAS IN SECONDS (S), WHERE "Ai" MEANS "$Area\_i$"

Method	Mean	A1	A2	A3	A4	A5	A6	A7
RG-CHM	31.01	45.82	20.93	52.37	3.94	60.72	26.46	6.89
Mask-RCNN	0.61	1.29	0.65	0.72	0.16	0.54	0.66	0.26
Ours	4.77	7.98	4.98	6.43	1.42	4.91	5.62	2.06

TABLE X
COMPARISON OF DIFFERENT REPRESENTATIONS AND OUR METHOD.

	Real dataset		
	Precision (%)	Recall (%)	F1 score (%)
Net_D	93.85	70.86	80.06
Net_H	**98.06**	46.37	61.75
Net_{HG}	96.54	61.83	73.73
Ours	85.99	**89.23**	**87.04**

D. Efficiency testing

We firstly evaluate the time efficiency of different methods. Specifically, we compare our method with RG-CHM and Mask-RCNN by calculating the inference time on real dataset of seven areas. Table IX presents the comparison results. Obviously, our method achieves better performance than RG-CHM on each area. More specifically, RG-CHM is about six times slower than our method in terms of average time. The reason for the high efficiency of our method mainly lies in the acceleration of GPU, which can execute the network quickly. However, comparing with the DL-based method, the Mask-RCNN, our method has a high time consumption. Mask-RCNN is more than seven times faster than our method in terms of average time. This is because our method takes the multi-branch network and needs to fuse the features generated by multiply branches, which would bring lower time efficiency.

Secondly, we calculate the parameters of the proposed method. The number of parameters is about 1.11 M, where ¡°M¡± stands for million. Besides, we report storage required to save model parameters in half precision floating point format and the required space is about 2.1MB, which means our model has a lightweight size and can be applied for real project in the forest inventory.

E. Model design analysis

As discussed in Section 3, our designed framework consists of three representations: the density based, height based and height-gradient based grid images. To analyze our model, we evaluated the effects of each representation.

Net_D, Net_H, and Net_{HG} are used to denote the proposed method only with the density based, height based and height-gradient based grid images, respectively. We compared our method with these three baselines. As shown in Table X, our method achieves the best performance in terms of recall and F1 score. Specifically, our method significantly outperforms other three baselines by a margin on real dataset. Compared with Net_D, our method achieves significant improvements of 18.37% and 6.98% of recall and F1-score, respectively.

The above results show that our method has a better balance between precision and recall. The reason lies in the multi-branch fusion strategy. The three representations used in the designed framework preserve different information in three aspects, which can provide rich complementary structure features to enhance the descriptiveness. However, it needs to point out that in terms of precision, our method is worse than Net_D, Net_H and Net_HG. Besides, we can observe that Net_H achieves the best performance in terms of precision. This is because for Net_H, comparing our method, it takes only the height information as its representation, the tree points are extracted more strictly. This means Net_H would obtain a lower FP. According to the definition of precision, Net_H can obtain a higher precision.

To further investigate the effects of each representation, we evaluate the performance of different representations in each area. Figure 12 provides the details for the performance of different representations in seven areas. More specifically, the precision/recall/F1 score in Table X is the average value of Figure 12 (a)/(b)/(c) in seven areas. As shown in Figures 12. (b) and (c), we observe that our method obtains the highest recall and F1 values, respectively in each area. This shows that our method is more descriptive than three baselines. In addition, among three baselines, Net_D performs well in terms of recall, while NetH achieves better performance in precision, whereas Net_{HG} produces the worst results. This is consistent

with the expectation of these representations. Representations of Net_D and Net_H are designed to capture the distribution patterns of density and height of points respectively, while that of Net_{HG} encodes the gradient of height, which is mainly expected to provide additional supplementary information. Therefore, these representations make their own contributions to the designed framework.

IV. Conclusions

This work aims to extract more discriminative features for ground filtering and individual tree detection in complex forest. At the ground filtering stage, a local topological based GCN is designed to mine the relationship among neighbor points to improve the ground filtering performance. As a data-driven approach, the modified GCN avoids the parameter selection problem associated with most of the existing parametric methods. Compared to CSF and PointNet, our method increases the performance by 2.47% and 1.08% in overall average k coefficient, respectively. Compared to DGCNN, our method also achieves better results. At the detection stage, unlike most of the existing methods mainly focused on the height information of forest, we firstly developed a multi-channel representation (MCR) to preserve three kinds of distribution patterns of points in three complementary perspectives: density, height, and height-gradient of points. Secondly, based on MCR, a multi-branch network (MBNet) is designed to fuse the hierarchical structure features to detect trees. The proposed MBNet presents a promising way to apply the DL to extract deep features from complex forest accurately. Experimental results show the superiority of the proposed architecture.

Limitation and future work: Firstly, as the description in part II, our method mainly focuses on the point clouds with obvious differences in spatial distribution. More specifically, our method mainly deals with the trees with larger crown coverage and smaller trunk space. Therefore, generally speaking, the proposed method is more suitable for the large broad-leaved forest, such as the study area of this work, the Shaoguan, located in Guangdong, China, which belongs to the subtropical monsoon climate zone. For the common trees in other areas, such as coniferous forest in the north, our method will be limited due to the small difference in vertical spatial distribution. Secondly, since our method is designed mainly for individual tree detection, it cannot be used directly for more detailed analysis. However, the results obtained by our method can be used for downstream tasks. For example, building on our method, several parameters, such as the tree height and crown size, can be obtain by designing inverse mapping from 2D image to 3D point cloud, which would be an important topic in our future work. Thirdly, it also needs to point out that the point cloud quality has important influence on the performance of tree detection. For example, the point cloud quality, especially the point cloud density, has a significant impact on the second module, i.e., MCR module. More specifically, for the computation of density channel in the algorithm 1, fixing the image resolution r, if the point cloud density is low, the mapping image will be difficult to accurately preserve the distribution pattern of the point cloud. In future work, we will explore an adaptive resolution setting method, so that the mapping image resolution r can adapt to the point cloud density. In addition, as shown in Table IX, comparing with Mask-RCNN, because of the use of multiply branches, our method has lower time efficiency. Besides, as a data-driven approach, the performance of our method would be dependent heavily on the requirement of large number of labeled samples, which would consume a lot of time and labor costs. Therefore, as the further work, generating more samples from real forest environments and investigating unsupervised methods, such as domain adaptation, will be focused.

References

[1] F. H. Wagner, M. P. Ferreira, A. Sanchez, M. C. Hirye, M. Zortea, E. Gloor, O. L. Phillips, C. R. Filho, Y. E. Shimabukuro, and L. E. Aragao, "Individual tree crown delineation in a highly diverse tropical forest using very high resolution satellite images," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 362–377, 2018.

[2] F. E. Gonzalez, U. Dieguez-Aranda, L. Barreiro-Fernandez, S. Bujan, M. Barbosa, J. Suarez, I. Bye, and D. Miranda, "A mixed pixel- and region-based approach for using airborne laser scanning data for individual tree crown delineation in pinus radiata d. don plantations," *ISPRS J. Photogramm. Remote Sens.*, vol. 34, pp. 7671–7690, 2013.

[3] H. Lee, K. Slatton, B. Roth, and W. Cropper, "Adaptive clustering of airborne lidar data to segment individual tree crowns in managed pine forests," *Int. J. Remote Sens.*, vol. 31, pp. 117–139, 2010.

[4] A. Chang, Y. Eo, Y. Kim, and Y. Kim, "Identification of individual tree crowns from lidar data using a circle fitting algorithm with local maxima and minima filtering," *Remote Sens. Lett.*, vol. 4, pp. 29–37, 2012.

[5] J. White, N. Coops, M. Wulder, M. Vastaranta, T. Hilker, and P. Tompalski, "Remote sensing technologies for enhancing forest inventories: A review," *Can. J. Remote Sens.*, vol. 42, no. 5, pp. 619–641, 2016.

[6] R. Pack, V. Brooks, J. Young, N. Vilac? a, S. Vatslid, P. Rindle, S. Kurz, C. Parrish, R. Craig, and P. Smith, "An overview of als technology," pp. 7–98, 2012.

[7] P. Tompalski, J. White, N. Coops, and M. Wulder, "Demonstrating the transferability of forest inventory attribute models derived using airborne laser scanning data," *Remote Sens. Environ.*, vol. 227, pp. 110–124, 2019.

[8] S. Xu, G. Vosselman, and S. Oude Elberink, "Multiple-entity based classification of airborne laser scanning data in urban areas," *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 1–15, 2014.

[9] D. Mongus, N. Lukac¡¦, and B. Z¡¦alik, "Ground and building extraction from lidar data based on differential morphological profiles and locally fitted surfaces," *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 145–156, 2014.

[10] W. G. B. D. Nurunnabi, A., "Robust locally weighted regression techniques for ground surface points filtering in mobile laser scanning three dimensional point cloud data," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 4, pp. 2181–2193, 2016.

[11] S. Popescu, R. Wynne, and R. Nelson, "Measuring individual tree crown diameter with lidar and assessing its in?uence on estimating forest volume and biomass," *Can. J. Remote Sens.*, vol. 29, pp. 564–577, 2003.

[12] M. Maltamo, K. Eerik?inen, J. Pitk?nen, J. Hyyppa, and M. Vehmas, "Estimation of timber volume and stem density based on scanning laser altimetry and expected tree size distribution functions," *Remote Sens. Environ.*, vol. 90, no. 3, pp. 319–330, 2004.

[13] C. Lin, G. Thomson, C.-S. Lo, and M.-S. Yang, "A multi-level morphological active contour algorithm for delineating

tree crowns in mountainous forest," *Photogramm. Eng. Rem. S.*, vol. 77, pp. 241–249, 2011.

[14] L. Ene, E. N?sset, and T. Gobakken, "Single tree detection in heterogeneous boreal forests using airborne laser scanning and area-based stem number estimates," *Int. J. Remote Sens.*, vol. 33, no. 16, pp. 5171–5193, 2012.

[15] R. Palenichka, F. Doyon, A. Lakhssassi, and M. B. Zaremba, "Multi-scale segmentation of forest areas and tree detection in lidar images by the attentive vision method," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1313–1323, 2013.

[16] W. Li, Q. Guo, M. Jakubowski, and M. Kelly, "A new method for segmenting individual trees from the lidar point cloud," *Photogramm. Eng. Rem. S.*, vol. 78, pp. 75–84, 2012.

[17] F. Hosoi, Y. Nakai, and K. Omasa, "3-D voxel-based solid modeling of a broad-leaved tree for accurate volume estimation using portable scanning lidar," *ISPRS J. Photogramm. Remote Sens.*, vol. 82, pp. 41–48, 2013.

[18] X. Lu, Q. Guo, W. Li, and J. Flanagan, "A bottom-up approach to segment individual deciduous trees using leaf-off lidar point cloud data," *ISPRS J. Photogramm. Remote Sens.*, vol. 94, pp. 1–12, 2014.

[19] C. Vega, A. Hamrouni, S. E. Mokhtari, J. B. Morel, and J., "Ptrees: A point-based approach to forest tree extraction from lidar data," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 33, no. 1, pp. 98–108, 2014.

[20] A. M. Ramiya, R. R. Nidamanuri, and R. Krishnan, "Individual tree detection from airborne laser scanning data based on supervoxels and local convexity," *Remote Sens. Environ.*, vol. 15, pp. 100–242, 2019.

[21] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, 2012, pp. 1097–1105.

[23] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE T. Pattern Anal.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[24] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 6230–6239.

[25] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2015, pp. 1440–1448.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.

[27] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 206–215.

[28] Y. Wang, Y. Sun, Z. Liu, S. Sarma, M. Bronstein, and J. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 146.1–146.12, 2019.

[29] A. Gressin, C. Mallet, J. Demantke, and N. David, "Towards 3d lidar point cloud registration improvement using optimal neighborhood knowledge," *ISPRS J. Photogramm. Remote Sens.*, vol. 79, pp. 240–251, 2013.

[30] H. Lin, J. Chen, P. Su, and C. Chen, "Eigen-feature analysis of weighted covariance matrices for lidar point cloud classi?cation," *ISPRS J. Photogramm. Remote Sens.*, vol. 94, pp. 70–79, 2014.

[31] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 1912–1920.

[32] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *IEEE Int. Conf. Intell. Robots Sys. (RIOS)*, 2015, pp. 922–928.

[33] H. Su, S. Maji, E. Kalogerakis, and E. Learnedmiller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2015, pp. 945–953.

[34] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 6526–6534.

[35] C. Ma, Y. Guo, J. Yang, and W. An, "Learning multi-view representation with lstm for 3-d shape recognition and retrieval," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1169–1182, 2018.

[36] 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)*, 2016, pp. 77–85.

[37] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5099–5108.

[38] H. You, Y. Feng, R. Ji, and Y. Gao, "Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition," in *ACM MM.*, 2018, pp. 1310–1318.

[39] Z. Luo, J. Li, Z. Xiao, G. Mou, X. Cai, and C. Wang, "Learning high-level features by fusing multi-view representation of mls point clouds for 3d object recognition in road environments," *ISPRS J. Photogramm. Remote Sens.*, vol. 150, pp. 44–58, 2019.

[40] A. Nurunnabi, G. West, and D. Belton, "Outlier detection and robust normal-curvature estimation in mobile laser scanning 3d point cloud data," *Pattern Recognit.*, vol. 48, no. 4, pp. 1404–1419, 2015.

[41] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv:1511.07122.*, 2016.

[42] C. Szegedy, V. Vanhoucke, S. S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2818–2826.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Med. Image Comput. Assis. Interv. (MICCAI)*, 2015, pp. 234–241.

[44] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *4th Deep Learning in Medical Image Analysis (DLMIA) Workshop*, 2018, pp. 3–11.

[45] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv:1606.02147.*, 2016.

[46] B. D. Brabandere, D. Neven, and L. Gool, "Semantic instance segmentation for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, 2017, pp. 478–480.

[47] W. Zhang, J. Qi, W. Peng, H. Wang, D. Xie, X. Wang, and G. Yan, "An easy-to-use airborne lidar data filtering method based on cloth simulation," *Remote Sensing*, vol. 8, no. 6, pp. 501.1–501.22, 2016.

[48] G. Sithole and G. Vosselman, "Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 59, no. 1-2, pp. 85–101, 2004.

[49] R. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote Sens. Environ.*, vol. 37, no. 1, pp. 35–46, 1991.

[50] J. Silvan-Cardenas and L. Wang, "A multi-resolution approach for filtering lidar altimetry data," *ISPRS J. Photogramm. Remote Sens.*, vol. 61, no. 1, pp. 11–22, 2006.

[51] T. Pingel, K. Clarke, and W. McBride, "An improved simple morphological filter for the terrain classification of airborne

lidar data," *ISPRS J. Photogramm. Remote Sens.*, vol. 77, pp. 21–30, 2013.

[52] C. Chen, Y. Li, W. Li, and H. Dai, "A multi-resolution hierarchical classification algorithm for filtering airborne lidar data," *ISPRS J. Photogramm. Remote Sens.*, vol. 82, pp. 1–9, 2013.

[53] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255.

[54] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask RCNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2980–2988.

Zhipeng Luo (S'19) received the Ph.D. degree in photogrammetry and remote sensing from Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China in July 2020.

He is now a postdoctoral researcher with The Hong Kong Polytechnic University. His current research interests include autonomous driving, mobile laser scanning, intelligent processing of point clouds, 3D computer vision, and machine learning.

Ziyue Zhang received the Bachelor of Science with Honors in Computer Science with Artificial Intelligence from the University of Nottingham, Ningbo China.

His research interests include LiDAR data processing, three-dimensional computer vision, graph neural network, explainable neural network, and machine learning.

Wen Li (S'20) received the B.Eng. degree in communication engineering from the School of Information Science and Technology, Shandong Agricultural University, Taian, China. He is currently working toward the Ph.D. degree in computer science and technology with the Fujian Key Laboratory Sensing and Computing for Smart Cities and School of Informatics, Xiamen University, Xiamen, China.

His research interests include LiDAR data processing, three-dimensional computer vision, and machine learning.

Yiping Chen (Senior Member, IEEE) received the Ph.D. degree in information and communications engineering from the National University of Defense Technology, Changsha, China, in 2011.

She is a Senior Engineer with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, China. From 2007 to 2011, she was an Assistant Researcher with The Chinese University of Hong Kong, Hong Kong. Her research interests include image processing, mobile laser scanning data analysis, point cloud computer vision, and autonomous driving.

Cheng Wang (M'07-SM'16) received the Ph.D. degree in signal and information processing from the National University of Defense Technology, Changsha, China, in 2002.

He is currently a Professor with the School of Informatics, and the Executive Director with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen, China. He has coauthored more than 150 papers in referred journals and top conferences including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, PR, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE Conference on Computer Vision and Pattern Recognition, Association for the Advancement of Artificial Intelligence (AAAI), and *International Society for Photogrammetry and Remote Sensing (ISPRS) Journal of Photogrammetry and Remote Sensing*. His current research interests include point cloud analysis, mobile mapping, and geospatial big data.

Prof. Wang is a Fellow of the Institution of Engineering and Technology and Associate Editor of IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. He is also the Chair of the Working Group I/6 on Multi-Sensor Integration and Fusion of the ISPRS.

Abdul Awal Md Nurunnabi (M'00-SM'11) received the B.Sc., M.Sc., and M.Phil. degrees from the University of Rajshahi, Bangladesh, in 1997, 1998, and 2008, respectively, in statistics, and the Ph.D. degree in spatial sciences from Curtin University, Australia, in 2015. He was funded by the prestigious International Postgraduate Research Scholarship (IPRS) for his Ph.D. He also obtained a top-up scholarship from the Cooperative Research Centre for Spatial Information, Australia. His Ph.D. research was on robust statistical approaches for feature extraction in laser scanning 3D point clouds. He was offered several Ph.D. research scholarships and postdoc positions in well reputed universities. He was awarded a Japan Society for the Promotion of Science (JSPS) postdoctoral research scholarship at The University of Tokyo, Japan, May 2016-April 2018. Presently, he is a researcher for the SOLSTICE project at the Department of Geodesy and Geospatial Engineering, University of Luxembourg, Luxembourg.

Dr Nurunnabi has authored over 60 peer reviewed papers in international high-impact journals including Pattern Recognition, ISPRS Journal of Photogrammetry and Remote Sensing and IEEE TGRS, and conference proceedings. His research interests include outlier analysis, robust statistics, feature extraction, 3D modelling, point cloud processing, pattern recognition, machine learning, deep learning, computer vision, data mining, photogrammetry, and remote sensing. He was a recipient of several publications and travel grant awards, including the best paper award in the Journal of Applied Statistics in 2010. He is a regular reviewer for several high impact journals.

Jonathan Li (Senior Member, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa, in 2000. He is currently a Professor with the Department of Geography and Environmental Management and cross-appointed with the Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada. He is also a Founding Member of the Waterloo Artificial Intelligence Institute.

He has coauthored more than 500 publications, over 300 of which were published in refereed journals, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, ISPRS Journal of Photogrammetry and Remote Sensing, and Remote Sensing of Environment. His research interests include artificial intelligence (AI) techniques for information extraction from LiDAR point clouds and Earth observation images and their applications in geospatial mapping, transportation, and urban digital twins. He is President-elect of Canadian Institute of Geomatics (CIG), Chair of the ISPRS WG I/2 on LiDAR, Air- and Space-borne Optical Sensing from 2016 to 2022 and the ICA Commission on Sensor-Driven Mapping from 2015 to 2023. He is Editor-in-Chief of the International Journal of Applied Earth Observation and Geoinformation, an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and Canadian Journal of Remote Sensing. Dr. Li is a recipient of the 2021 CIG Geomatica Award, the 2020 ISPRS Samuel Gamble Award, and the 2019 Outstanding Achievement Award in Mobile Mapping Technology.