Remote Sensing Letters

ISSN: 2150-704X (Print) 2150-7058 (Online) Journal homepage: http://www.tandfonline.com/loi/trsl20

Deep learning-based tree classification using mobile LiDAR data

Haiyan Guan, Yongtao Yu, Zheng Ji, Jonathan Li & Qi Zhang

To cite this article: Haiyan Guan, Yongtao Yu, Zheng Ji, Jonathan Li & Qi Zhang (2015) Deep learning-based tree classification using mobile LiDAR data, Remote Sensing Letters, 6:11, 864-873, DOI: 10.1080/2150704X.2015.1088668

To link to this article: http://dx.doi.org/10.1080/2150704X.2015.1088668

Remote Sensing Letters, 2015
Vol. 6, No. 11, 864–873, http://dx.doi.org/10.1080/2150704X.2015.1088668

Taylor & Francis
Taylor & Francis Group

Deep learning-based tree classification using mobile LiDAR data

Haiyan Guan[a]*, Yongtao Yu[b], Zheng Ji[c], Jonathan Li[b], and Qi Zhang[d]

[a]*College of Geography and Remote Sensing, Nanjing University of Information Science & Technology, Nanjing, China;* [b]*Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen, China;* [c]*School of Remote Sensing & Information Engineering, Wuhan University, Wuhan, China;* [d]*Public Administration College, Zhejiang Gongshang University, Hangzhou, China*

(*Received 7 February 2015; accepted 22 August 2015*)

Our work addresses the problem of extracting and classifying tree species from mobile LiDAR data. The work includes tree preprocessing and tree classification. In tree preprocessing, voxel-based upward-growing filtering is proposed to remove ground points from the mobile LiDAR data, followed by a tree segmentation that extracts individual trees via Euclidean distance clustering and voxel-based normalized cut segmentation. In tree classification, first, a waveform representation is developed to model geometric structures of trees. Then, deep learning techniques are used to generate high-level feature abstractions of the trees' waveform representations. Quantitative analysis shows that our algorithm achieves an overall accuracy of 86.1% and a kappa coefficient of 0.8 in classifying urban tree species using mobile LiDAR data. Comparative experiments demonstrate that the uses of waveform representation and deep Boltzmann machines contribute to the improvement of classification accuracies of tree species.

1. Introduction

In the case of urban areas, tree species classification is gaining increasing attention for safety studies, noise modelling, and environmental and ecological analysis because trees play a critical role in urban ecosystems for the maintenance of environmental quality, aesthetic beauty of urban landscape, and social service for inhabitants (Tooke et al. 2009; Zhang and Hu 2012). As cities grow rapidly, urban forests are increasingly displaced by infrastructure. Therefore, municipal governments desire to control development near greenbelt areas by using land cover maps (Jim and Liu 2001; Sugumaran, Pavuluri, and Zerr 2003). Moreover, biodiversity parameters such as tree species and age and height distributions are primarily chosen for ecosystem analysis (Maas 2010).

Since last decade, high-sampling density LiDAR data have been widely used for single tree extraction and tree species classification with explorations of varying algorithms (Maas 2010; Puttonen et al. 2011, Dinuls et al. 2012; Li, Hu, and Noland 2013). Currently, retrieval of tree structural attributes are largely from airborne LiDAR data (Jones, Coops, and Sharma 2012; Chang et al. 2013; Kwaka et al. 2014). However, airborne LiDAR systems mainly capture the tops of structures and have potentially limited access to objects under the canopy, leading to incomplete data coverage of trees. Compared to terrestrial and mobile LiDAR systems, airborne LiDAR systems have relatively low-point sampling densities. Because terrestrial LiDAR systems capture very

*Corresponding author. Email: guanhy.nj@nuit.edu.cn, guanhy.nj@gmail.com

dense 3D representations of tree topologies and local forest structures, terrestrial LiDAR is also a very valuable means for forest attribute measurements on single tree and plot levels. However, terrestrial LiDAR systems frequently capture data from multiple views, leading to a relatively time-consuming data acquisition process.

Mobile LiDAR has attracted much attention for urban vegetation detection and modelling (Jaakkola et al. 2010) because it acquires data at a much higher point density and more complete data coverage than an airborne LiDAR system and at a higher efficiency than a terrestrial LiDAR system. Consequently, studies have been conducted using mobile LiDAR data. 3D segmentation methods, such as 3D Hough transform (Rutzinger et al. 2010), minimum spanning tree (Shen et al. 2008), and stepwise voxel-based marked neighbourhood searching (Wu et al. 2013), were developed for identifying street trees from scattered mobile LiDAR points. In addition, algorithms combining mobile LiDAR data with digital images or videos captured by on-board digital camera(s) or video camera(s) were also exploited for tree detection (Puttonen et al. 2011; Zhong et al. 2013). However, less attention is paid to tree species classification from mobile LiDAR data. Thus, the importance of single-tree-based information for updating urban tree species maps is the motivation for developing algorithms from the unprecedented detailed 3D mobile LiDAR data for tree detection and species identification in this letter.

In this letter, we propose a tree classification method, which applies a deep learning technique to waveform representations of 3D tree points. The proposed method includes tree preprocessing and tree classification. In tree preprocessing, ground points are first removed from mobile LiDAR data through voxel-based upward-growing filtering; then individual trees are isolated from the filtered off-ground points via Euclidean distance clustering and voxel-based normalized cut segmentation. In tree classification, the extracted individual trees are profiled in height to form waveform representations. Based on waveform representation, deep learning using deep Boltzmann machines (DBMs) is used to generate high-level feature abstractions, which are further classified by a support vector machine (SVM) classifier.

2. Methodology

2.1. *Tree preprocessing*

2.1.1. *Ground point removal*

Usually, most filtering methods, which were originally developed for airborne LiDAR data, assume that the lowest point in a neighbourhood is a ground point. However, compared to the looking-down view patterns of airborne LiDAR systems, which are more likely to generate uniform point densities, mobile LiDAR systems with side-view patterns collect very dense data close to the scanner path and less dense data farther away from the scanner path. Points belonging to road surface account for a great portion of the collected mobile LiDAR data. Thus, the established filtering algorithms are unsuitable for retrieving non-ground points from mobile LiDAR data. To improve computational efficiency, we develop a rapid and effective method, namely voxel-based upward-growing filtering, for removing ground points. This method is implemented as follows:

(1) Grid the entire point cloud into a set of data blocks, D_j ($j = 0, 1, \ldots N$, where N is the number of data blocks), with a block size of w_b in the XY plane, as shown in Figure 1(*a*). The block size is determined by ground fluctuations of the study areas of interest.

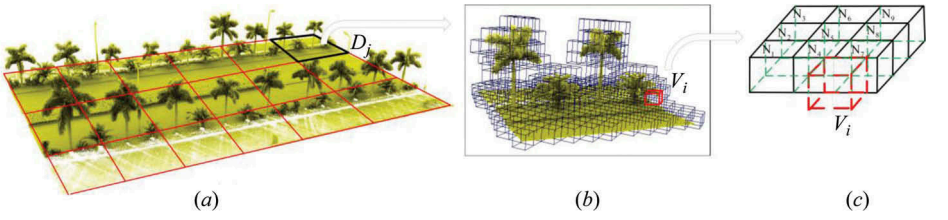

Figure 1. (*a*) Gridding raw data points of a road section into data blocks in the XY plane, (*b*) octree partition structure for block D_j, and (*c*) nine upper neighbours (N_1 to N_9) for each voxel v_i.

(2) Organize each data block, D_j, into an octree partition structure with a spacing of w_v to generate a set of voxels, v_i ($i = 0, 1, \ldots M$, where $M$is the number of voxels), as shown in Figure 1(*b*). The voxel size is empirically determined based on the point density of the acquired mobile LiDAR data and the computational efficiency of the proposed ground removal method.
(3) Each voxel, v_i, grows upward to its nine neighbours, N_1 to N_9, which are located above the voxel, as shown in Figure 1(*c*). Then, each neighbour continues to grow upward to its corresponding nine neighbours. The upward-growing process stops when no more voxels can be reached. Finally, a voxel, v_e, with the highest elevation within the grown region is determined to justify whether voxel v_i contains ground points or non-ground points based on the following criteria:
   (a) If the elevation of v_e lies below a predefined ground threshold, H_{thresh}, label all points in voxel v_i as ground points, which are further removed.
   (b) Otherwise, label all points in voxel v_i as non-ground points, which are retained.

The proposed voxel-based upward-growing filtering method has the advantage of rapidly and effectively handling large scenes with strong ground fluctuations. Figure 2(*a*) shows a visual example of the retrieved off-ground points after ground removal.

2.1.2. *Tree segmentation*

To rapidly group off-ground points into clusters representing individual objects, we use a Euclidean distance clustering approach, which clusters discrete points based on their Euclidean distances. Theoretically, an unclustered point is grouped into a specific cluster if and only if its shortest Euclidean distance to the points in this cluster lies below a threshold, d_t. Otherwise, a new cluster is created to contain this point. Such a Euclidean distance clustering approach starts at an unclustered off-ground point and then iteratively processes each unclustered point until all off-ground points are grouped into specific

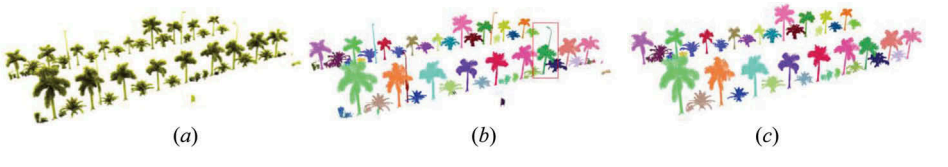

Figure 2. (*a*) Off-ground points after ground removal, (*b*) clustered off-ground points, and (*c*) detected individual trees after VNCut segmentation and prior knowledge filtering.

clusters. Through Euclidean distance clustering, most separated objects are successfully segmented (see Figure 2(*b*)). However, as shown by the red box in Figure 2(*b*), by using Euclidean distance clustering, some adjacent or overlapped objects cannot be well segmented.

Before tree classification, individual trees must be separated from the clusters containing multiple objects. In our previous study (Yu et al. 2015), we proposed a voxel-based normalized cut (VNCut) segmentation method, which effectively segments point cloud clusters, containing multiple adjacent or overlapped objects, into separated objects. Thus, in this letter, VNCut is used to segment individual trees.

First, by using the octree partition strategy, the clusters, containing more than one object, are partitioned into a voxel structure with a certain voxel spacing. Then, the generated voxels are formed into a weighted graph, whose nodes are represented by the voxels and edges are connected between each pair of voxels. The weight on the edge, measuring the similarity of the connected voxels, is computed using the features associated with the voxels. Next, after solving a generalized eigenvalue problem with respect to the weighted graph, a set of eigenvalues and their associated eigenvectors are obtained. Finally, by applying a threshold to the eigenvector associated with the second smallest eigenvalue, a cluster is divided into two components. By using VNCut segmentation, the clusters containing more than one object are effectively segmented into individual objects. After segmentation, prior knowledge (e.g. crown size) is used to filter out non-tree objects such as light poles. Figure 2(*c*) shows a visual example of the detected individual trees.

2.2. *Tree classification*

2.2.1. *Waveform generation*

The extracted individual trees are then profiled vertically to generate waveform representations. In elevation, using a predefined waveform dimension, n, a tree is subdivided vertically into n profiles, within each of which points are counted, and then the values of point statistics are normalized at the continuous range of 0–1. Figure 3 shows three subtropical tree types. The left figures are tree models in the form of 3D points, and the right figures are their corresponding waveform representations based on point statistics. As seen in Figure 3, the generated waveform data deliver a good representation of tree's geometric structures. Similar to real waveform digitalization, which delivers a full-intensity profile of the echo of a laser pulse, the proposed waveform representation offers a unique geometric profile of a tree. We classify trees directly from waveform data, rather than indirect information or parameters deduced from waveform data, which reduces classification errors caused by human interpretation.

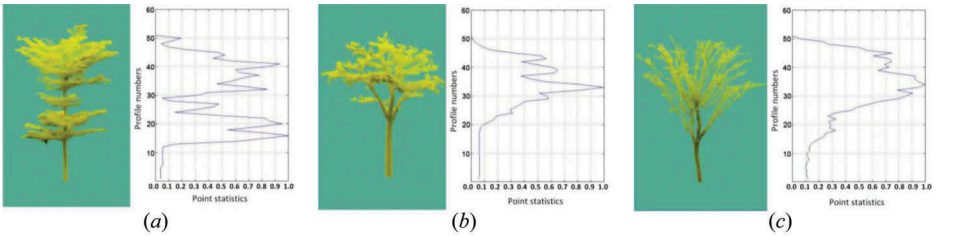

Figure 3. Tree samples and their corresponding waveform data: (*a*) tree 1, (*b*) tree 2, and (*c*) tree 3.

2.2.2. Tree classification

Recently, deep learning techniques have become attractive for their superior performance in learning hierarchical features from high-dimensional unlabelled data. By learning multilevel feature representations, deep learning models have been proved to be an effective tool for rapid object-oriented classification problems. DBMs (Salakhutdinov, Tenenbaum, and Torralba 2013) are an important breakthrough in the requirement for powerful deep feature representation models. In this letter, we construct a deep feature generation model for generating high-level feature abstractions of the trees' waveforms using a DBM model.

As shown in Figure 4(*a*), we construct a two-layer DBM model. Denote $\boldsymbol{v} \in [0,1]^n$ as the real-valued visible units representing a tree's waveform. Denote $\boldsymbol{h}^1 \in \{0,1\}^{F_1}$ and $\boldsymbol{h}^2 \in \{0,1\}^{F_2}$ as the lower- and higher-layer hidden units, respectively. F_1 and F_2 are the numbers of hidden units in the lower and higher hidden layers, respectively. Then, the energy of the configuration, $\{\boldsymbol{v}, \boldsymbol{h}^1, \boldsymbol{h}^2\}$, is defined as follows:

$$E(\boldsymbol{v}, \boldsymbol{h}^1, \boldsymbol{h}^2; \phi) = \frac{1}{2}\sum_{i=1}^{n}\frac{v_i^2}{\sigma_i^2} - \sum_{i=1}^{n}\sum_{j=1}^{F_1}\frac{v_i}{\sigma_i}w_{ij}^1 h_j^1 - \sum_{j=1}^{F_1}\sum_{m=1}^{F_2} h_j^1 w_{jm}^2 h_m^2, \tag{1}$$

where $\phi = \{\mathbf{W}^1, \mathbf{W}^2, \sigma\}$ are the model parameters. v_i is the ith element of $\boldsymbol{v}$; h_j^1 is the jth element of $\boldsymbol{h}^1$; h_m^2 is the mth element of $\boldsymbol{h}^2$. $\mathbf{W}^1$and $\mathbf{W}^2$are the visible-to-hidden and hidden-to-hidden symmetric interaction terms, respectively. w_{ij}^1 is the element on the ith row and jth column of $\mathbf{W}^1$.w_{jm}^2 is the element on the jth row and mth column of $\mathbf{W}^2$. σ^2 represents the variances of the visible units. σ_i is the ith element of $\boldsymbol{\sigma}$. The conditional distributions over the visible and two sets of hidden units are expressed as follows:

$$p(h_j^1 = 1|\boldsymbol{v}, \boldsymbol{h}^2) = g\left(\sum_{i=1}^{n}\frac{v_i}{\sigma_i}w_{ij}^1 + \sum_{m=1}^{F_2} h_m^2 w_{jm}^2\right), \tag{2}$$

$$p(h_m^2 = 1|\boldsymbol{h}^1) = g\left(\sum_{j=1}^{F_1} h_j^1 w_{jm}^2\right), \tag{3}$$

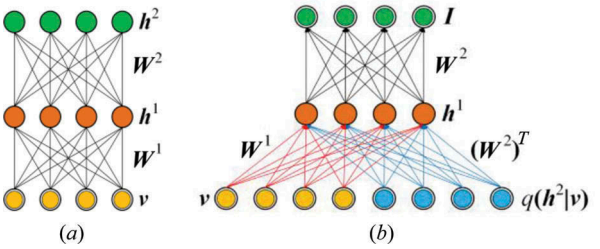

Figure 4. (*a*) Two-layer DBM model and (*b*) deep feature generation model.

$$p(v_i = x|\boldsymbol{h}^1) = \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{\left(x - \sigma_i \sum_{j=1}^{F_1} h_j^1 w_{ij}^1\right)}{2\sigma_i^2}\right), \tag{4}$$

where $g(x) = 1/(1 + e^{-x})$ is the logistic function (Salakhutdinov, Tenenbaum, and Torralba 2013).

To rapidly, effectively train the model parameters, ϕ, first, a greedy layer-wise pre-training is performed to initialize the model parameters. Then, an iterative training algorithm combined with variational and stochastic approximation approaches (Salakhutdinov, Tenenbaum, and Torralba 2013) is used to fine-train the model parameters.

Once the model parameters are trained, the stochastic activities of binary features in the hidden layers of the DBM are replaced by real-valued probability estimations to construct a deep feature generation model (Figure 4(*b*)). Considering the feedbacks from hidden layers, for each visible vector $\boldsymbol{v}$, mean-field inference (Salakhutdinov and Hinton 2012) is adopted to generate an approximate posterior distribution, $Q(\boldsymbol{h}^2|\boldsymbol{v})$. Then, the marginal, $q(\boldsymbol{h}^2|\boldsymbol{v})$ of the approximate posterior serves as an augment to the deep feature generation model. Finally, the top layer of the deep feature generation model produces the following high-level feature representation:

$$\boldsymbol{I}^{\mathrm{T}} = g\left(g\left(\frac{\boldsymbol{v}^{\mathrm{T}}}{\sigma^{\mathrm{T}}}\mathbf{W}^1 + q(\boldsymbol{h}^1|\boldsymbol{v})^{\mathrm{T}}(\mathbf{W}^2)^{\mathrm{T}}\right)\mathbf{W}^2\right) \in [0, 1]^{F_2}. \tag{5}$$

To classify the segmented trees into specific tree species, first, the trees' waveforms are characterized by the deep feature generation model to generate high-level feature representations. Then, by using an SVM classifier, the generated high-level feature representations are classified into specific classes. In this way, all the trees are classified into specific tree species.

3. Results and conclusion

In this study, using a RIEGL VMX-450 mobile LiDAR system, the mobile LiDAR data were acquired along the urban roads in Xiamen City, China. The specification of the RIEGL VMX-450 system claims that the system can achieve a maximum effective measurement rate of 1.1 million points per second and a scan speed of 400 lines per second. In this letter, point density stands for the number of points per square metre and sharply drops perpendicular to the line of travel. For example, with a vehicle driving speed of 50 km hour^{-1}, the average point density is around 250–500 points m^{-2}. For urban landscaping, a total number of 10 different species of trees (including T1: *Elaeocarpus apiculatus* mast; T2: *Sago cycas*; T3: palm; T4: *Roystonea regia*; T5: *Bischofia polycarpa*; T6: *Delonix regia*; T7: *Euonymus japonicas*; T8: mango; T9: *Ficus microcarpa*; and T10: *Ficus macrophylla*) are planted along both sides of the road corridor, as shown by the examples in Figure 5. At training stage, 50,000 tree samples from 10 different tree species, each of which includes 5000 tree samples, were selected for training the deep feature generation model and the SVM classifier. At test stage, from the collected data, a road segment of approximately 4 km, containing 2013 trees from 10 species, was selected for evaluating the proposed classification algorithm.

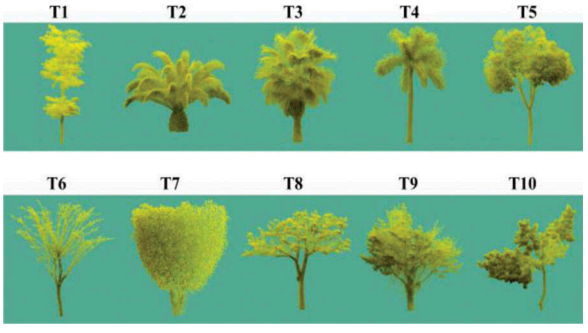

Figure 5. Illustration of 10 different tree species (termed as T1, T2, ... T10) in the study area.

3.1. *Tree classification*

We applied the proposed algorithm to the selected point cloud data set to classify tree species. Through parameter sensitivity analysis on computational and classification performances, the parameters and their configurations used in this study are listed in Table 1. In practice, first, the selected point cloud data set was partitioned into a set of data segments with a road length of about 50 m. Then, in the tree preprocessing step, each data segment was handled separately to segment individual trees. The block size (w_b) and voxel size (w_v) were set to 3 m and 5 cm, respectively. The ground threshold (H_{thresh}) was set to 0.3 m for removing ground points. In addition, the clusters were created from the extracted off-ground points based on a clustering threshold (d_t) of 0.15 m.

Next, waveform representations were generated from the segmented individual trees with the given waveform dimension, $n = 150$. Finally, the segmented individual trees were classified into different tree species based on the high-level features of their waveform representations. The numbers of hidden units in the lower and higher hidden layers (F_1 and F_2) of the DBM were set to 500 and 50, respectively. In the letter, a Gaussian radial basis function (RBF) kernel was chosen for our SVM classifier since RBF kernels yield extremely high-accuracy rates for the most challenging high-dimensional image classifications. Two parameters should be specified while using the RBF kernels: penalty parameter (C) and kernel function (γ). We selected the SVM parameters empirically by trying a finite number of values and keeping those that provide the least test error. The optimal values of C and γ were found to be 300 and 0.4, respectively. In the selected data set, there are a total number of 2013 trees from 10 different tree species. Figure 5 illustrates some tree examples of the 10 different tree species. The tree species classification results are illustrated in Table 2 by using a confusion matrix. The overall tree classification accuracy is 86.1% and the kappa value is 0.8. The producer's accuracies for all trees are >81%, and the user's accuracies for all trees are >78%. The misclassification errors were mainly caused by the waveform similarities of some trees among different species; thus, these trees were falsely classified into other species.

Table 1. Parameters and their configurations used in this study.

w_b	w_v	H_{thresh}	d_t	n	F_1	F_2
3 m	0.05 m	0.3 m	0.15 m	150	500	50

Table 2. Overall results of classification accuracies of 10 tree species (see Figure 5) by using confusion matrix, showing the numbers of classified versus validation trees.

		Validation trees											Accuracy
	Species	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Total	UA (%)
Classified trees	T1	180	3	3	2	5	11	0	7	0	2	210	85.7
	T2	3	165	4	3	0	0	15	0	0	0	190	86.8
	T3	2	3	159	4	0	5	12	2	2	0	189	84.1
	T4	4	3	2	186	3	4	0	2	4	2	210	88.6
	T5	0	0	5	3	185	2	3	0	3	2	203	91.1
	T6	15	3	0	3	4	165	3	8	8	2	211	78.2
	T7	0	17	8	0	3	2	161	2	0	0	193	83.4
	T8	3	1	1	4	3	2	2	176	0	16	208	84.6
	T9	0	2	0	0	1	1	0	2	179	1	186	96.2
	T10	3	5	3	4	0	1	2	12	5	178	213	83.6
	Total	210	199	185	209	204	193	198	211	201	203	2013	
Accuracy	PA (%)	85.7	82.9	85.9	89.0	90.7	85.5	81.3	83.4	89.1	87.7		
	OA (%) = 86.1; kappa coefficient = 0.8												

Note: PA, producer's accuracy; UA, user's accuracy; OA, overall accuracy.

3.2. *Comparative experiments*

In this study, a DBM-based deep feature generation model is used to generate high-level feature abstractions of the tree's waveform data generated from mobile LiDAR data. Thus, to demonstrate the superior performance of the deep feature generation model, we compared it with the following three methods.

Method 1 – in which the waveform representation is directly used as features. The waveform dimension, n, was set to 150 to generate a 150-dimensional (150-D) feature vector based on point statistics.

Method 2 – in which a total of 134 features covering four types of LiDAR features were designed, including (1) the 3D texture of a tree (14 features), (2) the relative degree of foliage clustering (80 features for 20 height layers), (3) the relative scale (20 features), and (4) the gap distribution within a tree crown (20 features). For more details about the used features, refer to Li, Hu, and Noland (2013).

Method 3 – in which the proposed DBM-based feature generation model is used to learn high-level feature abstractions from the total of 134 features in method 2.

In pursuing the objectives of this study, an SVM classifier was used in methods 1–3. The comparative results are detailed in Table 3. As seen in Table 3, method 1 achieved the lowest classification accuracies with an overall accuracy of 73.4% and a kappa value of 0.7. This is caused by the low-level, low-distinctiveness features of the 150-D waveform representation. Method 2 outperformed method 1 with an overall accuracy of 79.2% and a kappa value of 0.8. The producer's accuracies for all species were between 71.9% and 83.1%, and the user's accuracies for all species were between 75.6% and 82.4%. Moreover, method 2 achieved better classification accuracies than those reported in Li, Hu, and Noland (2013). This is because the point density of the used mobile LiDAR data is larger than that of the airborne LiDAR data in Li, Hu, and Noland (2013). The airborne LiDAR points cover tree canopies and miss the most structure of a tree under the canopy, while the mobile LiDAR data almost cover a complete structure of a tree. Compared to methods 1 and 2, method 3 achieved relatively higher classification accuracies. This might be benefited from the use of the deep feature generation model for generating high-level feature representations. Finally, the

Table 3. Comparative results of classification accuracies of four methods.

Method	OA (%)	Kappa coefficient	UA (%)		PA (%)	
			Minimum	Maximum	Minimum	Maximum
Method 1	73.4	0.7	65.1	77.3	63.6	80.9
Method 2	79.2	0.8	75.6	82.4	71.9	83.1
Method 3	83.1	0.8	76.5	91.2	80.0	91.3
Proposed method	86.1	0.8	78.2	96.2	81.3	90.7

Note: PA, producer's accuracy; UA, user's accuracy; OA, overall accuracy.

proposed tree classification method outperformed all the three methods. The reasons behind this phenomenon might be (1) the use of the DBM-based feature generation model for the high-level feature representation contributes to the improvement of classification accuracies, and (2) the use of waveform representation as data rather than classification feature vectors contributes to the complete description of a tree from mobile LiDAR data.

4. Conclusion

In this letter, we proposed a novel algorithm for classifying tree species in urban areas using mobile LiDAR data, which consists of (1) ground point removal and individual tree detection, and (2) waveform representation of individual trees and deep learning-based feature abstractions using DBMs for SVM classification. Our algorithm achieved an overall accuracy of 86.1% and a kappa coefficient of 0.8 on the selected 4-km urban road data set. In addition, comparative studies also demonstrated that the use of waveform representation and DBMs in the proposed algorithm obtained promising and high performance in classifying tree species from mobile LiDAR data. We will continue to develop novel waveform representations, with the goal of improving tree classification accuracy in support of urban forest attribute modelling.

Acknowledgement

The authors would like to thank Mr M. McAllister for his assistance in proofreading the article and the anonymous reviewers for their constructive comments and suggestions in improving the clarity of the article.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Startup Foundation for Introducing Talent of Nanjing University of Information Science & Technology (NUIST) and the National Natural Science Foundation of China [grant numbers 41501501, 41471379].

References

Chang, A., Y. Eo, Y. Kim, and Y. Kim. 2013. "Identification of Individual Tree Crowns from LiDAR Data Using a Circle Fitting Algorithm with Local Maxima and Minima Filtering." *Remote Sensing Letters* 4 (1): 29–37. doi:10.1080/2150704X.2012.684362.

Dinuls, R., G. Erins, A. Lorencs, I. Mednieks, and J. Sinica-Sinavskis. 2012. "Tree Species Identification in Mixed Baltic Forest using LiDAR and Multispectral Data." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5 (2): 594–603. doi:10.1109/JSTARS.2012.2196978.

Jaakkola, A., J. Hyyppä, A. Kukko, X. Yu, H. Kaartinen, M. Lehtomaki, and Y. Lin. 2010. "A Low-Cost Multi-Sensoral Mobile Mapping System and Its Feasibility for Tree Measurements." *ISPRS Journal of Photogrammetry and Remote Sensing* 65 (6): 514–522. doi:10.1016/j.isprsjprs.2010.08.002.

Jim, C., and H. Liu. 2001. "Species Diversity of Three Major Urban Forest Types in Guangzhou City, China." *Forest Ecology and Management* 146: 99–114. doi:10.1016/S0378-1127(00)00449-7.

Jones, T. G., N. C. Coops, and T. Sharma. 2012. "Assessing the Utility of LiDAR to Differentiate among Vegetation Structural Classes." *Remote Sensing Letters* 3 (3): 231–238. doi:10.1080/01431161.2011.559289.

Kwaka, D.-A., G. Cui, W.-K. Lee, H.-K. Cho, S. W. Jeon, and S.-H. Lee. 2014. "Estimating Plot Volume using LiDAR Height and Intensity Distributional Parameters." *International Journal of Remote Sensing* 35 (13): 4601–4629. doi:10.1080/01431161.2014.915592.

Li, J., B. Hu, and T. Noland. 2013. "Classification of Tree Species Based on Structural Features Derived from High Density LiDAR Data." *Agricultural and Forest Meteorology* 171-172: 104–114. doi:10.1016/j.agrformet.2012.11.012.

Maas, H. 2010. "Forestry Applications." Chapter 6. In *Airborne and Terrestrial Laser Scanning*, edited by G. Vosselman, and H.-G. Maas, 213–235. Scotland, UK: Whittles Publishing.

Puttonen, E., A. Jaakkola, P. Litkey, and J. Hyyppä. 2011. "Tree Classification with Fused Mobile Laser Scanning and Hyperspectral Data." *Sensors* 11: 5158–5182. doi:10.3390/s110505158.

Rutzinger, M., A. Pratihast, S. Oude Elberink, and G. Vosselman. 2010. "Detection and Modelling of 3D Trees from Mobile Laser Scanning Data." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 38 (Part 5): 520–525. Newcastle upon Tyne, UK, September 1–2.

Salakhutdinov, R., and G. Hinton. 2012. "An Efficient Learning Procedure for Deep Boltzmann Machines." *Neural Computing* 24 (8): 1967–2006.

Salakhutdinov, R., J. B. Tenenbaum, and A. Torralba. 2013. "Learning with Hierarchical-Deep Models." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (8): 1958–1971. doi:10.1109/TPAMI.2012.269.

Shen, Y., Y. Sheng, K. Zhang, Z. Tang, and S. Yan. 2008. "Feature Extraction from Vehicle-Borne Laser Scanning Data." Proceeding of SPIE 7285, International Conference on Earth Observation Data Processing and Analysis (ICEODPA), Wuhan, China, December 28–30. doi:10.1117/12.814958.

Sugumaran, R., M. Pavuluri, and D. Zerr 2003. "The Use of High-Resolution Imagery for Identification of Urban Climax Forest Species Using Traditional and Rule-Based Classification Approach." *IEEE Transactions on Geoscience and Remote Sensing* 41: 1933–1939. doi:10.1109/TGRS.2003.815384.

Tooke, T., N. Coops, N. Goodwin, and J. Voogt. 2009. "Extracting Urban Vegetation Characteristics using Spectral Mixture Analysis and Decision Tree Classifications." *Remote Sensing of Environment* 113: 398–407. doi:10.1016/j.rse.2008.10.005.

Wu, B., B. Yu, W. Yue, S. Shu, W. Tan, C. Hu, Y. Huang, J. Wu, and H. Liu. 2013. "A Voxel-Based Method for Automated Identification and Morphological Parameters Estimation of Individual Street Trees from Mobile Laser Scanning Data." *Remote Sensing* 5: 584–611. doi:10.3390/rs5020584.

Yu, Y., J. Li, H. Guan, C. Wang, and J. Yu. 2015. "Semiautomated Extraction of Street Light Poles from Mobile LiDAR Point-Clouds." *IEEE Transactions on Geoscience and Remote Sensing* 53 (3): 1374–1386. doi:10.1109/TGRS.2014.2338915.

Zhang, K., and B. Hu. 2012. "Individual Urban Tree Species Classification using Very High Spatial Resolution Airborne Multi-Spectral Imagery using Longitudinal Profiles." *Remote Sensing* 4: 1741–1757. doi:10.3390/rs4061741.

Zhong, R., J. Wei, W. Su, and Y. Chen. 2013. "A Method for Extracting Trees from Vehicle-Borne Laser Scanning Data." *Mathematical and Computer Modelling* 58: 733–742. doi:10.1016/j.mcm.2012.12.038.