AN EFFICIENT DEEP LEARNING APPROACH FOR GROUND POINT FILTERING IN AERIAL LASER SCANNING POINT CLOUDS

A. Nurunnabi[1 *], F. N. Teferle[1], J. Li[2], R. C. Lindenbergh[3], A. Hunegnaw[1]

[1] Geodesy and Geospatial Engineering, Faculty of Science, Technology and Medicine, University of Luxembourg, 6, rue Richard Codenhove-Kalergi, L-1359 Luxembourg − (abdul.nurunnabi, norman.teferle, addisu.hunegnaw)@uni.lu
[2] Geography and Environmental Management, University of Waterloo, Waterloo ON N2L 3G1,Canada − junli@uwaterloo.ca
[3] Geosciences and Remote Sensing, Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CN Delft, The Netherlands − r.c.lindenbergh@tudelft.nl

Commission I, WG I/2

KEY WORDS: Classification, CNN, Feature Extraction, LiDAR, Local Feature, Neural Network, PointNet, Semantic Analysis

ABSTRACT:

Ground surface extraction is one of the classic tasks in airborne laser scanning (ALS) point cloud processing that is used for three-dimensional (3D) city modelling, infrastructure health monitoring, and disaster management. Many methods have been developed over the last three decades. Recently, Deep Learning (DL) has become the most dominant technique for 3D point cloud classification. DL methods used for classification can be categorized into end-to-end and non end-to-end approaches. One of the main challenges of using supervised DL approaches is getting a sufficient amount of training data. The main advantage of using a supervised non end-to-end approach is that it requires less training data. This paper introduces a novel local feature-based non end-to-end DL algorithm that generates a binary classifier for ground point filtering. It studies feature relevance, and investigates three models that are different combinations of features. This method is free from the limitations of point clouds' irregular data structure and varying data density, which is the biggest challenge for using the elegant convolutional neural network. The new algorithm does not require transforming data into regular 3D voxel grids or any rasterization. The performance of the new method has been demonstrated through two ALS datasets covering urban environments. The method successfully labels ground and non-ground points in the presence of steep slopes and height discontinuity in the terrain. Experiments in this paper show that the algorithm achieves around 97% in both F_1-score and model accuracy for ground point labelling.

1. INTRODUCTION

Ground surface point filtering in laser scanned point clouds closely relates to generate a digital terrain model (DTM) or digital elevation model (DEM), which is essential for many applications including three-dimensional (3D) city modelling, urban infrastructure monitoring, flood and earthquake management, change detection, and corridor mapping (Weinmann et al., 2015; Vosselman et al., 2017; Zhang et al., 2018). Ground surface can be defined as the boundary surface between the solid ground and off-ground objects (such as buildings, trees, and bridges). Filtering the ground points means a classification of the laser scanning point clouds into ground (terrain) and non-ground (off-terrain) points. Filtering ground points and removing them from the non-ground points can reduce huge data volumes, saves cost and labour, and simplifies further relevant analysis such as segmentation and feature extraction for above ground objects modelling and scene understanding. Similar benefits arise from removing non-ground points when interested in getting information about ground points such as details about the road surface, kerbs, and footpaths in road environment.

Over the years, many ground point filtering methods have been developed in georeferenced laser scanning measurements known as point clouds. One of the first methods for filtering ground points in aerial laser scanning (ALS) point clouds was developed by Lindenberger (1993) that used the concept of mathematical morphology (Haralick and Shapiro, 1992). Later progressive densification and surface-based filtering approaches were developed for ground point filtering. The International Society of Photogrammetry and Remote Sensing (ISPRS), Working Group III, organized a performance test of eight different filtering algorithms (c.f., Sithole and Vosselman, 2004) in ALS point clouds. The test report concludes that although some methods perform better than the others in rural areas where terrain is flat, there are still a lot of opportunities for significant improvement of filtering methods in areas of complex urban environment and in the presence of steep slopes and height discontinuity.

The deep neural network (known as deep learning; DL) has been recognized as the most successful machine learning (ML) approach in applications of areas in computer vision, image processing, pattern recognition, semantic analysis and many more. Recently, DL has become the most dominant technique for 3D point cloud classification and semantic analysis. The most popular form of DL is the supervised learning approach that requires a sufficient number of reliable training data to learn the underlying data pattern of the unseen area. Two general categories of DL are end-to-end and non end-to-end approaches. The most common approach in DL is the Convolutional Neural Network (CNN), which brings an unprecedented success in image classification. Classic CNN architecture needs a regular data format like those of image grids or 3D voxels, unfortunately point clouds are not in a regular format. Most existing works typically transform such data to regular 3D voxel grids, however this renders the resulting data unnecessarily voluminous. One of the pioneering end-to-end DL works is PointNet (Qi et al., 2017a), used for segmentation and classification. This method

*Corresponding author

directly feeds a point cloud to the network without any transformation.

Alternatively, to the transformation into a regular format (like voxel grids), many researchers transform the raw data to get useful features and develop DL architectures based on the features. The feature-based DL architecture can be categorized as a non end-to-end approach. The main advantage of using a non end-to-end approach is that it requires less training data. However, feature-based approaches do depend on a clear understanding of the features and their relation to the underlying problem. Feature-based DL has been used in point cloud classification (Zhang et al., 2018; Kumar et al., 2019). Kumar et al. (2019) developed a multi-scale non end-to-end method for classification of terrestrial laser scanning (TLS) data. In (Kumar et al. 2019), because of multi-scaling, four normal vector based features (normal, plane-fit quality, eigenvalues and eigenvectors) are used a number of times. The authors claim that their method performs better than PointNet++ (Qi et al., 2017b), which is a multi-scale approach. Since, the method (Kumar et al., 2019) requires local features from multiple (radii) sizes of neighborhoods, it increases the computational burden. Additionally, we do not see any specific guideline on how the authors (Kumar et al., 2019) fix the size of the different radii to get the local neighborhood and corresponding features of multi-scale interest points.

In this paper, we are inspired by the feature-based DL approach, and develop an efficient non end-to-end DL algorithm that creates a binary classifier for ground and non-ground points classification in ALS point clouds. This method does not require transforming data into regular 3D voxel grids nor any rasterization. The scientific contributions of the paper are as follows. The method can successfully label the ground and non-ground points in cases of: (i) unstructured and unordered point clouds having inconsistent point densities, (ii) a smaller number of points comparing with the end-to-end DL approach, (iii) densely populated complex urban areas, (iv) presence of steep slopes, and (v) height discontinuity in the terrain.

The remainder of the paper is arranged as follows. Related works are briefly reviewed in Section 2. Section 3 proposes the methodology of the new algorithm. Section 4 demonstrates the proposed algorithm through two experiments on ALS point clouds, and presents analyses, evaluation and discussions on the experiments. Section 5 concludes the paper.

2. RELATED LITURATURE

Existing ground filtering methods can be categorized into morphological, progressive densification, surface-based, and segment-based filtering (Nurunnabi et al., 2016a; Qin et al., 2021). Morphological filtering is based on the idea of mathematical morphology, which is a set-theoretical approach of image analysis that provides information about geometrical structures. In connection with morphological filtering, Vosselman (2000) developed a slope-based filter incorporating the idea of planimetric distance as the neighborhood criteria. Morphological filters generally investigate height differences between neighboring points, whereas surface-based filters consider the surface inclination as well. Zaksek and Pfeifer (2004) noticed that the morphological filtering algorithm is active in areas with small differences but struggles in areas with steep slopes. Progressive densification begins with a small subset of the given data and iteratively increases the amount of information used to classify the whole data. Axelsson (2000) introduced a progressive triangular irregular network (TIN) densification combining the offset of a point with respect to the TIN. In contrast to the progressive densification, surface-based filtering assumes that all given points belong to the terrain surface and then iteratively remove the points that are inconsistent to the surface model by a step-by-step refinement of the surface characteristics. Robust interpolation based filtering integrates the filtering and DTM interpolation to determine an individual weight for each irregularly distributed point in such a fashion that the modelled surface represents the terrain (Kraus and Pfeifer, 1998). Sithole and Vosselman (2005) introduced segment based filtering approach in ALS data. Besides, there are many algorithms developed based on statistical tools, for example, skewness balancing was proposed by Bartels et al. (2006) for DTM generation. Nurunnabi et al. (2016a) introduced a statistically robust filtering algorithm using robust locally weighted regression. The authors (Nurunnabi et al., 2016b) also developed a region growing based robust segmentation algorithm that can be used for ground points filtering. In Nurunnabi et al. (2016b), the robust principal component analysis (PCA) is used for region growing based segmentation that starts from a seed point and examines *k* neighborhood points to see whether they fulfil certain criteria to be in a same region or not. The method provides robust segmentation in the presence of outliers and noise.

Among many other ML approaches, Random Forest (Chehata et al., 2009), Conditional Random Field (Vosselman et al., 2017), Support Vector Machines (Zhang et al., 2013), and Multi-Layer Perceptron (Weinmann et al. 2015) have been used to develop classification algorithms. Weinmann et al. (2015) conducted a study to investigate the performance of different ML techniques. Nowadays, DL is the most dominant ML technique. The two most well-known and revolutionary end-to-end DL techniques used in point cloud classification are PointNet and PointNet++ (Qi et al., 2017a, b). They have been evaluated mainly with indoor point clouds. These are the first attempts to feed 3D points to the network without any transformation. The basic idea behind the PointNet algorithm is to learn each and every point's spatial encoding and then combine all individual point features to a global signature. One important criterion that was missing in the PointNet algorithm, was that it did not consider local structure induced by the metric. Soon after the publication of the PointNet, Qi et al. (2017b) improved the PointNet algorithm, by exploiting local metric space distances where points are present, leading to PointNet++. Rizaldy et al. (2018) used a fully Convolutional Neural Network (FCNN) approach for ground point classification. The authors claimed that their method reduces computational cost on typical CNN based methods. Soilan et al. (2019) used the PointNet algorithm for automatic classification in ALS point clouds of an outdoor environment. Inspired by the idea of using different sizes of local neighborhoods in PointNet++ (Qi et al., 2017b) architecture, Kumar et al. (2019) developed a non end-to-end feature-based multi-class classification algorithm in TLS point clouds. Interested readers are referred to Guo et al. (2014) and Biasotti et al. (2016) for an overview on hand-crafted point features. Xiu et al. (2019) developed an algorithm based on PointNet together with intensity and color (red, green and blue; r, g, b) information. Wang et al. (2019) introduced a promising approach named dynamic graph CNN (DGCNN) for learning in point clouds including classification and segmentation. This approach acts on graphs, dynamically calculated in each layer of the network. Ma et al. (2019) provided a feature extraction framework for 3D point cloud segmentation. This method designs a 3D point-based convolutional operation together with an adaptive graph CNN using both local and global features in multiple scales.

Widyaningrum et al. (2020) developed a dynamic graph CNN based semi-automatic algorithm that uses LiDAR features as well as color information to increase the accuracy of the proposed classifier.

3. METHODOLOGY

Some researchers believe that modern DL techniques do not need feature engineering, because end-to-end DL can extract useful features automatically from raw data. This is not always true, many feature-based algorithms evidence that appropriate features can solve the problems more smartly while using fewer resources with far less numbers of training data. Moreover, feature based DL can make a problem easier if the user has a deep understanding about the problem and knowledge of useful features. This section proposes a feature-based DL architecture to label the ground and non-ground points in ALS point clouds. The proposed architecture develops a binary classifier, and follows the basic workflow of a DL approach.

We derive the algorithm in three steps. First step is for feature engineering that consists of two stages: extraction of relevant features, and finding the optimum feature space. Second step develops a binary classifier by constructing a DL architecture. The third step evaluates the classifier and fine-tunes the necessary hyper-parameters for the network.

3.1 Step 1. Feature engineering

This section comprises two tasks: generating related local features and selecting the optimum feature set. Local saliency features have been used for many fundamental tasks in point cloud processing including high level and global feature extraction, classification, segmentation, geometric primitive (e. g., planes, and circles) fitting, and surface reconstruction (Nurunnabi et al., 2016b, 2018, 2019). We extract 1D, 2D and 3D local saliency features. Local saliency features are usually estimated by studying the characteristics of the points in a local neighbourhood. Getting optimum neighbourhood is a crucial issue (Weinmann et al., 2015). In this paper, we empirically study similar data to fix neighbourhood size (radius). We use a spherical neighborhood to get 3D local shape features (e.g., normal and curvature), such neighborhood is obtained by considering all the points in a sphere with a predefined radius. The shape features are derived from the local covariance matrix generated by the neighbors of an interest point. The most common 3D shape features are known as Covariance Features (*CovF*). Principal component (PC) analysis is used to get the saliency features, mainly because of its computational simplicity. The local covariance matrix Σ based on a local neighborhood of an interest point p_i (x_i, y_i, z_i) is defined as

$$\Sigma_{3\times3} = \frac{1}{k}\sum_{i=1}^{n}(p_i - \bar{p})(p_i - \bar{p})^T \quad (1)$$

where k and $\bar{p}$ are the number of points and the mean of the points in the neighborhood (sample), respectively. The covariance matrix is decomposed into three eigenvectors (principal components; PCs) denoted by v_2, v_1 and v_0 that are sorted w. r. t. their corresponding eigenvalues λ_2, λ_1 and λ_0 that are arranged usually in descending order, i.e., $(\lambda_2 \geq \lambda_1 \geq \lambda_0 \geq 0)$. The first and third PCs (v_2 and v_0) explain the directions of the most and the least of the data variability for the sampled surface. The v_0 approximates the surface normal. Pauly et al. (2002) define the variation along the surface normal as λ_0, and surface curvature as $\sigma(p_i) = \lambda_0/(\lambda_0 + \lambda_1 + \lambda_2)$. In this paper, the eigenvalues are normalized by their totality and defined as $e_i = \lambda_i/\sum_0^2 \lambda_i$ (i = 0, 1, 2). The most common 3D shape features are: three eigenvalues, first eigenvector (PC1), surface normal, curvature, linearity, planarity, scattering, omnivariance, eigentropy, plane offset and verticality of the point of interest (Weinmann et al., 2015; Becker et al., 2017; Kumar et al. 2019; Ozdemir et al., 2019).

We are interested to extract ground points, and the ground points likely lie on a locally planar surface. Therefore, we consider 2D features. These features are generally derived by using a cylindrical neighborhood, where the neighbor points are those for which the 2D projection onto the ground (*x*-*y*) plane are within a circle with a given (same to sphere) radius. Most 2D features for filtering ground points are based on points' height (z) information, namely *z*-features (*zF*): the minimum, range, mean, variance, point height (P_z) values, and relative position (*RP*) of the interest point w.r.t. the neighbors. Point density (*PD*), positive openness (*PO*), and echo ratio (*ER*) are additionally useful (Kumar et al., 2019). Point density is the number of points per unit volume of the 2D cylindrical neighborhood. The openness is a measure to express the degree of power or enclosure of a location on an irregular surface (Yokoyama et al., 2002). According to viewer's perspective, positive openness expresses openness above the surface. The *ER* is defined for local transparency and roughness, which is an improved version of the point density ratio (Hofle et al., 2009). Furthermore, we consider two 1D LiDAR features (*LiF*): intensity (*I*) and return numbers (*RN*) in search of an appropriate feature space (Soilan et al., 2019). The feature sets are summarized in Table 1. To balance between the feature vectors, we standardize all the feature vectors before feeding them into the network. Since, we have a number of features to get the optimum set, it is not possible to evaluate every combination of the features. We investigate the relevance of the features by grouping them into three models, and train the architecture based on the models. Model 1 considers all features. Model 2 studies the features that were considered for a multi-scale feature-based method in TLS point cloud classification, proposed by Kumar et al. (2019). Unlike Model 2, Model 3 emphasizes on linearity, planarity, scattering, omnivariance and eigentropy rather than distinct eigenvalues. Table 2 defines the three models. In our experiment, we follow Kumar et al. (2019)'s rules for selecting multi-scale features for the feature combination from multiple neighborhoods.

3.2 Step 2. Developing deep learning architecture

In this step, we create a binary classifier by developing a DL architecture. Our inputs for the network are the feature vectors and the targets: 1 and 0 (1 for ground and 0 for non-ground). Dense layers are used to generate the network. We use the Rectified Linear Unit (*ReLU*) activation function for the hidden layers, and the Sigmoid function for the output layer (Goodfellow, et al., 2016). We use binary cross entropy as the loss function and the *Adam* (a method for stochastic optimization) optimizer to train the model. We use the '*He initialization'* strategy with the *ReLU* activation function, and *Batch Normalization* is used to reduce the influence of vanishing and exploding gradients. We perform the proposed network for the three feature sets (Models 1, 2 and 3) in Python software supported by the *TensorFlow* DL module.

3.3 Step 3. Tuning hyper-parameters

We conduct an extensive study based on our input data, and evaluate developed architecture to fine-tune the necessary hyper-parameters. The hyper-parameters used in this proposed network

are: number of hidden layers, number of neurons for the hidden layers, number of epochs and the mini-batch size. We start to fit the model form 7 layers and 70 neurons for each hidden layer. After an extensive empirical study, we see 5 layers with 50 neurons for the hidden layers are good enough to fit the model with the optimum level of accuracy. We train the model with 10 epochs and a mini-batch size of 128. We investigated L2 regularization with a learning rate γ= 0.2, 0.1, and 0.01 and finally fix with γ = 0.01 to avoid the possible of overfitting during training the model. At the end of every epoch, the model computes its' evaluation metrics (loss and accuracy) on the validation dataset. We finalize the hyper-parameters of the architecture with the least and highest values of loss and accuracy, respectively.

	Features	Definition
Covariance Features ($CovF$)	Three eigenvalues	$\lambda_2,\ \lambda_1,\ \lambda_0$
	1st Eigenvector, PC1	v_2
	Surface normal	v_0
	Curvature, σ_p	$\lambda_0/(\lambda_0+\lambda_1+\lambda_2)$
	Linearity, L_λ	$(e_2-e_1)/e_2$
	Planarity, L_p	$(e_1-e_0)/e_2$
	Scattering, S_λ	e_0/e_2
	Omnivariance, O_λ	$\sqrt[3]{e_0e_1e_2}$
	Eigentropy, E_λ	$-\sum_{i=0}^{2}\lambda_i ln(\lambda_i)$
	Plane offset (PO)	Sum of the residual distances to the fitted plane
	Verticality, θ	The angle between the normal and the *x-y* plane
Z Features (zF)	Minimum *z*, M_z	Lowest *z* value among the neighbour points
	Range, R_z	Distance between the maximum and the minimum *z* values of the neighbours
	Mean *z*, $\bar{z}$	Average of *z* values of the neighbour points
	Variance *z*, σ_z^2	Variance of *z* values of the neighbour points
	Point height (*z*), P_z	*z* distance from the local *x-y* plane
	Relative position (RP_z)	Relative position of *z* value of the interest point from *x-y* plane
	Point density (PD)	No. of points per unit volume in 2D neighborhood
	Positive openness (PO)	Openness above the surface
	Echo ratio (ER)	The ratio (in %) of no. of points in 3D and 2D neighborhoods
LiF	Intensity (I)	Return strength of a laser pulse
	Return number (RN)	*i*th return number for a given pulse

Table 1. Extracted features used in the proposed network.

Models	Required Features
Model 1	All features; $(CovF\ \cup\ zF\ \cup\ LiF)\ \cup\ (PD, PO, ER)$
Model 2	$(CovF\ \cup\ zF\ \cup\ LiF) \cup (PD, PO, ER)$ $-(\theta, \sigma_p,\ L_\lambda, L_p, S_\lambda, O_\lambda, E_\lambda,\ \bar{z}, RN)$
Model 3	$(CovF\ \cup\ zF\ \cup\ LiF) \cup (PD, PO, ER)$ $-\ (\lambda_2,\ \lambda_1,\ \lambda_0, v_2, M_z, RP_z, RN)$

Table 2. Three groups of features; Models 1, 2 and 3.

4. EXPERIMENTS, EVALUATION AND DISCUSSION

This section demonstrates the proposed algorithm. We conducted several experiments. Here, we analyse and explain the results of two ALS datasets, and evaluate the performance of the new algorithm for Model 1, Model 2, and Model 3.

We analyse and discuss the quantitative and qualitative outcomes of the experiments. Once the model is developed and trained with an optimum level of success, the trained model is used to classify the test data. We get the required predicted labels (ground and non-ground) for each point in the dataset. Quantitative performance for the developed methods is evaluated through four well-known performance metrics used in the classification task. We compare the results (predicted labels) with the ground truth and calculate the values for the performance metrics: precision, recall, F_1-score and model accuracy. The metrics are defined as follows.

$$Precision = \frac{TP}{TP+FP}, \tag{2}$$

$$Recall = \frac{TP}{TP+FN}, \tag{3}$$

$$F_1 - score = 2 \times \frac{Preision \times Recall}{Precision + Recall}, \tag{4}$$

$$Model\ accuracy = \frac{TP+TN}{TP+FP+TN+FN}, \tag{5}$$

where the number of True Positive, False Positive, True Negative, and False Negative cases are abbreviated as *TP*, *FP*, *TN* and *FN*, respectively.

4.1 Data set 1: AHN data

We perform our first experiment on the *Actueel Hoogtebestand Nederland* (AHN) data. This is a well-known open source data used in Soilan et al. (2019) and available at https://www.pdok.nl/nl/ahn3-downloads. AHN data is acquired by an ALS system, covers the whole area of The Netherlands. It has an average point density of around 20/m^2, and records up to five returns. The data come with class labels (ground, vegetation, building, water, and bridge) and is divided into tiles of 500m × 500m. We select two tiles in urban areas with complex objects like residential-commercial buildings, vegetation, car and a big church. Ten training, one validation and one test samples (Figure 1) are sampled from different locations. Test and validation sets are chosen randomly. Training sets are chosen of different sizes and from different locations to cover different landscape and to contain various objects. The selected areas contain 6,968,776 points in total, 14% of them are used for the test set, and the rest are used for the training and validation sets. Excluding the test set, the validation set contains 13% of the points.

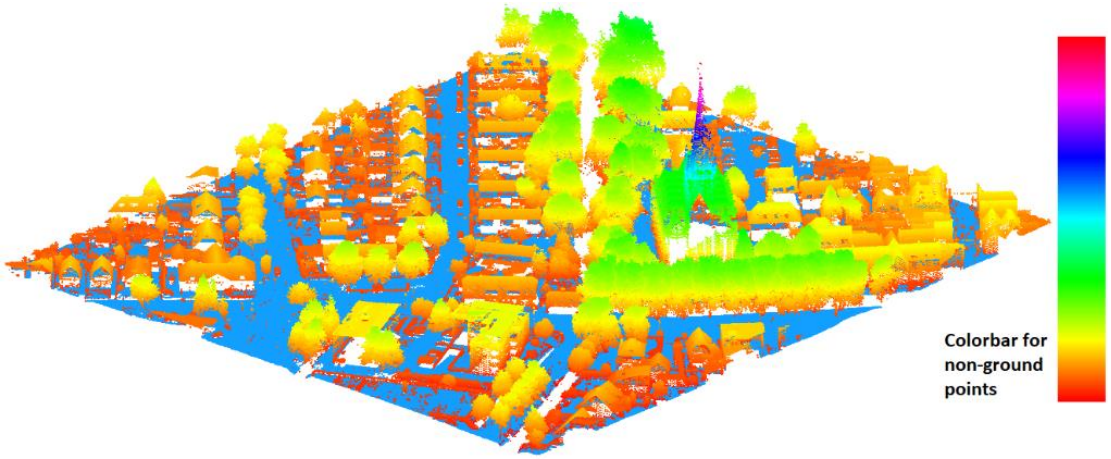

Figure 1. AHN test dataset in a complex urban area with trees, cars, small-large residential-commercial buildings, and a church.

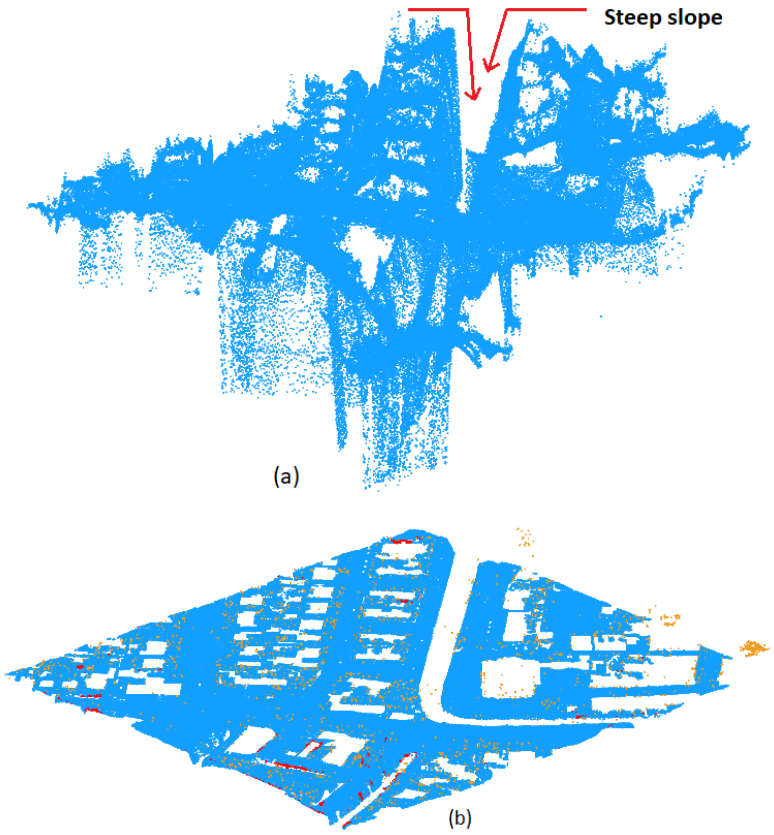

Figure 2. (a) Presence of steep slopes at a drainage system in the terrain, (b) extracted ground surface; ground points (blue), false positives (yellow) and false negatives (red).

Radius (cm)	Model	Ground			Non-Ground			Model Accuracy (%)
		Precision (%)	Recall (%)	F_1-score (%)	Precision (%)	Recall (%)	F_1-score (%)	
100	Model 1	92.5	99.8	96.0	99.8	95.0	97.4	96.7
	Model 2	89.9	98.9	94.6	99.9	93.0	96.3	95.6
	Model 3	93.3	99.7	96.2	99.7	95.6	97.6	97.2
50	Model 1	93.0	99.6	96.2	99.8	95.4	97.5	97.0
	Model 2	89.8	99.2	94.2	99.9	**92.4**	**96.0**	95.3
	Model 3	**93.8**	99.7	**96.7**	99.8	**95.6**	**97.7**	**97.3**
100 ∪ 50	Model 1	91.6	99.7	95.7	99.9	94.3	97.1	96.5
	Model 2	92.0	99.7	95.7	99.8	94.6	97.1	96.4
	Model 3	93.3	99.7	96.5	99.8	95.8	97.7	97.2

Table 3. Results show the performance of the proposed algorithm for the AHN test dataset.

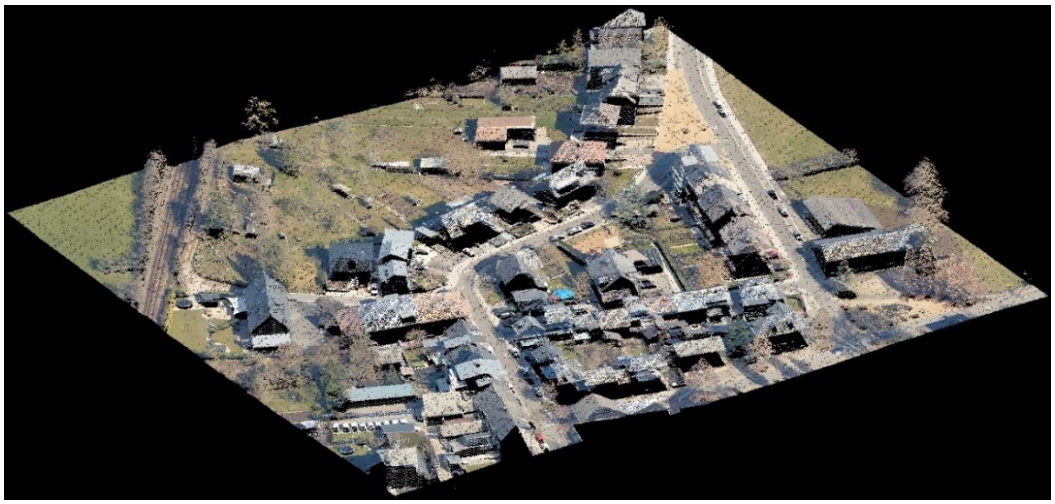

Figure 3. ACT test dataset in an urban area with trees, cars, and small-medium residential-commercial buildings.

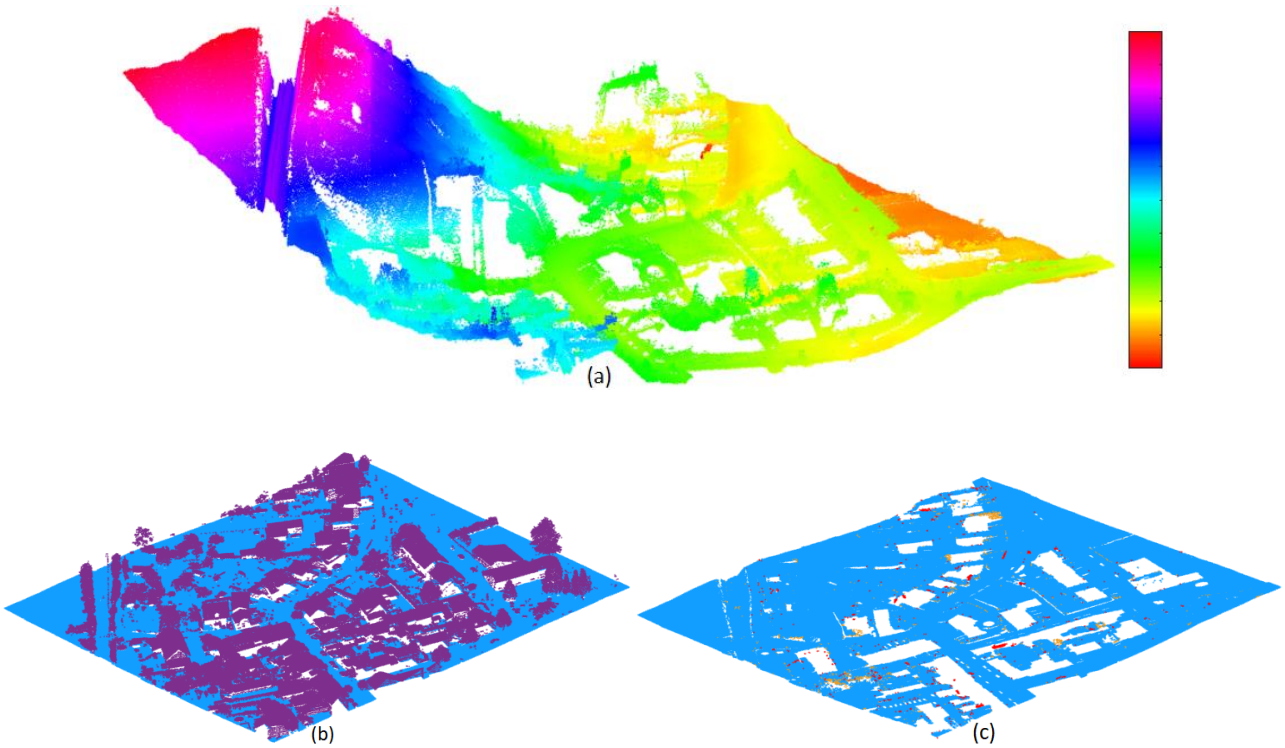

Figure 4. Results for ACT data; (a) Presence of height discontinuity in the terrain, (b) classified ground (blue) and non-ground points (purple), (c) false positives (yellow) and false negatives (red) cases.

Radius (cm)	Model	Ground			Non-Ground			Model Accuracy (%)
		Precision (%)	Recall (%)	F_1-score (%)	Precision (%)	Recall (%)	F_1-score (%)	
100	Model 1	99.6	98.0	98.8	96.2	99.4	97.7	98.4
	Model 2	99.7	93.5	96.5	89.7	99.2	94.2	95.7
	Model 3	99.7	98.4	**99.0**	96.8	99.6	98.2	**98.6**
50	Model 1	99.7	97.2	98.4	96.5	98.6	97.5	98.1
	Model 2	99.7	93.8	96.6	90.2	99.5	94.6	95.9
	Model 3	99.7	97.6	98.7	96.1	99.6	97.8	98.4
100 U 50	Model 1	99.4	97.1	98.2	96.2	99.0	97.6	98.4
	Model 2	99.0	96.2	97.6	93.7	99.2	96.4	96.3
	Model 3	99.6	97.9	98.7	96.4	99.4	97.9	98.5

Table 4. Results show the performance of the proposed algorithm for the ACT test dataset.

The test dataset consists of different sizes of complex objects including big and small trees, many cars, small-large buildings, and a large church (Fig. 1). According to our interest of classifying data into ground and non-ground points, all given labels are grouped into two: one group is only for ground points, and all the other points are re-labelled as non-ground points. We calculate features sets with the help of OPALS (Pfeifer et al., 2014) and the MATLAB software. The proposed network is trained using the training and validation datasets, and the trained model is used on the test dataset. Quantitative performance for the developed method is evaluated through the metrics: precision, recall, F_1-score and model accuracy. The performance of the algorithm is assessed through the three models using the features from neighbourhoods of radii 100cm, and 50cm, and combining (multi-scale) features from radii 100cm and 50cm following Kumar et al. (2019). Results are shown in Figure 2 and Table 3. For every case of radius 100cm, 50cm, and (100cm ∪ 50cm), Model 1 and Model 3 perform almost similar for every metric, and sometime Model 3 performs slightly better than Model 1. However, both of Model 3 and Model 1 are better than almost all evaluation metrics of Model 2. For ground filtering with 50cm radius, Model 3 achieves the best value for precision, F_1-score and model accuracy of 93.8%, 96.7% and 97.3%, respectively. Model 3 also produces a better F_1-score than the others for non-ground surface extraction. With 50cm radius, Model 3 has recall and F_1-score of 95.6% and 97.7%, whereas, for non-ground surface, Model 2 gets the metric values of 92.4% and 96.0% respectively. Fig. 2(a) indicates the presence of steep slopes (of a drainage system) in the data. Fig. 2(b) shows the identified ground points, with false positive and false negative cases using the Model 3.

4.2 Data set 2: ACT data

In the second experiment, we use the LiDAR 2019 data, the 3D survey of the Luxembourg territory and provided by Administration of Cadastre and Topography (ACT). These are available at https://data.public.lu/en/datasets/lidar-2019-releve-3d-du-territoire-luxembourgeois/. We name it ACT data. They come from a 3D aerial survey based on LiDAR technology. They have an average point density of around 15/m^2 with a horizontal precision of $\pm$ 3 cm and vertical precision of $\pm$ 6 cm. The data are structured into 500m×500m tiles, which contain on average five to seven million points. The raw tiles are grouped together up to nine tiles representing a spatial extent of 1500m×1500m. The points in the ACT data are usually labelled into soil, low vegetation, medium vegetation, high vegetation, buildings, low points (noise), water, bridges, footbridges, viaducts, high voltage lines and unclassified points.

We selected two tiles of 500m×500m in urban areas. The training dataset is selected from one tile, and the test dataset is from the other tile. To get more accurately classified data, further, we manually edit the data using the LAStools software. Following our objective in this paper, we reclassified the data as the previous experiment: ground and non-ground points. The data we consider for our experiment contain 5,110,261 points. The training dataset is collected by five segments (samples) consist of 4,131,522 points. From the five training segments, we randomly choose one segment with 650,764 points as validation dataset to validate the developed model. The validation set is 15.75% of all the training points. In the test set, there are 977,739 points, which is the 19.13% of all the points selected for the experiment. The necessary feature vectors for the data are calculated as the previous experiment. We perform the proposed algorithm for the Models 1, 2, and 3. This time, we get slightly better results for Model 3 with radius of 100cm than 50cm and their combination (100cm ∪ 50cm). Using the features generated from the neighborhood size of 100cm, Model 3 achieves F_1-score of 99.0% and 98.2% for labelling ground and non-ground points, respectively, and model accuracy of 98.6%. Model 2 finds ground and non-ground points with significantly less F_1-score than Model 1 and Model 3 for each neighbourhood size. With 100cm radius, Model 2 achieves recall of 93.5% for ground point labelling, whereas Model 1 and Model 3 achieve recall of 98.0% and 98.4%, respectively.

5. CONCLUSIONS

This paper proposes a non end-to-end DL algorithm to classify ground and non-ground points in ALS point clouds. The new algorithm constructs a fully connected deep neural network to develop a binary classifier. The algorithm performed on three models which are the combinations of extracted pointwise local features. The inputs for the networks are point features, rather than the raw point clouds (x, y, z). In the results of the first experiment in Section 4, of 50cm radius, it is explored that Model 3 is the best group of features that achieves more than 97% model accuracy with F_1-score for ground and non-ground point filtering are 96.7% and 97.7%, respectively. Similar decisions (better performance of Model 3) can be made from Experiment 2. The algorithms perform better for ACT than AHN data, e.g., in terms of model accuracy; Model 3 achieves highest performance of 97.3% and 98.6% for AHN and ACT data, respectively. Model 1 and Model 3 perform almost similar, but the advantage of the Model 3 is that it uses 19 features, whereas Model 1 uses 28 features. Therefore Model 1 creates more computational burden for the proposed architecture. The new algorithm does not require multi-scaling, as the models developed from the features generated by multi-scaling (radius) do not perform better. Reasonably, all the models also make a significant amount of computational complexity when features are from multi-scale (radii) neighborhoods. We finally suggest Model 3 as the best for labelling ground and non-ground points in ALS data. Experiments in Section 4 show that the proposed filtering methods finds both ground and non-ground points successfully in the presence of steep slopes and height discontinuity in the terrain. Potential applications of the proposed algorithm include city modelling, building footprint identification, digital surface and terrain modelling, disaster like flood management, corridor mapping and urban assets monitoring and management.

Like other non end-to-end DL methods, the proposed algorithm needs through understanding about the related features. Future research will continue to classify non-ground objects like vegetation, buildings and road-corridor infrastructure.

ACKNOWLEDGEMENTS

This study is supported by the Project 2019-05-030-24, SOLSTICE - Programme Fonds Européen de Developpment Régional (FEDER)/Ministère de l'Economie of the G. D. of Luxembourg.

REFERENCES

Axelsson, P., 2000. DEM generation from laser scanner data using adaptive TIN models. *Int. Arch. of Photogramm. Remote Sens. Spatial Inf. Sci.*, 33(part B4/1), 110–117.

Bartles, M., Wei, H., Mason, D. C., 2006. DTM generation from LiDAR data using skewness balancing. *In Proc. IAPR 18th Int. Conf. Pattern Recognit.*, 20–24 August, Hong-Kong, China.

Becker, C., Rosinskaya, E., Häni, N., D'Angelo, E., Strecha, C., 2018. Classification of aerial photogrammetric 3d point clouds. *Photogramm. Eng Remote Sensing*, 84(5), 287–295.

Biasotti, S., Cerri, A., Bronstein, A., Bronstein, M., 2016. A Recent trends, applications, and perspectives in 3D shape similarity assessment. *Computer Graphics Forum*,35(6),87–119.

Chehata, N., Guo, L., Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. *Laserscanning*, Paris, France.

Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT Press.

Guo, Y., Bennamoun, M. Sohel, F., Lu, M., Wan, J., 2014. 3D object recognition in cluttered scenes with local surface features: a survey. IEEE *TPAMI.*, 36(11), 2270–2284.

Haralick, R., Shapiro, L., 1992. *Computer and Robot Vision*. Addison-Wesley, Reading, Mass.

Hofle, B., Mucke, W., Dutter, M., Rutzinger, M., Dorninger, M., 2009. Detection of building regions using airborne LiDAR – a new combination of raster and point cloud-based GIS methods. In *GI-Forum 2009, Intl. Conf. applied geoinformatics*, Salzburg.

Kraus, K., Pfeifer. N., 1998. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS J. Photogramm. Remote Sens.*, 53(4), 193–203.

Kumar, A., Anders, K., Winiwarter, L., Hofle, B. 2019. Feature relevance analysis for 3D point cloud classification using deep learning, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-2/W5.

Lindenberger, J., 1993. Laser-Profilmessungen zur topographischen gelaedeaufnahme, Deutsche Geodaetische Kommission, Series C, No. 400, Munich, Germany.

Ma, L., Li, Y., Li, J., Tan, W., Yu, Y., Chapman, M. A., 2019. Multi-scale point-wise convolutional neural networks for 3d object segmentation from lidar point clouds in large-scale environments. *IEEE Trans. Intell. Transp. Syst.*, 22(2), 821–836.

Nurunnabi, A., West, G., Belton, D., 2016a. Robust locally weighted regression techniques for ground surface points filtering in mobile laser scanning 3D point cloud data. *IEEE Trans. Geosci. Remote Sens.* 54(4), 2181–2193.

Nurunnabi, A., Belton, D., West, G., 2016b. Robust segmentation for large volumes of laser scanning three-dimensional point cloud data. *IEEE Trans. Geosci. Remote Sens.*, 54(8), 4790–4805.

Nurunnabi, A., Sadahiro, Y., Laefer, D.F., 2018. Robust statistical approaches for circle fitting in laser scanning three-dimensional point cloud data. *Pattern Recognit.*, 81(9), 417–431.

Nurunnabi, A., Sadahiro, Y., Lindenbergh, R., Belton, D., 2019. Robust cylinder fitting in laser scanning point cloud data. *Measurement*, 138, 632–651.

Özdemir, E., Remondino, F., 2019. Classification of aerial point clouds with deep learning. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, XLII-2/W13.

Pauly, M., Gross M., Kobbelt, L. P., 2002. Efficient simplification of point sample surface. *Proc. IEEE Conf. Vis.*, Washington, D.C., 27 October – 1 November, 163–170.

Pfeifer, N., Mandlburger, G., Otepka, J., Karel, W., 2014. OPALS - A framework for Airborne Laser Scanning data analysis. Computers, Env. and Urban Systems 45:125 –136.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. *Proc. IEEE Comput. Soc. CVPR.*, 652–660.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*.

Qin, N., Tan, W., Ma, L., Zhang, D., Li, J., 2021. OpenGF: An ultra-large-scale ground filtering dataset built upon open ALS point clouds around the world. https://arxiv.org/pdf/2101.09641.

Rizaldy, A., Persello, C., Gevaert, C.M. Oude Elberink, S.J., 2018. Fully convolutional networks for ground classification from lidar point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, 4(2).

Soilán, M., Lindenbergh, R., Riveiro, B., Sánchez Rodríguez, A., 2019. Pointnet for the automatic classification of aerial point clouds. *ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci.*, IV-2/W5.

Sithole, G., Vosselman, G., 2004. Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens*., 59(1-2), 85–101.

Sithole, G., Vosselman, G., 2005. Filtering of airborne laser scanner data based on segmented point clouds. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, 36(part 3), W19.

Vosselman, G., 2000. Slope based filtering of laser altimetry data. *Int. Arch. Photogramm. Remote Sens*.33(Part3), 935–942.

Vosselman, G., Coenen, M., Rottensteiner, F., 2017. Contextual segment-based classification of airborne laser scanner data. *ISPRS J. Photogramm. Remote Sens.*, 128, 354–371.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.*, 105, 286–304.

Widyaningrum, E., Fajari, M. K., Lindenbergh, R. C., Hahn, M., 2020. Tailored features for semantic segmentation with a DG-CNN using free training samples of a colored airborne point cloud. *Int. Arch. of the Photogramm. Remote Sens. Spatial Inf. Sci.*, 43(B2).

Xiu, H., Vinayaraj, P., Kim, K.S., Nakamura, R., Yan, W., 2019. 3D semantic segmentation for high-resolution aerial survey derived point clouds using deep learning. *26th ACM SIGSPATIAL Conf. Adv. in Geographic Inf. Systems*, 588–591.

Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M., Solomon, J., 2019. Dynamic graph CNN for learning on point clouds, *ACM Trans. Graph*., 1(1).

Yokoyama, R., Shirasawa, M., Pike, R.J., 2002. Visualizing topography by openness: a new application of image processing to digital elevation models. *PE&RS*, 68(3), 257–265.

Zakšek, K., Pfeifer, N., 2004. An improved morphological filter for selecting relief points from a LIDAR point cloud in steep areas with dense vegetation. *Technical Report, Delft Institute of Earth Observation and Space systems*, TU Delft.

Zhang, J., Lin, X., Ning, X., 2013. SVM-based classification of segmented airborne LiDAR point clouds in urban areas. *Remote Sens*., 5, 3749 – 3775.

Zhang, L., Li, Z., Li, A., Liu, F., 2018. Large-scale urban point cloud labelling and construction. *ISPRS J. Photogramm. Remote Sens.*, 138, 86 –100.