A FEATURE-BASED DEEP LEARNING APPROACH FOR THE EXTRACTION OF GROUND POINTS FROM 3D POINT CLOUDS

Y. Dogan [1]*, A.Ö. Ok [2]

[1] Vocational School of Technical Sciences, Karamanoglu Mehmetbey University, Karaman, Türkiye - yusufdogan@kmu.edu.tr
[2] Department of Geomatics Engineering, Hacettepe University, Ankara, Türkiye - ozgunok@hacettepe.edu.tr

KEY WORDS: Point Cloud, Classification, Deep Learning, Feature Extraction, Neural Networks.

ABSTRACT:

Extracting ground points from 3D point clouds is important for sustainable development goals, infrastructure planning, disaster management, and more. However, the irregularity and complexity of the data make it challenging. Deep learning techniques, particularly end-to-end and non-end-to-end approaches, have shown promise for 3D point cloud segmentation and classification, but both require a comprehensive understanding of the features and their relationship to the problem. This paper presents a study on the filtering of 3D LiDAR point clouds into ground and non-ground points using a non-end-to-end deep learning approach. The aim of this research is to investigate the effectiveness of utilizing geometric features and a binary classifier-based deep learning model in accurately classifying point clouds. The publicly available ACT benchmark datasets were employed for training, validation, and testing purposes. The study utilized a k-fold cross-validation method to address the limited availability of training data. The results demonstrated highly satisfactory performance, with validation averages reaching 96.83% for the divided Dataset-1 and an accuracy of 97% for the test set. Furthermore, an independent dataset, Dataset-2, was used to evaluate the generalizability of the trained model, achieving an accuracy of 93%. These findings highlight the potential of the proposed non-end-to-end approach to filtering point cloud data and its applicability in various domains such as DEM and DTM production, city modeling, urban planning, and disaster management. Moreover, this study emphasizes the need for accurate data to achieve sustainable development goals, positioning the proposed approach as a viable option in various studies.

1. INTRODUCTION

Light Detection And Ranging (LiDAR) is a technique that uses laser pulses to measure the distance and reflectance of objects in the environment, producing a 3D representation of the scene (Huang et al., 2019b). The accurate filtering of 3D LiDAR point clouds is a fundamental task in numerous geospatial-related applications, including terrain analysis, urban planning, and disaster management. These practices are essentially indispensable to ensuring the sustainability of the Sustainable Development Goals (SDGs). Point cloud processing plays a crucial role in extracting meaningful information from the acquired data, enabling applications such as terrain modeling, object detection, and environmental monitoring (Huang et al., 2019a). Of particular interest is the accurate separation of ground and non-ground points, as this classification is fundamental for various geospatial tasks, especially for digital terrain model (DTM) generation.

In recent years, deep learning techniques have demonstrated remarkable success in various domains, including image classification, natural language processing, speech recognition, and point cloud object detection. The end-to-end deep learning approach, which involves training a model to directly map raw input data to desired outputs, has gained significant popularity due to its ability to automatically learn hierarchical representations and extract discriminative features from complex data (Qi et al., 2017b; Shi et al., 2019; Liu et al., 2019). However, for certain tasks such as point cloud classification, the end-to-end approach may suffer from limitations, particularly when dealing with limited training data and imbalanced class distributions (Winiwarter et al., 2019; Schmohl & Sörgel, 2019).

In this study, we present a non-end-to-end deep learning approach specifically tailored for ground and non-ground point cloud filtering. By leveraging geometric features and a binary classifier-based architecture, the proposed model aims to accurately classify individual points as either ground or non-ground. The use of geometric features allows the model to capture intrinsic characteristics of the point cloud data, thereby improving classification accuracy (Nurunnabi et al., 2021; Weinmann et al., 2014). Furthermore, the adoption of a binary classifier enables the separation of point clouds into two distinct classes, facilitating subsequent analyses and applications.

To evaluate the effectiveness of the proposed approach, we utilize the publicly available ACT benchmark datasets, which provide diverse and challenging point cloud data for testing and validation purposes. Given the limited availability of training data, we employ the k-fold cross-validation technique to ensure reliable performance assessment (Wong & Yeh, 2019). This technique involves dividing the training data into multiple folds, training separate models on each fold, and evaluating their performance on the validation set. The evaluation measures, including accuracy, precision, recall, and F1-score, provide a comprehensive assessment of the proposed approach's classification capabilities.

The results of this study demonstrate the efficacy of the non-end-to-end deep learning approach in accurately classifying ground and non-ground points in 3D LiDAR point clouds. The obtained accuracy values showcase the potential of the proposed methodology for various applications, including DTM

[1]* Corresponding author

generation, city modeling, urban planning, disaster management, land management, and corridor mapping.

In conclusion, this research paper presents a novel non-end-to-end deep learning approach for ground and non-ground point cloud filtering. The proposed methodology demonstrates promising results, highlighting the significance of leveraging geometric features in point cloud classification tasks. The findings contribute to the existing body of knowledge in the field of point cloud analysis and provide valuable insights for researchers and practitioners interested in the accurate and efficient processing of 3D LiDAR point clouds.

The remainder of this paper is organized as follows: Section 2 presents the study area and dataset, including information about the data utilized. Section 3 describes the methodology, data processing, the deep learning model architecture, and the evaluation measures employed. Section 4 presents the results and discussions, followed by a discussion on the implications and limitations of the proposed approach. Finally, Section 5 concludes the paper and outlines future research directions.

2. STUDY AREA AND DATASET

In this study, publicly available aerial LiDAR dataset provided by the Luxembourg Administration of Cadastre and Topography (ACT) was used. The data can be accessed at the following address: https://data.public.lu/en/datasets/LiDAR-2019-releve-3d-du-territoire-luxembourgeois/. These data were collected through measurements conducted in 2019 and consist of 3D point clouds covering the entire country. The measurements were performed with an average point density of 15 points per square meter and horizontal and vertical accuracy of ± 3 cm and ± 6 cm, respectively. To facilitate easy downloading, these large data sets, which cover a large area, were divided into 500-m-wide sections. Each download file contains nine sections, resulting in a point cloud dataset of 1500 x 1500 m. Two of these sections were randomly selected, one for training data and the other for testing data (Figure 1).


(a) (b)

Figure 1. a) Training dataset (Dataset-1) b) Test dataset (Dataset-2)

In the ACT dataset, the points are labeled as soil, low vegetation, medium vegetation, high vegetation, buildings, low points (noise), water, bridges, footbridges, viaducts, high voltage lines, and unclassified points. In this study, the points were manually assigned into ground and non-ground categories. This process is detailed in Section 3.1. The training dataset consists of 3,395,251 points, with 70% used for training and 30% for testing. This training set is further divided into four parts, and validation values are calculated using a 4-fold cross-validation method. The independent test dataset contains 4,122,627 points, which are used as validation data to measure the accuracy of the trained deep learning model.

3. METHODOLOGY

In this section, we detail our strategy in two sub-sections: (i) feature extraction, and (ii) deep learning architecture. A number of preprocessing operations were applied to the downloaded raw data using the open-source software CloudCompare. The labels of very low points (noise) and unclassified points were removed from the point clouds of the training and test data. This is because these points do not belong to any class and would negatively affect the accuracy of deep learning training.

3.1 Feature Extraction

After pruning the point clouds, features needed to be calculated. Feature extraction for 3D point cloud data is a process of identifying and extracting meaningful geometric shapes or structures from the raw 3D point data collected by LiDAR sensors. This feature extraction can be useful for various applications, such as object recognition, scene understanding, mapping, registration, segmentation, classification, and reconstruction (Baek et al., 2017).

There are different methods and algorithms for geometric feature extraction from 3D point cloud LiDAR data, depending on the type of features, the quality of the data, the computational complexity and the desired accuracy (Dittrich et al., 2017; Weinmann et al., 2017; Grilli et al., 2019; Zeybek & Biçici, 2021). Principal Component Analysis (PCA) can be used for geometric feature extraction, which involves transforming the original features of a dataset into a new set of features (Sawyer et al., 2021). This transformation is done by projecting the data onto the principal components identified through PCA. In the context of geometric feature extraction, PCA helps to identify the most informative and relevant features in the dataset. It achieves this by finding a smaller set of uncorrelated variables, known as principal components, that capture the maximum variance in the data. These principal components represent a new feature space that retains the essential characteristics of the original data while reducing its dimensionality. By selecting a subset of the top-ranked principal components, one can effectively reduce the number of features in the dataset. This dimensionality reduction can be beneficial for various reasons, including simplifying subsequent analysis, alleviating computational complexity, and removing noise or redundant information. PCA-based feature extraction can be particularly valuable when dealing with high-dimensional datasets where the number of features exceeds the number of samples. It allows for a more concise representation of the data while retaining the most important information, making subsequent modeling or analysis tasks more efficient and effective (Lin et al., 2014; Gilani et al., 2016).

The utilized features and their descriptions are provided in Table 1. These features were extracted with the help of the CloudCompare software, and the calculation of these features was based on the studies by Douros and Buxton (2002) and Hackel et al. (2016). Note that we calculated 8 out of the 9 3D local shape features with a specific radius of 0.5 m spherical neihgborhood, while intensity, which is one of the nine features, was not calculated as it was already present in the dataset. In total, the features utilized for training the deep learning model include intensity, mean curvature, volume density, sum of eigenvalues, omnivariance, eigenentropy, planarity, linearity, and verticality. These processes were both applied to Dataset-1 and Dataset-2.

Features	Definition
Intensity	Return strength of a laser pulse
Curvature	$\lambda_3 \div (\lambda_1 + \lambda_2 + \lambda_3)$
Volume density	Number of neighbors divided by the neighborhood volume
Sum of eigenvalues	$\lambda_1 + \lambda_2 + \lambda_3$
Omnivariance	$(\lambda_1 \cdot \lambda_2 \cdot \lambda_3)^{\frac{1}{3}}$
Eigenentropy	$-\sum_{i=1}^{3} \lambda_i \cdot ln(\lambda_i)$
Planarity	$(\lambda_2 - \lambda_3) \div \lambda_1$
Linearity	$(\lambda_1 - \lambda_2) \div \lambda_1$
Verticality	$1 - (\lambda_3 \div (\lambda_1 + \lambda_2 + \lambda_3))$

Table 1. Features utilized in the proposed approach.

3.2 Deep Learning Architecture

The architectural structure of the deep learning model has been specifically devised to function as a binary classifier (Figure 2). The utilization of a binary classifier is motivated by the objective of categorizing point clouds into two distinct classes, namely ground and non-ground. The neural network was trained using feature vectors and targets. The target values comprise binary digits, specifically 1s and 0s, where 0 represents non-ground and 1 represents ground. The network architecture comprises a total of five densely connected layers. The Rectified Linear Unit (ReLU) activation function is employed for the hidden layers, while the sigmoid function is utilized for the output layer. Each dense layer is comprised of 50 neurons. The model was trained using the binary cross-entropy loss function and the Adam optimizer. The learning rate utilized by the Adam optimizer is fixed at 0.01.

The k-fold cross-validation technique was utilized during the training of the model. This approach is frequently employed in situations where there is a limited availability of accessible data (Srinivasan et al., 2019). The variability of the validation scores is dependent on the specific validation dataset. In such cases, the best approach to apply is K-fold cross-validation (Figure 3). The data at hand is partitioned into K segments, with K being equal to 4 in our study. Subsequently, K models are constructed, wherein each model is trained on K-1 parts of the data and assessed on the remaining parts. The validation score is computed as the arithmetic mean of the K validation scores.

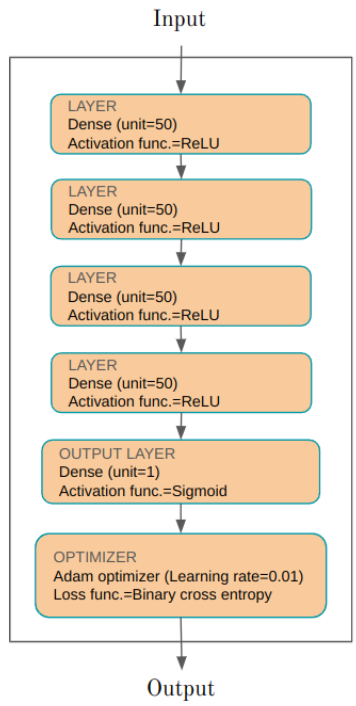

Figure 2. Structure of the deep learning algorithm to train the model

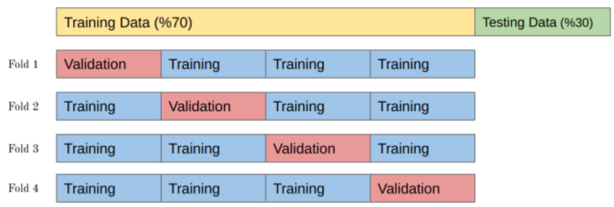

Figure 3. 4-Fold cross validation for Dataset-1

In the current study, due to the limited availability of 3D point data for training, we utilized the 4-fold cross-validation method. We trained the model for 20 epochs and set the mini-batch size to 128. At the end of each epoch, the evaluation measures (loss and accuracy) were computed on the validation dataset. The accuracy measures for the 4-folds were calculated separately, and the overall accuracy measure was obtained by computing their average. After training the model in this manner, the evaluation measures were computed on the 30% test data that we had split.

Furthermore, the point cloud from Dataset-2, which was not used in any way during training, was used to independently evaluate the accuracy of the model. The purpose was to observe the performance of the model on a similar but different dataset. The model was developed in Python using Google Colaboratory.

4. RESULTS AND DISCUSSION

This section discusses the findings of the study and highlights the advantages and disadvantages of the method presented. As mentioned earlier, Dataset-1 was divided into two parts (part 1: 70% training and part 2: 30% test), and part 1 was further divided into 4 folds for 4-fold cross-validation during training. Additionally, Dataset-2 was used separately to evaluate the accuracy of the trained model.

Table 2 presents the average accuracy values for each fold. Figure 4 shows the average training and validation accuracy values for each fold at the end of each epoch. Since each fold was trained for 20 epochs, the average values of all epochs from the first to the last epoch in each fold were calculated as the accuracy values for each epoch.

# of Fold	Accuracy %
Fold 1	96.85
Fold 2	96.84
Fold 3	96.81
Fold 4	96.81
Mean	**96.83**

Table 2. Value of accuracy for each fold and mean accuracy

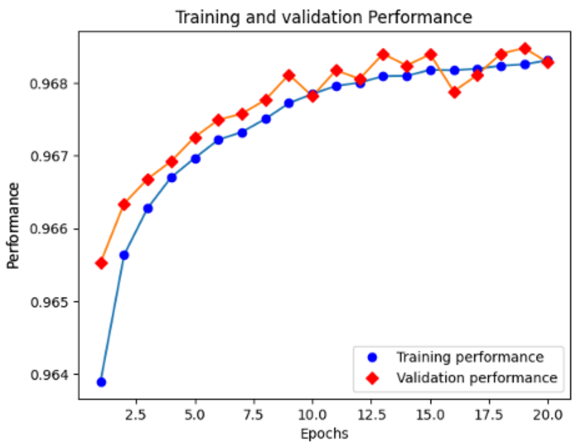

Figure 4. Training and validation performance of 4-fold cross validation

The accuracy values for the test data of part 2, which accounts for 30% of Dataset-1, are provided in Table 3. The table also includes precision, recall, and F1-score values for the ground and non-ground points. In part 2, there are a total of 1,018,576 points, with 305,827 points labeled as non-ground and 712,749 points labeled as ground. The given evaluation values represent the correct prediction rates for these points. Therefore, the overall probability of correctly predicting these points is calculated to be 97%.

The evaluation measures calculated for Dataset-2 are provided in Table 4. In this case, there are a total of 4,122,627 points, with 1,231,622 points labeled as non-ground and 2,891,005 points labeled as ground. The overall performance of correctly predicting classes of these points is calculated to be 93%.

	Precision	Recall	F1-score	# of Points
Non-ground	0.94	0.95	0.95	305,827
Ground	0.98	0.97	0.98	712,749
Accuracy			0.97	1,018,576
Macro Average	0.96	0.96	0.96	1,018,576
Weighted Average	0.97	0.97	0.97	1,018,576

Table 3. Results of the validation dataset (Dataset 1)

	Precision	Recall	F1-score	# of Points
Non-ground	0.82	0.97	0.89	1,231,622
Ground	0.99	0.91	0.95	2,891,005
Accuracy			0.93	4,122,627
Macro Average	0.90	0.94	0.92	4,122,627
Weighted Average	0.94	0.93	0.93	4,122,627

Table 4. Results of the testing dataset (Dataset 2)

In Table 4, the precision value for non-ground points appears as 0.82. The primary factor contributing to the comparatively lower value in relation to other values presented in the table can be associated to the chosen features that possess the characteristic of ground points. This is because the model was not trained using any features formed by elevation data. Upon visual examination of the misclassified points, it is evident that these particular points, which are known to belong to buildings, have been erroneously classified as ground points (refer to Figure 5). Roofs on these structures are all relatively flat rather than sloping. The points associated with flat roofs exhibit similarities to ground points, which may account for this finding. The misclassifications in flat roofs are represented by the point clouds enclosed within a red ellipse in Figures 5 and 6. Furthermore, within the point cloud representation of the forested region, it has been observed that certain points have been inaccurately classified. A possible explanation for this particular problem could be attributed to the specific parameterization of the spherical neighborhood value utilized during the computation of the features. These findings will be further evaluated in future studies.

Figure 5. Points on flat roofs in the red ellipse has ground characteristic

One notable benefit of employing this approach is its utilization of pre-existing geometric and physical knowledge, leading to enhanced performance and relatively less dependence on labeled data. Moreover, it exhibits significantly shallow architecture in contrast to end-to-end deep learning methodologies, thereby resulting in reduced memory consumption and less computational load. Nevertheless, one critical issue is that it is necessary to calculate the features prior to training the deep learning model. The process of feature computation is dependent on predetermined parameters, such as the radii of spheres. Besides, it is necessary to compute the features for both the training and test data prior to training the model.

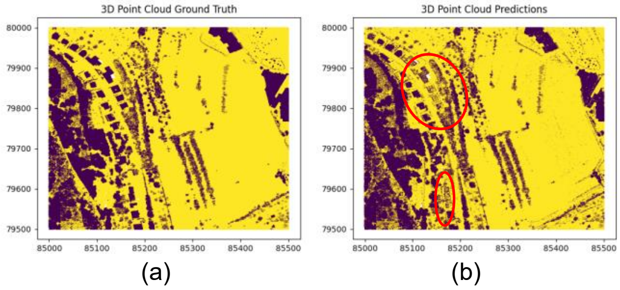

Figure 6. (a) ground truth and (b) predictions of the test dataset (Dataset-2) (ground: yellow, non-ground: purple)

Figure 6 illustrates the ground truth and prediction results of the points provided in the Dataset-2. Yellow color is used for ground points, while purple color is used to represent non-ground points. Figures 7 and 8 display the ground and non-ground points of the same dataset in a 3D graphical format.

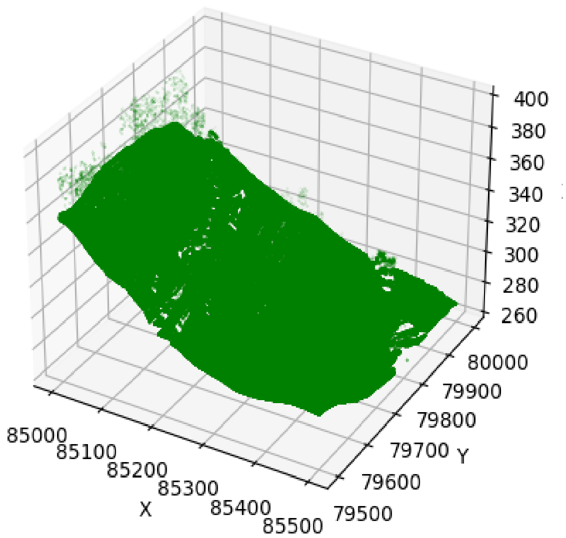

Figure 7. Prediction results of ground points

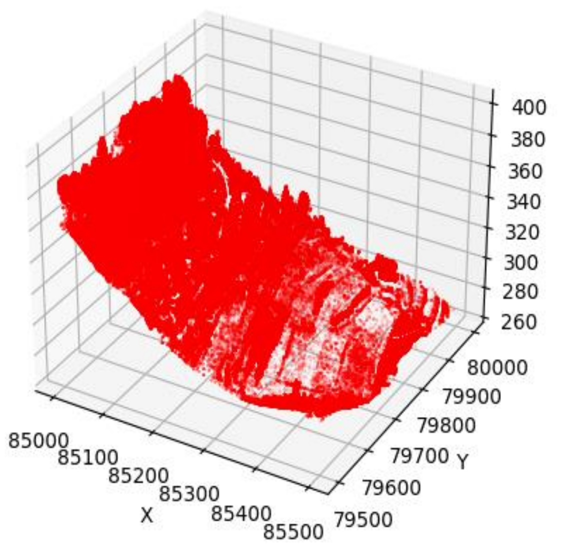

Figure 8. Prediction results of non-ground points

5. CONCLUSION

This study focuses on the filtering of 3D LiDAR point clouds into ground and non-ground points. For this purpose, a non-end-to-end deep learning algorithm is preferred. This algorithm constructs a deep learning model based on a binary classifier. A total of 9 features and point labels are used as inputs, and the output is the predictions of point classes. The datasets used in the study are publicly available ACT (Luxembourg Administration of Cadastre and Topography) benchmark datasets. Due to the limited number of data used in the study, the k-fold cross-validation method is employed, resulting in highly satisfactory results. In the cross-validation method, the training data of Dataset-1 is divided into 4 equal folds, and their validation averages are calculated as 96.83%. The accuracy result for the test data of Dataset-1 is 97%, and for the other data used solely for testing the model (Dataset-2), the accuracy result is computed to be 93%. Based on these findings, it is possible to conclude that the use of geometric features as a non-end-to-end approach for filtering a limited number of point cloud data is a feasible method. This proposed approach can be utilized in various studies such as DTM production, city modeling, urban planning, disaster management (e.g., flood and earthquake), land management, and corridor mapping. In order to achieve sustainable development goals, this approach has the potential to be applied in every stage of various studies. Considering the need for high-accuracy data to reach these goals, this approach can be considered as a suitable option.

Similar to other non-end-to-end methods, the proposed approach also requires some improvements. Future research will explore different feature combinations and different deep learning models. Additionally, point clouds will be classified into multiple classes such as vegetation, building, and road to test the performance of multi-class classification.

REFERENCES

Baek, N., Shin, W. S., & Kim, K. J., 2017. Geometric primitive extraction from LiDAR-scanned point clouds. Cluster Computing, 20, 741-748.

Dittrich, A., Weinmann, M., & Hinz, S., 2017. Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3D point cloud data. ISPRS journal of photogrammetry and remote sensing, 126, 195-208.

Douros, I., & Buxton, B. F. (2002). Three-dimensional surface curvature estimation using quadric surface patches. Scanning, 44(0).

Grilli, E., Farella, E. M., Torresani, A., & Remondino, F., 2019. Geometric feature analysis for the classification of cultural heritage point clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42, 541-548.

Hackel, T., Wegner, J. D., & Schindler, K. (2016). Contour detection in unstructured 3D point clouds. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1610-1618).

Huang, R., Hong, D., Xu, Y., Yao, W., & Stilla, U., 2019a. Multi-scale local context embedding for LiDAR point cloud classification. IEEE Geoscience and Remote Sensing Letters, 17(4), 721-725.

Huang, R., Xu, Y., & Stilla, U., 2019b. Extraction of multi-scale geometric features for point cloud classification. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium (pp. 2499-2502). IEEE.

Lin, C. H., Chen, J. Y., Su, P. L., & Chen, C. H., 2014. Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification. ISPRS Journal of Photogrammetry and Remote Sensing, 94, 70-79.

Liu, Z., Tang, H., Lin, Y., & Han, S., 2019. Point-voxel cnn for efficient 3d deep learning. Advances in Neural Information Processing Systems, 32.

Nurunnabi, A., Teferle, F. N., Li, J., Lindenbergh, R. C., & Hunegnaw, A., 2021. An efficient deep learning approach for ground point filtering in aerial laser scanning point clouds. Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci, 24, 1-8.

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.

Sawyer, T. W., Diaz, A., Salcin, E., & Friedman, J. S., 2021. Using principle component analysis to estimate geometric parameters from point cloud LIDAR data. Laser Radar Technology and Applications XXVI, 11744, 1174403.

Gilani, S. A. N., Awrangjeb, M., & Lu, G., 2016. Robust building roof segmentation using airborne point cloud data. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 859-863). IEEE.

Schmohl, S., & Sörgel, U., 2019. Submanifold sparse convolutional networks for semantic segmentation of large-scale ALS point clouds. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4, 77-84.

Shi, S., Wang, X., & Li, H., 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 770-779).

Srinivasan, K., Cherukuri, A. K., Vincent, D. R., Garg, A., & Chen, B. Y., 2019. An efficient implementation of artificial neural networks with K-fold cross-validation for process optimization. Journal of Internet Technology, 20(4), 1213-1225.

Weinmann, M., Jutzi, B., & Mallet, C., 2014. Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(3), 181.

Weinmann, M., Jutzi, B., Mallet, C., & Weinmann, M., 2017. Geometric Features and Their Relevance for 3D Point Cloud Classification. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4.

Winiwarter, L., Mandlburger, G., Schmohl, S., & Pfeifer, N., 2019. Classification of ALS point clouds using end-to-end deep learning. PFG–journal of photogrammetry, remote sensing and geoinformation science, 87, 75-90.

Wong, T. T., & Yeh, P. Y., 2019. Reliable accuracy estimates from k-fold cross validation. IEEE Transactions on Knowledge and Data Engineering, 32(8), 1586-1594.

Zeybek, M., & Biçici, S., 2021. Geometric feature extraction of road from UAV based point cloud data. In Innovations in Smart Cities Applications Volume 4: The Proceedings of the 5th International Conference on Smart City Applications (pp. 435-449). Springer International Publishing.