FEATURE RELEVANCE ANALYSIS FOR 3D POINT CLOUD CLASSIFICATION USING DEEP LEARNING

Ashutosh Kumar [1,2], Katharina Anders [3,4], Lukas Winiwarter [3], Bernhard Höfle [3,4*]

[1]Institute of Industrial Science, The University of Tokyo, Komaba, Japan – ashutosh@iis.u-tokyo.ac.jp
[2]School of Engineering, The University of Tokyo, Hongo, Japan
[3]3D Geospatial Data Processing Research Group (3DGeo), Institute of Geography, Heidelberg University, Heidelberg, Germany –
(katharina.anders, lukas.winiwarter, hoefle)@uni-heidelberg.de
[4]Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Heidelberg, Germany

KEY WORDS: Terrestrial Laser Scanning, Machine Learning, Semantic Labelling, Neural Network, Fully Connected Network

ABSTRACT:

3D point clouds acquired by laser scanning and other techniques are difficult to interpret because of their irregular structure. To make sense of this data and to allow for the derivation of useful information, a segmentation of the points in groups, units, or classes fit for the specific use case is required. In this paper, we present a non-end-to-end deep learning classifier for 3D point clouds using multiple sets of input features and compare it with an implementation of the state-of-the-art deep learning framework PointNet++. We first start by extracting features derived from the local normal vector (normal vectors, eigenvalues, and eigenvectors) from the point cloud, and study the result of classification for different local search radii. We extract additional features related to spatial point distribution and use them together with the normal vector-based features. We find that the classification accuracy improves by up to 33% as we include normal vector features with multiple search radii and features related to spatial point distribution. Our method achieves a mean Intersection over Union (mIoU) of 94% outperforming PointNet++'s Multi Scale Grouping by up to 12%. The study presents the importance of multiple search radii for different point cloud features for classification in an urban 3D point cloud scene acquired by terrestrial laser scanning.

1. INTRODUCTION

3D point clouds are sets of point coordinates (x, y, z) in space which can be acquired by Light Detection and Ranging (LiDAR), or by Dense Image Matching (DIM) using multiple images taken from different points of view. Laser scanning uses LiDAR measurements to record vast numbers of points in a geographic scene that represent the geometry and textural information of objects (Otepka et al., 2013). It can even capture (micro-)topography of the surface that might be occluded by vegetation and is a powerful remote sensing technique that has been used, for example, in generating 3D models of cities (Rebecca et al., 2008), for flood and glacier modelling (Telling et al., 2017), sand dune modelling (Dong, 2015), and extraction of vegetation parameters (Rosette et al., 2009).

For extraction of information from a point cloud containing different objects, typically a classification step is required. By classification, we refer to the assignment of a semantic label to each point in the point cloud. For example, if vegetation should be separated from other classes, we can use a label (e.g., the numeric value 1) for vegetation points while assigning a different label to other classes (e.g., 0). The assignment of labels depends on the type of information we want to retrieve. In this paper, we assign labels to points by the semantic object class they represent. In the context of an urban scene, typical classes are *Building, Vegetation, Natural terrain, Man-made terrain, Water, Vehicle, and Scanning artefact (Noise)*.

After classification, the point cloud can serve as data basis for analyses that may answer specific questions, such as the separation of natural and man-made terrain in urban planning tasks. With recent advances in technology, Mobile Laser Scanning (MLS) has emerged to scan large areas, such as entire cities (Balado et al., 2018). The 3D model produced from such scanning may, for example, serve as an input map to autonomous systems for navigation (Whitty et al., 2010; Yue et al., 2018), or automatic inspection of power transmission lines (Qin et al., 2018). For such autonomous systems, the classification is important to recognize the type of object represented by the point cloud. Thus, automatic classification of point clouds is crucial to achieve a higher rate of automation and is therefore a frontier research topic.

Several traditional machine learning classification algorithms such as Support Vector Machine (SVM), Decision Trees and Random Forest are commonly applied to 3D point cloud data. However, the capacity of these models to learn is limited and they cannot extract additional features from the training data other than the ones obtained beforehand and input to the model. Neural networks (Rumelhart et al., 1986) have been around for many years, but have recently evolved as deep neural networks because of increased computational capacities and larger amounts of data readily available for training. Such deep neural networks can model highly varying and complex functions to learn the mapping between the input and the output (Brahma et al., 2016). Deep learning methods have recently been shown to be successful in various real-world problems such as image classification, object detection, image captioning, language translation, and text classification in computer vision and natural language processing (LeCun et al., 2015).

There are two ways of classification using deep learning techniques. One approach learns directly from the raw data, which we call end-to-end learning. The other approach is to first extract features from the data which are input to the neural network for

---

* Corresponding author

learning. We call this approach non-end-to-end learning. Attempts have been made for the classification of 3D point clouds using both approaches, for example, by classifying 3D point clouds on a voxel grid as in VoxNet (Maturana and Scherer, 2015), or in a graph-based approach as in SEGCloud (Tchapmi et al., 2017). These methods are non-end-to-end as they do not work directly on the primary point cloud data for classifying them through neural networks. Recently, end-to-end methods such as PointNet (Qi et al., 2017a), and PointNet++ (Qi et al., 2017b) have been developed that work directly on the 3D coordinates of the point cloud for classification. A drawback of end-to-end methods is that larger amounts of data are required for training than for non-end-to-end methods, and the features used for classification by the network are difficult to interpret. Few attempts have been made to classify 3D point clouds by first extracting geometrical and statistical features from the point cloud before classifying the points using a neural network (Zhang et al., 2018). However, the study by Zhang et al. (2018) does not give an interpretation of how different combinations of extracted features affect the result of the neural network-based classification. Furthermore, they only used the normal vector and height above ground as geometrical and statistical features.

In our research, we investigate the neural network-based classification of a terrestrial laser scanning point cloud by extracting meaningful point cloud features. We examine (i) how different combinations of features affect the result of classification, (ii) which sets of features are important for the classification of different class types, and (iii) compare our results to the deep learning framework PointNet++ (cf. Winiwarter and Mandlburger, 2019). The advantage of our approach is that it does not require a large amount of data to train the neural network as we extract useful features from our geographic knowledge of the point cloud scene. From the results in this study, however, we cannot interpret effects of distance from the scanner on the results of classification which can have an impact on the classification results as the objects closer to the sensor will have a higher local point density than those who are far away from the sensor (Kaasalainen et al., 2011).

2. METHODOLOGY

2.1 Dataset

For our research, we use the Semantic3D dataset (Hackel et al., 2017), consisting of around four billion points acquired by terrestrial laser scanning in 30 scenes of diverse urban places, including market squares and streets. Of this dataset, we select the *Bildstein* scene and use the first scan ('bildstein1') only. For each 3D point coordinate in the Semantic3D dataset, the laser return intensity (I), as well as Red (R), Green (G), and Blue (B) values corresponding to the true colors of the scene are provided. The data is classified according to the types in Tab. 1.

Class number	Class type
1	Man-made terrain
2	Natural terrain
3	High vegetation
4	Low vegetation
5	Building
6	Hard scape
7	Scanning artefact
8	Car

Table 1. Classes in the Semantic3D dataset

For our study, we perform two experiments. For both, we exclude unlabelled points from the dataset. In the first experiment, we randomly split the points in the dataset into 72% of points for training, 8% for validation, and 20% for testing. In the second experiment, we divide the scene spatially into two parts, one of which we use as training set and the other one as test set. From the training set and test set of experiment 2, we remove all examples from class 4, class 7, and class 8 which together constitute less than 1.5% of the total points in the dataset. The training set consists of 37% points of the dataset and the test set of the remaining 63%.

2.2 Extraction of Point Cloud Features

The features we use in this paper are established in the 3D point cloud domain and commonly used for 3D point cloud classification. The surface normals of points are estimated based on the local neighborhood in the point cloud via plane fitting, which is solved with an eigenvalue decomposition of the covariance matrix (structure tensor) of the points. The normal vector features (NF) include the normal vector, eigenvalues (Weinmann et al., 2013), eigenvectors and quality of the plane fit derived from residual point distances to the plane (Dorninger and Nothegger, 2007). The feature Echo Ratio (ER) (Höfle et al., 2009) is a measure of the penetrability of the objects represented by the points. Intensity (I) is the strength of the laser signal scattered back from an object (Höfle and Pfeifer, 2007).

Feature		Search Radius [cm]					
		1	5	10	50	100	200
NF	Normal vector	✔	✔	✔	✔	✔	✔
	Plane-fit quality	✔	✔	✔	✔	✔	✔
	Eigenvalues	✔	✔	✔	✔	✔	✔
	Eigen vectors	✔	✔	✔	✔	✔	✔
ER	Echo Ratio	X	✔	X	X	X	X
NStat	Minimum Z	X	X	X	✔	X	X
	Normalized height	X	X	X	✔	X	X
	Range	X	X	X	✔	X	X
	Variance	X	X	X	✔	X	X
	Rank	X	X	X	✔	X	X
	Positive Openness	X	X	X	✔	X	X
	Points / Volume	X	X	X	✔	X	X

Table 2. Features extracted for different search radii

We derive further features using Neighborhood Statistics (NStat) in the local 2D neighborhood of points, namely the range, normalized height, variance, rank, points/volume, and openness. Range is the vertical distance from the highest to the lowest z value in the 2D neighborhood. Normalized height is the relative height of the current point's z value to the minimum z value in its 2D neighborhood. Variance describes the variance of all z values in the local 2D neighborhood. The rank of a point represents its relative vertical position 2D neighborhood and accordingly is 100% if the point is a local maximum and 0% if the point is a local minimum. Points/volume describes the point count per unit volume of a cylinder of the neighborhood radius. Openness is defined as the degree of dominance or enclosure of a location on an irregular surface (Yokoyama et al., 2002).

We calculate the described features on the point cloud data for different search radii as shown in Tab. 2. For the calculation of

the point cloud features, we use the point cloud processing software OPALS (Pfeifer et al., 2014).

2.3 Preparation of Training Dataset

We calculate the features in neighborhoods of different search radii, which we denote as a subscript to the name of the feature set. For example, $(NF)_{1,5}$ means the normal vector, eigenvalues, eigenvectors, and quality of the plane fit are calculated for search radii of 1 cm and 5 cm, respectively. Additionally, we use the intensity value for each point. We do not use the RGB values from the dataset as our objective is to study the classification only from features derived directly from the LiDAR point clouds.

It is important to note that the number of points per class in the training dataset for experiment 1 is highly imbalanced as shown in the histogram of point numbers per class in the training set (Fig. 1). The balance is improved in the training dataset of experiment 2, achieved by the alternative approach of creating it (section 2.1).
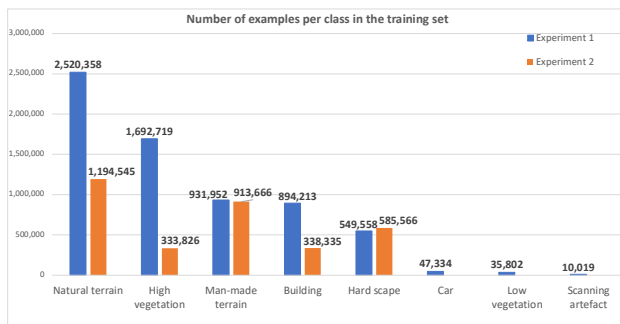

Figure 1. Histogram showing the number of examples in the training set

We prepare the training set by combining different features. The feature sets are listed in Tab. 3 with IDs, to which we subsequently refer. For instance, $\{(NF)_1, (ER)_5, (I)\}$ represents one feature set of training data.

ID	Features
A	$\{(NF)_1, (ER)_5, (I)\}$
B	$\{(NF)_{1,5,10,50,100,200}, (ER)_5, (I)\}$
C	$\{(NF)_{1,5,10,50,100,200}, (ER)_5, (NStat)_{50}, (I)\}$
PN	PointNet++ Multi Scale Grouping using search radii of 25, 50, 100 and 200 cm

Table 3. Different sets of features used in the experiment

2.4 Training the Neural Network

After calculating features for the training datasets, we input them to a fully connected neural network. For meaningful comparisons of different sets of features, we use a fixed strategy for defining the neural network architecture for all sets of features. The fixed hyperparameters for experiment 1 and 2 are listed in Tab. 4.

Since we aim to achieve a correct classification for all classes and the dataset is highly imbalanced, we chose Intersection over Union (IoU, Eq. 1), which is a per-class measure, as evaluation metric. The evaluation measure per class i is calculated as

$$IoU_i = \frac{a_{ii}}{a_{ii} + \sum_{j \neq i} a_{ij} + \sum_{k \neq i} a_{ki}} \qquad (1)$$

where a_{ii} is the number of samples from the ground-truth class i predicted as class i in the confusion matrix, a_{ij} is the number of

samples from the ground-truth class i predicted as class j in the confusion matrix, and a_{ki} is the number of samples from the ground-truth class k predicted as class i in the confusion matrix. To quantify overall performance, we use the mean IoU (mIoU) over all classes without weighting. The same evaluation metric is used by the Semantic3D benchmark (Hackel et al., 2017). We study sets of different feature combinations and their effect on the classification result by training a neural network. The results of our study are presented in the subsequent section.

Hyperparameter	Experiment 1	Experiment 2
Regularization	$\lambda = 0$	L2 ($\lambda = 0.5$)
Hidden layers	5	3
Nodes in hidden layers	60	60
Mini-batch size	128	128
Epochs	10	10
Optimizer	Adam	Adam
Feature Scaling	Standard Scaling	Standard Scaling
Learning Rate	0.001	0.001
Batch Normalization	True	True
Dropout	False	False
Classifier	Softmax	Softmax
Loss Function	Multinomial Cross-entropy	Multinomial Cross-entropy

Table 4. Hyperparameters of the Neural Network

2.5 PointNet++

As a state-of-the art baseline, we employ PointNet++. Since it automatically extracts features from the neighborhood, we assume comparison to accurately depict the quality of our feature choices. However, PointNet++ does not calculate the full neighborhood representation for every point, but employs two different grouping techniques: Multi-Resolution Grouping (MRG) and Multi-Scale Grouping (MSG). Because of the smaller memory footprint, we used MSG (Qi et al., 2017b).

In MSG, the whole point cloud is only used for feature extraction on the smallest search radius (e.g. 25 cm). This information is gathered at a subset of the total points. For the next radius (e.g. 50 cm), only this subset is used for the extraction. The subset sizes and search radii used in our study are presented in Tab. 5. This inevitably leads to a loss in information, but allows the network to consider larger neighborhoods at all, which would not be possible without subsampling due to high memory requirements (Winiwarter and Mandlburger, 2019).

Level	Number of points	Search radius [m]	Max. neighbors
0	200,000		
1	8,192	0.25	16
2	4,096	0.5	16
3	2,048	1	16
4	512	2	16

Table 5. Number of points per PointNet++-Level, search radii and neighbor count.

The parameters were selected to best represent the search radii used in feature set C, but may be tuned to achieve better results.

3. RESULTS

3.1 Experiment 1

We use the hyperparameters shown in Tab. 4 for training the neural network. In Tab. 6, we present results corresponding to different combinations of features for experiment 1. The results shown here are for the test dataset, and the IoU as well as mean of IoUs (mIoU) scores have been rounded to two digits for readability.

ID	IoU for each class								mIoU
	1	2	3	4	5	6	7	8	
A	0.91	0.84	0.76	0.21	0.84	0.70	0.52	0.71	0.69
B	0.97	0.99	0.99	0.76	0.98	0.95	0.78	0.95	0.92
C	0.98	0.99	0.99	0.75	0.99	0.96	0.87	0.97	0.94
PN	0.94	0.97	0.99	0.61	0.96	0.88	0.48	0.89	0.84

Table 6. Classification results of the test dataset for different combinations of features

From the classification results, we observe that as we increase the number of features, the IoU increases for each class. It is evident that when we use feature set A for training the neural network, class 4 (Low vegetation) and class 7 (Scanning artefact) are being classified with low IoUs of 0.21 and 0.52, which results in a low mIoU of 0.69. When including normal vector features derived from multiple search radii as in feature set B, the IoU increases for all classes. The increase is highest in the case of class 4 (Low vegetation) and class 7 (Scanning artefact).

To also consider the height of points in their local neighborhood, we include normalized height, as well as other features derived from point cloud statistics as in the case of feature set C. Although this does not affect the IoU of class 4 (Low vegetation), the IoU of class 7 (Scanning artefact) increases from 0.78 to 0.87. This increase is because without additional information from the neighborhood statistics, many points of class 7 (Scanning artefact) were misclassified as class 5 (Building) and class 6 (Hard scape) in case of feature set B.

Figure 2. Ground truth labels of the test dataset

Classification using the PointNet++ Multi Scale Grouping framework results in a mIoU of 0.84 compared to 0.94 for feature set C. PN achieves a higher mIoU than feature set A where only a single search radius is taken into account. The performance of PN is good with all classes except class 4 and class 7 where the number of training examples per class is less compared to other classes.

The ground truth test dataset is visualized in Fig. 2. The classification results for feature set C and feature set A are shown in Fig. 3.

(a) $\{(NF)_{1,5,10,50,100,200}, (ER)_5, (NStat)_{50}, (I)\}$
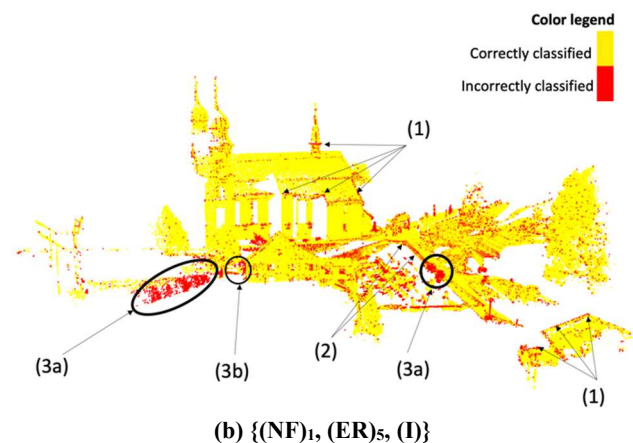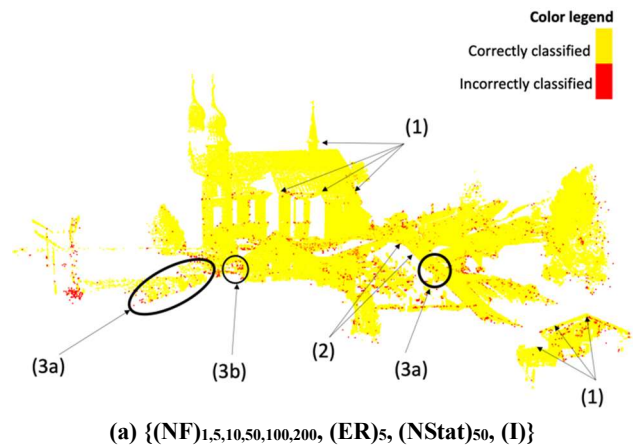

(b) $\{(NF)_1, (ER)_5, (I)\}$

Figure 3. Classification by network with arrows showing the locations of misclassification for (a) feature set C and (b) feature set A

When comparing the classification results in Fig. 3a and Fig. 3b to the ground truth data in Fig. 2, it becomes visible that misclassification occurs at boundaries of the classes such as the corners of the building. This is shown by the arrows numbered (1) in Fig. 3b. When there is overlap between similar classes, such as class 6 (Hard scape) and class 1 (Man-made terrain), classification is dominated by the class which has more training samples. Arrows marked with (2) in Fig. 3 point to an example, where class 6 (Hard scape) points are classified as class 1 (Man-made terrain) near boundaries between the classes. The low IoU of class 4 (Low vegetation) by feature set A is due to misclassification of class 4 into class 3 (High vegetation) as shown by arrows numbered (3a). The arrow labeled (3b) shows misclassifications of class 8 (Car) into class 5 (Building). These misclassifications are reduced when we consider multiple search radii as in Fig. 3a. Fig. 3a also shows that the misclassification is reduced at the corners and boundaries as shown by arrows numbered (1) and (2). Also, the class 4 (Low vegetation) and class 8 (Car) is now correctly classified as shown by the arrows (3a) and (3b), respectively. Confusion matrices normalized by reference with the classification results by feature set A, feature set C, and PN are shown in Tab. 7, Tab. 8, and Tab. 9, respectively.

Predicted / True	1	2	3	4	5	6	7	8
1	**96.39**	2.37	0.22	0.07	0.19	0.76	0.01	0.01
2	0.71	**91.78**	6.59	0.13	0.42	0.34	0.01	0.02
3	0.09	10.21	**87.40**	0.33	1.24	0.62	0.02	0.08
4	3.44	15.91	37.92	**29.50**	4.21	7.83	0.02	1.17
5	0.40	1.09	3.01	0.10	**90.59**	4.67	0.02	0.12
6	5.34	2.31	5.51	0.61	5.09	**80.50**	0.08	0.55
7	0.68	0.68	11.44	0.07	10.58	16.80	**59.42**	0.32
8	0.47	0.95	4.38	0.91	3.66	8.48	0.04	**81.10**

Table 7. Confusion matrix normalized by reference (in %) showing the classification result by feature set A $\{(NF)_1, (ER)_5, (I)\}$

Predicted / True	1	2	3	4	5	6	7	8
1	**98.65**	0.80	0.01	0.12	0.03	0.38	0.01	0.00
2	0.20	**99.60**	0.01	0.02	0.05	0.11	0.00	0.00
3	0.00	0.03	**99.72**	0.03	0.19	0.02	0.01	0.00
4	3.65	1.00	0.37	**91.03**	0.35	3.33	0.06	0.20
5	0.07	0.18	0.32	0.03	**99.30**	0.06	0.02	0.01
6	0.42	0.94	0.04	0.87	0.15	**97.46**	0.05	0.07
7	0.94	0.50	1.47	0.47	1.65	1.83	**92.52**	0.61
8	0.23	0.05	0.02	0.22	0.14	0.25	0.10	**98.99**

Table 8. Confusion matrix normalized by reference (in %) showing the classification result by feature set C $\{(NF)_{1,5,10,50,100,200}, (ER)_5, (NStat)_{50}, (I)\}$

Predicted / True	1	2	3	4	5	6	7	8
1	**96.88**	0.72	0.04	0.74	0.26	1.02	0.16	0.18
2	0.94	**97.77**	0.05	0.05	0.39	0.79	0.01	0.01
3	0.02	0.22	**99.01**	0.14	0.42	0.09	0.02	0.08
4	1.15	0.42	1.47	**91.39**	1.55	3.42	0.17	0.42
5	0.04	0.32	0.51	0.23	**97.98**	0.83	0.03	0.05
6	0.97	1.68	0.25	0.94	0.48	**94.57**	0.91	0.20
7	7.09	0.91	0.73	0.36	0.73	3.45	**85.82**	0.91
8	0.32	0.02	0.06	0.21	0.08	0.49	0.15	**98.67**

Table 9. Confusion matrix normalized by reference (in %) showing the classification result of PN

3.2 Experiment 2

For experiment 2, we use L2 regularization in all layers and also reduced the number of hidden layers to three as compared to experiment 1. The architecture of PN was unchanged. In Tab. 10, we present the results for feature ID A, C and PN. The classification results for feature set C and feature set A are shown in Fig. 4.

From Tab. 10, we can see that IoUs for class 1 and class 3 have strongly improved with feature set C resulting in an overall increase in the mIoU by 31% compared to feature set A. Comparing PN and C to A shows that the improvement in class 3 is similar, but PN does not improve as much as C in class 1, in fact performing worse than feature set A. Most misclassification occurs in class 5 (Building) and class 6 (Hard scape) with all the three feature sets. They are emphasized by arrows marked with (3) in Fig. 4. The normalized confusion matrix in Tab. 11 shows 33.93% misclassification of class 5 into class 6, and 26.29% misclassification of class 6 into class 5 using feature set C.
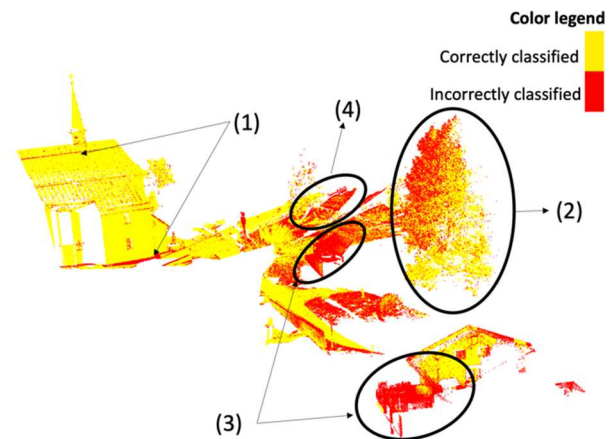
(a) $\{(NF)_{1,5,10,50,100,200}, (ER)_5, (NStat)_{50}, (I)\}$
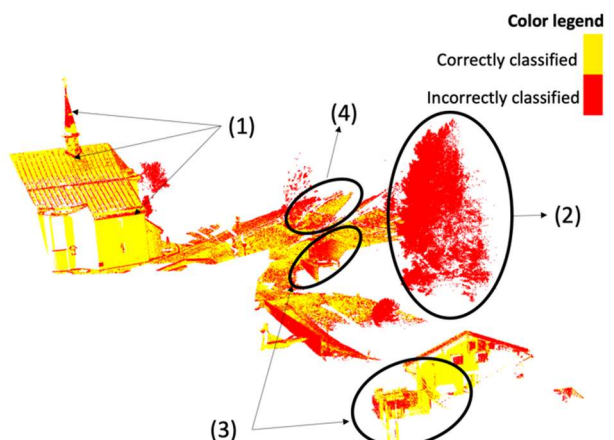
(b) $\{(NF)_1, (ER)_5, (I)\}$

Figure 4. Classification by network with arrows showing the locations of misclassification for (a) feature set C and (b) feature set A

The arrow marked with (2) in Fig. 4b shows the misclassification of class 3 (High vegetation), for which almost all points are predicted as class 5 (Building). This misclassification is largely reduced with feature set C as shown by arrows marked with (2) in Fig. 4a. Similar to experiment 1, the misclassification at locations of geometry change is reduced when we consider NF with multiple search radii and NStat as shown by arrows marked with (1) in Fig. 4a and Fig. 4b. An occurrence of misclassification of class 2 (Natural terrain) into class 6 (Hard scape) is shown by arrows marked with (4) in Fig. 4. From the normalized confusion matrix, it can be seen that 4.10% of the points from class 2 got predicted as class 6.

ID	IoU for each class					mIoU
	1	2	3	5	6	
A	0.58	0.74	0.56	0.38	0.12	0.48
C	0.87	0.84	0.83	0.42	0.19	0.63
PN	0.52	0.81	0.84	0.50	0.20	0.57

Table 10. Classification results of the test dataset for different combinations of features

Predicted True	1	2	3	5	6
1	**96.13**	1.83	0.04	0.96	1.04
2	1.05	**89.14**	3.80	1.91	4.10
3	0.02	3.08	**89.04**	7.06	0.80
5	0.81	6.02	6.84	**52.40**	33.93
6	3.60	4.70	1.87	26.29	**63.54**

Table 11. Confusion matrix normalized by reference (in %) showing the classification result for feature set C $\{(NF)_{1,5,10,50,100,200}, (ER)_5, (NStat)_{50}, (I)\}$

4. DISCUSSION

4.1 Experiment 1

Results of the classification point out that when we consider feature set A as input to the neural network, it results in a low IoU especially in the classes 4 (Low vegetation) and 7 (Scanning artefact). This is due to the features being only calculated based on a single search radius. Thus, they cannot take into account how the same feature will vary depending on the size of the object as we adapt the search radius. The difference in the IoU of class 3 (High vegetation) and class 4 (Low vegetation) can be attributed to the fact that these classes have similar properties and are only distinguished based on a threshold criterion of height. As the number of examples in class 4 is higher than in class 3 (cf. Fig. 1), this resulted in a low IoU for class 4 as most of the points of class 4 got classified as class 3, as shown by the normalized confusion matrix in Tab. 7. Class 7, representing artefacts/noise, is difficult to classify using only one search radius for feature calculation. By calculating NF on multiple search radii, we are able to strongly increase the classification accuracy. This results in an average increase of 33% with respect to the IoU for each class, with an increase in IoU for class 4 (Low vegetation) of 261%, and for class 7 (Scanning artefact) of 34%. This increase can be attributed to the fact that normal vector-based features for different search radii extract information from the point cloud on different scales, representing objects of different sizes. The pattern of variation, however, is similar for the same type of object. For example, in the case of class 6 (Hard scape), change is expected to be small in the normal vector features when increasing the search radius from 1 cm to 200 cm. However, in the case of class 8 (Car), the NF will change as the search radius increases, since a search radius of 200 cm, for example, might include points from around the object itself. Since there will be a distinct pattern for NF derived for multiple search radii of each object, it can be recognized and learned by the Neural Network, effectively improving the classification result. This has been observed in the case of feature set B. The classification result further improves when we additionally consider NStat features as in the case of feature set C. This particularly improves the classification of class 7 (Scanning artefact). Class 7 includes artefacts caused by moving objects. NStat can take such cases into account. For example, if we have an artefact because of a moving car then the local point cloud density in the neighborhood of the artefact will be lower compared to other objects. Such concepts can be learned by the neural network when including NStat features in the classification task. NStat further helps the classification of class 4 (Low vegetation) and class 3 (High vegetation). However, when including NStat features, points from class 6 (Hard scape) are classified into class 4 (Low vegetation). This constituted 13.38% of the points classified as class 4 as shown in Tab. 8, resulting in a decrease of 1.31% IoU for class 4.

4.2 Experiment 2

The results of classification for experiment 2 are consistent with those of experiment 1. The mIoU for experiment 2 compared to experiment 1 is lower due to the fact the we used only 37% of the points in the training set compared to 72 % in experiment 1 and, more importantly, the points in training and test set are not correlated in experiment 2. Strong misclassification occurs in class 5 (Building) and class 6 (Hard scape) with all feature sets, which is expected as they have similar geometric properties. The number of examples for class 5 in the training set is lower than for class 6 (Fig. 1). The training samples for class 5 seem to be insufficient to learn different examples of the test set, which results in misclassifications. Misclassifications in class 3 (High vegetation) and also at locations of geometry change are reduced with feature set C as also seen in experiment 1. Features with multiple search radii as in feature set C and PN have improved learning in the neural network and generalization on the test set.

4.3 Comparison of Experiments with PointNet++

Comparison of our result with PointNet++ shows that our model with feature set C outperforms PN by 12% and 11% in mIoU for experiment 1 and 2, respectively.

In experiment 1, PN especially suffers from misclassifications in class 4 (Low vegetation), of which more than 3% (cf. Tab. 9) are classified as Hard scape. Class 7 points (Scanning artefact) are also misclassified as class 1 (Man-made terrain) in more than 7% of the test cases, leading to a lower mIoU. Since artefacts do not have a clear geometric signature and make up only 0.7% of the dataset, they cannot be effectively learned by the end-to-end approach of PointNet++, which would need more training data (Winiwarter and Mandlburger, 2019).

In experiment 2, we find that classification result with PN is comparable to feature set C in all classes except for class 1 (Man-made terrain). Using feature set C, the IoU in class 1 improves by 67% when compared to PN. PN, however, surpasses the mIoU of feature set A by 19%.

When comparing our model with feature set A to PN or our model with feature set C, we can see the clear improvement of classification metrics, showing the significance of features derived from multiple neighborhood definitions, i.e. search radii. This is especially the case for class 4 (Low vegetation), which is expected to exhibit very different feature values on different search radii due to the penetrability and volume of the objects.

5. CONCLUSION

In this paper, we present a method for the classification of 3D point clouds using deep learning techniques by manually extracting features from the point cloud and then feeding them through a Fully Connected Neural Network. We show that multiple search radii for normal vector features improve the result of classification compared to considering only a single search radius. The result further improves when we include features derived from neighborhood statistics of the point cloud. We can see that even though our training dataset is highly imbalanced in experiment 1, we achieve comparable classification accuracies for all classes. With only 37% of the data as training set in experiment 2, we find that our model generalizes well on the test set. The state-of-the-art deep learning framework PointNet++ in experiment 1 performs equally well as our model except for classes 4 and 7,

two classes with exceptionally low frequency in the dataset. Similar results are observed in experiment 2 except for classification of class 1 where PointNet++ performs even worse than that with feature set A. For the overall classification of point clouds, we can conclude that features extracted with multiple search radii either manually or using the end-to-end deep learning framework PointNet++, greatly improves the classification result. An interesting continuation of this work would be to investigate the further generalization of the model for similar objects. For example, it could be examined if the model can also learn to classify other vehicles such as a mini bus or truck as car, which were not part of the training set.

For the neural network architecture in experiment 1 and 2, we can use Automated Machine Learning (AutoML) techniques such as Adanet (Cortes et al., 2017) for the best neural network architecture search as part of further research.

Further, the ground truth data we use is labelled by humans and always contains some human error associated with the labelling process. An approach to overcome this could be to use synthetic data generated by LiDAR simulation, for example using HELIOS (Bechtold and Höfle, 2016). Such data will have no error for ground truth examples, and it is further possible to augment training examples by scanning a scene from different viewpoints. Using simulated data, the training examples can be easily balanced over the classes, improving the classification result of the rare classes. Furthermore, using different search radii not only for the normal vector-based features, but also for the neighborhood statistics may give better results.

ACKNOWLEDGEMENTS

REFERENCES

Balado, J., Díaz-Vilariño, L., Arias, P. and González-Jorge, H., 2018. Automatic classification of urban ground elements from mobile laser scanning data. *Automation in Construction*, 86, pp. 226-239.

Bechtold, S. and Höfle, B., 2016. HELIOS: A multi-purpose LiDAR simulation framework for research, planning and training of laser scanning operations with airborne, ground based mobile and stationary platforms. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3, pp. 161-168.

Brahma, P.P., Wu, D. and She, Y., 2016. Why Deep Learning Works: A Manifold Disentanglement Perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 27(10), pp. 1997-2008.

Cortes, C., Gonzalvo, X., Kuznetsov, V., Mohri, M. and Yang, S., 2017. Adanet: Adaptive structural learning of artificial neural networks. *Proceedings of the 34th International Conference on Machine Learning,*-Vol. 70, pp. 874-883.

Dong, P., 2015. Automated measurement of sand dune migration using multi-temporal LiDAR data and GIS. *International Journal of Remote Sensing*, 36(21), pp. 5426-5447.

Dorninger, P. and Nothegger, C., 2007. 3D segmentation of unstructured point clouds for building modelling. *International Archives of Photogrammetry and Remote Sensing*, 35(3/W49A), pp. 191-196.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K. and Pollefeys, M., 2017. Semantic3D. net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847.*

Höfle, B., Mücke, W., Dutter, M., Rutzinger, M. and Dorninger, P., 2009. Detection of building regions using airborne LiDAR–A new combination of raster and point cloud based GIS methods. *Proceedings of GI-Forum 2009—International Conference on Applied Geoinformatics*, pp. 66-75.

Höfle, B. and Pfeifer, N., 2007. Correction of laser scanning intensity data: Data and model-driven approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(6), pp. 415-433.

Kaasalainen, S., Jaakkola, A., Kaasalainen, M., Krooks, A. and Kukko, A., 2011. Analysis of incidence angle and distance effects on terrestrial laser scanner intensity: Search for correction methods. *Remote Sensing*, 3(10), pp. 2207-2221.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *nature*, 521(7553), p. 436.

Maturana, D. and Scherer, S., 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 922-928.

Otepka, J., Ghuffar, S., Waldhauser, C., Hochreiter, R. and Pfeifer, N., 2013. Georeferenced point clouds: A survey of features and point cloud management. *ISPRS International Journal of Geo-Information*, 2(4), pp. 1038-1065.

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

Qi, C.R., Su, H., Mo, K. and Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2), p. 4.

Qi, C.R., Yi, L., Su, H. and Guibas, L.J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pp. 5099-5108.

Qin, X., Wu, G., Lei, J., Fan, F., Ye, X. and Mei, Q., 2018. A Novel Method of Autonomous Inspection for Transmission Line based on Cable Inspection Robot LiDAR Data. *Sensors*, 18(2), p. 596.

Rebecca, O. C., Gold, C., Kidner, D., 2008. 3D City modelling from LiDAR Data. *Advances in 3D geoinformation systems*, pp. 161-175.

Rosette, J.A., North, P.R.J., Suarez, J.C. and Armston, J.D., 2009. A comparison of biophysical parameter retrieval for forestry using airborne and satellite LiDAR. *International Journal of Remote Sensing*, 30(19), pp. 5229-5237.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986. Learning representations by back-propagating errors. *nature*, 323(6088), p. 533.

Tchapmi, L.P., Choy, C.B., Armeni, I., Gwak, J. and Savarese, S., 2017. SEGcloud: Semantic segmentation of 3d point clouds. *arXiv preprint arXiv:1710.07563*.

Telling, J.W., Glennie, C., Fountain, A.G. and Finnegan, D.C., 2017. Analyzing glacier surface motion using LiDAR data. *Remote Sensing*, 9(3), p. 283.

Weinmann, M., Jutzi, B. and Mallet, C., 2013. Feature relevance assessment for the semantic interpretation of 3D point cloud data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. II-5/W2, pp. 313–318.

Whitty, M., Cossell, S., Dang, K.S., Guivant, J. and Katupitiya, J., 2010. Autonomous navigation using a real-time 3d point cloud. In 2010 *Australasian Conference on Robotics and Automation* pp. 1-3.

Winiwarter, L. and Mandlburger, G., 2019. Classification of 3D Point Clouds using Deep Neural Networks. *Dreiländertagung der DGPF, OVG und SGPF, Wien. DGPF Tagungsband 28*.

Yokoyama, R., Shirasawa, M. and Pike, R.J., 2002. Visualizing topography by openness: a new application of image processing to digital elevation models. *Photogrammetric engineering and remote sensing*, 68(3), pp. 257-266.

Yue, X., Wu, B., Seshia, S.A., Keutzer, K. and Sangiovanni-Vincentelli, A.L., 2018. A LiDAR Point Cloud Generator: from a Virtual World to Autonomous Driving. *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval* pp. 458-464.

Zhang, L., Li, Z., Li, A. and Liu, F., 2018. Large-scale urban point cloud labeling and reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 138, pp. 86-100.