A TWO-STEP FEATURE EXTRACTION ALGORITHM: APPLICATION TO DEEP LEARNING FOR POINT CLOUD CLASSIFICATION

A. Nurunnabi[1 *], F. N. Teferle[1], D. F. Laefer[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] Center for Urban Science and Progress and Department of Civil and Urban Engineering, Tandon School of Engineering,
New York University, 370 Jay St., #1301C, Brooklyn, NY 11201 USA − debra.laefer@nyu.edu
[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

KEY WORDS: Dimension Reduction, Feature Selection, LiDAR, Machine Learning, Neural Network, PCA, Semantic Segmentation

ABSTRACT:

Most deep learning (DL) methods that are not end-to-end use several multi-scale and multi-type hand-crafted features that make the network challenging, more computationally intensive and vulnerable to overfitting. Furthermore, reliance on empirically-based feature dimensionality reduction may lead to misclassification. In contrast, efficient feature management can reduce storage and computational complexities, builds better classifiers, and improves overall performance. Principal Component Analysis (PCA) is a well-known dimension reduction technique that has been used for feature extraction. This paper presents a two-step PCA based feature extraction algorithm that employs a variant of feature-based PointNet (Qi et al., 2017a) for point cloud classification. This paper extends the PointNet framework for use on large-scale aerial LiDAR data, and contributes by (i) developing a new feature extraction algorithm, (ii) exploring the impact of dimensionality reduction in feature extraction, and (iii) introducing a non-end-to-end PointNet variant for per point classification in point clouds. This is demonstrated on aerial laser scanning (ALS) point clouds. The algorithm successfully reduces the dimension of the feature space without sacrificing performance, as benchmarked against the original PointNet algorithm. When tested on the well-known Vaihingen data set, the proposed algorithm achieves an Overall Accuracy (OA) of 74.64% by using 9 input vectors and 14 shape features, whereas with the same 9 input vectors and only 5PCs (principal components built by the 14 shape features) it actually achieves a higher OA of 75.36% which demonstrates the effect of efficient dimensionality reduction.

1. INTRODUCTION

Feature extraction is crucial in pattern recognition and machine learning (ML) that has been used for object detection, classification, and segmentation (Bishop et al., 2006; Murphy, 2012). Recently, deep learning (DL) has drawn a remarkable attention as a top ML technique for point cloud classification and segmentation (Qi et al., 2017a; Hu et al., 2020; Zou et al., 2021). Some DL algorithms for classification are considered end-to-end as they use point coordinates, normalized coordinates and/or a few features such as intensity and colors (Qi et al., 2017a, b; Thomas et al., 2019; Hu et al., 2020), but many are not (Hsu and Zhuang, 2020; Nurunnabi et al., 2021a) and rely upon using hand-crafted features such as point normal and curvatures as inputs, instead of just points. Many of the latter feature-based DL algorithms use multi-scale (Thomas et al., 2018; Cabo et al., 2019; Atik et al., 2021) and/or multi-type (Blomely et al., 2016; Weinmann and Weinmann, 2019) features to improve classification performance. Laser scanning based point clouds are challenging to classify as they are usually unstructured, having highly variable point density and irregular data format, and are typically capturing sharp features (e.g., edges and corners) and arbitrary surface shapes. Multi-scale and multi-type neighborhood-based features can describe the points' local structure with more detail and help to understand their correspondence with neighboring objects. However, the variable and unstructured nature of point clouds makes the different scales and different types of hand-crafted feature-based network heavy, complex and challenging. The main advantage of a hand-crafted feature-based DL approach is that it involves less training data. It is frequently the case that many candidate features are unnecessary and redundant to the learning process, which hinder the performance of the learning model and can lead to overfitting. Therefore, clear knowledge about the features and their relation to the underlying problem is needed (Nurunnabi et al., 2021a). Besides feature extraction, feature selection methods are used to increase learning capability and to better generalize supervised models, both of which can reduce computational complexity and required storage. Zebari et al. (2020) noted that feature extraction is less influenced by overfitting and achieves better accuracy for the classification tasks compared to feature selection methods. Thus, this paper concentrates on feature extraction.

Several common feature extraction techniques are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Partial Least Squares (PLS), PLS Discriminant Analysis (PLS-DA), Independent Component Analysis (ICA), Canonical Correlation Analysis (CCA), kernel PCA (KPCA) and kernel LDA (KLDA, Scholkopf et al., 1998; Mika et al., 1999; Hastie et al., 2017; Wang et al., 2019). These methods can be grouped into linear (e.g., PCA, PLS-DA) and non-linear approaches (e.g., KPCA and KLDA). Both groups employ dimension reduction and are frequently used in medical image processing, data visualization, text categorization, bioinformatics, chemometrics and astronomy (Wang and Paliwal, 2003; Thomas et al., 2014; Liu et al., 2021). The most popular two feature extraction methods are PCA and LDA. Both optimize through a transformation matrix. PCA optimizes the transformation matrix by maximizing the variance in the projected space, on the other hand, LDA searches the largest ratio of between-class variation and within-class variation, when projecting the given variables

*Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVI-2-W1-2022-401-2022 | © Author(s) 2022. CC BY 4.0 License.

to a subspace (Wang and Paliwal, 2003). Reddy et al. (2020) intensively analysed PCA and LDA as dimension reduction techniques with several ML algorithms that showed most of the cases PCA outperforms LDA. However, usage of the above techniques has been either in supervised approaches or deployed in computationally intensive ways, and thus, poorly suited for feature-based DL algorithms for point cloud classification. To the best of our knowledge, to date a DL approach has not been employed that combines both dimension reduction and feature extraction for point cloud classification.

This paper integrates feature extraction and dimension reduction to improve a DL framework for per point classification of large-scale aerial LiDAR point clouds. Scientific contributions of this paper are: (i) developing a two-step feature extraction algorithm, (ii) understanding the impact of dimension reduction in feature extraction, and (iii) developing a PointNet variant for classification in point clouds that is not end-to-end. The performance of the proposed algorithm is demonstrated on aerial laser scanning (ALS) point clouds. The new PCA based algorithm is intended to reduce the dimensionality of the feature space and improves the performance of the original PointNet algorithm for large-scale point cloud classification.

The remainder of the paper is organized as follows. Section 2 includes a brief discussion on dimensionality reduction, feature extraction, PCA, and PointNet. Section 3 introduces the proposed algorithm. Section 4 demonstrates the newly developed algorithm through two experiments on large-scale ALS point clouds. Concluding remarks are given in Section 5.

2. RELATED METHODS AND PRINCIPLES USED IN THE PRPOSED ALGORITHM

This section presents a short discussion about the basics of feature extraction, feature selection, dimension reduction, PCA, and the PointNet algorithm.

2.1 Dimension reduction, feature extraction and feature selection

The number of variables (features) in a data set is referred to as its dimensionality. Dimension reduction is a widely used process of reducing the number of variables under consideration. This approach is widely applied in areas such as statistics, information theory, and machine learning (ML) and related artificial intelligence techniques. Within ML type approaches, dimensionality reduction is used for both feature extraction and feature selection (Jorgensen et al., 2019). Feature extraction is a process of creating new features based on the original input feature (variable) set to reduce the dimensionality of the feature space. In contrast, feature selection refers to the selection of a small group of features that maximizes relevance to the target (e.g., class labels in classification) and minimize redundancy between the features (Kotsiantis, 2011). In many algorithms, feature extraction, feature selection and dimension reduction are used together for feature optimization and to improve classification accuracy (Jorgensen et al., 2019; Liu et al., 2021).

2.2 Principal component analysis

PCA is one of the most classic statistical techniques, frequently used for point cloud processing, and is employed in feature extraction, outlier detection, regression and classification, as well as dimension reduction (Jolliffe, 2002; Nurunnabi et al., 2012; 2015; Hastie et al., 2017; Grilli et al., 2021). PCA aims to transform a set of possibly correlated variables into a set of uncorrelated variables that can be considered as a set of orthogonal linear combinations of variables that maximize the variance of each combination in rank. These uncorrelated variables, representing the reduced dimensions are called Principal Components (PCs). PCs can be computed by using the well-known Singular Value Decomposition (SVD) to the covariance matrix, *C*,

$$C = \frac{1}{n} X^T X, \tag{1}$$

where *X* is the mean centered data matrix, having *n* observations and *m* variables, $X = x_i - c$; (i = 1, 2, …, *n*), x_i is the *i*th row of the matrix *X*, and *c* is the centre (mean) of *n* observations. PCs are usually sorted into descending order of the non-negative eigenvalues. The 1st PC describes the largest proportion of the data variance. The 2nd PC explains the majority portion of the remaining variance, and the following PCs successively explain the highest variance possible under the orthogonality conditions. Hence, from the beginning, a small number of PCs may explain a sufficient portion of the variance in a data set without losing data information. Thus, PCA has been applied as a successful dimension reduction technique. Besides classical PCA, there are many variants (e.g., Hubert et al., 2005; Nurunnabi et al, 2014). Some produce robust PCs in the presence of outliers. For example, Hubert et al. (2005, 2012) developed robust PCA, ROBPCA that employed a robust covariance matrix together with projection pursuit (PP) to get robust PCs. Nurunnabi et al. (2013, 2014, 2015) showed the advantages of using outlier diagnostic approaches before performing classical PCA to achieve robust results in the presence of outliers and noise. The authors (Nurunnabi et al., 2014; 2015) also showed that the diagnostic PCA was significantly faster than the robust PCA. Additionally, the approach of Kernel Principal Component Analysis (KPCA) is a non-linear PCA has also been used for dimension reduction (Scholkopf et al., 1998; Sidhu et al., 2012). KPCA uses kernel methods to project the data in a higher dimensional space, and then perform PCA on that higher space.

2.3 PointNet, and deep learning in point cloud

PointNet (Qi et al., 2017a) was the first end-to-end DL algorithm that feeds point cloud of irregular data format directly into the network, and offers a simple and computationally efficient means for indoor point cloud classification. It provides a unified architecture applied for part segmentation, object classification, and scene-based semantic parsing. This deep neural network processes raw points in a point cloud *P* that can be arranged in a matrix of order $n \times d$, where *n* is the number of points and *d* is the number of features (variables). For a LiDAR based 3D (three-dimensional) point cloud, the common features are the three respective point coordinates (*x*, *y* and *z*). In PointNet, these *x*, *y*, and z are used as a basic setting with the choice of using additional features e.g., colors (*R*, *G*, *B*), intensity (*I*), and Return Number (*RN*). PointNet learns a spatial encoding of per point features using shared multi-layer perceptrons (MLPs) followed by a global max-pooling function. Shared MLPs transform features from lower dimensions (*d* = 3) to the higher dimension (*d* = 1024), and share their parameters among all the points of each layer. The max-pooling is computed as the global signature of the maximal response among all the points. To make the semantic labelling invariant to certain geometric transformation, this algorithm joins T-Net (a spatial transformer network; Jaderberg et al., 2015) in two stages. Thus, PointNet combines three modules: a symmetry function, local and global information aggregation, and two joint alignment networks to align input points and the point features. However, PointNet ignores a point’s local spatial structure which limits the learning of in-depth patterns in a complex scene.

A variety of DL methods for point cloud classification were developed that employing PointNet as a cornerstone (Qi et al., 2017b; Li et al., 2018; Hu et al., 2020). Qi et al. (2017b) introduced a method where point features are learned by aggregating the spatial structure of the neighbouring points. In PointCNN (Li et al., 2018), the approach changes point neighbors to a canonical order for enabling typical convolution process to a normal role. Thomas et al. (2019) presented a new type of spatially deformable kernel point convolution (KPConv) that learns to adapt kernel points to local geometry. Hu et al. (2020) developed a random point sampling-based algorithm, RandLA-Net, for point cloud classification. To avoid the possibility of discarding key features (because of random sampling) this method successfully incorporates a local feature aggregation module to effectively preserve geometric details. Su et al. (2022) proposed a learnable attention module-based network, DLA-Net, that can be easily implanted into various network architectures for point cloud segmentation. More on DL based point cloud classification methods is available in Bello et al. (2020) and Li et al. (2021). PointNet is employed herein because of its simplicity to understand and implement. Moreover, it is effective and sufficiently fast to demonstrate our method on large-scale point clouds.

3. METHODOLOGY

The proposed method consists of three main steps including two steps for feature collection, feature extraction and dimension reduction (see proposed workflow in Fig. 1). The third step implements PointNet for per point classification.

3.1 Feature collection

At the beginning of feature extraction, the relevant features are collected. In this paper, two types of features are used; one is those available from the laser scanner and the resulting point cloud. Herein, these are referred to as LiDAR feature (LiF). This group includes points coordinates (*x*,*y*,*z*), normalized coordinates (x_n, y_n, z_n), return number (*RN*), intensity (*I*) and points local height z_h. The point’s local height, z_h is the height difference between the *z* values of the interest point and the lowest point in a local neighbourhood. In the absence of *RN* and *I*, other features such as scan angle could be used (Nurunnabi et al., 2021b).

The second group of features are the local saliency features (e.g., normal and curvatures) that are usually derived based on a local neighborhood of each point p_i. To get the respective neighborhood in three dimensions, the well-known *k* nearest neighbour (*k*NN) search algorithm is employed, which avoids problems with point density variation and lack of adequate redundant observations (Nurunnabi et al., 2015). Local saliency features are known as shape features (SFs) and describe local geometry. SFs for a point of interest are derived from the local covariance matrix of the respective neighbors. Here PCA is used to generate the SFs. As per Eq. (1), for every point (p_i; p_x, p_y, p_z) in the point cloud, a covariance matrix *C* of order 3 with the point coordinates within their respective neighborhood of size *k* is generated. Then PCA is performed on *C* to ascertain the 3 PCs (PC1, PC2, PC3) with corresponding eigen values λ_2, λ_1 and λ_0, where $\lambda_2 \geq \lambda_1 \geq \lambda_0 \geq 0$. The most common SFs are: three eigenvalues (λ_2, λ_1, λ_0), surface point normal vector (n_x, n_y, n_z), curvature (σ), linearity (*L*), planarity (*P*), scattering (*S*), omnivariance (*O*), eigentropy (*E*), plan offset (*PO*) and verticality (θ) as defined in Pauly et al. (2002), Weinmann et al. (2015), and Nurunnabi et al. (2021a),

$$\text{curvature, } \sigma = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}, \tag{2}$$

$$\text{linearity, } L = \frac{e_2-e_1}{e_2}, \tag{3}$$

$$\text{planarity, } P = \frac{e_1-e_0}{e_2}, \tag{4}$$

$$\text{scattering, } S_\lambda = \frac{e_0}{e_2}, \tag{5}$$

$$\text{omnivariance, } O = \sqrt[3]{e_0 e_1 e_2}, \tag{6}$$

$$\text{eigentropy, } E = -\sum_{i=0}^{2} e_i ln(e_i), \tag{7}$$

$$\text{plane offset, } PO = \sum_{i=0}^{k} r_i \text{ , and} \tag{8}$$

$$\text{verticality, } \theta = arccos|a.b| \tag{9}$$

where, the eigenvalues λ_i (*i* = 0, 1, 2) are normalized as $e_i = \lambda_i/\sum_0^2 \lambda_i$, r_i is the *i*th residual distance of the interest point and to the fitted plane, and *a* and *b* are the normals of the fitted plane based on the local neighborhood and the *x-y* plane, respectively.

3.2 Dimension reduction

In this section, PCA is again used – this time to reduce the dimension of the feature space. Now the PCs (linear combinations of SFs) are used as the new features derived by the SFs matrix of the 14 columns involved in all of 12 SFs mentioned in Section 3.1. Following the basic principle of PCA, the orthogonal linear combinations (principal components) of the correlated variables in the feature space are sought. For this the LiFs are excluded as they are mutually uncorrelated to each other, whereas, local neighborhood based SFs should have correlation to each other. The number of required PCs is established by investigating the maximal variance, as explained by the PCs. The user can define the number of PCs based on their data and the variance explained by the PCs.

3.3 Implementation of DL, PointNet, algorithm

This last (i.e., third) step implements the PointNet architecture modified to be feature based. Unlike the classic PointNet, with the point coordinates (*x*, *y*, *z*), here normalized coordinates (x_n, y_n, z_n), *RN*, *I*, z_h and hand-crafted SFs based PCs are used as the input vectors. PointNet is implemented in four configurations using only (i) the three coordinates of each point, (ii) the LiFs, (iii) the LiFs and SFs, and (iv) the LiFs and different groups of PCs. In this paper, the standard structure of PointNet is followed. PointNet uses the Rectified Linear Unit (*ReLU*) and the Softmax activation functions for the hidden layers and the output layer, respectively. Multiple cross entropy is used as the loss function, and a stochastic optimizer (*Adam*) is used to train the model. To reduce the influence of vanishing and exploding gradients, this network uses the ‘*He initialization*’ strategy (He et al., 2015) with the *ReLU* activation function, while *Batch Normalization* (Ioffe and Szegedy, 2015) is used for all the layers. The reader can consult Goodfellow et al. (2016) for more technical details used in DL and PointNet algorithm.

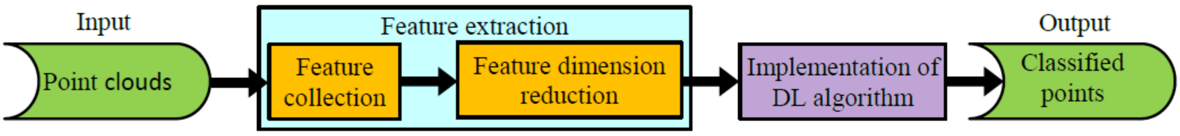

Figure 1: Workflow of the proposed algorithm.

Input features	Group 1	Group 2 (LiFs)	Group 3 (LiFs+SFs)	Group 4	Group 5	Group 6	Group 7
	x, y, z	*x, y, z, x_n, y_n, z_n, RN, I, z_h*	Group 2 + all SFs	Group 2+10PCs	Group 2+ 7PCs	Group 2+ 5PCs	Group 2+ 3PCs

Table 1. Groups of features that are used in the network as the input vectors. Point coordinates (*x*, *y*, *z*), normalized point coordinates (x_n, y_n, z_n), *RN* (return number), *I* (intensity), z_h (height of the interest point), and *N*PCs (*N*: number of PCs).

4. EXPERIMENTS, RESULTS AND EVALUATION

Two real world ALS point cloud data sets were used for per point classification. The results of the classifiers are presented here in brief based on standard evaluation metrics of F_1-score (F_1), mean F_1 (mF_1), Intersection over Union (IoU), mean IoU (mIoU), and the Overall Accuracy (OA), c.f., Hsu et al. (2020) and Zou et al. (2021),

$$F_{1_i} = \frac{2TP_i}{2TP_i+FP_i+FN_i}, \tag{10}$$

$$mF_1 = \frac{\sum_{i=1}^{C} F_{1_i}}{C}, \tag{11}$$

$$IoU_i = \frac{TP_i}{TP_i+FP_i+FN_i}, \tag{12}$$

$$mIoU = \frac{\sum_{i=1}^{C} IoU_i}{C}, \text{ and} \tag{13}$$

$$OA = \sum_{i=1}^{C} \frac{TP_i}{TP_i+TN_i+FP_i+FN_i}, \tag{14}$$

where True Positive (TP) is the number of predicted values correctly predicted as actual positive, False Positive (FP) means the number of negative values predicted as positive, if predicted values correctly predicted as an actual negative, we call it True Negative (TN), and False Negative (FN) is the number of positive values predicted as negative. $(\,.\,)_i$ means the results are for the *i*th class. *C* refers to the total number of classes.

4.1 Experiment 1: Vaihingen data set

The first experiment employed the ISPRS (International Society for Photogrammetry and Remote Sensing) benchmark Vaihingen data (Cramer, 2010; Niemeyer et al., 2014). This open access data set was collected in Vaihingen, Germany using a Leica ALS50 system with a scanning height of 500m having a field of view 45°. It has been frequently used for per point labelling (Blomley et al., 2016; Atik et al., 2021). This data set has point densities of 4-6/m^2. It is divided into training and test sets, these consist of 753,876 points and 411,722 points, respectively. Along with the coordinates (*x*, *y*, *z*), each point has *I*, *RN*, and the number of returns. The training area is predominantly residential with small multi-story buildings that covers a 399m×421m area. In contrast, the test area that consists of two scenes is in a city centre comprises high-rise, dense and complex buildings, and covers a 389m×419m area. The points are labelled power lines (PL), low vegetation (LV), impervious surface (IS), car, fence, roof, facade, shrub, and tree. This dataset has an uneven number of points in the groups. The groups of PL, car, and fence have significantly fewer points than the other groups such as tree and IS. To avoid the possible missing of sample points from the classes with fewer points, a stratified sampling approach was employed. First, 20% points of the training set were taken for the validation set. Then the PointNet architecture was run on the training set with a block size of 10m ×10m, a batch size of 32, and 2,048 points sampled per block. Other parameters were remained the same. The network fed the LiFs and SFs of the points, as the input vectors generated by the formulas defined in Section 3.1 with respective neighborhood of size *k* =15. The network was trained using the above-mentioned hyper-parameters for 100 epochs.

The model with the highest accuracy for the validation set was used to label the test data. Next, PCA was performed on the SFs to find PCs with their respective explained variance. Variances and cumulative variances (in percentage, %) explained by the PCs are portrayed by the bar and line diagrams, respectively in Fig. 2. The figure shows that 3PCs, 5PCs, 7PCs and 10PCs explain variance of 90.9%, 97.0%, 99.6% and 100%, respectively. The user can choose the desired level of variance and fix the number of PCs. The performance metrics (Eqs. 10-14) were then estimated. The algorithm is assessed through seven groups of features (defined in Table 1). The results are presented in Table 2 and Fig. 3. In Table 2, Group 4 with LiFs and 10PCs achieves the best overall accuracy (OA: 75.65%). Even Group 7, which considers LiFs and only 3PCs produces an OA of 73.52%, and Group 3 with all LiFs and SFs produces OA of 74.64%. While the original PointNet gets results only of OA = 53.74% and 66.78% using Group 1 (*x, y, z*) and Group 2 (LiFs), respectively. The reader can see that relatively similar performance is achieved by the other performance metrics of F_1, mF_1, IoU and mIoU.

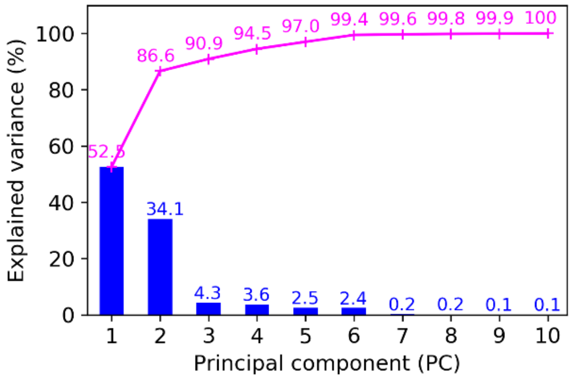

Figure 2. Bar diagram of explained variance versus PCs, and line diagram of cumulative variance of the groups of PCs for the Vaihingen test data set.

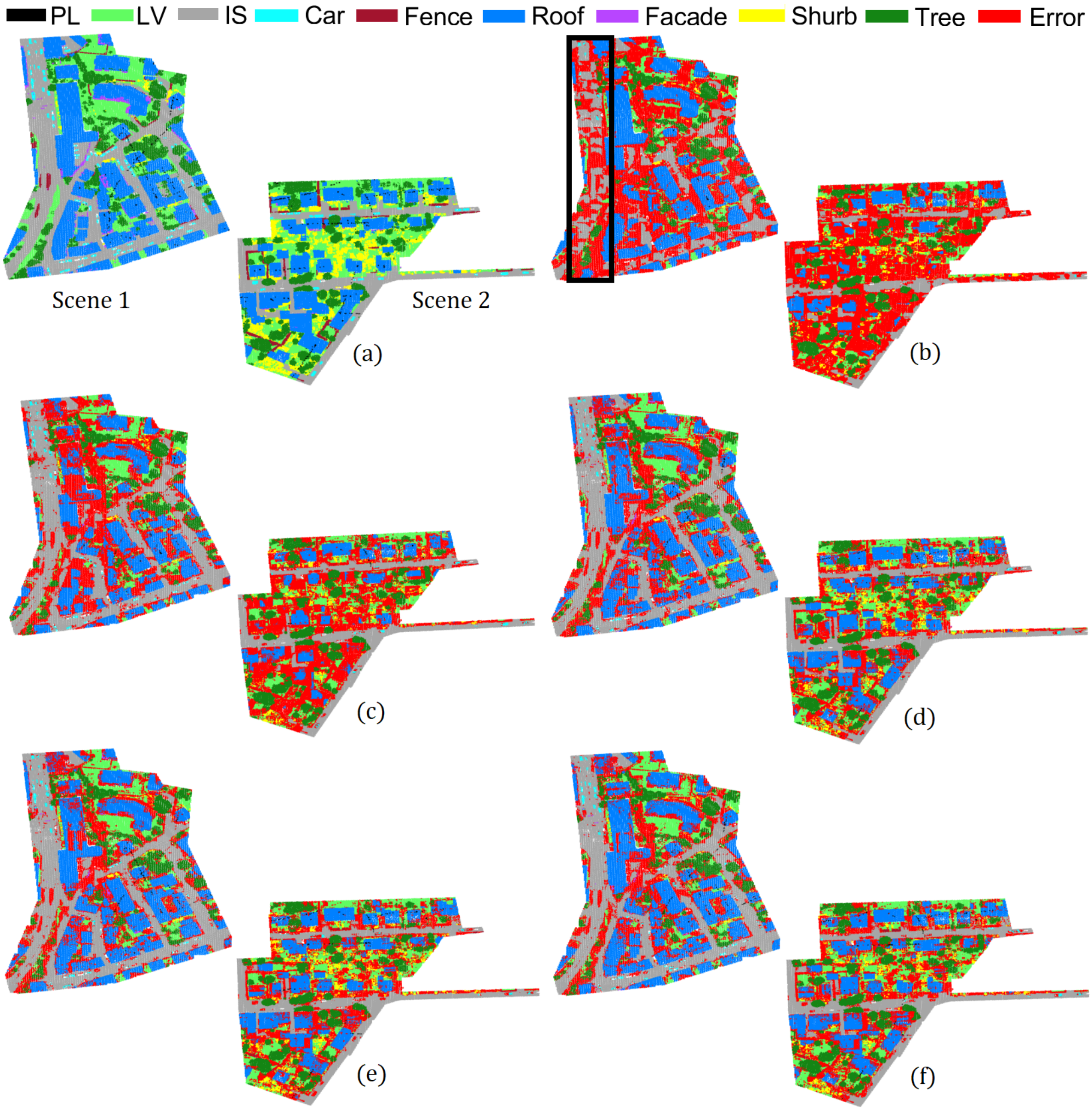

Figure 3: Vaihingen test data set, (a) ground-truth; classification results for the inputs (b) Group 1; (c) Group 2; (d) Group 3; (e) Group 4; (f) Group 6. Results in the black rectangle in (b) show that many points of the impervious surface, fence and low vegetation are misclassified (red). Misclassifications for the same part are almost similar in the figures (e) and (f), but significantly better than (b).

Class	Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7	
	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU
PL	07.35	03.82	27.88	16.19	52.79	35.86	59.65	42.50	58.50	41.34	54.65	37.60	49.09	32.49
LV	46.42	30.22	69.70	53.49	73.67	58.32	73.38	57.95	73.24	57.77	73.55	58.17	72.23	56.53
IS	59.71	42.57	87.66	78.04	88.64	79.59	89.31	80.68	88.87	80.10	88.92	80.05	88.04	78.63
Car	09.41	49.39	37.45	23.03	36.34	22.20	45.36	29.34	37.38	22.99	42.37	26.88	41.17	25.92
Fence	08.80	04.60	09.45	04.96	09.95	05.23	16.72	09.13	11.54	06.12	10.73	05.67	11.32	06.00
Roof	71.28	53.37	71.20	55.28	83.98	72.39	84.74	73.52	86.02	75.48	85.73	75.02	83.52	71.71
Facade	15.71	08.52	15.12	08.18	30.61	18.07	34.45	20.81	32.89	19.68	28.91	16.90	28.81	16.83
Shrub	19.71	10.93	19.34	10.71	29.35	17.20	31.36	18.60	26.79	15.47	25.72	14.76	24.90	14.22
Tree	51.75	34.91	54.92	37.8	67.47	50.91	68.61	52.21	66.83	50.19	67.01	50.39	63.33	46.34
mF_1, mIOU	32.24	21.76	43.64	32.10	52.53	39.97	55.95	42.75	53.56	40.99	53.07	40.60	51.38	38.74
OA	**53.74**		**66.78**		**74.64**		**75.65**		75.43		75.36		**73.52**	

Table 2. Classification results (in percentage, %) of the proposed algorithm for the Vaihingen test (Scenes 1, 2) data set. The performance metrics are F_1-score (F_1), mean F_1 (mF_1), Intersection over Union (IoU), mean IoU (mIoU), and Overall Accuracy (OA).

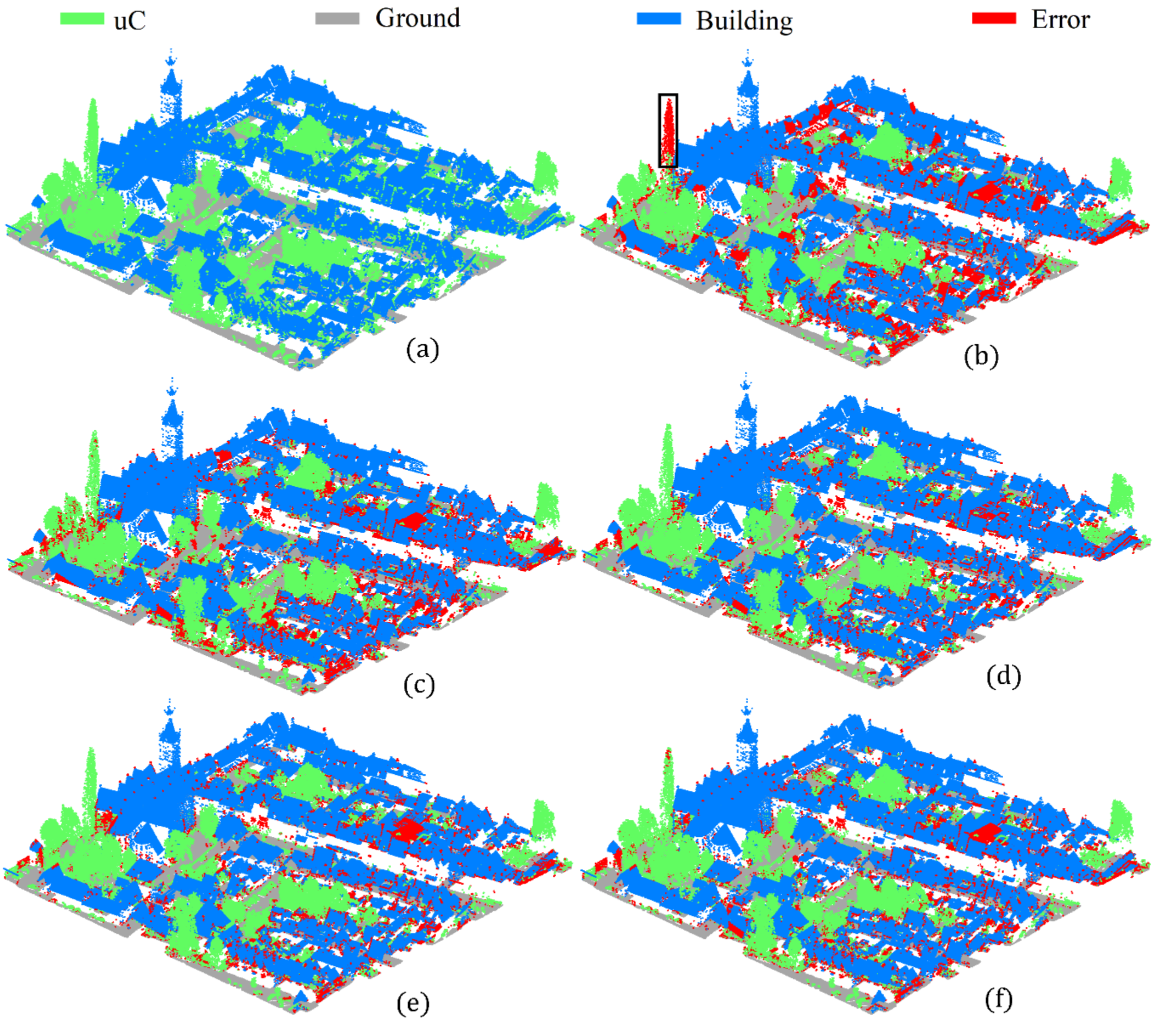

Figure 4: AHN test data set, (a) ground-truth, classification results for the inputs (b) Group 1, (c) Group 2, (d) Group 3, (e) Group 4, and (f) Group 7.

Class	Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7	
	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU
Vegetation	60.13	43.10	72.27	56.59	82.10	69.63	83.45	71.60	81.68	69.03	81.43	68.67	81.67	69.02
Ground	92.95	86.83	91.56	84.43	96.32	92.90	95.28	90.98	95.58	91.54	95.50	91.39	95.36	91.13
Building	78.54	64.66	88.10	78.73	93.16	87.20	92.14	85.42	92.17	85.47	92.17	85.47	91.91	85.04
mF_1, mIOU	77.21	64.82	83.98	73.25	**90.53**	83.25	90.29	82.67	**89.81**	82.01	89.7	81.85	**89.65**	81.73
OA	80.53		87.00		**92.59**		**91.87**		91.77		91.71		91.57	

Table 3. Classification results (in percentage %) of the proposed algorithm for the AHN test data set. The performance metrics are F_1-score (F_1), mean F_1 (mF_1), Intersection over Union (IoU), mean IoU (mIoU), and Overall Accuracy (OA).

4.2 Experiment 2: AHN data set

A second experiment was done using the widely used, publicly available *Actueel Hoogtebestand Nederland* (AHN) data. These data cover the whole of The Nederland (see AHN). This data set has a point density of around 20/m^2, and with up to five returns. Data points are labelled in five different classes: ground, vegetation, building, water, and bridge. These data are arranged into 500m × 500m tiles.

A total of nine differently sized chunks of data from two different tiles were used for training. The training data include urban, semi-urban and rural landscapes. The majority of objects were residential, commercial, and religious small and high-rise buildings with complex roofs, vegetation, and vehicles. The training data contain 4,846,707 points. We also select 1 validation data set and 1 test data set (Fig. 4a) of 927,046 points and 440,987 points respectively. Both data sets include churches, small houses, complex multi-story buildings and vegetation. These were relabelled into three classes: buildings, ground and unclassified (which included vegetation and other objects).

Similar to the previous experiment, to get the SFs, we perform PCA on the covariance matrices of all the points with their respective neighbors of size *k*. PCA was performed 2nd time on the SFs, and to find PCs. Up to the 10PCs were considered. For these the variances and cumulative variances were calculated as explained by the PCs. Bar diagram in Fig. 5 depicts that PC1 and

https://doi.org/10.5194/isprs-archives-XLVI-2-W1-2022-401-2022

PC2 explain the majority of variances, 56.2% and 35.2%, respectively. So, these 2 PCs cover 91.4% of cumulative variance. The line diagram in the same figure shows that 3PCs, 5PCs, 7PCs and 10PCs explain total variances of 94.6%, 99.1%, 99.9% and 99.99%, respectively.

The PointNet network was trained using the training data, and evaluated with the validation data. The same hyper-parameters as the first experiment were used. After 100 epochs the trained model was used to classify the test data (Fig. 4a). PointNet was applied again with the same groups as input vectors defined in Table 1. We calculate all the performance metrics: F_1, mF_1, IoU, mIoU and OA. The results are available in Table 3, and classified labels for different groups of inputs are visualized in different colors in Fig. 4. We see, when all the LiFs and SFs (Group 3) were considered as the input vectors, the DL algorithm achieved an OA of 92.59%. This OA (92.59%) is only 0.72% more than the OA of 91.87%, when 10PCs were used instead of all 14 SFs. OA of 91.77% and 91.57% were found for the input vectors of Group 5 (LiFs+7PCs) and Group 7 (LiFs+3PCs), respectively. If we consider the performance in terms of mF_1 (mean F_1 score), 90.53%, 89.81% and 89.65% were achieved for the input Group 3, Group 5 and Group 7, respectively.

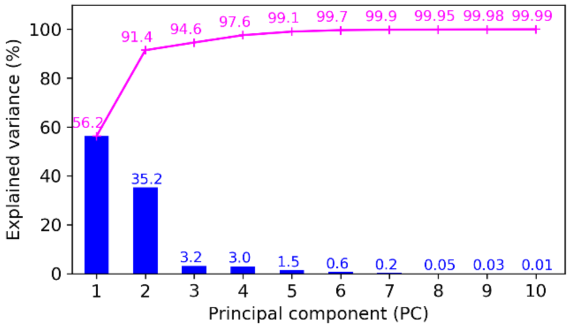

Figure 5. Bar diagram of explained variance versus PCs, and the line diagram shows cumulative variances of the groups of PCs for the AHN test data set.

5. DISCUSSION AND CONCLUSIONS

The proposed PCA based two-step feature extraction algorithm successfully extracted most useful features with reduced dimension of the feature space. At the first step, PCA was used for point coordinates (x, y, z) based local SFs (e.g., normal and curvatures) estimation, and at the second step, PCA was used for dimension reduction in SFs space by generating uncorrelated PCs (linear combinations of reasonably correlated SFs). Two experiments showed that dimension reduction in feature space (matrix) is useful for increasing the performance of feature-based DL algorithm. Results revealed that use of PCs instead of SFs is useful for per point classification as it achieved without significant reduction (even better for the Vaihingen data set) in classification accuracy. The new non-end-to-end (feature-based) DL variant outperforms the original PointNet. The algorithm has potential for classification and segmentation in large-scale outdoor point clouds.

It is known that in the presence of outliers and noise, classical PCA can produce non-robust PCs that can mislead the results of SFs based point cloud processing (Nurunnabi et al., 2014; 2015). Therefore, further study in this direction is needed to investigate the possibilities of robust and/or diagnostic variants of PCA (Hubert et al., 2012; Nurunnabi et al., 2015) for SFs estimation, dimension reduction in SFs space, and their potential for applying within the DL framework. Critically, as robust PCA approaches are usually computationally intensive (Nurunnabi et al., 2014; 2015), so developing fast and robust variants of PCA is useful for efficient DL based per point classification in large-scale point clouds.

ACKNOWLEDGEMENTS

This study is with the Project 2019-05-030-24, SOLSTICE - Programme Fonds Européen de Developpment Régional (FEDER)/Ministère de l'Economie of the G. D. of Luxembourg. Additional assistance was provided through the National Science Foundation award 1940145.

REFERENCES

AHN: *Actueel Hoogtebestand Nederland*. Available at: https://app.pdok.nl/ahn3-downloadpage/.

Atik, M. E., Duran, Z., Seker, D. Z., 2021. Machine learning based supervised classification of point clouds using multiscale geometric features. *Int. J Geo-Information*, 10 (187), 1–23.

Bello, S. A., Yu, S., Wang, C., Adam, J. M., Li, J., 2020. Review: Deep learning on 3D point clouds. *Remote Sensing*, 12(11), 1729.

Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*, Springer.

Blomley, R., Jutzi, B., Weinmann, M., 2016. 3D semantic labeling of ALS point clouds by exploiting multi-scale, multi-type neighborhoods for feature extraction. *GEOBIA, Solutions and Synergies*, 14–16 Sep.

Cabo, C., Ordóñez, C., Sáchez-Lasheras, F., Roca-Pardiñas, J., de Cos-Juez, J., 2019. Multiscale supervised classification of point clouds with urban and forest applications. *Sensors*, 19(20), 4523.

Cramer, M., 2010. The DGPF-test on digital airborne camera evaluation overview and test design. *PFG Photogrammetrie, Fernerkundung, Geoinformation*, 73–82.

Goodfellow, I., Yoshua B., Aaron, C., 2016. *Deep Learning*. MIT press.

Grilli, E., Poux, F., Remondino, F., 2021. Unsupervised object-based clustering in support of supervised point-based 3D point cloud classification. *Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci.*, Vol. XLIII-B2, ISPRS Congress, 5-9 July.

Hastie, T., Tibshirani, R., Friedman, J., 2017. *The Elements of Statistical Learning*. Springer.

He, K., Zhang, X., Ren, S., Sun., J., 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *IEEE ICCV*,1026–1034.

Hsu, P-H., Zhuang, Z-Y., 2020. Incorporating hand-crafted features into deep learning for point cloud classification. *Remote Sensing*, 12 (3713), 1–28.

Hu, Q., et al., 2020. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. CVPR, 11108–11117.

Hubert, M., Rousseeuw, P. J., Vanden Branden, K., 2005. ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47(1), 64–79.

Hubert, M., Rousseeuw, P. J., Verdonck, T., 2012. A deterministic algorithm for robust location and scatter. *Journal of Computational and Graphical Statistics*, 21(3), 618–637.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, 448–456.

Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K., 2015. Spatial transformer networks. ArXiv :1506.02025.

Jolliffe, I. T., 2002. *Principal Component Analysis*, Springer, New York.

Jorgensen, P. E., Kang, S., Song, M. S., Tian, F., 2019. Dimension reduction and kernel principal component analysis. *arXiv preprint arXiv*:1906.06451.

Kotsiantis, S., 2011. Feature selection for machine learning classification problems: a recent overview. *Artificial Intelligence Review*, 42(1), 157–176.

Li, Y., et al., 2018. PointCNN: Convolution on x-transformed points. *Advances in Neural Information Processing Systems*, 31. 820–830.

Li, N., Kähler, O., Pfeifer, N., 2021. A comparison of deep learning methods for airborne lidar point clouds classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 6467–6486.

Liu, W. B., Liang, S. N., Qin, X. W., 2021. A novel dimension reduction algorithm based on weighted kernel principal analysis for gene expression data. *PloS One*, 16(10), e0258326.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K. R., 1999. Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing* IX: Proc. of the 1999 IEEE Signal Processing Society Workshop (cat. no. 98th8468), 41–48.

Murphy, K. P., 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS J. Photogramm. Remote Sens*. 87, 152–165.

Nurunnabi, A., Belton, D., West, G., 2012. Robust segmentation for multiple planar surface extraction in laser scanning 3D point cloud data. *IAPR Int. Conf. on Pattern Recognition* (ICPR'2012), 1367–1370.

Nurunnabi, A., West, G., Belton, D., 2013. Robust outlier detection and saliency features estimation in point cloud data. *IEEE Int. Conf. on Computer and Robot Vision* (CRV), 28–31 May, Regina, SK, Canada, 98–105.

Nurunnabi, A., Belton, D., West, G., 2014. Robust statistical approaches for local planar surface fitting in 3D laser scanning data. *ISPRS J. of Photogramm. and Remote Sens.*, 96, 106–122.

Nurunnabi, A., West, G., Belton, D., 2015. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. *Pattern Recognition*. 48(4), 1404–1419.

Nurunnabi, A., Teferle, F. N., Li, J, Lindenbergh, R., Hunegnaw, A., 2021a. 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*., Vol. XLIII-B1-2021, 31–38, XXIV- ISPRS Congress, 5-9 July.

Nurunnabi, A., Teferle, F. N., Li, J., Lindenbergh, R., Parvaz, S., 2021b. Investigation of PointNet for semantic segmentation of large-scale outdoor point clouds. *Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci*. Vol. XLVI-4/W5, 397–404.

Pauly, M., Gross, M., Kobbelt, L. P., 2002. Efficient simplification of point-sampled surfaces. *IEEE Visualization*, 163–170.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. PointNet : Deep learning on point sets for 3d classification and segmentation. *IEEE 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:1706.02413.

Reddy, G. T., et al., 2020. Analysis of dimensionality reduction techniques on big data. *IEEE Access*, 8: 54776–54788.

Scholkopf B, Smola A, Muller K. R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5), 1299–1319.

Sidhu, G. S., Asgarian, N., Greiner, R., Brown, M. R., 2012. Kernel Principal Component Analysis for dimensionality reduction in fMRI-based diagnosis of ADHD. *Frontiers in Systems Neuroscience*, 6, 74.

Su, Y., et al., 2022. DLA-Net: Learning dual local attention features for semantic segmentation of large-scale building facade point clouds. *Pattern Recognition*, 123, 108372.

Thomas, M., De Brabanter, K., De Moor, B., 2014. New bandwidth selection criterion for Kernel PCA: approach to dimensionality reduction and classification problems. *BMC Bioinformatics*, 15(1), 1–12.

Thomas, H., Goulette, F., Deschaud, J. E., Marcotegui, B., LeGall, Y., 2018. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. *IEEE International Conference on 3D Vision* (3DV), 390–398.

Thomas, H., Qi, C. R., Deschaud, J-E., Marcotegui, B., Goulette, F., Guibas. L. J., 2019. KPConv: Flexible and deformable convolution for point clouds. *IEEE ICCV*, 6411–6420.

Wang, X., Paliwal, K. K., 2003. Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition. *Pattern Recognition*, 36(10), 2429–2439.

Wang, Y., Cang, S., Yu, H., 2019. Mutual information inspired feature selection using kernel canonical correlation analysis. *Experts Systems with Applications*: *X*, 4, 1–9.

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

Weinmann, M., Weinmann, M. 2019. Fusion of hyperspectral, multispectral, color and 3D point cloud information for the semantic interpretation of urban environments. *Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci.*, Vol. XLII-2/W13, 1899–1906.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., Saeed, J., 2020. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, 1(2), 56–70.

Zou, Y., Weinacker, H., Koch, B., 2021. Towards urban scene semantic segmentation with deep learning from LiDAR point clouds: A case study in Baden-Wurttemberg, Germany. *Remote Sensing*, 13, 3220, 1–20.