

# AERIAL POINT CLOUD CLASSIFICATION WITH DEEP LEARNING AND MACHINE LEARNING ALGORITHMS

E. Özdemir<sup>1,2</sup>, F. Remondino<sup>2</sup>, A. Golkar<sup>1</sup>

<sup>1</sup> Skolkovo Institute of Technology (SkolTech), Moscow, Russia  
Email: A.Golkar@skoltech.ru

<sup>2</sup> 3D Optical Metrology (3DOM) unit, Bruno Kessler Foundation (FBK), Trento, Italy  
Email: (eozdemir, remondino)@fbk.eu, web: http://3dom.fbk.eu

Commission II, WGII/4

**KEY WORDS:** point cloud, classification, machine learning, deep learning, urban areas, geometric features

## ABSTRACT:

With recent advances in technology, 3D point clouds are getting more and more frequently requested and used, not only for visualization needs but also e.g. by public administrations for urban planning and management. 3D point clouds are also a very frequent source for generating 3D city models which became recently more available for many applications, such as urban development plans, energy evaluation, navigation, visibility analysis and numerous other GIS studies. While the main data sources remained the same (namely aerial photogrammetry and LiDAR), the way these city models are generated have been evolving towards automation with different approaches. As most of these approaches are based on point clouds with proper semantic classes, our aim is to classify aerial point clouds into meaningful semantic classes, e.g. ground level objects (GLO, including roads and pavements), vegetation, buildings' facades and buildings' roofs. In this study we tested and evaluated various machine learning algorithms for classification, including three deep learning algorithms and one machine learning algorithm. In the experiments, several hand-crafted geometric features depending on the dataset are used and, unconventionally, these geometric features are used also for deep learning.

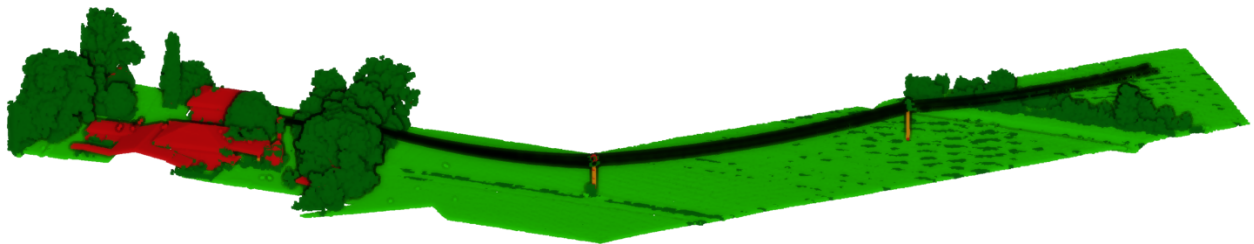


Figure 1. A LiDAR point cloud classified with our approach in 5 classes: buildings (red), powerline poles (orange), powerline cables (black), ground/soil (light green) and trees (dark green).

## 1. INTRODUCTION

Point cloud classification (Figure 1) is nowadays a very interesting research topic. Indeed, 3D geometric data alone is not very interesting for the majority of final users who need further ancillary information (i.e. semantic) to better exploit, use and further process point clouds. In the geospatial community various solutions were presented (Hackel et al., 2017a; Lafarge and Mallet, 2012; Weinmann et al., 2015a), with different benchmark available (Cavegn et al., 2014; Hackel et al., 2017b; Nex et al., 2015; Rottensteiner et al., 2014; Serna et al., 2014; Wichmann et al., 2018) and some commercial solutions also exist. Up today, some reliable solutions exist but, to our knowledge, they are confined to either specific data (e.g. only LiDAR) or scenarios (indoor vs outdoor, terrestrial vs aerial).

In this paper we present our approach for the classification of aerial point clouds in urban scenes. It is a part of our ongoing project on 3D city modelling with previous steps presented in (Özdemir and Remondino, 2019). We are implementing our approach (Section 3.1) with alternative deep and machine learning algorithms. We aim to have a bunch of complementary solutions, evaluating their performances and producing classified aerial point clouds for 3D building modelling purposes. Our aim is to ingest any type of aerial point cloud (either from LiDAR or from photogrammetry) and deliver a semantically segmented point cloud with specific classes.

In the following sections we will be summarizing the related works (Section 2), describing our experimental pipeline in Section 3, sharing our results in Section 4 and making some conclusions of the study in Section 5.

## 2. RELATED WORK

Point cloud classification, which is also named in the literature as semantic labelling, semantic segmentation or semantic classification of point clouds, has been a challenging research field for many years now. During these years, researchers came up with different solutions that could be grouped in (i) data source based and (ii) artificial intelligence (AI) based.

### 2.1 Classification Approaches Based on Data Source

Most of the studies employ LiDAR as data source. Charaniya et al. (2004) classify LiDAR point clouds using height and LiDAR features. Douillard et al. (2011) developed a classification approach based on voxelization. Niemeyer et al. (2012) introduced the Conditional Random Field (CRF) based classification approach whereas Zhang et al. (2013) presented an object-based method based on Support Vector Machines (SVM) together with a connected component analysis. Lin et al. (2014) examined the effects of using a weighted covariance matrix for

eigen-features (also called as eigenvalue features or covariance features) extraction on LiDAR point clouds. Ramiya et al. (2017) proposed a building detection method based on a point cloud segmentation approach.

There are fewer studies employing photogrammetric point clouds Dorninger and Nothegger (2007) presented their work focused on unstructured point cloud classification. Becker et al. (2017) examined the use of colour and geometry-based features for a classification with classic machine learning algorithms. Zhu et al. (2017) proposed a semantic relationship including homogeneity and adjacency. Özdemir and Remondino (2018) proposed a pipeline, based on machine learning, for the segmentation of the photogrammetric oblique point clouds for 3D building modelling purposes.

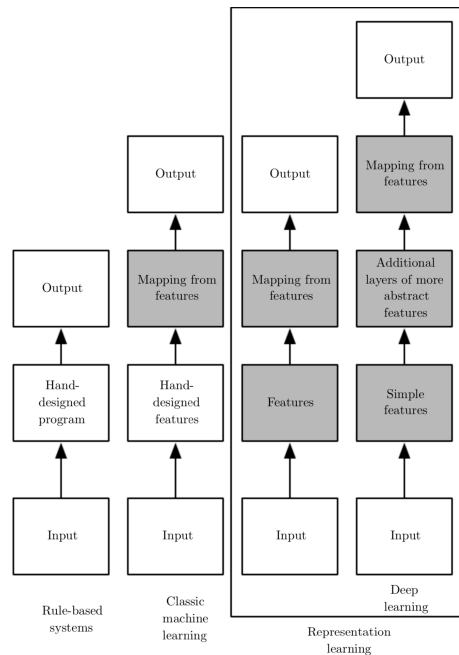


Figure 2. Different approaches in AI for data classification. Grey boxes show modules that can learn from data (Goodfellow et al., 2016).

## 2.2 Classification Approaches Based on Artificial Intelligence Method Used

There are several AI approaches popularly used by the community, which can be categorized as follows (Goodfellow et al., 2016): (i) rule-based, (ii) classic machine learning (ML), (iii) representation learning, (iv) representation learning with deep learning (DL) (Figure 2). LeCun et al. (2015) present a review of different deep learning concepts while a detailed theoretical background of all the aforementioned AI approaches can be found in the Deep Learning book by Goodfellow et al. (2016). These alternative AI approaches have been investigated/used by many communities, such as natural language processing, computer vision, geospatial, etc. While there are four main AI approaches (Fig. 1), we would like to focus on ML and DL approaches, as these are more related to our work. The studies on point cloud classification with ML focuses on different details: in their works, Weinmann et al. (2013) examined the relevance of features for TLS point cloud classification task, Dohan et al. (2015) established a method for a hierarchical approach in semantic segmentation, Weinmann et al. (2015b) developed a method for interpreting the optimal neighbourhood and relevant features for classification with ML, Hackel et al. (2016b) focused

on extracting contours from 3D point clouds, Hackel et al. (2016a), where they represent their approach on classifying point clouds with varying density using a random forest classifier, Thomas et al. (2018) developed their method for terrestrial laser scanning (TLS) classification using a multiscale neighbourhood approach.

Considering the advances in DL field in the recent years, it's implementations for point cloud classification has become as the state of the art in a short period of time and continuing to improve. Some of these works include: Wu et al. (2015) proposed their work on volumetric shapes analysis with deep representations that can handle shape completion and object detection, Qi et al. (2017) introduced a DL solution for point cloud classification, namely PointNet++, that can learn features in contextual scales within a metric space, Landrieu and Simonovsky (2018) came up with a DL framework that uses superpoint graphs to implement contextual relations among objects' parts, Yousefhussein et al. (2018) introduced a deep neural network (DNN) that learns local and global geometry from the points' coordinates of the point cloud.

For a further reading on point cloud segmentation and classification readers may refer to review articles of Nguyen and Le (2013) and Grilli et al. (2017). For a detailed review on deep learning studies on 3D data (including point clouds and RGB-Depth data) classification, readers may refer to review article by Griffiths and Boehm (2019).

## 3. APPROACHES AND EMPLOYED DATA

In order to have a generic and reliable approach, we wanted to examine how different algorithms react to data from different sources. Therefore, we included a dataset acquired with airborne laser scanning (Vaihingen) and one produced from oblique aerial photogrammetry (Dortmund).

The tested classification approaches include a machine learning classifier and three deep neural networks. For machine learning, we utilized a One versus One classifier (OvO, Section 3.4), and for deep learning we used Bidirectional Long Short-Term Memory Deep Neural Network (BiLSTM, Section 3.5) and two different Convolutional Neural Networks (CNN, Sections 3.6 and 3.7)

### 3.1 Aim and Overview of the Classification Approach

Analysing the ways of implementing AI methods (Figure 2), it can be seen that hand-designed (or hand-crafted) features are used in rule-based systems and ML. As representation learning approaches are designed to extract the needed features throughout computational modules (i.e. convolutional layers of a CNN), such features are not provided as input.

Additionally, reviewing the literature (Section 2), one can see that many of the developed approaches are designed for specific kind of data or acquisition hardware (data source): for example, various methods take advantage of LiDAR features (i.e. number of returns, intensity, etc.) or colour information in case of photogrammetric clouds.

Seeking for a wider applicability without depending on ancillary or source specific data, we focus on extracting all the necessary information from the point cloud itself, instead of exploiting existing databases or data source-related features. In addition, considering the computational power needed for a DL execution, we preferred to seek for a DL approach that can run with low computational power (i.e. a mid-class laptop). Therefore, in an unusual way, we used both extracted features and the data itself (only the 3D coordinates in our case) as input for DL methods (Figure 3).

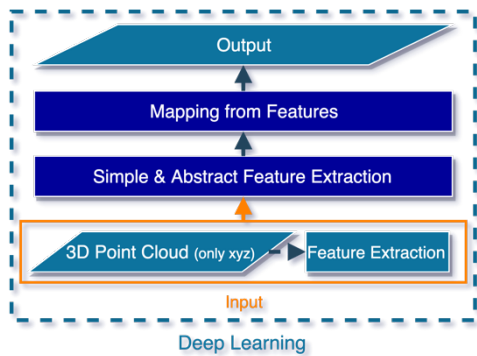


Figure 3. Our proposed framework. (Dark blue boxes show modules that can learn from data.)

### 3.2 Employed Data

Experiments were run using the ISPRS 3D Semantic Labeling Contest Dataset of Vaihingen (Niemeyer et al., 2014) and the ISPRS Benchmark Dataset of Dortmund City Center (Nex et al., 2015).

The Vaihingen dataset, acquired with a Leica ALS50 LiDAR sensor, contains separated point clouds for training (753,876 points, Figure 4) and evaluation (411,722 points). The points have an average density of ~5pts/sqm and they are labelled in nine classes, including: powerline, low vegetation, impervious surfaces, cars, fence/hedge, roof, facade, shrub and tree.

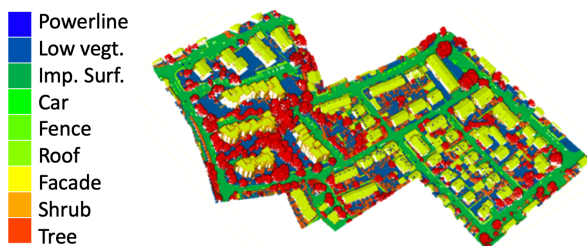


Figure 4. Vaihingen dataset, training data with original classes.

The Dortmund point cloud is generated with oblique photogrammetry, with images acquired by IGI PentaCam at 10cm GSD in the nadir and 8-12cm GSD in the oblique views. The point cloud has a density of ~50pts/sqm. The dataset is designed for dense image matching benchmark and it has no labels on the points for semantic classes. Considering the very high point density of the cloud, we initially down sampled the point cloud to ~6pts/sqm (5 million points in total) before labelling some portions for training (130,000 points) and evaluation (88,000 points, Figure 5).

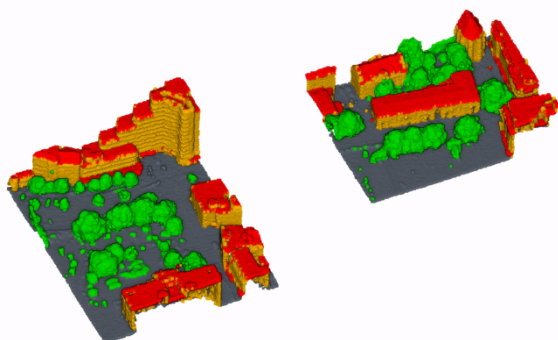


Figure 5. Dortmund point cloud training (left) and evaluation (right) sets. Points are manually labelled in GLO (blue), roof (green), facade (yellow) and vegetation (red).

### 3.3 Feature Extraction and Usage

The employed features can be categorized into two: (i) geometric features (including eigen-features) and (ii) height features. Among the geometric features, we used vertical angle (VA), local planarity (LP) and roughness (R). Within eigen-features, we used linearity (L), planarity (P), surface variation (SV), sphericity (S), omnivariance (O), anisotropy (A) and verticality (V). As height features, we used elevation change (EC) and height above ground (HAG).

Hackel et al. (2016a) report eigen-features computational aspects whereas details of LP, VA, EC and HAG (computed with a DEM extracted from the point cloud) are given in (Özdemir and Remondino, 2019). HAG is now extracted with a new and faster approach consisting of:

- retrieval of the highest ( $P_H$ ) and lowest ( $P_L$ ) points in the input point cloud,
- creation of two imaginary grids with predefined grid nodes and grid node intervals with the elevations of  $P_H$  and  $P_L$ , named as  $G_H$  and  $G_L$ ,
- centering  $G_H$  and  $G_L$  around each point in the point cloud, the point of interest ( $P_i$ ),
- searching for the closest points in the point cloud for each of the points in  $G_H$  and  $G_L$ , named as  $PC_H$ ,  $PC_L$ ,
- finding the highest point in  $PC_H$  and the lowest point in  $PC_L$ , named as  $P_{HH}$  and  $P_{LL}$ ,
- compute HAG as the elevation difference between  $P_i$  and  $P_{LL}$ .

Table 1 summarizes the employed features used for each dataset. With respect to (Özdemir and Remondino, 2019), the different combination of features used in the Dortmund datasets helped to (i) improve classification results and (ii) make the proposed framework suitable to oblique photogrammetric point cloud (i.e. Dortmund) in addition to LiDAR point cloud (i.e. Vaihingen).

Features/Dataset	Vaihingen	Dortmund
Linearity	-	+
Planarity	-	+
Surface Variation	+	+
Sphericity	+	+
Omnivariance	-	+
Anisotropy	+	-
Verticality	+	+
Vertical Angle	*	-
Local Planarity	+	+
Roughness	-	+
Elevation Change	*	+
HAG	+(DEM based)	+(grid based)
<b>Total</b>	<b>6</b>	<b>10</b>

Table 1. Sets of features used for each dataset (+ means used, - means not used) for classification, \* means used only during DEM extraction).

	$d_x$	$d_y$	$d_z$	$F_1$	$F_2$	...	$F_{m-1}$	$F_m$
$P_1$								
$P_2$								
.								
.								
$P_{n-1}$								
$P_n$								

Table 2. Input data matrix representation for DL applications: each row represents the 3D points,  $(d_x, d_y, d_z)$  represents PWDC,  $F_m$  represents extracted features.

These features are utilized in different ways with respect to the employed classification network. For instance, we used feature vectors for ML (Section 3.4) and 1D CNN (Section 3.6) classifier. For 2D CNN and bidirectional long short-term memory (BiLSTM) network, we created an n-by-m matrix (patch), where n points are represented by m number of features. For these cases, we also included patch-wise decentralized coordinates (PWDC) as additional data, so that the deep neural network (DNN) can also extract some local geometry based on the distribution of the points (Table 2).

### 3.4 One vs. One Machine Learning Classifier

The One vs One (OvO) classifier uses a basic approach for multi-class classification problem. Considering that the aim is to have N number of classes, the algorithm trains  $N*(N-1)/2$  binary classifiers. Each classifier votes for the input data for classification. For our classifier, we used kernel ridge regression, radial basis function, support vector machine and insensitive support vector regression trainers (King, 2009). For this classifier, each point is represented by a feature vector, which includes the features shown in Table 1.

### 3.5 Classification with BiLSTM

Sequence classification with Recurrent Neural Networks (RNN) is an approach where the input is a sequence (i.e. words for a sentence, video frames, etc.) and the relations between the items and their order in the sequence matters. Due to their recurrent structure, RNNs are capable of keeping the knowledge from past and relating it to present.

Given a 3D point, we used a certain amount of surrounding points in order to create a sequence. While creating these sequences, we used the input data matrix (Table 2), which includes PWDC and features.

The employed neural network contains five layers: sequence input layer, BiLSTM layer with 200 hidden units, dropout layer and a dense layer.

### 3.6 Classification with 1D CNN

CNNs are widely used by the deep learning community for many purposes, therefore the structure of the layers varies accordingly. In our case, starting from an identical input like the OvO classifier, we worked with 1D CNN for feature vector input. Our network consists of an input layer, two convolutional blocks (each consists of convolution, batch normalization and activation layers), one maximum pooling layer, two dropout layers, one global average pooling layer and one dense layer, as shown in Figure 6.

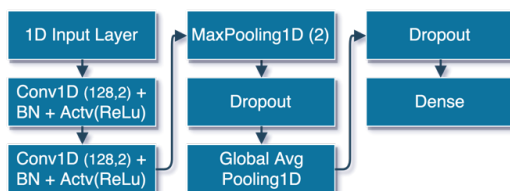


Figure 6. The employed 1D CNN architecture (number of filters and their dimensions are given next to each layer).

### 3.7 Classification with 2D CNN

2D CNNs are possibly the most widely used DNN structures, especially for computer vision purposes. In our case, we utilized 2D CNN for inputting patches. Our 2D CNN consists of twentyone layers as follows input layer, 4 2D convolutional

blocks (each consists of convolution, batch normalization and activation layers), two max pooling layers, three dropout layers, one flattening layer and two dense layers, as shown in Figure 7. The sequences with patch-wise decentralized coordinates defined in Section 3.3 are transformed into 2D matrices and used as input for 2D convolutions.

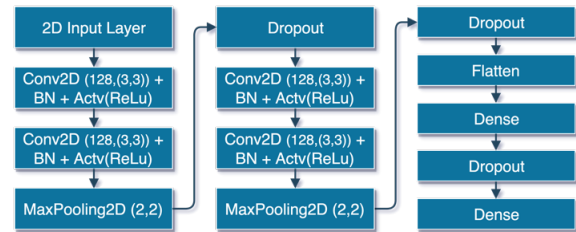


Figure 7. The employed 2D CNN architecture (number of filters and filter dimensions are given next to each layer).

## 4. RESULTS

Our classification aim is to have four classes, i.e. buildings' facades, buildings' roofs, vegetation and GLOs. Therefore, we made some adjustments (Table 3) in the Vaihingen dataset which is labelled for nine classes.

Original Class	New Class (Adjusted)
Powerline	Removed from training point cloud
Cars	
Fence/Hedge	
Roof	Roof
Façade	Facade
Low Vegetation	Ground Level Objects
Impervious surfaces	
Shrub	Vegetation
Tree	

Table 3. Adjustments in the original Vaihingen dataset in order to classify the available point cloud in only 4 classes.

In addition to these changes, there were some undesired point eliminations during feature extraction caused by point density issues. As we define the geometric features with the neighbouring points within a search radius, the points which do not have enough neighbouring points are eliminated (Table 4).

Class	Reference Data	After Elimination	Lost Data	% Data Loss
Powerline	600	98	502	84%
Low veget.	98690	93467	5223	5%
Imp. surf.	101986	97853	4133	4%
Car	3708	3235	473	13%
Fence	7422	7087	335	5%
Roof	109048	103897	5151	5%
Façade	11224	7533	3691	33%
Shrub	24818	23230	1588	6%
Tree	54226	47486	6740	12%
<b>Total</b>	<b>411722</b>	<b>383886</b>	<b>27836</b>	<b>7%</b>

Table 4. Eliminated points during feature extraction, due to low point density issues.

To our observations, this situation did not cause significant results for vegetation, GLOs and roofs classes. Yet, a significant amount of façade points is eliminated and we believe this had important effects on the results.

The training and evaluation data for Vaihingen is shown in Figure 8, along with classification results. For the accuracy assessment, we shared confusion matrices, F1 scores ( $(2 * Precision * Recall / (Precision + Recall))$ ) and balanced accuracies ( $(True Negative$



Rate + Recall/2). Results for Vaihingen set are shared in Figures 8-9 and Table 5, whereas for Dortmund set are similarly shared in Table 7, Figures 10-11.

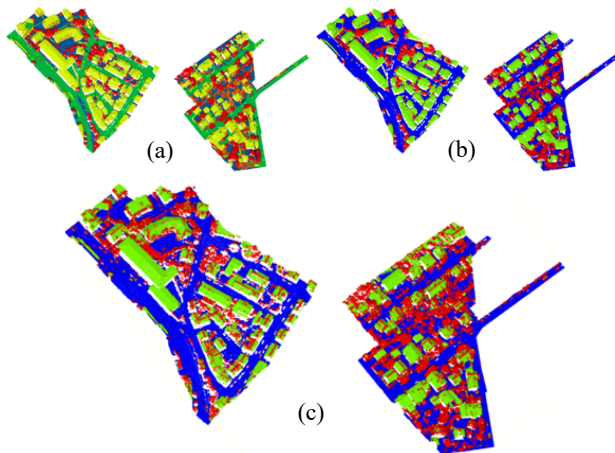


Figure 8. Evaluation data with: (a) original classes, (b) adjusted classes, (c) 2D CNN classification results for Vaihingen.

Classes	GLO	Roof	Facade	Veget.	F1 Score	Blncd. Acc.
GLO	173141	1794	35	16350	91.7%	93.5%
Roof	4113	85607	32	14145	87.7%	96.0%
Facade	1114	646	1615	4158	67.1%	92.8%
Veget.	5810	2503	107	62296	71.1%	74.0%
Others*	1925	728	96	7670		
Avg.						79.4% 89.1%

Table 5. 2D CNN classification results for Vaihingen set and confusion matrix with accuracy measures, overall accuracy 86.4% excluding others, 84.1% including others. Others\* include points from removed classes (powerline, cars and fence classes removed from the training set) which are classified as GLOs, roof, façade or vegetation.

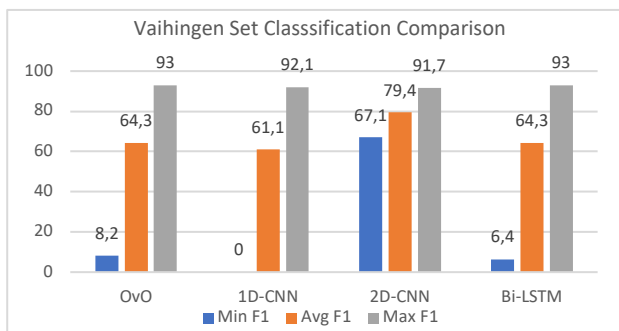


Figure 9. Minimum, average and maximum F1 scores comparisons by algorithm for Vaihingen dataset.

Abbrev.	Imp Surf.	Roof	Tree	Facade	Ovr. Acc.	Avg F1
NANJ2	91.2	93.6	82.6	42.6	85.2	70.3
WhuY4	91.4	94.3	82.8	53.1	84.9	69.2
<b>OURS</b>	<b>*91.7</b>	<b>87.7</b>	<b>*71.1</b>	<b>67.1</b>	<b>84.1</b>	<b>79.4</b>
LUH	91.1	94.2	83.1	56.3	81.6	68.39
RIT 1	91.5	94	82.5	49.3	81.6	63.33
BIJ W	90.5	92.2	78.4	53.2	81.5	60.3

Table 6. Our results among the ISPRS Benchmark results (ISPRS, 2019).

Table 6 shows our results compared to previous studies reported in the ISPRS benchmark (ISPRS, 2019). Results are sorted by overall accuracy with highest values per column shown in bold. Values with asterisk (\*) represent not the exact classes, but the matching classes (GLOs to impervious surfaces, vegetation to tree).

Classes	GLO	Roof	Facade	Veget.	F1 Score	Blncd. Acc.
GLO	31085	139	178	314	97.9%	98.3%
Roof	15	15178	1843	1634	86.0%	93.0%
Facade	491	208	10878	1049	80.3%	86.3%
Veget.	184	1086	1559	15860	84.5%	89.8%
Avg.						87.2% 91.9%

Table 7. 2D CNN classification results for Dortmund set, overall accuracy 89.4%.

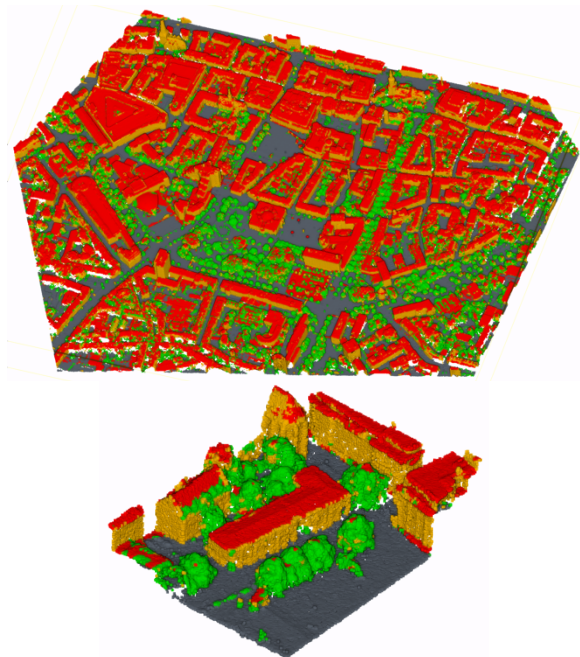


Figure 10. 2D CNN classification results for Dortmund: entire city centre (top) and evaluation part (bottom).

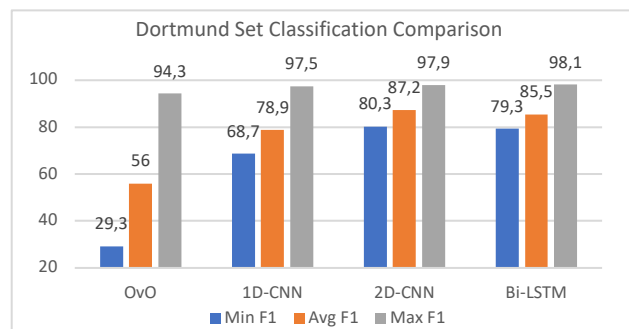


Figure 11. Minimum, average and maximum F1 scores comparisons by algorithm for Dortmund dataset.

All processes were run on a mid-performances laptop and, for each algorithm, the times spent for the classification of the Vaihingen dataset are as follows:

- OvO classifier ~1min for training and ~1min for classification,
- 1D CNN algorithm ~15min for training and ~5secs for classification,

- 2D CNN took ~45min for training and <10min for classification,
- BiLSTM DNN took ~2 hours for training and <10min for classification.

For the feature extraction, our previous approach took >1 hour for Vaihingen set, while our new approach took ~6min.

In the processes, we used Dlib C++ library (King, 2009) for the OvO classifier implementation, Keras (Chollet, 2015) with PlaidML (2019) for DL implementations and Point Cloud Library (Rusu and Cousins, 2011) for point cloud processing.

## 5. CONCLUSIONS AND FUTURE WORK

The paper aimed to examine how way various data classification algorithms react to point clouds from different sources (aerial laser scanning over Vaihingen and oblique aerial photogrammetry over Dortmund), and to experiment how DL algorithms performs providing hand-crafted features. The reported tests show that in general deep learning algorithms are capable of reaching better accuracies, even if it takes relatively longer time to train and evaluate the employed networks.

Considering the number of features and average F1 scores for OvO classifier, we noticed how the classifier features a decrease in accuracy as the complexity increases.

The classification failed when 2D CNN did not include PWDC as features, therefore we shared results only achieved with PWDC.

In our experiments, while extracting the necessary features, we used k-nearest points search for Dortmund set and radius search for Vaihingen. We observed some disadvantages of both approaches: (i) when search radius is applied, and there are very few points within the search radius that is not enough for feature computations, this causes some data loss as shown in Table 4; (ii) when k-nearest search is applied, and similarly there are very few points within the close neighbourhood, even the nearest points are far away and this ends up with noise in the features (features can be extracted, but not useful); (iii) k-nearest search is observed to be highly effected by point density variations. Therefore, as future work, it is planned a method able to extract the optimum neighbouring points and to use these optimum points for feature extraction purposes.

## ACKNOWLEDGEMENTS

The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF): <http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html>.

The authors would like to acknowledge the provision of the Downtown Toronto data set by Optech Inc., First Base Solutions Inc., GeoICT Lab at York University and ISPRS WG II/4.

## REFERENCES

Becker, C., Häni, N., Rosinskaya, E., d'Angelo, E., Strecha, C., 2017. Classification of Aerial Photogrammetric 3D Point Clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 3-10.

Cavegn, S., Haala, N., Nebiker, S., Rothermel, M., Tutzauer, P., 2014. Benchmarking High Density Image Matching for Oblique Airborne Imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3, 45-52.

Charaniya, A.P., Manduchi, R., Lodha, S.K., 2004. Supervised parametric classification of aerial LiDAR data. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30-30.

Chollet, F., et al., 2015. Keras, <https://keras.io>.

Dohan, D., Matejek, B., Funkhouser, T., 2015. Learning hierarchical semantic segmentations of LIDAR data. *Proc. IEEE International Conference on 3D Vision*, pp. 273-281.

Dorninger, P., Nothegger, C., 2007. 3D Segmentation of Unstructured Point Clouds for Building Modelling. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35, 191-196.

Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A., 2011. On the Segmentation of 3D LiDAR Point Clouds. *Proc. IEEE Robotics and Automation (ICRA)*, pp. 2798-2805.

Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep learning*. MIT press.

Griffiths, D., Boehm, J., 2019. A review on deep learning techniques for 3D sensed data classification. *Remote Sensing*, 11, 1499.

Grilli, E., Menna, F., Remondino, F., 2017. A review of point clouds segmentation and classification algorithms. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(2/W3), 339-344.

Hackel, T., Wegner, J.D., Schindler, K., 2016a. Fast Semantic Segmentation of 3D Point Clouds with Strongly Varying Density. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3, 177-184.

Hackel, T., Wegner, J.D., Schindler, K., 2016b. Contour Detection in Unstructured 3D Point Clouds. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1610-1618.

Hackel, T., Wegner, J.D., Schindler, K., 2017a. Joint classification and contour extraction of large 3D point clouds. *ISPRS Journal of Photogrammetry Remote Sensing*, 130, 231-245.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M., 2017b. Semantic3D. Net: A New Large-Scale Point Cloud Classification Benchmark. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1-W1, 91-98.

ISPRS, 2019. ISPRS Vaihingen 3D Semantic Labeling Contest Results, <http://www2.isprs.org/commissions/comm2/wg4/vaihingen-3d-semantic-labeling.html>.

King, D.E., 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10, 1755-1758.

Lafarge, F., Mallet, C., 2012. Creating Large-Scale City Models From 3D-Point Clouds: A Robust Approach with Hybrid Representation. *International Journal of Computer Vision*, 99, 69-85.

- Landrieu, L., Simonovsky, M., 2018. *Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs*.
- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature*, 521, 436-444.
- Lin, C.-H., Chen, J.-Y., Su, P.-L., Chen, C.-H., 2014. Eigenfeature analysis of weighted covariance matrices for LiDAR point cloud classification. *ISPRS Journal of Photogrammetry and Remote Sensing* 94, 70-79.
- Nex, F., Remondino, F., Gerke, M., Przybilla, H.-J., Bäumker, M., Zurhorst, A., 2015. ISPRS Benchmark for Multi-Platform Photogrammetry. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3/W4), 135-142.
- Nguyen, A., Le, B., 2013. 3D Point Cloud Segmentation: A Survey. *Proc. 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 225-230.
- Niemeyer, J., Rottensteiner, F., Soergel, U., 2012. Conditional random fields for LiDAR point cloud classification in complex urban areas. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1, 263-268.
- Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of LiDAR data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 152-165.
- Özdemir, E., Remondino, F., 2018. Segmentation of 3D Photogrammetric Point Cloud for 3D Building Modeling. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W10, 135-142.
- Özdemir, E., Remondino, F., 2019. Classification of Aerial Point Clouds with Deep Learning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13, 103-110.
- PlaidML, 2019. PlaidML, <https://github.com/plaidml/plaidml>.
- Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. Pointnet++: Deep Hierarchical Feature Learning On Point Sets In A Metric Space. *Advances in Neural Information Processing Systems*, pp. 5099-5108.
- Ramiya, A.M., Nidamanuri, R.R., Krishnan, R., 2017. Segmentation Based Building Detection Approach from LiDAR Point Cloud. *The Egyptian Journal of Remote Sensing and Space Science*, 20, 71-77.
- Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J.D., Breitkopf, U., Jung, J., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 256-271.
- Rusu, R.B., Cousins, S., 2011. 3D is here: Point cloud library (PCL). *Proc. IEEE Robotics and automation (ICRA)*, pp. 1-4.
- Serna, A., Marcotegui, B., Goulette, F., Deschaud, J.-E., 2014. Paris-rue-Madame database: a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. *Proc. 4th International Conference on Pattern Recognition, Applications and Methods (ICPRAM)*.
- Thomas, H., Goulette, F., Deschaud, J.-E., Marcotegui, B., 2018. Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods. *Proc. IEEE International Conference on 3D Vision (3DV)*, pp. 390-398.
- Weinmann, M., Jutzi, B., 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*, 5, 1-7.
- Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C., 2015a. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. *Computers & Graphics*, 49, 47-57.
- Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015b. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105, 286-304.
- Wichmann, A., Agoub, A., Kada, M., 2018. Roofn3D: Deep Learning Training Data For 3D Building Reconstruction. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(2), 1191-1198.
- Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D Shapenets: A Deep Representation for Volumetric Shapes. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912-1920.
- Yousefhusien, M., Kelbe, D.J., Ientilucci, E.J., Salvaggio, C., 2018. A multi-scale fully convolutional network for semantic labeling of 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143, 191-204.
- Zhang, J., Lin, X., Ning, X., 2013. SVM-Based Classification of Segmented Airborne LiDAR Point Clouds in Urban Areas. *Remote Sensing*, 5.
- Zhu, Q., Li, Y., Hu, H., Wu, B., 2017. Robust point cloud classification based on multi-level semantic relationships for urban scenes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 129, 86-102.