Tailored features for semantic segmentation wit a DGCNN using free training samples of a colored airborne point cloud

Widyaningrum, E.; Fajari, M.K.; Lindenbergh, R.C.; Hahn, M.

DOI
10.5194/isprs-archives-XLIII-B2-2020-339-2020

Publication date
2020

Document Version
Final published version

Published in
International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives

Citation (APA)
Widyaningrum, E., Fajari, M. K., Lindenbergh, R. C., & Hahn, M. (2020). Tailored features for semantic segmentation wit a DGCNN using free training samples of a colored airborne point cloud. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *43*(B2), 339-346. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-339-2020

TAILORED FEATURES FOR SEMANTIC SEGMENTATION WITH A DGCNN USING FREE TRAINING SAMPLES OF A COLORED AIRBORNE POINT CLOUD

E. Widyaningrum [1, 3*], M. K.Fajari [2,3], R. C. Lindenbergh [1], M. Hahn [2]

[1] Dept. of Geoscience and Remote Sensing, Delft University of Technology, The Netherlands - (e.widyaningrum, r.c.lindenbergh)@tudelft.nl

[2] Photogrammetry and Geoinformatics, Faculty of Geomatics, Computer Science and Mathematics, Hochschule für Technik Stuttgart, Germany – (82fama1mpg, Michael.Hahn)@hft-stuttgart.de

[3] Centre for Topographic Base Mapping and Toponyms, Geospatial Information Agency, Indonesia – (elyta.widyaningrum, marda.khoiria)@big.go.id

Commission II, WG II/3

KEY WORDS: Airborne point cloud, aerial photos, semantic segmentation, feature combinations, DGCNN.

ABSTRACT:

Automation of 3D LiDAR point cloud processing is expected to increase the production rate of many applications including automatic map generation. Fast development on high-end hardware has boosted the expansion of deep learning research for 3D classification and segmentation. However, deep learning requires large amount of high quality training samples. The generation of training samples for accurate classification results, especially for airborne point cloud data, is still problematic. Moreover, which customized features should be used best for segmenting airborne point cloud data is still unclear. This paper proposes semi-automatic point cloud labelling and examines the potential of combining different tailor-made features for pointwise semantic segmentation of an airborne point cloud. We implement a Dynamic Graph CNN (DGCNN) approach to classify airborne point cloud data into four land cover classes: bare-land, trees, buildings and roads. The DGCNN architecture is chosen as this network relates two approaches, PointNet and graph CNNs, to exploit the geometric relationships between points. For experiments, we train an airborne point cloud and co-aligned orthophoto of the Surabaya city area of Indonesia to DGCNN using three different tailor-made feature combinations: points with RGB (Red, Green, Blue) color, points with original LiDAR features (Intensity, Return number, Number of returns) so-called IRN, and points with two spectral colors and Intensity (Red, Green, Intensity) so-called RGI. The overall accuracy of the testing area indicates that using RGB information gives the best segmentation results of 81.05% while IRN and RGI gives accuracy values of 76.13%, and 79.81%, respectively.

1. INTRODUCTION

Up to now, automatic object classification of large-scale point cloud data is still challenging due to high variations in object shape, size, color, and texture. Airborne point clouds and aerial photos have been used as main input data for various 3D mapping activities, as both provide high-resolution earth surface data. LiDAR point clouds and aerial photos have different characteristics and capabilities. The combination of 3D point clouds and aerial photos is believed to increase the degree of automation as well as object detection accuracy. Spectral information from photos provides essential features for classification, while highly detailed 3D information provided by LiDAR point clouds will increase the results accuracy.

PointNet, proposed by Qi et al., (2016), pioneered pointwise deep learning approaches for point cloud classification and segmentation. This high computationally effective and efficient network still suffers from a lack of capability to make use of local information on the point sets [Jiang and Ma, 2019]. For segmentation, the local context is crucial for labelling the categories semantically. Qi et al. (2017) next presented PointNet++ to improve the basic model by adding a hierarchical neural network to capture local geometric features. Acknowledging that encoding geometric relations between a single point to its nearby points is still a problem in PointNet++, Wang et al. (2018) proposed Dynamic Graph CNN (DGCNN) by incorporating a graph-based CNN approach to capture the local geometry of points by an edge convolution operation on a *k*-nn graph which is iteratively updated by the nearest neighbors. We used such a DGCNN, as this network architecture achieves state-of-the-art performance on semantic analysis.

Previous work on semantic segmentation of airborne 3D point cloud data includes Soilan et al. (2019) that classified the Actueel Hoogtebestand Nederland (AHN) airborne point cloud data into three land cover classes: ground, vegetation, and buildings in a PointNet architecture. However, though the classification accuracy result achieved 87.77%, there is high confusion between vegetation and building classes. Wicaksono et al. (2019) used a similar architecture to this study, DGCNN, to classify building and non-building points using two different feature combinations: with color and without color features. Based on their results, they stated that color features do not improve results but even affects the semantic segmentation results. In contrast, using so-called sparse manifold CNN, Schmohl and Soergel obtained a 0.8% higher overall accuracy when using additional spectral information on their test set segmentation. Xiu et al., (2019) also implemented PointNet using Intensity (depth) and

* Corresponding author

spectral (RGB) features obtained by fusing a LIDAR point cloud with an orthophoto. By applying this data fusion, overall accuracy increased by 2%, from 86% to 88%. Poliyapram et al. (2019) examined something similar by comparing the use of intensity features with the combination of intensity and color features (RGB) on PointNet on point cloud data of Osaka City, Japan. The results show that simply applying data fusion at the observation level can improve overall accuracy from 65% to 79%. Even with the characteristics of the city of Osaka, which is a complex urban area with dense buildings, the use of a combination of Intensity and RGB features can provide improved performance in identifying buildings by 4%.

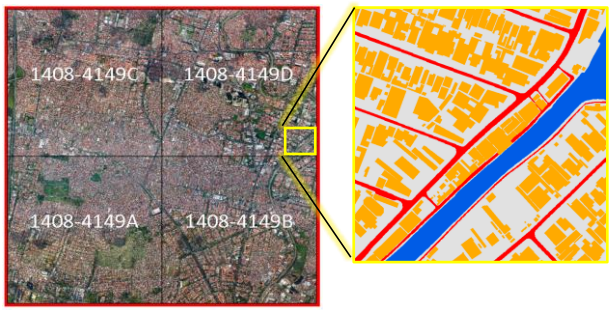
Based on the various results of the aforementioned works when using different feature combinations, it is essential to provide more research on the use of various tailored features for semantic segmentation of airborne point clouds. Furthermore, labelling 3D training samples using 2D data is worth to examine. Thus, this paper addresses two contributions as follows:

1. to provide a method for creating high-quality free training samples for 3D semantic segmentation of new airborne point cloud data using existing 2D GIS base maps for 4 land cover classes: bare-land, buildings, trees, and roads;
2. to exploit different combinations of discriminative features from off-the-shelf features of airborne LiDAR point clouds and airborne photos for 3D pointwise semantic segmentation.

2. DATASET AND STUDY AREA

A set of airborne LiDAR point cloud data and aerial photos acquired at the same time are used in this study. The airborne LiDAR point cloud, obtained by an Optech Orion H300 instrument, has an average point density of about 30 points/m², while the aerial photos captured by a tandem camera have a ground sampling distance of 8 cm. Both LiDAR point cloud and aerial photos are acquired at the same time in 2016. The LiDAR point cloud used for our research has a total number of 354.197.545 points.

Two base map versions are used in this study. First, the 1:5.000 base map extracted by manual delineation from WorldView2satellite images acquired in 2012 and the 1:1.000 base map Year 2017 generated by manual stereo-plotting from the same aerial photos used in this study.



(**a**) Airborne orthophoto of the study area covering the four indicated 1:5000 map sheets.

(**b**) The 1:1000 base map Buildings in orange, the road in red, water in blue, and bare-land in grey.

Figure 1. The study area in the city of Surabaya, Indonesia.

The urban landscape selected for the study area is located in Surabaya city, Java Island, Indonesia. This city has a typical metropolitan character with dense and complex high-rise buildings. The study area covers 21.5 km². According to Indonesian 1:5000 base map index, the study area consists of four map sheets: 1408-4149A, 1408-4149B, 1408-4149C, and 1408-4149D, as shown in Figure 1.

3. METHODOLOGY

This study exploits multi-class labelling used for training in a semi-automatic way and evaluates different tailor-made features trained by the DGCNN architecture for segmenting a colored airborne LiDAR point cloud using the dataset described in Section 2. Our methodological framework, as shown in Figure 2, consists of three main steps: (1) semi-automated point cloud labelling; (2) data training by DGCNN; and (3) evaluation. In the first step, a colored airborne point cloud is obtained by projecting color information of orthophotos to the LiDAR point cloud. Next, semi-automated point cloud labelling for obtaining training samples is conducted by using available base map vector data combined with roughness filtering. After training by the DGCNN, results are evaluated and discussed.

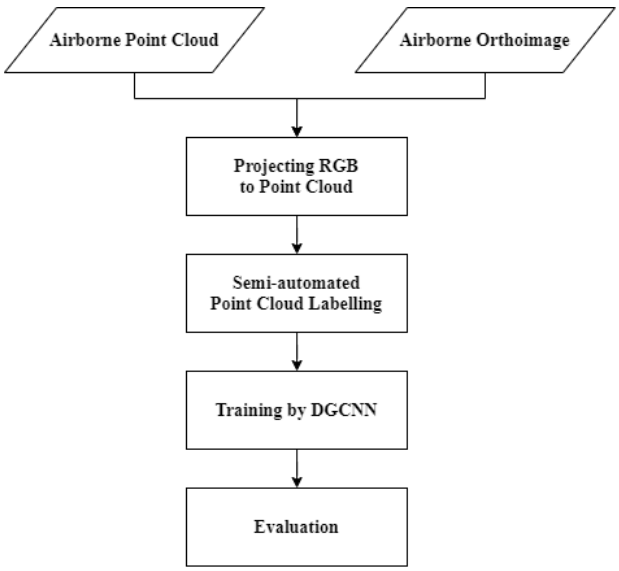


Figure 2. The methodological framework.

3.1 Labelling Free Training Samples

Labelling the point cloud to provide sufficiently accurate training samples for semantic segmentation is a non-trivial task. Variations in object characteristics and different landscapes may introduce confusion and noise in the training data. Moreover, using training samples from a different domain may require additional or completely new training samples. An existing base map is a useful information source to label the training data. However, there are several challenges to extract free training samples using a 2D base map. First, the base map data used to label the point cloud does not provide trees and bare-land information. Second, as we use base map polygons to label the points, the labelled building and road points may include many mislabelled points as there are trees covering buildings or roads. We propose a simple hierarchical approach to label tree and bare-land classes as well as to improve the quality of training samples by filtering likely mislabelled points.

For semantic segmentation, the study area is divided into 8 parts by dividing each of four map sheets into two parts. We use 7 out of 8 parts in our study area for training and the remaining part (lower part of 1408-4149D map sheets) for testing.

The labelling procedure is as follows:

- After projecting spectral information (RGB color) to the point cloud data, the points are labelled into four classes (bare-land, tree, building, and road). The airborne point cloud used in this study has been automatically classified into the ground and non-ground points by TerraScan software.
- From the non-ground points, building points are labelled using building polygons of the base map. Using the same method, road points are labelled from the ground points, and remaining points are labelled as bare-land.
- However, using base map polygons to label buildings points may introduce mislabelling in case of trees exist near to the buildings. Therefore, we apply tree filtering based on point cloud roughness. In this case, any point that is either labelled as building or road or unclassified is labelled as a tree when its roughness is above a threshold. The surface roughness is estimated for each point based on the distance between the corresponding point to the best fitting plane estimated using all neighbouring points inside an area of 2m × 2m. The roughness threshold is set empirically to 0.5.
- In the final step, the point cloud dataset is systematically downsampled to 1m × 1m, and remaining outliers are removed using a statistical outlier removal algorithm (with standard deviation = 2 times the mean distance of 30 points).

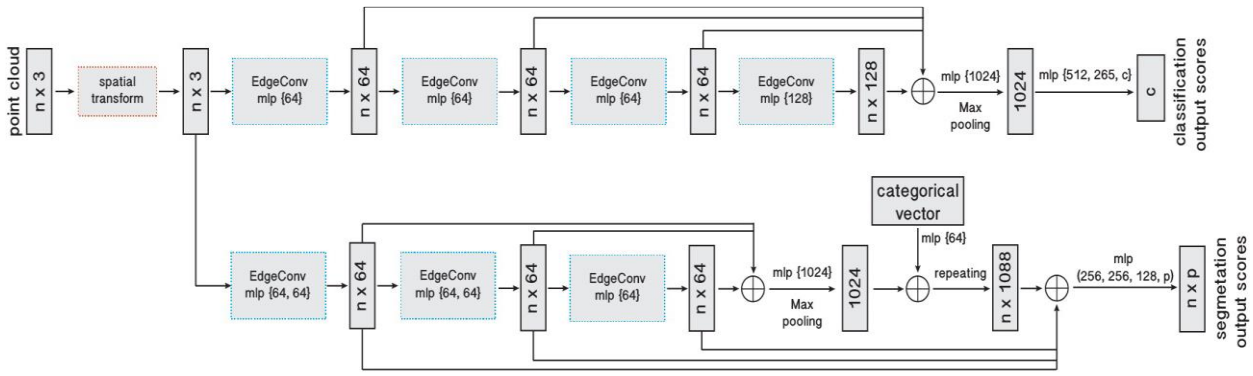
To prepare the training data, we then split each area into 30m × 30m blocks with a stride or overlap of 15m to ensure local geometry is captured efficiently by the network.

3.2 Pointwise Semantic Segmentation of Different Tailored Features

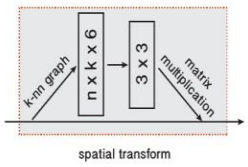
We use a DGCNN architecture [Wang et al., 2018] to perform airborne point cloud semantic segmentation. Using a similar network architecture as basic PointNet architecture, the DGCNN incorporated the so-called EdgeConv to construct a local neighbourhood graph describing the relationship of a point to the neighbouring points. The EdgeConv operation aggregates local geometric features from the neighboring points by k-nearest neighbour and applies convolution-line operations on the edges.

As seen in Figure 3, the DGCNN uses a spatial transformation component to compute a global shape transformation, followed by EdgeConv, which acts as MLP (Multi Layer Perceptron), to learn local geometric features for each point. For semantic segmentation, three sequential EdgeConv steps followed by three fully connected layers are used to make a prediction score for each segmented point. A max pooling or downsampling operation is then performed as a symmetric edge function to make the model permutation invariant and to capture global feature.

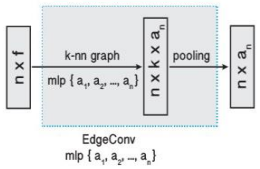
Suppose a F-dimensional point cloud is given with n points, denoted by $X = \{x_1, \ldots, x_n\} \subseteq R^F$. Consider a feature dimensionality of $F = 9$, where each point consists of 3D coordinates and 6 other features (e.g. RGB color, surface normals, Intensity, etc). Consider a directed graph $G = (V, E)$ representing local point cloud connectedness, where $V = \{$



(**a**) Network architecture

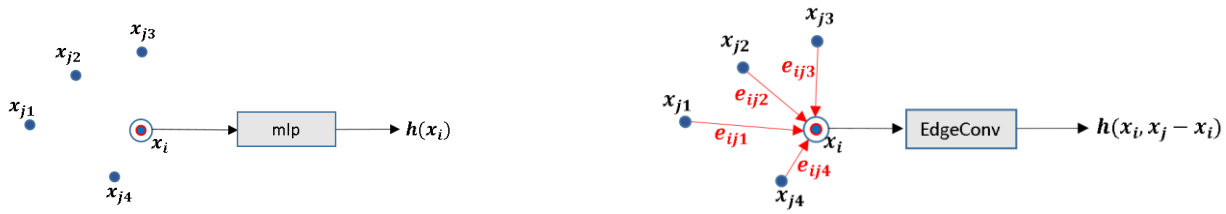


(**b**) Spatial transformer module



(**c**) EdgeConv

Figure 3. The DGCNN architecture (Wang et al., 2018). (**a**) The top branch shows the classification model architecture while the bottom branch shows a semantic segmentation model architecture; (**b**) A spatial transformation module is used to learn specific transformations from the input point cloud, i.e. to estimate a 3 × 3 transformation matrix by concatenating the coordinates of each point and the coordinate differences between its k-nn neighbors, and then apply point convolutions; (**c**) EdgeConv takes an output tensor of shape $(n \times f)$, and then applies multi layer perceptron (MLP) operations with a number of neurons defined as $\{a_1, a_2, \ldots, a_n\}$ to compute edge features which finally results in a tensor shape of $(n \times a_n)$ after pooling the neighboring edge features.



(**a**) PointNet (**b**) EdgeConv operation results in edge features

Figure 4. Basic differences between PointNet and DGCNN. (**a**) The PointNet output of the feature extraction $\boldsymbol{h(x_i)}$, is only related to the point itself, (**b**) The DGCNN incorporates local geometric relationship $\boldsymbol{h(x_i, x_j - x_i)}$ of a point $\boldsymbol{x_i}$ to its neighbourhood $\boldsymbol{x_{ji}}$ to $\boldsymbol{x_{j4}}$.

$1, \ldots, n\}$ and $E \subseteq V \times V$ denotes the vertices and edges, respectively. For each point, DGCNN uses the k-nearest neighbour graph to construct graph G containing directed edges of the $(i, j_1), \ldots, (i, j_k)$ such that points $x_{j1}, \ldots, x_{jk}$ are closest to x_i in G (see Figure 4). The k-nearest neighbours of a point dynamically change from layer to layer of the network and are computed sequentially. The edge function is then defined as $e_{ij} = h\Theta(x_i, x_j)$, where $h\Theta: R^F \times R^F \rightarrow R^{F'}$ is a non-linear function that contain learnable parameters Θ, and $\Theta = (\theta_i, \ldots, \theta_k)$ are the weights of the filter to be optimized in each edge convolutional layer.

DGCNN adopts an asymmetric edge function $h\Theta(x_i, x_j) = h\Theta(x_i, x_j - x_i)$ across all layers to combine both the global shape structure (by capturing the coordinates of the patch center x_i) and the local neighbourhood information (by capturing $x_j - x_i$). Similar to PointNet and PointNet++, the aggregation operation to downsample the input representation in DGCNN is max pooling.

3.3 Feature Combinations

As this paper aims to investigate different feature combinations to classify earth objects in an airborne point cloud, we provide three different sets of feature vector for training, consisting of:

1. RGB points: each point is represented by true color (Red, Green, and Blue) from an orthophoto $(X, Y, Z, R, G, B, nx, ny, nz)$.
2. IRN points: each point is represented by original LiDAR features consisting of Intensity, return number, and number of returns $(X, Y, Z, I, R, N, nx, ny, nz)$.
3. RGI points: each point is represented by Red and Green color and Intensity $(X, Y, Z, R, G, I, nx, ny, nz)$.

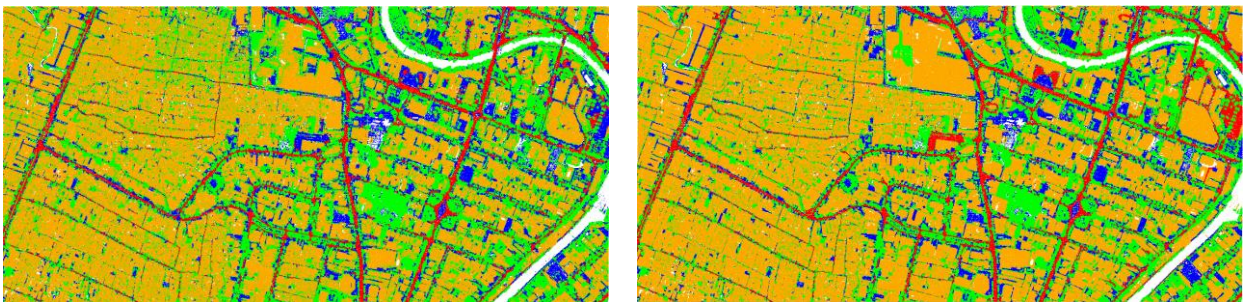
All feature combinations are complemented by normalized 3D coordinates (nx, ny, nz) in which the point cloud original coordinates are transformed to local coordinates by subtracting from centroid XYZ values to boost the translational invariance of the algorithm [Qi et al., 2018].

During training, 4096 points are uniformly sampled from each block to form data batches with a consistent number of points, while all points are used during testing time. As we used 9 features for training, therefore, the input data size to the network is 4096×9. We use $k = 20$ nearest neighbours for each point to construct the edge graphs. For all experiments, the final model is obtained after running 50 epochs, optimized by a so-called Adam optimizer with an initial learning rate of 0.001, a momentum of 0.9, and a mini batch size of 24.

The 3D point cloud semantic segmentation using DGCNN is performed on a High Performance Computing (HPC) environment of Delft University of Technology consisting of 26 computing nodes. During training, two Tesla P100 GPUs available in the cluster were used.

3.4 Evaluation

To evaluate the performance of different training models from different feature combinations, we determine the confusion matrix and overall accuracies. Along with overall accuracy, there three other criteria metrics are estimated: the recall (also known as completeness), the precision (also known as correctness), and F1-score. The recall refers to the percentage of the total points correctly predicted by the model, while the precision refers to the percentage of the results that area relevant. The F1-score is a harmonic mean of precision and recall. In addition, the overall accuracy indicates the percentage of all correctly classified points of all classes from the total number reference points is also estimated.



(**a**) Classification result using RGB features (**b**) Ground truth

Figure 5. Semantic segmentation result over the test set area in comparison to the ground truth. Blue points represent bare-land, orange represents building, green represents vegetation, and red represents road.

4. RESULTS AND DISCUSSIONS

This study aims to provide a cheap but effective method to label a considerable amount of training samples and analyse optimal features using point-wise deep learning approach. Figure 5 shows the classification result over our test area.

The test data comprises of 46.477.312 points and these points are excluded for training. Overall accuracy evaluated on the test area for RGB points, IRN points, and RGI points are 81.05%, 76.13%, and 79.81%, respectively (Table 1). Table 2 to Table 4 show the confusion matrix of the results using different feature combinations.

	F1-score (%)				Overall Accuracy (%)
	bare-land	tree	building	road	
RGB	74.2	79.8	84.2	73.9	81.05
IRN	74.7	75.4	77.4	70.4	76.13
RGI	74.5	78.8	82.4	74.6	79.81

Table 1. Comparison of the overall accuracy and F1-score per-classes of different feature combinations.

RGB		**ground truth**				**comp./ recall**
		bare-land	**tree**	**building**	**road**	
prediction	**bare-land**	4,305,343	151,379	1,437,965	665,654	65.63%
	tree	77,020	11,798,573	3,855,133	3,434	74.99%
	building	169,987	1,868,355	19,798,584	6,220	90.64%
	road	498,846	4,246	69,909	1,766,664	75.51%
corr./ precision		85.23%	85.36%	78.69%	72.35%	**81.05%**

Table 2. Confusion matrix of segmentation result using RGB composition.

IRN		**ground truth**				**comp./ recall**
		bare-land	**tree**	**building**	**road**	
prediction	**bare-land**	4,303,394	73,926	1,149,484	948,231	66.46%
	tree	108,541	12,711,073	7,075,664	4,619	63.88%
	building	414,761	1,039,166	16,902,283	23,763	91.96%
	road	223,507	1,134	31,702	1,466,064	85.12%
corr./ precision		85.21%	91.94%	67.18%	60.02%	**76.13%**

Table 3. Confusion matrix of segmentation result using IRN composition.

RGI		**ground truth**				**comp./ recall**
		bare-land	**tree**	**building**	**road**	
prediction	**bare-land**	4,383,401	148,951	1,519,290	661,463	65.30%
	tree	79,409	12,160,087	4,791,228	3,331	71.39%
	building	129,737	1,509,060	18,777,942	5,719	91.95%
	road	457,338	4,981	73,448	1,771,927	76.78%
corr./ precision		86.80%	87.97%	74.63%	72.55%	**79.81%**

Table 4. Confusion matrix of segmentation result using RGI composition.

Based on the confusion matrix, the detected building and road points have higher recall rate than the precision. In contrast, the bare-land and tree points have higher precision rate than the recall rate. In our case, it is likely that using the base map to label the training samples induces a higher recall rate. A lower recall but higher precision means that the model is accurate enough to detect the object (in this case, bare-land and tree) but misses a significant number of the corresponding object points.

A more detailed analysis based on the results of different feature combinations in connection to the training samples labelling procedure discusses as follows:

a. Prediction of building points

We discover that the number of building points detected as the tree is at least three times more than the number of tree points detected as building. This portion is the biggest contribution to the precision rate of the building class. We assume that the use the 2D polygons to label the buildings may still include mislabelled points. For example, points on the building façade are labelled as building points but in the result, many façade points are predicted as trees (see Figure 6.c). A significant number of building points falsely classified as road is likely caused by many ground points within building surrounding that are labelled as building. In this case, the training model correctly predicts the ground points that were mislabelled in the ground truth. However, the worst building precision rate is achieved when no spectral information is used, in this case, when using IRN features.

b. Prediction of road points

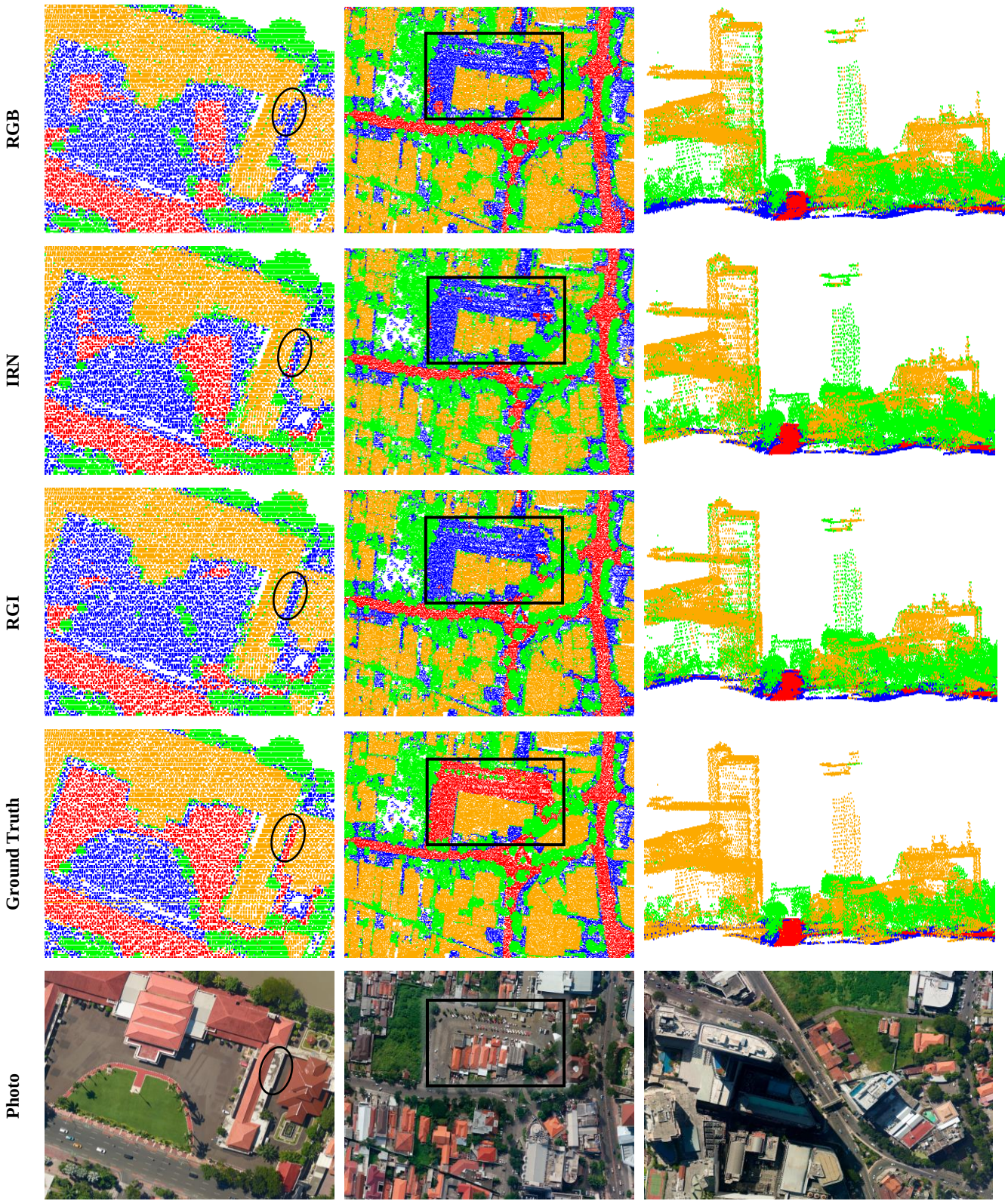
Using spectral information for road points prediction returns a higher precision but lower recall. On the other hand, using LiDAR features gives a higher recall but lower precision result. This means, using only off-the-shelf LiDAR features increases the over-classification chance. Such road and bare-land confusion exist particularly when open yard with asphalt surface is labelled as road, for example, points on parking lot or front yard offices with asphalt surface are labelled as the road (see Figure 6.a and 6.b). However, the model has a low rate to detect such points on asphalt yard as the road. IRN features the worst precision rate for road.

c. Prediction of bare-land points

For all feature combinations, a lower recall rate is mostly caused because many road and building points are detected as bare-land points. We observe there were many ground points remain in the non-ground points after filtering process, which later induce mislabelling. The recall and precision rate of the prediction results from all combinations are similar. We assume that other features than XYZ coordinates may have small weights for predicting the ground points.

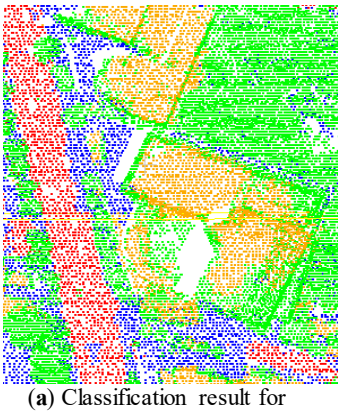
d. Prediction of tree points

The lower recall rate of the tree class is because many building points are classified as tree. The reason why the tree class has lower recall is the same as why the building points have lower precision. It is because points on building facades labelled as buildings in ground truth, are predicted as tree (see Figure 6.c). However, among other classes of all feature combinations, the tree class always has the highest precision rate. This means that the errors in predicting tree points as other class are very low, in other words, the model can predict the tree class accurately. The highest precision rate of the tree class is achieved by the IRN feature combination. The use of return number and number of returns appears to increase the chance to predict tree points correctly.

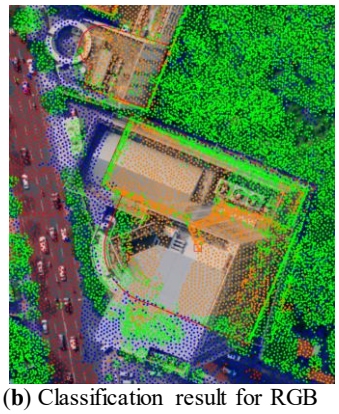


(**a**) Subset 1 (**b**) Subset 2 (**c**) Subset 3

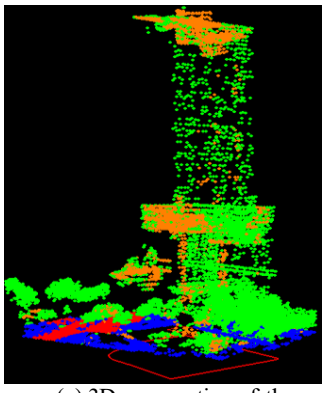
Figure 6. Comparison of DGCNN point cloud semantic segmentation results using different feature combinations (RGB, IRN, RGI) to the reference for three different subset areas. Black ellipses and rectangles highlight the differences in result between three feature combinations and the ground truth. Blue points represent bare-land, orange represents building, green represents vegetation, and red represents road.



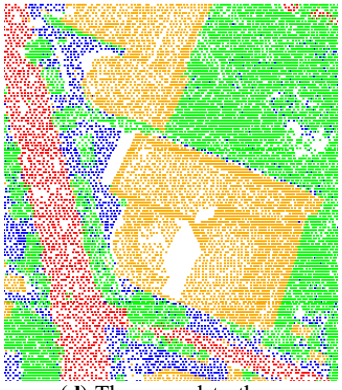
(**a**) Classification result for RGB composition



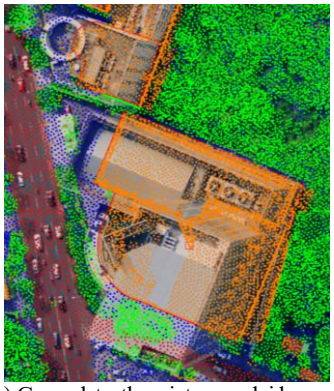
(**b**) Classification result for RGB overlaid on orthophoto



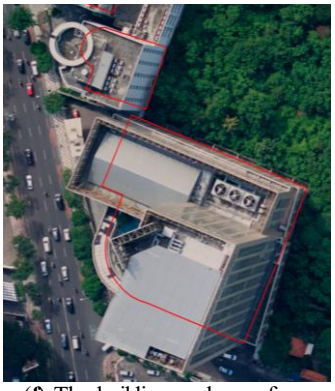
(**c**) 3D perspective of the classification results overlaid on 2D base map polygon



(**d**) The ground truth



(**e**) Ground truth points overlaid on orthophoto



(**f**) The building polygon from base map (red outline) overlaid on orthophoto

Figure 7. Results in an area where relief displacement occurs. The classification method still correctly predicts ground and tree points that may have incorrect spectral color.

Figure 6 shows a comparison between three different subsets of three different classification setups and the ground truth. The rectangles and ellipses highlight areas where differences occur. The black ellipses indicate the differences over the building roof area. The RGB features failed to detect several building points of a particular area inside the ellipse but not for IRN and RGI features. It is likely that the black building roof color causes false predictions when using RGB features. Other differences are indicated by the black rectangles in the second column of Figure 6. There is confusion to classify an asphalt surface covering a big front yard from all feature combinations. Although the asphalt front yard has a similar color as the road and is defined as the road in the ground truth, but based on its shape, it can be considered as non-road.

As we use a ground orthophoto to color the points cloud, it is possible that points are having incorrect colors (or color misalignment) due to building relief displacement. This problem usually happens in the surrounding of high-rise building areas, where some particular areas were blocked by the leaning building roof (see Figure 7). However, we found out that the DGCNN network is still able to detect bare-land points correctly even though these points were assigned with an incorrect color.

5. CONCLUSION AND RECOMMENDATION

This study exploits the use of a 2D base map to label the training samples and use different feature combinations to perform semantic segmentation using existing pointwise deep learning architecture, a DGCNN, from a colored airborne LiDAR point cloud. Three different feature combinations (RGB, IRN, and RGI) are used to classify the point cloud into four classes: bare-land, tree, building, and road points.

The best overall accuracy, 81.05%, is achieved when using a point cloud attributed with spectral information to the DGCNN network Based on our results, using a 2D base map to label the training samples is indeed a cheap and effective approach. However, the accuracy rate may not indicate the correct value as our ground truth labelled from 2D base map still include mislabels. In our study, we discover that using spectral information (RGB) is increasing overall accuracy, in particularly for building and tree points. Surprisingly, spectral information is likely not affecting bare-land prediction as it has a similar recall and precision rates for all feature combinations. The combination of two spectral bands (Red and Green) with Intensity has the highest detection rate for the road class. The use of original LiDAR features (Intensity, return number, and number of

returns) gives an impressive accuracy result in detecting tree points. As the study area is located in a big city, where many high-rise buildings exist, points on the façade heavily affect the detection result. However, incorrect color due to relief displacement is not affecting the result.

When labelling training samples using a base map one should consider additional steps to refine the labels and reduce mislabels especially in case trees exist in the surrounding of buildings and roads. Improving the proposed labelling method and combining a complete set of original LiDAR point clouds and aerial photos features for training are believed to increase the detection accuracy. Domain and time shift using the trained model is also interesting to be investigated further. In addition, we also address the need of improved neural network architecture for more robust 3D point cloud semantic segmentation for future research.

ACKNOWLEDGEMENT

The authors acknowledge Surabaya municipal government (Pemerintah Kota Surabaya) of Republic Indonesia and Geospatial Information Agency (Badan Informasi Geospasial) of Republic Indonesia for providing the datasets for this research. The authors gratefully acknowledge support from the Indonesia Endowment Fund for Education (LPDP), Ministry of Finance of Republic of Indonesia, for scholarship support to the first author.

REFERENCES

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652-660.

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

Qi, C. R., Liu, W., Wu, C., Su, H., & Guibas, L. J. 2018. Frustum pointnets for 3D object detection from RGB-D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918-927.

Soilán Rodríguez, M., Lindenbergh, R., Riveiro Rodríguez, B., & Sánchez Rodríguez, A. 2019. Pointnet for the automatic classification of aerial point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W5, ISPRS Geospatial Week, Enschede, the Netherlands.

Schmohl, S., & Sörgel, U. 2019. Submanifold sparse convolutional networks for semantic segmentation of large-scale ALS point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W5, ISPRS Geospatial Week, Enschede, The Netherlands.

Wicaksono, S. B., Wibisono, A., Jatmiko, W., Gamal, A., & Wisesa, H. A. 2019. Semantic Segmentation on LiDAR Point Cloud in Urban Area using Deep Learning. In *IEEE 2019 International Workshop on Big Data and Information Security (IWBIS)*, pp. 63-66.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. 2019. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5), pp.1-12.

X. Jiang, and X. Ma. 2019. Dynamic graph CNN with attention module for 3D hand pose estimation. In Proceedings: *16th International Symposium on Neural Networks*, ISSN 2019, 10-12 July, Moscow, Russia.

Xiu, H., Poliyapram, V., Kim, K. S., Nakamura, R., & Yan, W. 2018. 3D semantic segmentation for high-resolution aerial survey derived point clouds using deep learning. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 588-591.

Poliyapram, V., Wang, W., & Nakamura, R. 2019. A point-wise lidar and image multimodal fusion network (PMNet) for aerial point cloud 3D semantic segmentation. *Remote Sensing*, 11(24), 2961.