

RESEARCH

Open Access



Automatic LIDAR building segmentation based on DGCNN and euclidean clustering

Ahmad Gamal², Ari Wibisono^{1*} , Satrio Bagus Wicaksono¹, Muhammad Alvin Abyan¹, Nur Hamid¹, Hanif Arif Wisesa¹, Wisnu Jatmiko¹ and Ronny Ardhianto³

*Correspondence:

ari.w@cs.ui.ac.id

¹ Faculty of Computer

Science, Universitas

Indonesia, Kampus UI Depok,

Indonesia

Full list of author information

is available at the end of the article

Abstract

There has been growing demand for 3D modeling from earth observations, especially for purposes of urban and regional planning and management. The results of 3D observations has slowly become the primary source of data in terms of policy determination and infrastructure planning. In this research, we presented an automatic building segmentation method that directly uses LIDAR data. Previous works have utilized the CNN method to automatically segment buildings. However, the existing body of works have relied heavily on the conversion of LIDAR data into Digital Terrain Model (DTM), Digital Surface Model (DSM), or Digital Elevation Model (DEM) formats. Those formats required conversion of LIDAR data into raster images, which poses challenges to the evaluation of building volumes. In this paper, we collected LIDAR data with unmanned aerial vehicle and directly segmented buildings utilizing the said LIDAR data. We utilized a Dynamic Graph Convolutional Neural Network (DGCNN) algorithm to separate buildings and vegetation. We then utilized Euclidean Clustering to segment each building. We found that the combination of these methods are superior to prior works in the field, with accuracy up to 95.57% and an Intersection Over Union (IOU) score of 0.85.

Keywords: LiDAR, DGCNN, Building segmentation, Euclidean clustering

Introduction

At this time, the demand for 3D modeling using earth observations is high. The results of 3D observations become the primary data in terms of policy determination. Planning infrastructure is something that can be supported by 3D data observations. Traffic policies, utilities, permits, and urban planning are some of the problems that can be solved with 3D data observations.

3D city modeling has three stages: building detection, building extraction, and building reconstruction. The purpose of the first stage is to detect that a collection of points is a building. The second stage is to compute the polygons that fit the set of building points. The third stage determines the best plane for the roof. Some researchers have tried to conduct building extraction by using image processing from high-resolution images [1–3].

S.M. Abdullah conducted building detection using a segmentation method [4]. The technique divides LIDAR data into the ground and non-ground points. After that, they iterated from the highest LIDAR point and searched for the planar shape of the roof segment. The region growing method was used for roof identification. The correctness and completeness of this study reached 70–90%.

The Digital Terrain Model (DTM) and Digital Surface Model (DSM) techniques have been used to extract data from point cloud data. The DSM and DTM techniques convert 3D point cloud data to a raster image. This was done by filtering using the Sohn filter generated by the DTM. Then, the DSM technique is applied to get the roof pixels. The classification obtained from this method reached 95.1% accuracy, 98.1% correctness, and 89.5% efficiency. Z. Hao used the DSM and DTM techniques to produce a Digital Elevation Model (DEM). To perform building extraction, he used the Gray Level Co-occurrence Matrix (GLCM), which is an unsupervised clustering technique. The GLCM technique is used to separate candidates from vegetation and trees [5].

LIDAR (Light Detection and Ranging) data are the data obtained from a laser sensor, combined with several sensors, and they include laser and GPS data. LIDAR can be placed on the bottom of the aircraft and pointed to the ground. It will generate points that represent the surface below the aircraft. The topology includes buildings, plants, and vegetation. In the problem regarding the detection of urban buildings, some researchers have taken several approaches [6–11]. In LIDAR data or point cloud data, the data formed are only a collection of 3-dimensional points with additional RGB data. There is no information about the surface of an object from those points.

Some alternatives to obtain information from LIDAR generated data have been studied by researchers. The DSM is a standard method to get visual information about the surface from raw LIDAR data. In the DSM, the height feature becomes the primary component that can be used to differentiate each point. The results of the differentiation of each point are represented in the form of a raster image. Zhao et al. employed the Gray Level Co-occurrence matrix [5] to the images resulting from the DSM process.

In addition to using the image processing method, the use of deep learning methods to conduct building extractions has also been studied by several researchers. Hsiuhan Lexie Yang et al. applied deep learning to remote sensing images to conduct building extraction [12]. Several experiments have been carried out, including the Fully Convolutional Network (FCN) method and SegNet (Semantic Segmentation). Experimental results show that the value of the F-Score is between 0.62 and 0.73.

Detecting buildings using deep learning was also studied by Faten et al. [13]. They merged LIDAR data with Orthophoto data. Some features extracted from LIDAR data include the density, the boxy fit, the shape index, the DEM, and the DSM. The best accuracy result by this method is 86.19%. The use of the DSM to identify changes in an area by using aerial images and LIDAR data has also been studied. The resulting proposed method achieves 93% completeness and 90.2%.

Building segmentation methods have been divided into methods that rely on the extracted images from the DTM, the DSM, and the DEM [14–20]. In this research, we promote a building segmentation method that directly uses LIDAR data, which is in the form of point clouds. Many researchers have used the CNN method for image

processing and segmentation [21–33]. When processing LIDAR data, several researchers convert LIDAR data to the DTM, DSM, or DEM formats, which are then processed using a CNN [12] [13]. This causes an important feature to be missing, the z feature. In this paper, we strive to ensure that we provide a process that directly uses LIDAR data. Our research position is described in Fig. 1.

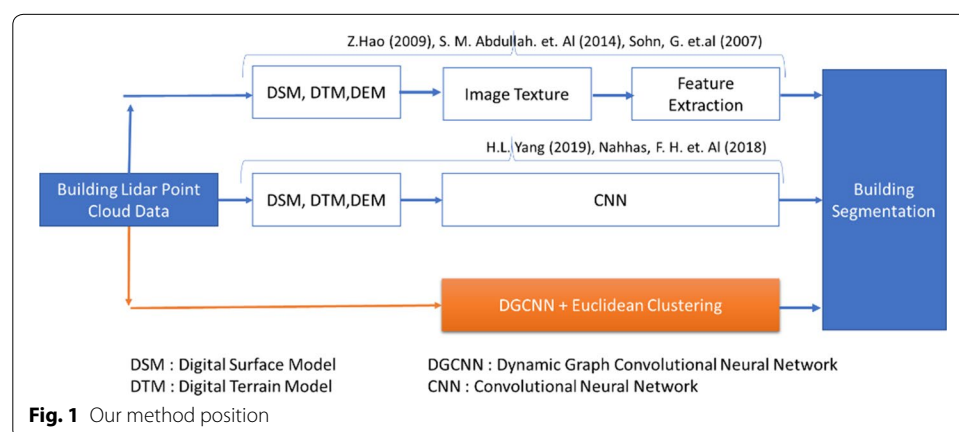
Anandkumar Ramaya applied the Euclidean clustering technique to open datasets to detect buildings [36]. The best accuracy results that were obtained range from 82% to 89% when the number of buildings is $25 + 25 = 50$. In this research, we used four datasets with a total number of 229 buildings. We captured the dataset ourselves using drones and LIDAR equipment. We conducted the building extraction process using the DGCNN, and the building segmentation is done using the Euclidean clustering method. Our results achieve accuracies from 74.28% to 95.57%.

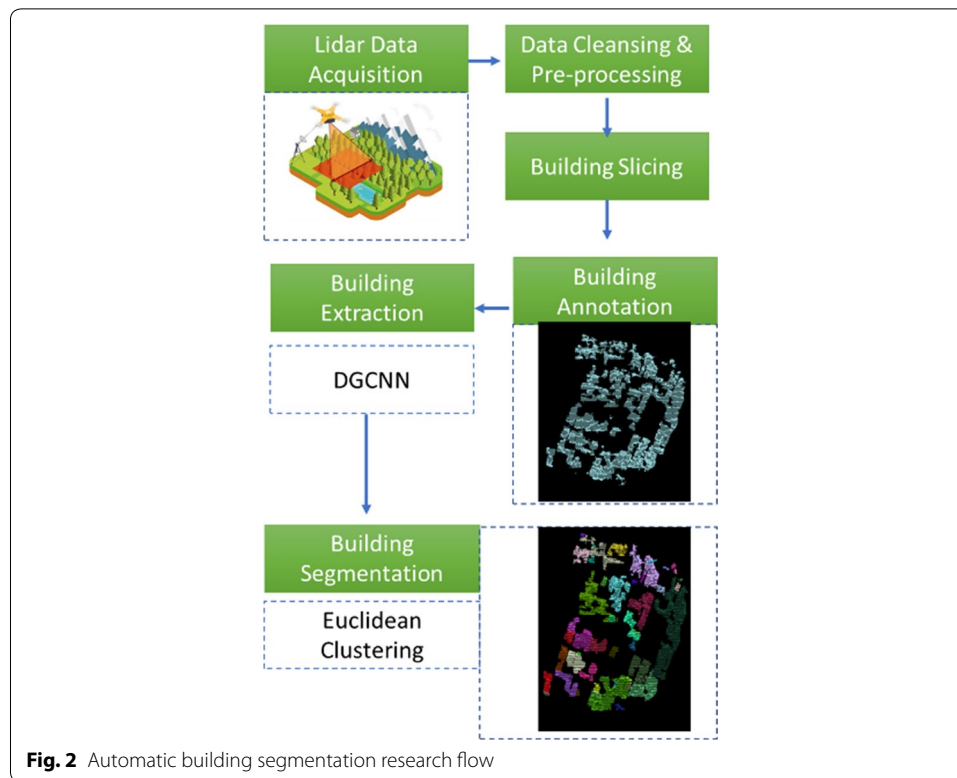
Several researchers fused LIDAR Data and High-Resolution images to extract building from the data. They utilized deep learning for training the neural network. Huang, Jianfeng et al. used the gate residual refinement network to extract high-resolution aerial images and LIDAR data [37]. They converted the data to a pixel-based format to fit the data into the residual refinement network. Pan, Xuan, et al. utilized a fine segmentation network to do semantic labeling of aerial images [38]. It converted the LIDAR data to the DSM form then transformed into the neural network structure.

Chen Shanxiong et al. employed Adaptive Iterative Segmentation to extract the building [39]. They utilized LIDAR Data, which is in 3D- form to Digital Surface Model (2D-form). Based on these several references. We propose a new method that utilizes directly raw LIDAR data, which is 3D-data into the DGCNN network to extract building then utilize PCL to segment each building. Instead of losing some detail in the Z dimension, we try to preserve the x,y,z raw data. So the DGCNN network process 3-D data rather than 2-D data.

Proposed method

In this research, we perform building segmentation using LIDAR data. First, we perform data collection using LIDAR, it utilizes a drone to get the LIDAR point data. The drone flies approximately 200 meters above the ground. After the data collection has been completed in the field, the data will be cleaned and preprocessed. The next process





is preparing the training data in a building or house in a point cloud format. Each home or building will become training data for the DGCNN. The data will be used to train the DGCNN for 100 epochs. Figure 2 gives the overall description of our proposed method. Also, we used a public dataset to evaluate our metric from the city of Dublin [40]. We used this data in our experiment because it gives the same object with our dataset. It provides building objects, land, and vegetation in the city area. Besides, the data format is the same as our datasets, and it has the same properties and features.

After we extract the points of the building by using the DGCNN, then we utilize Euclidean Clustering to segment each of the buildings. Rusu developed Euclidean clustering based on point cloud data by implementing the clustering method using kd-tree data structures [35]. The Euclidean clustering method works by segmenting objects that are on the same plane. For this reason, it is necessary to define what distinguishes one cluster from another cluster [35].

We also evaluated a pixel-based method to compare it with the DGCNN + Euclidean Cluster. LIDAR data is first converted into DSM form, so that lidar data changes to the 2-D format. To perform building extraction, we utilize the binarization of the pixel-based DSM image. The segmentation for each building was done using the Haralick and Shapiro method [41]. After the segmentation is done, we can get the label results for each building. The evaluation of this method is described in Tables 4 and 5.

We utilize the DGCNN algorithm to extract the building points from our dataset. It differentiates between the point collections of building and nonbuilding points, such as vegetation or land. The Dynamic Graph Convolutional Neural Network (DGCNN) is a neural network architecture inspired by PointNet that can perform classification and

segmentation directly on data in the form of point clouds. In PointNet, each point is processed locally to maintain the invariant permutation by ignoring the geometric relations between points [34]. The DGCNN architecture uses a convolution layer called EdgeConv, which has edge features where the relations between points, including geometric structures, are considered [34].

EdgeConv exploits the local geometric structures by constructing graphs at adjacent points and applying convolution operations on each connected edge [34]. The DGCNN classification process uses the architecture shown in Fig. 3. Input data are in the form of N inputs, and M features, where features can be in the form of point clouds with point x , y , z coordinate data, or they can have other features added such as a color representation in the form of Red, Green, and Blue (RGB) data. For each point, the edge feature on the EdgeConv layer will be calculated, and each feature will be aggregated to calculate the EdgeConv calculation result for each point [34].

The Edge Convolution Operation (EdgeConv) applies asymmetric aggregation operation in the operator to determine the edge features that correspond to all edges of the reference point to each surrounding point. EdgeConv has the property that it is able to

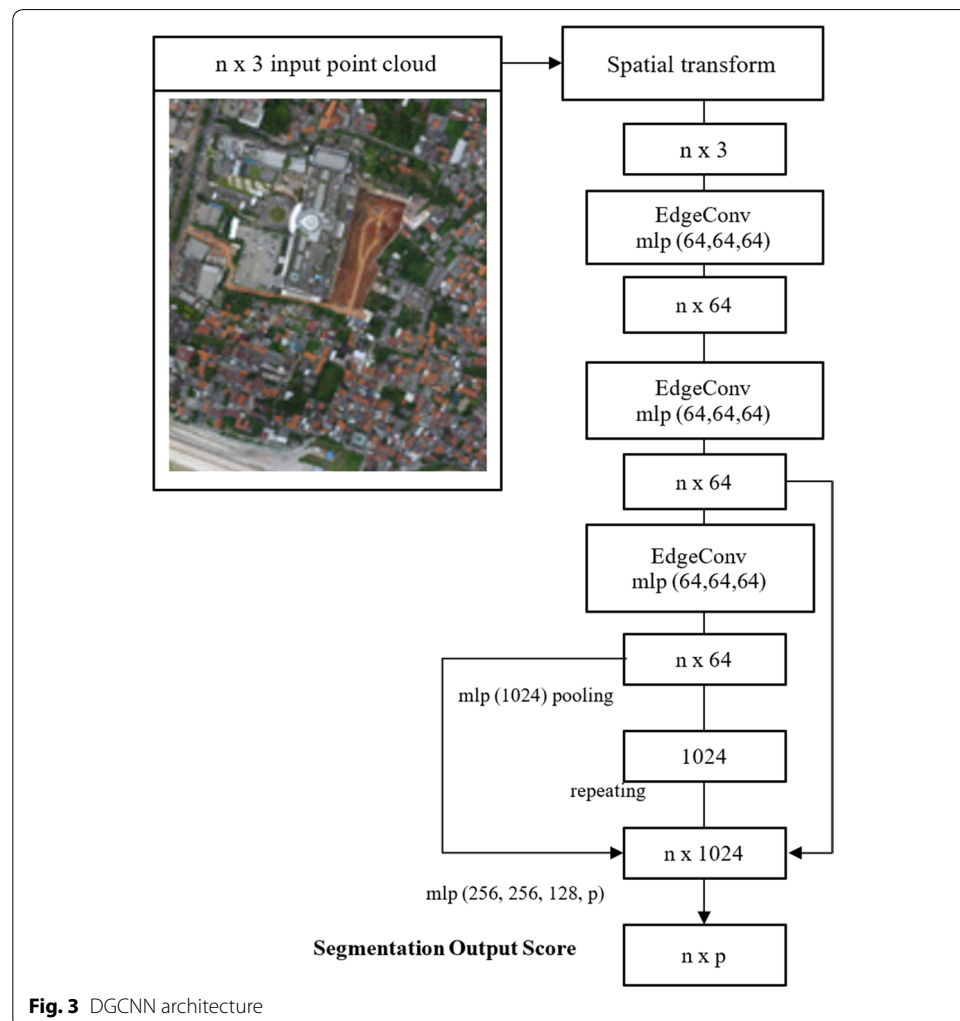


Fig. 3 DGCNN architecture

recognize an object in an area, even if the object has been moved, rotated, or scaled, (enlarged or reduced) and then determine its nonlocal feature [34]. There are several options for determining the edge and operator functions used in a model. The PointNet model applies $h\Theta(x_i, x_j) = h\Theta(x_i)$. The DGCNN adds edge detection using asymmetric edge functions in the form of $h\Theta(x_i, x_j) = h\Theta(x_i, x_j - x_i)$. This function was chosen to be implemented in the DGCNN because of its ability to combine the global structure form (formed by t reference points of x_i) and the surrounding information (obtained from the closest point relations x_j, x_{ij}). EdgeConv takes the tensor input of the input $n \times f$. The 3D edge features are obtained by applying a multilayer perceptron (MLP). The output of the EdgeConv block is formed after the pooling process, and it is a tensor with the size of $n \times a_n$.

Unlike static CNN graphs, the DGNN graph is dynamically updated for each network layer. The DGCNN finds a set of k -nearest neighbors from the point of the layer changes in the network, which is calculated based on the embedding sequence. The distance in the feature space is different from the input distance, thus leading to the diffusion of nonlocal information across the point cloud data. Experiments using the DGCNN model provide the advantage of recalculating the graph using the nearest neighbors in the feature space generated from each layer. This is what distinguishes the DGCNN from CNN graphs that work with input fixes. This algorithm is called the DGCNN because the graph is dynamically processed with updates. With the updated graph, the receptive area will be spaced as wide as the point cloud distance so that it becomes scattered.

The formulation of our method starting, if there is an input in the form of point cloud P and O is the representation of a cluster, then the equation can be defined as follows:

$$\min p_i - p_{j2} \geq d_{th} \quad (1)$$

where clusters $O_i = \{p_i \in P\}$ and $O_j = \{p_j \in P\}$ have different points between them and d_{th} is the maximum distance between the points in different clusters. The purpose of the above Eq. 1 is that if the difference between the points $\{p_i \in P\}$ with $\{p_j \in P\}$ is greater than d_{th} , then p_i is the part of cluster O_i and p_j is the part of cluster O_j [35]. The processing algorithm of Euclidean Clustering is described in Fig. 4.

After all data points of a building or house are available, we will annotate each point cloud datum as building, vegetation, or land. Furthermore, the DGCNN training process is carried out using the following parameters: 150 epochs, and 4.096 points are sampled for each

1. Initialize kd-tree for point cloud P 's input data
2. Initialize the list of empty clusters C and queue Q that contains the points that need to be calculated
3. For each $p_i \in P$:
 - a. Add p_i to Q
 - b. For each $p_i \in Q$:
 - i. Find collection of P_i^k points that neighbor p_i with the distance $r \leq d_{th}$
 - ii. For each $p_i^k \in P_i^k$, check that the point has been processed if it has not been added to Q
 - c. If all points in Q have been processed, add all points in Q to C , then clear Q
4. The algorithm stops when all points, $p_i \in P$ has been processed and is part of the cluster C list

Fig. 4 Euclidean clustering pseudocode

building. This step is carried out to divide separate buildings, land, and vegetation. We also evaluate the accuracy of this division. After the DGCNN process is performed, now we get only the point cloud data of the buildings. Then, we perform semantic segmentation using Euclidean Clustering. The results we get from Euclidean Clustering are labeled as buildings with different colors for the visualization. The visualization of the segmentation is shown in Fig. 7.

The explanation of Eq. 1 provides the process flow through Eq. 9.

$$k = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1)$$

$$K = [k_0, k_1, k_2, \dots, k_n] \quad (2)$$

$$B = [b_0, b_1, b_2, \dots, b_n] \quad (3)$$

$$L = [l_0, l_1, l_2, \dots, l_n] \quad (4)$$

$$DGCNN(B, L) = \text{EdgeModelBL} \quad (5)$$

If K is an array of 3D vectors of point cloud data, k consists of the horizontal (x), vertical (y), and depth (z) of a variable. Variable B is a collection of variables k that make up a building. Variable L is a collection of points from k that are land. Notation B is a collection of points for each building, and L is a collection of points for each nonbuilding. They are used to train the DGCNN, which produces a neural network model with the name EdgeModelBL. EdgeModelBL is a model that is used to determine the points classified as a building or nonbuilding.

$$n = \begin{bmatrix} k \\ e \end{bmatrix} \quad (6)$$

$$N = [n_0, n_1, n_2, \dots, n_n] \quad (7)$$

$$\text{EdgeModelBL}(K) = N. \quad (8)$$

If there are multiple points k and we want to predict the points classified as building or nonbuilding, we can do that by using $\text{EdgeModelBL}(K)$ in Eq. 8. The result of the $\text{EdgeModelBL}(K)$ model is that it generates a set of points that are automatically labeled as building or nonbuilding with the notation e in Eq. 6. The set of points labeled as building is NB and the set labeled as nonbuilding is NL .

$$N = [NB, NL]. \quad (9)$$

$$EC(NB) = P \quad (10)$$

$$p = \begin{bmatrix} b \\ r \end{bmatrix} \quad (11)$$

$$P = [p_0, p_1, p_2, \dots, p_n] \quad (12)$$

N consists of the dots labeled buildings and the dots marked as nonbuilding. To segment each building, we just need a dot labeled as a building with the *NB* notation. Therefore, *NB* point sets will be segmented individually by Euclidean Clustering or *EC* in Eq. 10. Euclidean clustering will generate *b* points as buildings and *r* as the labels of those buildings, e.g., building 1, building 2, and building *n*. The set of building points *b* and the labels *r* of each building are included in *P*.

Results and discussions

The measurement results are calculated in the building extraction and building segmentation process. The process of removing objects other than buildings is carried out in the building extraction process, and the process of segmenting each building is carried out in the building segmentation process. We perform data acquisition using drones that are equipped with LIDAR sensing devices. The tool will produce a collection of 3-dimensional point clouds. A point cloud contains the data that we use as input data for the DGCNN algorithm. The data are gathered by flying a drone at the height of 200 m above the ground. The drone will move according to the specified flight plan. In addition, the LIDAR laser will shoot the laser light down to the ground and produce points that describe the condition of the surface under the drone. At this time, we perform surveillance in 4 different locations for our own dataset. Dataset 1–4 is our own dataset, which acquiesced in the Depok area in Indonesia. The surveillance is illustrated in Fig. 5. Dataset 5–8 are public datasets from the city of Dublin.

Building extraction

In the building extraction process, we utilized the DGCNN method. There are four datasets that we have gathered. The DGCNN configuration that we use is as follows: 50 epochs, a batch size of 12, a learning rate of 0.001, the Adam optimizer, and a momentum



Fig. 5 LIDAR data acquisition drone

of 0.9. The metrics we use in measuring the performance of the building extraction process are Accuracy, Precision, Recall, F-Score, and Intersection Over Union (IOU). For each dataset, we use the x , y , and z features as our training features in the DGCNN.

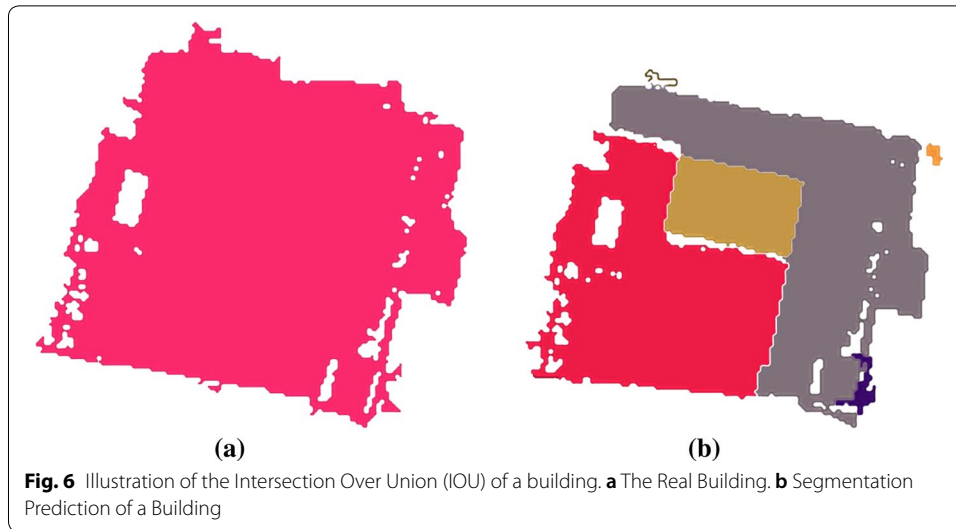
The accuracy, precision, and recall metrics are used to evaluate how good the DGCNN model is at predicting a collection of points that are categorized as buildings or nonbuildings. Meanwhile, the Intersection Over Union metric seeks to assess the performance regarding the intersection of the points of the predicted data and ground truth data.

Table 1 shows the performance results of the building extraction process using the DGCNN. It can be seen that the separation between building points and nonbuilding points is highly accurate for datasets 1 and 2. Datasets 1 and 2 are city areas. Thus, the height of the building is higher than the surrounding vegetation. Therefore, this makes the accuracy, precision, and recall results for datasets 1 and 2 better than those of the other datasets. The model can obtain good results that differentiate high buildings and the ground. In Table 1, the accuracies for datasets 3 and 4 are above 84%. However, the IOU shows that in these datasets, buildings and nonbuildings is not separated perfectly. It can be seen that the IOUs are 0.68 and 0.63, respectively. Datasets 3 and 4 consists of various objects, high buildings, small houses, vegetation, trees, and bushes. Therefore, it is difficult for the models to perfectly separate buildings and nonbuildings. We also measure the IOU score since the IOU will give a clear picture of the quality of the classification results. It can be seen in dataset four that although the accuracy obtained is relatively high, the intersection of the classification results generated with the ground truth data is 0.63. This result gives us information that only 60% of the points of the ground truth object and the predicted object intersect. A detailed illustration of the IOU can be seen in Fig. 6. The real building is illustrated in Fig. 6(a), and the segmentation prediction is illustrated in Fig. 6(b).

Dataset 5–8 is the open LIDAR dataset from Dublin city. Based on Table 1, we can see that the Dublin LIDAR dataset (dataset 5,6,7,8) generally has almost the same building extraction performance compared to the Depok dataset (dataset 1–4). Table 2 is the result of the evaluation metric using the DSM (pixel-based) method. From Table 2, we can see that the Accuracy, Precision, Recall, F1-Score, and IOU consistently have smaller values than the DGCNN + Euclidean clustering method in Table 1. Image processing or pixel-based on LIDAR data causes the removal feature z

Table 1 Building extraction performance results (DGCNN)

Dataset	Accuracy	Precision	Recall	F-Score	IOU
Dataset 1	89.00%	0.90	0.86	0.87	0.78
Dataset 2	91.50%	0.90	0.89	0.89	0.81
Dataset 3	85.20%	0.79	0.81	0.80	0.68
Dataset 4	84.30%	0.79	0.74	0.76	0.63
Dataset 5	83.4%	0.83	0.77	0.79	0.67
Dataset 6	78.4%	0.77	0.82	0.77	0.63
Dataset 7	85.8%	0.82	0.83	0.83	0.71
Dataset 8	78.2%	0.76	0.77	0.76	0.62

**Table 2** Building extraction performance results (DSM)

Dataset	Accuracy	Precision	Recall	F-Score	IOU
Dataset 1	54.01%	0.81	0.54	0.64	0.55
Dataset 2	60.50%	0.87	0.61	0.71	0.63
Dataset 3	52.03%	0.78	0.52	0.62	0.51
Dataset 4	51.06%	0.76	0.51	0.61	0.48
Dataset 5	48.00%	0.63	0.45	0.52	0.47
Dataset 6	44.00%	0.58	0.38	0.46	0.39
Dataset 7	51.03%	0.67	0.48	0.55	0.51
Dataset 8	41.00%	0.56	0.32	0.40	0.32

in the LIDAR data. In DSM, z features should be removed because DSM only uses the image-based principal. In the DSM method, the z is converted into color density. The process of the building extraction is not optimal with the converted z feature. Thus the evaluation metric has a reasonably large error compared to the DGCNN + Euclidean Clustering method.

Building segmentation

In the building segmentation process, the separation between building objects is done by utilizing Euclidean clustering [35]. Euclidean clustering is utilized because this method has been developed for point cloud data specification. The Euclidean clustering input is the output data from the building extraction process. The point cloud data only consist of building points. Semantic segmentation is done using a model that has been trained using the data in each dataset. The parameters of the Euclidean Clustering of point cloud data are given in Table 3.

We use the parameters in Table 3 to perform the building segmentation process based on the output from the building extraction process. Our LIDAR data do not give a well-ordered city mapping. They consist of various types of objects that are included in our

Table 3 Euclidean clustering parameters

Search method	Kd-tree
Minimal cluster size	1000
Maximal cluster size	2000,000
Tolerance	1.1 meters

dataset. The resolution of our LIDAR sensing data is 1 m for a 40 point cloud. The distance between one building and another can be less than 1 m. Some buildings are enormous, such as offices and malls.

Additionally, some buildings are tiny, such as security posts and small shops (3×3 meters). Therefore, we have to choose a large range of cluster sizes. Each of the clusters represents the points of a building.

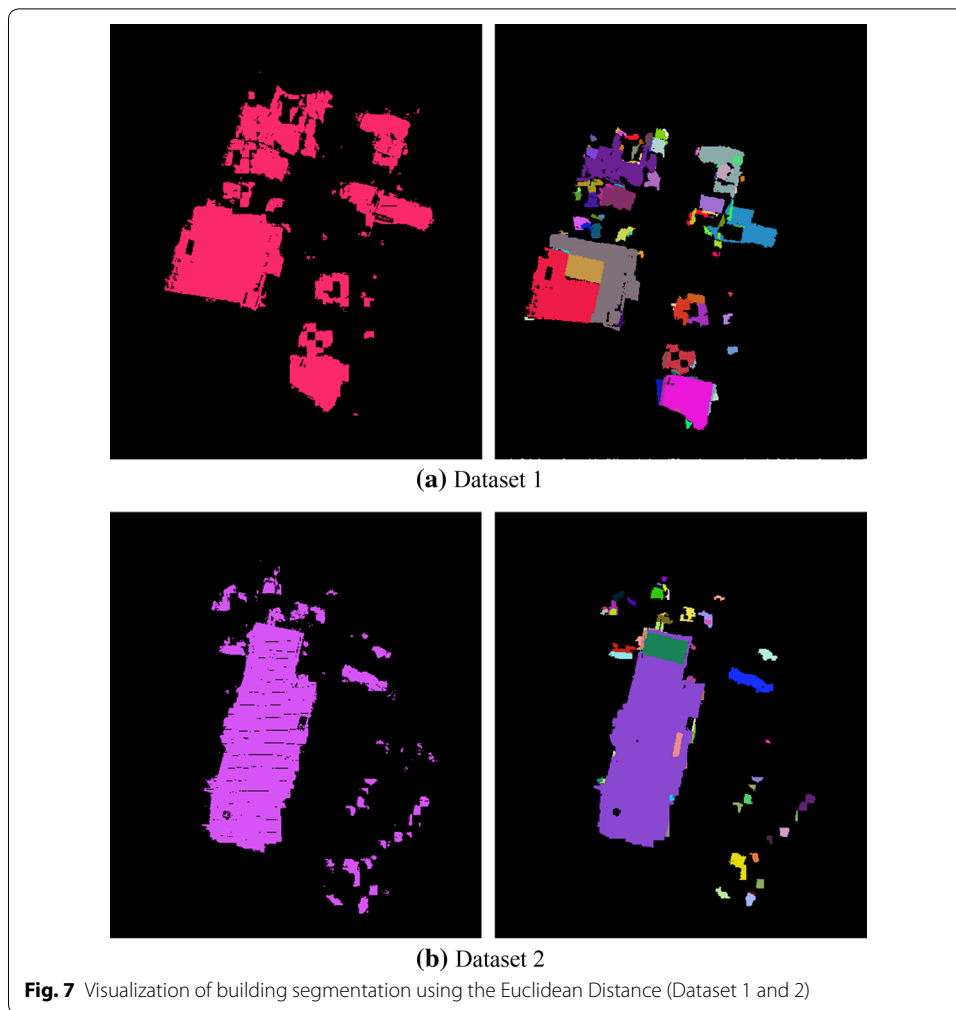
The search method is a data structure used to find the nearest neighbors at each point; the data structure that is used is the kd-tree. The minimum cluster size is the lowest number of points in a cluster or building. The maximum cluster size is the maximum number of points in a cluster. Tolerance is a tolerance variable for the distance between the points in a cluster. If the tolerance value is too small, a label that should be labeled as one building can be labeled as several buildings. Conversely, if the tolerance value is too large, some buildings can be labeled as one building only. The visualization results of the building segmentation utilizing Euclidean Clustering are described in Figs. 7 and 8 for the Depok dataset, and Figs. 9 and 10 for Dublin Dataset.

The metrics that we used to evaluate the segmentation results for each building are the Accuracy and IOU Score. In this evaluation, we want to test the success of the separation between one building and another. The accuracy metric aims to measure whether the results of the cluster are in the form of a building. Meanwhile, the Intersection Over Union metric is used to test the results of the cluster and assess if it has a large enough intersection with the original building.

Illustrations and examples of IOU evaluations can be seen in Fig. 6. In Fig. 6, we can see a building that is identified as a unit using pink. Meanwhile, the results of the Euclidean Clustering produced three buildings that are colored pink, brown, and gray. In this case, the IOU value obtained for the building is 0.45 because the points are separated into three different buildings. Only 45% of the Euclidean clustering result has an intersection with the real points of the building. However, the accuracy is near 100% because all of the clusters are in the form of a building.

Table 4 explains the Euclidean clustering results for datasets 1–8. The building segmentation is visualized in Figs. 7, 8, 9 and 10. Datasets 1–3 have 32, 35, and 61 buildings, respectively. The display of the buildings in datasets 1–3 can be seen clearly in Fig. 6a–c, respectively. Visually, we can see that the gap or distance between one building and another building is quite tight in these datasets. The Euclidean clustering algorithm cannot maximize the separation between one building and another building. therefore, the accuracies and IOUs for datasets 1–3 are from 74% to 81% and 0.65–0.69, respectively.

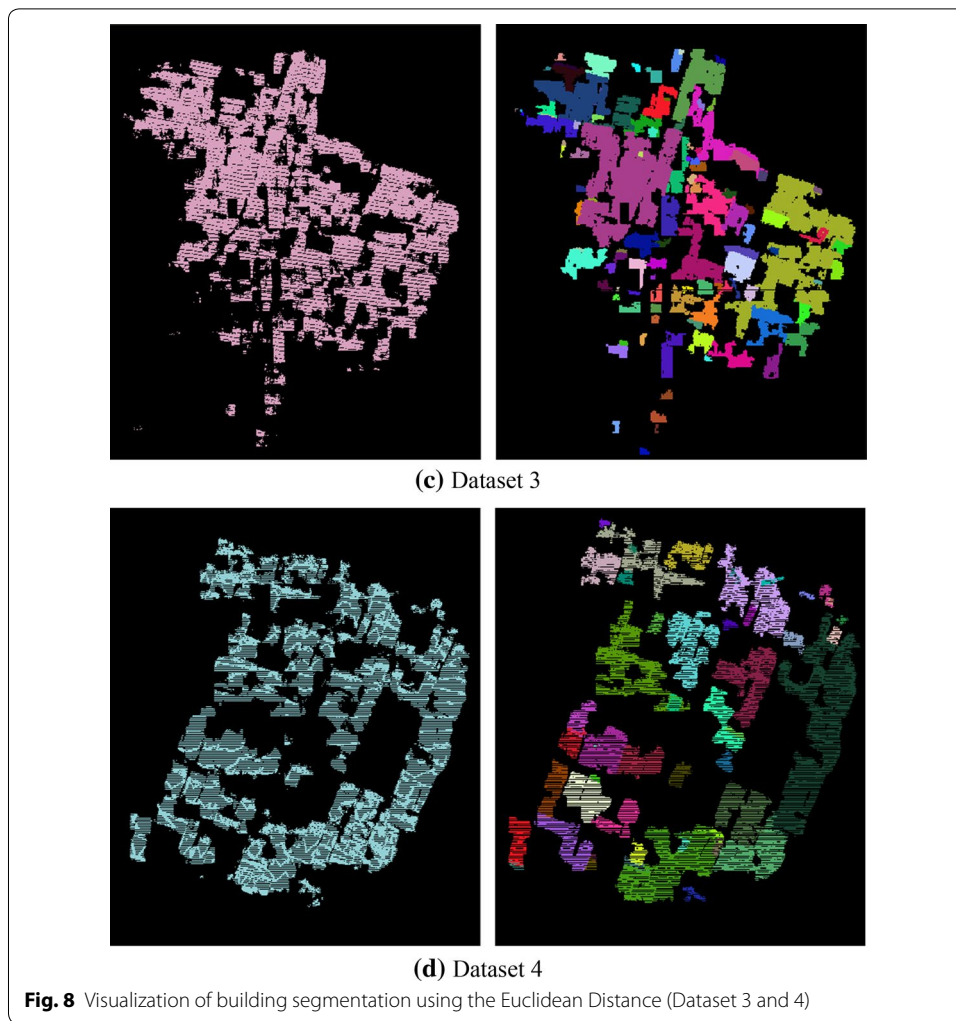
We can see in datasets 1–3 that the accuracy obtained is between 74.28% and 82.00%. 6–9 buildings cannot be segmented as buildings. This happens because in dataset 1-3, several buildings have several points that are too small, and so they are not segmented as



buildings. The IOU results also show values from 0.65 to 0.72. These results indicate that several buildings have been segmented and can be separated from each other. However, the results of the separation are not perfect. As explained in Fig. 5, some buildings are not completely segmented. There are segmentation results that include only a few parts of the building.

As we can see in Table 4, the dataset achieves an accuracy of 91.67%. From the results of this accuracy, it can be explained that the number of buildings that can be segmented as buildings is quite good. However, good results were obtained because there was a large gap between one building and another, as shown by Fig. 7d. Considerable differentiation between buildings gives Euclidean Clustering the ability to maximally separate the building points of one building from those of another. If we measure the IOU in dataset 4, the obtained results are between 0.84. The IOU results show relatively agreement between the segmented buildings and the ground truth buildings. Therefore, the intersection between the segmented buildings and the ground truth buildings reflects a good result.

Dataset 5–8 is the Dublin area dataset. In the Dublin dataset, we can see that the result of the metric evaluation value is better than the dataset 1–4, which is the Depok area



dataset. There are several factors that cause this to happen. The city layout of the Depok area is not well designed, so there is no clustering between office buildings, parks, and housing. Meanwhile, in a relatively developed city such as Dublin, the urban planning and arrangement are excellent, so that the placement of areas and Clustering of building designations have been appropriately implemented. In addition to city planning, the resolution of the LIDAR data also affects the results of the evaluation. In the Dublin dataset, the LIDAR dataset resolution is 300 points per meter. In comparison, in the Depok dataset, the resolution possessed by the dataset is 45 points per meter. So that the building segmentation on data 5–8 has a better performance compared to the dataset 1–4.

As a comparison, we also conducted a segmentation evaluation using DSM (pixel-based image). Pixel-based image evaluation is carried out by utilizing the Haralick method for segmenting each building. Table 5 shows the results of the evaluation using the DSM. From Table 5, in general, the evaluation metrics show a performance result that is not better than Table 4 for all datasets. This is due to the conversion of the z dimension. It happened when converting LIDAR data into DSM format.

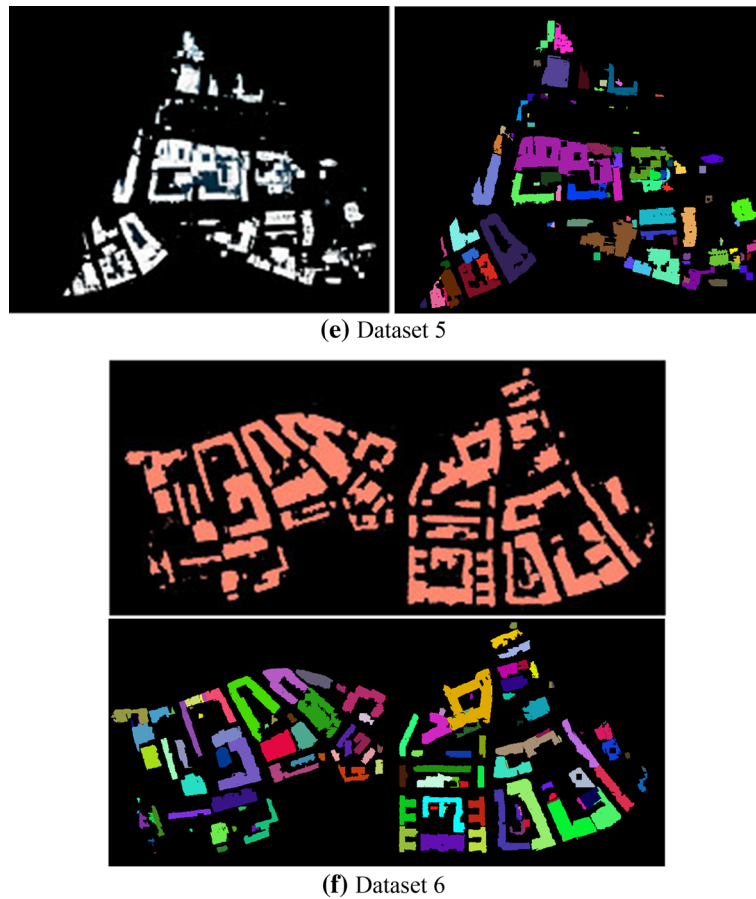


Fig. 9 Visualization of building segmentation using the Euclidean distance (Dataset 5 and 6)

Based on several references, which uses the DSM format + deep learning such as [37–39]. These researchers used a combination of LIDAR data and High-Resolution aerial images. The fusion between lidar data that is converted into DSM with a high-resolution image is one way to segment buildings. Deep learning and its variations are perfect if implemented for image processing needs. High-resolution images can help the performance of deep learning to classify results. In our research, we use LIDAR data with minimal resolution and do not use high-resolution images. Hence, the computation we do uses only three main features, namely x , y , z , in the point cloud data. We do not convert from 3-Dimensions to 2-Dimensions. All data remains in 3 Dimensional formats. This can make the process shorter than converting to DSM.

We compare the process between the DSM conversion + CNN method and using the raw Lidar data. The usual method for building extraction is done by converting point cloud data to the DSM image (2-Dimensions). It converted 3-dimensional data into 2-dimensional data. The use of high-resolution images also requires considerable computing power in the processing. It utilized deep learning and employed hidden layers when training the data in several epochs.

Our method does not convert to DSM (2-Dimensions). The original features of the Point cloud are retained and fed directly into the neural network for results. So that



(g) Dataset 7



(h) Dataset 8

Fig. 10 Visualization of building segmentation using the Euclidean distance (Dataset 7 and 8)

in terms of process, the method we do has a process that is shorter than the usual method. In terms of environmental factors, the method we use does not depend on photos or aerial images. As we know that aerial images are of low quality when dealing with bad weather. The method we use only considers the point cloud of the lidar data from the laser sensor on the lidar, so that problems in bad weather can be resolved.

From the evaluation results of the eight datasets, we can conclude that the merging of the building extraction process using the DGCNN and the building segmentation

Table 4 DGCNN + Euclidean clustering results

No	Dataset	Number of buildings	Buildings detected	Buildings undetected	Accuracy	IOU
1	Dataset 1	32	26	6	81.25%	0.72
2	Dataset 2	35	26	9	74.28%	0.65
3	Dataset 3	61	52	9	82.00%	0.69
4	Dataset 4	36	33	3	91.67%	0.84
5	Dataset 5	62	58	4	93.55%	0.80
6	Dataset 6	75	73	2	92.23%	0.87
7	Dataset 7	103	95	8	88.89%	0.84
8	Dataset 8	54	48	6	93.00%	0.80

Table 5 DSM Results

No	Dataset	Number of buildings	Buildings detected	Buildings undetected	Accuracy	IOU
1	Dataset 1	32	17	15	53.12%	0.46
2	Dataset 2	35	15	20	42.85%	0.34
3	Dataset 3	61	31	30	50.81%	0.42
4	Dataset 4	36	22	14	61.11%	0.54
5	Dataset 5	62	40	22	64.51%	0.53
6	Dataset 6	75	47	28	62.67%	0.51
7	Dataset 7	103	60	43	58.25%	0.49
8	Dataset 8	54	33	21	61.11%	0.53

process using Euclidean Clustering can segment buildings. According to the test results, the best results give an accuracy of 91.67% and an IOU score of 0.84. Based on our evaluation, the building extraction and building segmentation process produce an accuracy that varies from 81.25% to 91.67% and an IOU score that varies from 0.65 to 0.84. This result has shown that the process of combining the DGCNN and Euclidean clustering can be used to automatically segment buildings.

Conclusion

The automatic segmentation of buildings can be accomplished by using a combination of two methods: the DGCNN as the method to separate buildings and nonbuildings and the Euclidean clustering method to segment the buildings. The evaluation process is done by using two metrics, the accuracy and IOU score, for each building. From the evaluation results, according to those metrics, the best results obtained achieve an accuracy of 91.67% and an average IOU score of 0.84. The test results of several datasets show that the IOU varies from 0.65 to 0.84. In the future, we will try to implement and explore some methods that can solve the problem of clustering buildings with small gaps between one building and another.

Abbreviation

CNN: Convolutional neural network; DEM: Digital Elevation Model; DGCNN: Dynamic Graph Convolutional Neural Network; DSM: Digital Surface Model; DTM: Digital Terrain Model; IOU: Intersection Over Union; LIDAR: Light Detection and Ranging.

Acknowledgements

We would like to express our gratitude for the grant received from Indonesia Ministry of Higher Education (RISTEK-BRIN) No: NKB-363/UN2.RST/HKP.05.00/2020. Faculty of Computer Science Universitas Indonesia and PT. Pangripta Geomatika Indonesia.

Authors' contributions

AG Contributed the idea of building extraction and building segmentation using engineering perspective. AW Technical guidance, infrastructure, and algorithm design. SBW, MAA, NH, and HAW Perform data cleansing, data verification, simulation, and manuscript proofreading. WJ Contributed the idea in the perspective of computer science and algorithm. RA Data acquisition using drone and LIDAR equipment. All authors read and approved the final manuscript.

Funding

Indonesia Ministry of Higher Education (RISTEK-BRIN).

Availability of data and materials

<https://drive.google.com/drive/folders/12-3taOn9dL7hWwnC6yQW8BqINOCY8cEF?usp=sharing>.

Competing interests

Not Applicable.

Author details

¹ Faculty of Computer Science, Universitas Indonesia, Kampus UI Depok, Indonesia. ² Faculty of Engineering, Universitas Indonesia, UI Depok, Indonesia. ³ PT Pangripta Geomatika Indonesia (UAV LIDAR Specialist), Jakarta, Indonesia.

Received: 23 June 2020 Accepted: 1 November 2020

Published online: 17 November 2020

References

1. Rottensteiner, F., Trinder, J., Clode, S., Kubik, K., 2005. Using the Dempster–Shafer method for the fusion of LIDAR data and multi-spectral images for building detection. *Information Fusion*, Volume 6, Issue 4, December 2005, pp 283–300.
2. Vosselman, G., 1999. Building reconstruction using planar faces in very high density height data. *International Archives of Photogrammetry and Remote Sensing* 32 (Part 2), 87–92.
3. Sohn, G., Dowman, I., 2007. Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction. *ISPRS Journal of Photogrammetry & Remote Sensing* 62 (2007) pp.43–63, 15 February 2007.
4. S. M. Abdullah, M. Awrangjeb and Guojun Lu, "Automatic segmentation of LiDAR point cloud data at different height levels for 3D building extraction," 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Chengdu, 2014, pp. 1–6.
5. Hao Z, Yongsheng Z, Jun L, Song J. Automatic building detection using airborne LIDAR data. *Int Forum Inform Technol Appl*. 2009;2009:668–71.
6. Vosselman G. Fusion of laser scanning data, maps, and aerial photographs for building reconstruction. Toronto: International Geoscience and Remote Sensing Symposium; 2002.
7. Vosselman G, Dijkman S. "3D building model reconstruction from point clouds and ground plans", IAPRS, XXXIV-3/W4. US: Annapolis; 2001.
8. I. Suveg, G. Vosselman, "3D building reconstruction by map based generation and evaluation of hypotheses," 12th Proceedings of the British Machine Vision Conference, Manchester, UK, 2001.
9. Haala Norbert, Brenner Claus. Extraction of buildings and trees in urban environments. *ISPRS J Photogrammetry Remote Sensing*. 1999;54:130–7.
10. Baltisavius E. Object extraction and revision by image analysis using existing geodata and knowledge : current status and steps towards operational systems. *ISPRS J Photogrammetry Remote Sensing*. 2004;58:129–51.
11. F. Tarsha-Kurdi, T. Landes, P. Grussenmeyer, et al., "New approach for automatic detection of building in airborne laser scanner data using first echo only," IAPRS, vol. 36, part 3, pp. 25–30, 2006.
12. Yang HL, Yuan J, Lunga D, Laverdiere M, Rose A, Bhaduri B. Building Extraction at Scale Using Convolutional Neural Network: mapping of the United States. *IEEE J Selected Topics Applied Earth Observations Remote Sensing*. 2018;11(8):2600–14.
13. Nahhas, F. H., Shafri, H. Z., Sameen, M. I., Pradhan, B., & Mansor, S. (2018). Deep learning approach for building detection using lidar–orthophoto fusion. *Journal of Sensors*, 2018.
14. Lafarge F, Descombes X, Zerubia J, Pierrot-Deseilligny M. Automatic building extraction from DEMs using an object approach and application to the 3D-city modeling. *ISPRS J Photogrammetry Remote Sensing*. 2008;63(3):365–81.
15. Tomljenovic I, Höfle B, Tiede D, Blaschke T. Building extraction from airborne laser scanning data: an analysis of the state of the art. *Remote Sensing*. 2015;7(4):3826–62.
16. Li Y, Wu H, An R, Xu H, He Q, Xu J. An improved building boundary extraction algorithm based on fusion of optical imagery and LiDAR data. *Optik Int J Light Electron Optics*. 2013;124(22):5357–62.
17. Li H, Zhong C, Hu X, Xiao L, Huang X. New methodologies for precise building boundary extraction from LiDAR data and high resolution image. *Sensor Review*. 2013;33(2):157–65.
18. V. Saeidi, B. Pradhan, M. O. Idrees, and Z. Abd Latif, "Fusion of airborne LiDAR with multispectral spot 5 image for enhancement of feature extraction using Dempster–Shafer theory," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6017–6025, 2014. .
19. Awrangjeb M, Zhang C, Fraser CS. Building detection in complex scenes thorough effective separation of buildings from trees. *Photogrammetric Eng Remote Sensing*. 2012;78(7):729–45.

20. R. Wang, Y. Hu, H. Wu, and J. Wang, "Automatic extraction of building boundaries using aerial LiDAR data," *Journal of Applied Remote Sensing*, vol. 10, no. 1, article 016022, 2016.
21. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, <http://www.deeplearningbook.org>.
22. LeCun Yann, Bottou Leon, Bengio Yoshua, Haffner Patrick. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278–323.
23. Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference in Learning Representation*, 2015, pp. 1–14.
24. C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9.
25. Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.
26. Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.
27. Tara N Sainath, Brian Kingsbury, George Saon, Hagen Soltau, Abdelrahman Mohamed, George Dahl, and Bhuvana Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks*, vol. 64, pp. 39–48, 2015.
28. Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, pp. 1–14, 2015.
29. Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell*. 2017;39(4):640–51.
30. van Noord Nanne, Postma Eric. Learning scale-variant and scaleinvariant features for deep image classification. *Pattern Recogn*. 2017;61:583–92.
31. Jamie Sherrah, "Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery," *arXiv*, pp. 1–22, 2016.
32. Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez, "Fully convolutional neural networks for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 5071–5074, 2016.
33. Volpi Michele, Tuia Devis. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Trans Geosci Remote Sens*. 2017;55(2):881–93.
34. Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & M.Solomon, J. (2018, January 24). Dynamic Graph CNN for Learning on Point Clouds. <https://arxiv.org/pdf/1801.07829.pdf>.
35. Rusu, R. B. (2009). *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. München.
36. Ramiya, Anandakumar M., Rama Rao Nidamanuri, and Ramakrishan Krishnan. "Segmentation based building detection approach from LiDAR point cloud." *The Egyptian Journal of Remote Sensing and Space Science* 20.1 (2017): 71–77.
37. Huang, Jianfeng, et al. "Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network." *ISPRS journal of photogrammetry and remote sensing* 151 (2019): 91–105.
38. Pan, Xuran, et al. "Semantic labeling of high resolution aerial imagery and LiDAR data with fine segmentation network." *Remote Sensing* 10.5 (2018): 743.
39. Chen, Shanxiong, et al. "Automatic Building Extraction via Adaptive Iterative Segmentation With LiDAR Data and High Spatial Resolution Imagery Fusion." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020): 2081–2095.
40. Laefer DF, Abuwarda S, Vo A-V, Truong-Hong L, Gharibi H. 2015 Aerial Laser and Photogrammetry Survey of Dublin City Collection Record. Center for Urban Science and Progress: New York University; 2015.
41. R. M. Haralick and L. G. Shapiro, *dalam Computer and Robot Vision*, vol. I, Addison-Wesley, 1992, pp. 28–48.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)