sensors

MDPI

Article

ALS Point Cloud Classification by Integrating an Improved Fully Convolutional Network into Transfer Learning with Multi-Scale and Multi-View Deep Features

Xiangda Lei [1], Hongtao Wang [1,*], Cheng Wang [1,2], Zongze Zhao [1], Jianqi Miao [1] and Puguang Tian [1]

[1] School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China; 211804010013@home.hpu.edu.cn (X.L.); wangcheng@radi.ac.cn (C.W.); zhaozongze@hpu.edu.cn (Z.Z.); 211904020001@home.hpu.edu.cn (J.M.); 211904020027@home.hpu.edu.cn (P.T.)

[2] Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China

* Correspondence: wht_31@hpu.edu.cn; Tel.: +86-391-3987-665

Received: 21 October 2020; Accepted: 4 December 2020; Published: 6 December 2020

Abstract: Airborne laser scanning (ALS) point cloud has been widely used in various fields, for it can acquire three-dimensional data with a high accuracy on a large scale. However, due to the fact that ALS data are discretely, irregularly distributed and contain noise, it is still a challenge to accurately identify various typical surface objects from 3D point cloud. In recent years, many researchers proved better results in classifying 3D point cloud by using different deep learning methods. However, most of these methods require a large number of training samples and cannot be widely used in complex scenarios. In this paper, we propose an ALS point cloud classification method to integrate an improved fully convolutional network into transfer learning with multi-scale and multi-view deep features. First, the shallow features of the airborne laser scanning point cloud such as height, intensity and change of curvature are extracted to generate feature maps by multi-scale voxel and multi-view projection. Second, these feature maps are fed into the pre-trained DenseNet201 model to derive deep features, which are used as input for a fully convolutional neural network with convolutional and pooling layers. By using this network, the local and global features are integrated to classify the ALS point cloud. Finally, a graph-cuts algorithm considering context information is used to refine the classification results. We tested our method on the semantic 3D labeling dataset of the International Society for Photogrammetry and Remote Sensing (ISPRS). Experimental results show that overall accuracy and the average F1 score obtained by the proposed method is 89.84% and 83.62%, respectively, when only 16,000 points of the original data are used for training.

Keywords: ALS point cloud; classification; transfer learning; fully convolutional neural network; graph-cuts; small training samples

1. Introduction

ALS (airborne laser scanning) technology can obtain high precision and high-density 3D point cloud for a large area, which are used widely in topographic mapping, forest monitoring, power line detection, 3D building reconstruction and so on [1–4]. However, due to the fact that ALS data are discretely, irregularly distributed and contain noise, it is still a challenge to accurately identify various typical surface objects in complex scenarios [5].

Currently, when classifying 3D point cloud data, there are typically three kinds of approaches that have been widely investigated, namely rule-based point cloud classification methods, traditional machine learning methods and deep learning methods. Among them, rule-based point cloud classification methods first extract the features of each 3D point and then the corresponding classification rules are designed to classify the point cloud [6–9]. These methods are relatively stable, but are only applicable when large category attributes have been derived and cannot work well in complex scenes [10]. Methods based on traditional machine learning are relatively mature point cloud classification methods. Among these methods, SVM (support vector machine) and RF (random forest) classifiers are often used to perform point cloud classification with artificial features [11–15]. However, when applying these methods, the local features of each point need to be estimated independently, which requires elaborate designing. Moreover, the classification results of these methods tend to suffer from noise because the label consistency among neighborhood points is not considered [16]. To solve this problem, researchers began to use CRF (conditional random fields) and MRF (Markov random fields) frameworks to integrate contextual information into the classification process [17,18]. However, the classification accuracy of these methods is limited by the feature quality and the performance of the classifier, which cannot be applied to the large-scale regions, especially in complex environment.

In recent years, deep learning methods have become the most prevalent approach and achieved great success in image processing and related fields [19–21]. Following this trend, scholars began to apply CNN (convolutional neural network) on point cloud classification. In order for the existing CNN models to be used for point cloud classification directly, researchers often transform the unevenly distributed point cloud into regular images or voxels [22–24]. However, these methods will lead to feature loss and information redundancy in the process of transforming point cloud into a non-point cloud. Therefore, researchers began to study the classification methods on the original 3D point cloud. Qi et al. designed the PointNet model, which was the first deep learning framework to directly classify the original 3D point cloud [25]. It extracts the local features and global features by using convolutional and max-pooling operation, and then these features are fed into MLP (multiple layer perception) for point clouds classification. With the success of PointNet, more and more studies focus on PointNet-like architectures for point cloud classification. Qi et al. further designed a PointNet++ model to extract the multi-scale local features by point feature down-sampling and up-sampling [26]. Ben et al. used an improved Fisher vector to represent point cloud and design a CNN for point classification by these vectors [27]. Li et al. propose a PointCNN model, a new point feature learning architecture by using the X-transformation to standardize the disordered point cloud [28]. Tomas et al. designed a new point convolution named KPConv, which could be applied directly to point cloud [29]. Wen et al. proposed a novel, directly constrained point convolution module named D-Conv, which extracts locally representative features of 3D point sets from the projected 2D receptive fields [16]. Moreover, Arief et al. used the Atrous XCRF model to alleviate the overfitting problem of PointCNN [30]. Although these 3D point-based methods have achieved success in point cloud classification, such methods require a large number of training samples and manual labeling, which is time consuming and laborious. Zhao et al. [31] proposed an ALS point cloud classification method based on transfer learning. In this method, it used the deep residual network to extract deep features, and then a CNN was designed for point cloud classification. Nonetheless, a small range of multi-view projection led to the lack of detailed description on each 3D point, which affected classification accuracy. Moreover, in the process of generating feature maps, a large amount of computing time was required.

To overcome these problems, this paper proposes an ALS point cloud classification method which integrates an improved fully convolutional network into transfer learning with multi-scale and multi-view deep features. The main contributions of this paper are as follows:

(1) We proposed a novel feature map generation method using multi-scale voxel and multi-view projection. This algorithm can quickly and accurately extract the features and reduce the effect of ground object size on classification results.

(2) We proposed a novel point cloud classification method by integrating an improved fully convolutional neural network into the transfer learning. This method uses the pre-trained DensNet201 model to derive deep features, which can characterize the object effectively. Then, the full CNN is improved by using max-pooling operation to extract the global features and average-pooling operation to derive the final feature of each object. Finally, the improved fully CNN is used for the classification of the ALS point cloud.

(3) Our method achieves the highest accuracy in the ISPRS reference dataset [5]. When only 16,000 points of the original data are used as training samples, our overall accuracy is 89.84% and the average F1 score is 83.62%.

The rest of the paper is organized as follows. In Section 2, we describe the proposed methods in detail. Section 3 shows the experimental data, settings, and results. Section 4 discusses the experimental results and evaluates the performance of the proposed method on ISPRS point cloud segmentation benchmark dataset. Finally, we conclude our work in Section 5.

2. Methodology

2.1. Overview

Figure 1 shows the overview of the proposed method. The whole procedure is composed of three parts: (1) feature map generation, (2) point cloud classification based on transfer learning, (3) post processing.

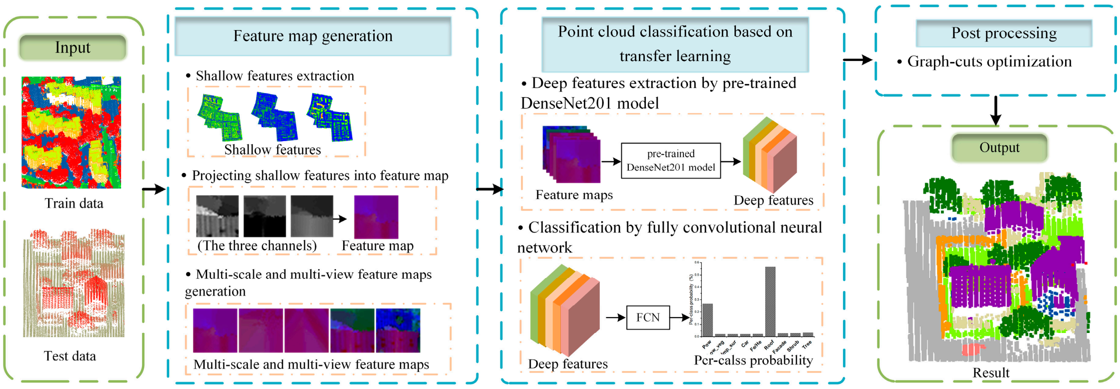
Input
Train data
Test data
Feature map generation
• Shallow features extraction
Shallow features
• Projecting shallow features into feature map
(The three channels)
Feature map
• Multi-scale and multi-view feature maps generation
Multi-scale and multi-view feature maps
Point cloud classification based on transfer learning
• Deep features extraction by pre-trained DenseNet201 model
Feature maps
pre-trained DenseNet201 model
Deep features
• Classification by fully convolutional neural network
Deep features
FCN
Per-class probability (%)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Pow
Low_veg
Imp_sur
Car
Fe/He
Roof
Facade
Shrub
Tree
Pcr-calss probability
Post processing
• Graph-cuts optimization
Output
Result

Figure 1. The flow chart of the proposed classification method.

(1) Feature map generation includes the shallow feature extraction and the generation of feature maps by multi-scale voxel and multi-view projection.

(2) Point cloud classification based on transfer learning, which includes deep feature extraction based on a pre-trained DenseNet201 model and a classification procedure using a fully convolutional neural network with convolutional and pooling layers.

(3) Post processing, in which a graph-cuts algorithm considering context information is used to refine the classification result.

A detailed explanation of the whole procedure, as well as the respective methods is presented in the following subsections.

2.2. Feature Map Generation by Multi-Scale Voxel and Multi-View Projection

2.2.1. Shallow Feature Extraction

At present, the main shallow features used in ALS point cloud classification are elevation features, spectral features and geometric features. The literature [5] evaluates the importance of various features, and the results show that the normalized height in elevation features is the most important feature in ALS point cloud classification. Intensity information in spectral features is widely used for classification because it can effectively distinguish ground objects of different materials. Similarly, the change of curvature in geometric features can distinguish flat objects such as buildings and roads from irregular objects such as vegetation and shrubs [31]. Therefore, the above three features, namely the normalized height, intensity and change of curvature are selected as the shallow features in this paper. To calculate the normalized height, an excellent and efficient CSF filtering algorithm is used to identify ground and non-ground points from the original point cloud [32]. Then, the ground points are interpolated to generate a DEM (digital elevation model), and the normalized height of each point is obtained by reducing the corresponding height in DEM. Intensity features can be derived by searching in the original file. The change of curvature is obtained by the covariance analysis. In the calculation of covariance, the selection of neighborhood points is very important. Our proposed method uses Shannon entropy to adaptively select the optimal number of neighborhood points for each point [33] and then the optimal neighborhood point is used to obtain the change of curvature.

2.2.2. Projecting Shallow Feature into Feature Map

Due to the requirement of subsequent transfer learning, we project the shallow features into feature maps. The projection method combining point cloud voxelization and cube neighborhood projection, which can preserve the original spatial features and reduce the computational complexity. The projection of feature map is shown in Figure 2.

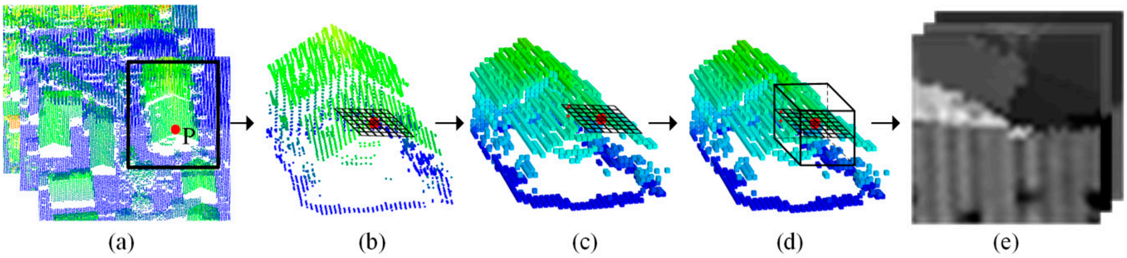

Figure 2. Projection of a feature map: (**a**) ALS point cloud display by the normalized height, intensity and change of curvature, (**b**) two-dimensional grid generation of point P, (**c**) Z value of each grid node is calculated in the voxelized point cloud, (**d**) calculating the normalized height, intensity and change of curvature of each grid node in the cube neighborhood, (**e**) feature map of point P that indicating the normalized height, intensity and change of curvature.

Figure 2 shows the projection process from the normalized height, intensity and change of curvature to feature map of point P. First, point P is projected to an X-Y plane, and then a regular grid is constructed with point P as the center, as shown in Figure 2b. The coordinates of each grid node are calculated by Equation (1).

$$\begin{aligned} X_{p_{i,j}} &= X_P + (j - [(N-1)/2]) * S, \\ Y_{p_{i,j}} &= Y_P + (i - [(N-1)/2]) * S, \end{aligned} \tag{1}$$

In Equation (1), i and j are the row and column number corresponding to the grid node $p_{i,j}$. ($Xp_{i,j}$,$Yp_{i,j}$) are the two-dimensional coordinates of point $P_{i,j}$. (X_P, Y_P) are the coordinates of point P; S is the size of the grid; N is the height of feature map, which is defined as 33 in this paper.

Second, a voxelized point cloud is generated by down sampling the original point cloud with grid size S. Then, two-dimensional adjacent points of each grid node are searched in the voxelized point cloud. By using 3D coordinates of these adjacent points, an interpolation process is conducted to derive the Z value of the corresponding grid node, as shown in Figure 2c. For each 3D grid node calculated from above process, three points closest to it are searched in a cube neighborhood with point

P serving as the center. Then these points are used to calculate the normalized height, intensity and change of curvature of the corresponding grid node, as shown in Figure 2d.

Finally, the three features above can be re-assigned to a range of 0~255, that is, a feature map with three channels can be generated, which can be seen from Figure 2e.

2.2.3. Multi-Scale and Multi-View Feature Maps Generation

To derive a more accurate description of 3D objects in point cloud, a multi-scale voxel and multi-view feature maps generation method is proposed in this paper. For each 3D point, three feature maps are generated with different grid sizes by using aforementioned feature map projection method, which can be seen from Figure 3. In this work, three grid sizes including S1 = 0.1 m, S2 = 0.3 m and S3 = 0.5 m are adopted for the generation of multi-scale feature maps, which are determined by the point cloud density and the size of ground objects in the experimental area. Differently from the method of Zhao et al. [31], we generated the multi-scale feature maps based on the voxelized point cloud, which can improve the neighborhood searching efficiency during the generation of feature maps.

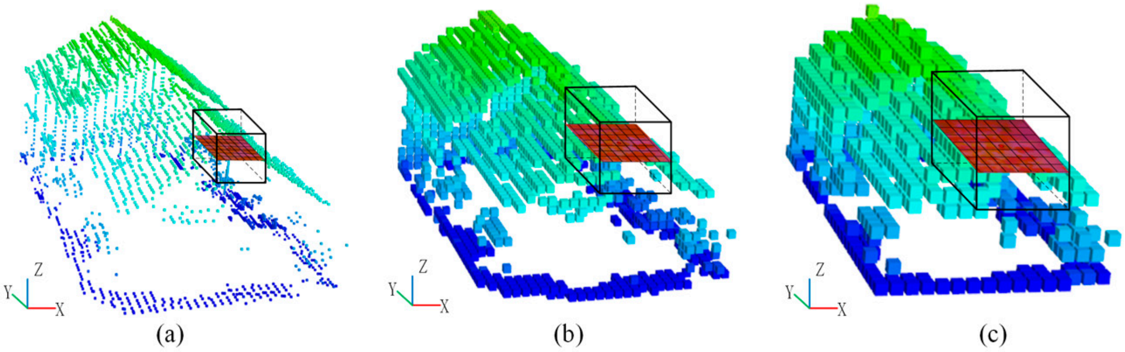

Figure 3. Multi-scale feature maps: (**a**) feature map with scale of S1, (**b**) feature map with scale of S2, (**c**) feature map with scale of S3.

Note that the aforementioned feature maps are derived by projecting the point cloud into X-Y plane. Figure 4a shows the feature map with a grid size of S1 based on the projection plane of X-Y plane. In addition, other two feature maps are then generated by projecting the point cloud to Y-Z plane and X-Z plane by using the feature map projection method described in Section 2.2.2, as shown in Figure 4b,c. In the method of Zhao et al. [31], they rotated the point cloud with two small angles to generate multi-view feature maps. In our method, we project the voxelized point cloud into X-Y, Y-Z, and X-Z plane for the generation of multi-view feature maps, which can derive a more accurate description of each 3D point.

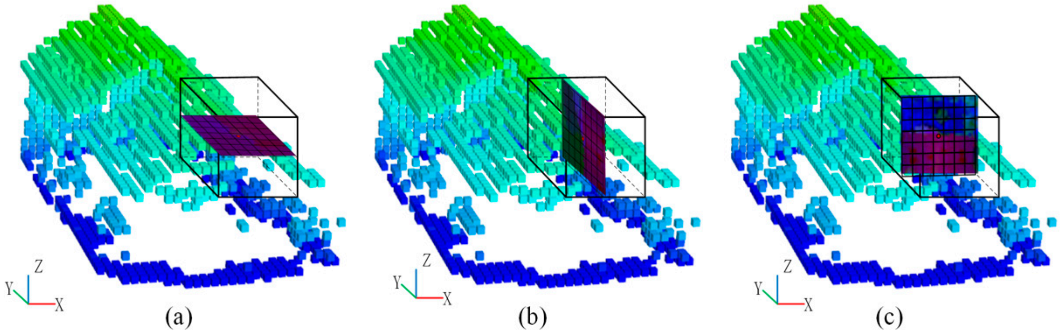

Figure 4. Multi-view feature maps: (**a**) feature map with view of X-Y plane, (**b**) feature map with view of Y-Z plane, (**c**) feature map with view of X-Z plane.

In summary, five feature maps are obtained as the inputs for the transfer learning. Among them, three feature maps are generated by projecting the point cloud into X-Y plane with three grid sizes, and other two feature maps are generated by projecting the point cloud into Y-Z and X-Z plane with the same grid size.

2.3. Point Cloud Classification Based on Transfer Learning

2.3.1. Deep Feature Extraction

In order to obtain a well-trained deep neural network, a large number of training samples are usually required and many parameters need to be tuned in this process. To solve these problems, scholars began to apply the transfer learning method for point cloud classification. In transfer learning, it is critical to choose an appropriate pre-trained neural network, which can extract deep features with good generalization performance [31]. The well-known network-based transfer learning model includes AlexNet, VggNet, and ResNet which are commonly used in various classification tasks [34]. Zhao et al. use the ResNet50 model pre-trained on ImageNet to extract the deep features [31]. In this paper, we use DensNet201 model instead of ResNet50 model to extract deep features, because the pre-trained DensNet201 model achieves better performance in the following classification task. This can be explained that the dense connection and feature reuse of DenseNet can extract deep features more efficiently. Meanwhile, the ImageNet dataset can also provide data guarantee for transfer learning [35]. As shown in Figure 5, the feature maps (5 × 33 × 33 × 3) generated in Section 2.2 are fed into the pre-trained DesNet201 model to generate the multi-scale and multi-view deep features (5 × 1920).

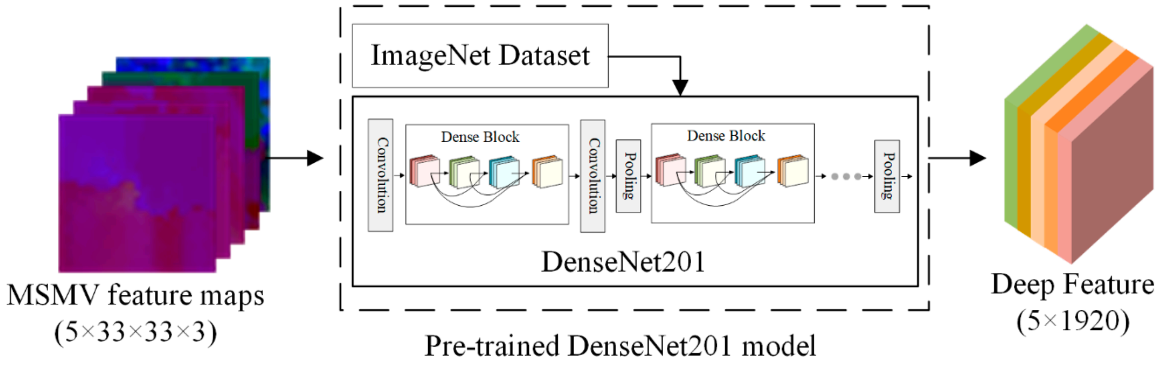

Figure 5. Deep feature extraction based on pre-trained DenseNet201 model.

2.3.2. Classification by Using an Improved Fully Convolutional Neural Network

Multi-scale and multi-view deep features derived from the pre-trained DenseNet201 model have a good generalization ability, which is critical to improve classification accuracy. Based on these deep features, a classifier is often needed for subsequent point cloud classification. In this paper, an improved fully convolutional neural network was designed to carry out the classification task. Compared with the fully-connected layer, the convolutional and pooling layers can effectively conduct data mining, reduce computing cost and increase classification accuracy [25,36]. We designed a fully convolutional neural network with convolutional and pooling layers for the classification of ALS point cloud, as can be seen from Figure 6. First, the deep features derived from pre-trained DenseNet201 model (5 × 1920) were used to generate the global features (1 × 1920) by the max-pooling operation. Second, the combined features (6 × 1920), which were formed by global features and deep features were fed into multiple convolutional layers with different convolutional kernels to learn high-level features (2 × 8). After that, the average-pooling layer was used to derive the final feature of each object (1 × 8). Finally, a softmax layer was used to obtain the per-class probability of each point and the prediction label was obtained.

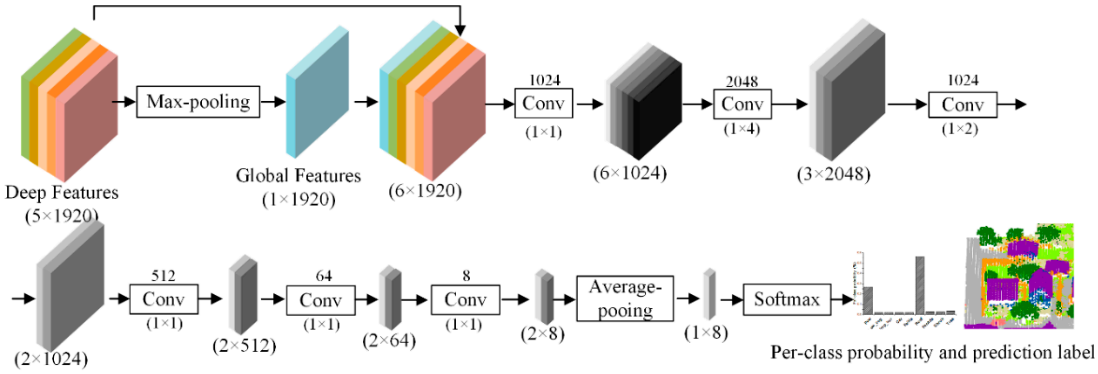

Figure 6. Point cloud classification based on the fully convolution neural network with pooling and convolutional layers (the Conv in the figure represent a convolutional layer, the number above Conv represents the number of the convolution kernel and the number under Conv represents the size of the convolutional kernel).

2.4. Post Processing with Graph-Cuts Optimization

Although the context information is taken into account in the generation of feature maps, the point cloud classification adopted in this paper is conducted point by point, so it is easy to make the point cloud classification result containing noise, which affects the classification accuracy. To further improve the classification accuracy, a post processing is conducted by using graph-cuts optimization. In this process, the adjacent points around the interest point are searched to build an undirected graph model. The nodes in the graph represent each 3D points, while the edges indicate the neighborhood environment. Based on this graph model, a cost function is defined as:

$$\mathrm{E(L)} = \sum\nolimits_{p\in P} D_p\left(l_p\right) + \lambda \sum\nolimits_{\{\mathrm{p,q}\}\in N} V_{p,q}\left(l_p, l_q\right)\cdot B_{(p,q)}, \tag{2}$$

In Equation (2), the first item on the right side of the equation is the data item. $D_p\left(l_p\right)$ represents the cost when marking point p as the corresponding category l_p, which is defined as the per-class probability of point p derived from the transfer learning. The second item on the right of the equation is a smooth item. $V_{p,q}\left(l_p, l_q\right)$ represents the cost of determining adjacent points (p, q) as the same or different category. When marked as the same category, its value is 0; otherwise, its value is 1. The parameter λ shows the contribution of the data and smooth terms to the total energy function. P is the set of all points; N is the geometry of neighborhood points. $B_{(p,q)}$ can be expressed as Equation (3). In Equation (3), k is defined as the 3D distance between adjacent point p and q.

$$B_{(p,q)} = \mathrm{e}^{-\mathrm{k}}, \tag{3}$$

The aforementioned cost function can be solved by employing a graph-cuts strategy using the alpha-expansion, which can minimize the cost function to refine the classification results [37].

3. Experimental Results

3.1. Dataset

We use two ALS dataset to evaluate the performance of the proposed classification method. The first dataset is provided by the ISPRS Test Project on Urban Classification and 3D Building Reconstruction [5]. The ISPRS benchmark dataset is collected by the Leica ALS50 system and has a point density of about 4/m^2. Each point contains information such as 3D coordinates, reflection intensity and category label. As shown in Figure 7, the dataset is divided into training data and test data. Among them, the training data have 753,876 points and the testing data have 411,722 points. The dataset defines nine different classes, including powerline (Pow), low vegetation (Low_veg), impervious surface (Imp_sur), car (Car), fence/hedge (Fe/He), roof (Roof), facade (Facade), shrub (Shrub) and tree (Tree). In our experiment, we ignore class powerline, which has a limited number of occurrences (around a thousand) in the ISPRS benchmark dataset.

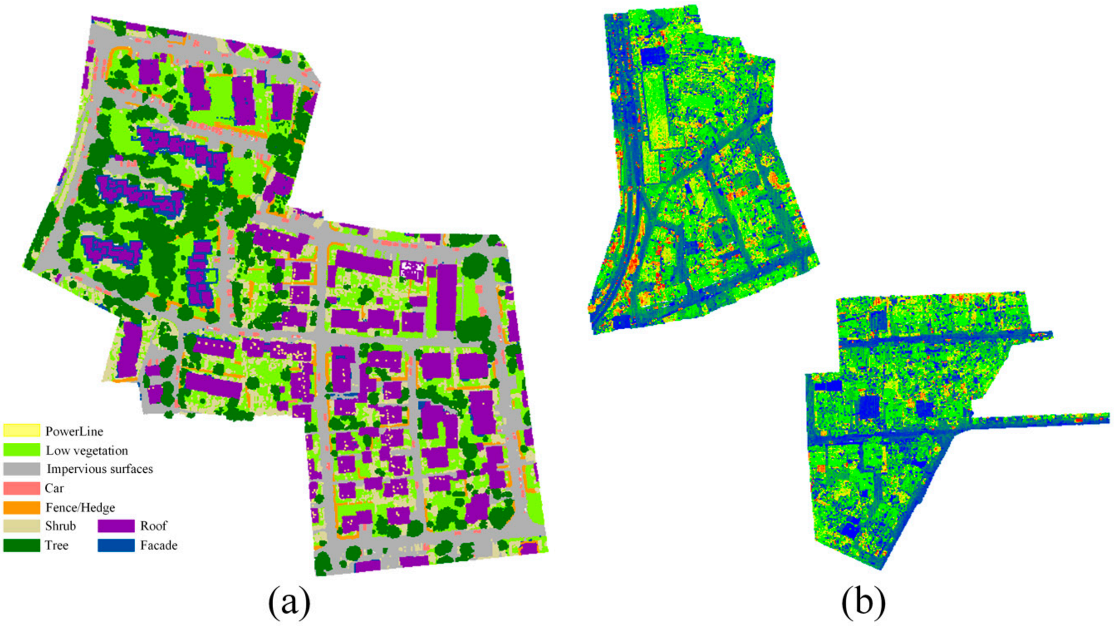

Figure 7. International Society for Photogrammetry and Remote Sensing (ISPRS) dataset: (**a**) training dataset displayed by label, (**b**) testing dataset displayed by intensity.

In order to further verify the generalization ability of the proposed classification method, experiments are conducted on the 2019 IEEE GRSS Data Fusion Competition (DFC) 3D point cloud classification dataset. The dataset contains information such as 3D coordinates, intensity and labels for each point, and are labeled into six classes: ground, high vegetation, building, water, road/bridge and unclassified points. As shown in Figure 8, in our experiment, we chose two areas from the dataset as training data and testing data, respectively, and kept only the first five kinds of point clouds. Among them, there are 748,569 points in the training data and 799,881 points in the testing data.

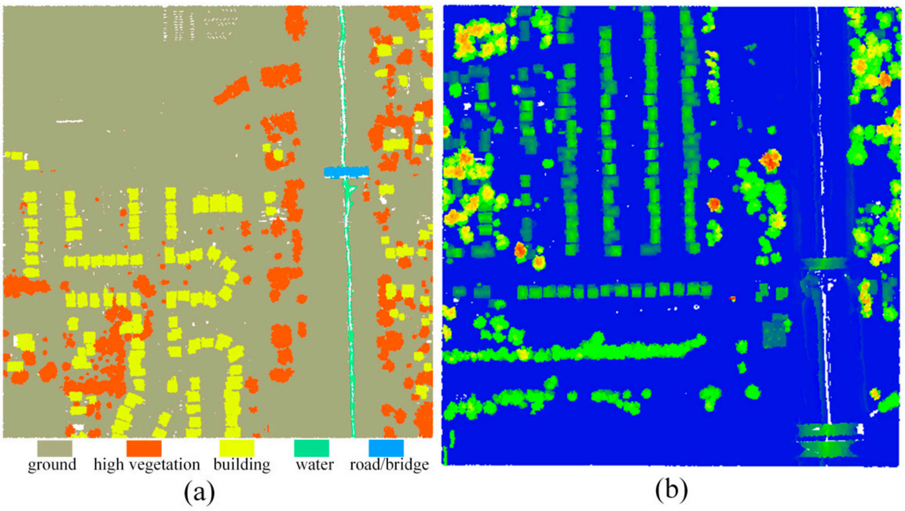

Figure 8. The dataset derived from the 2019 IEEE GRSS Data Fusion Competition: (**a**) training dataset displayed by label, (**b**) testing dataset displayed by the normalized height.

3.2. Experiment Settings

In the experiment, we randomly selected 2000 samples per-class from ISPRS benchmark dataset and IEEE training dataset for training. By using these training data, the proposed network is trained 100 epochs. During the training, we adopted the optimization method of Adam [38], with the initial learning rate set as 0.0001 and batch size set as 128. Overall accuracy (OA), F1 score and average F1 score (Avg. F1) are used as accuracy evaluation indexes. The F1 score is an evaluation index that comprehensively considers the precision and the recall. F1 score is defined as Equation (4):

$$\mathrm{F1} = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{4}$$

In this study, Python 3.5, Tensorflow 1.13.1 and Keras 2.4.3 were used to generate the feature maps and classify the point cloud. All the algorithms were implemented on a computer with an Intel I7-6700HQ CPU, 16 GB RAM, and a GeForce GTX1070 (8 GB\Nvidia) GPU.

3.3. Classification Result

Based on the above experiment settings, two aforementioned datasets are used to validate the proposed method. The qualitative evaluation of the experimental results can be seen from Figure 9. We can tell by the Figure 9 that various objects have been effectively identified by using the proposed method in this paper by our proposed model.

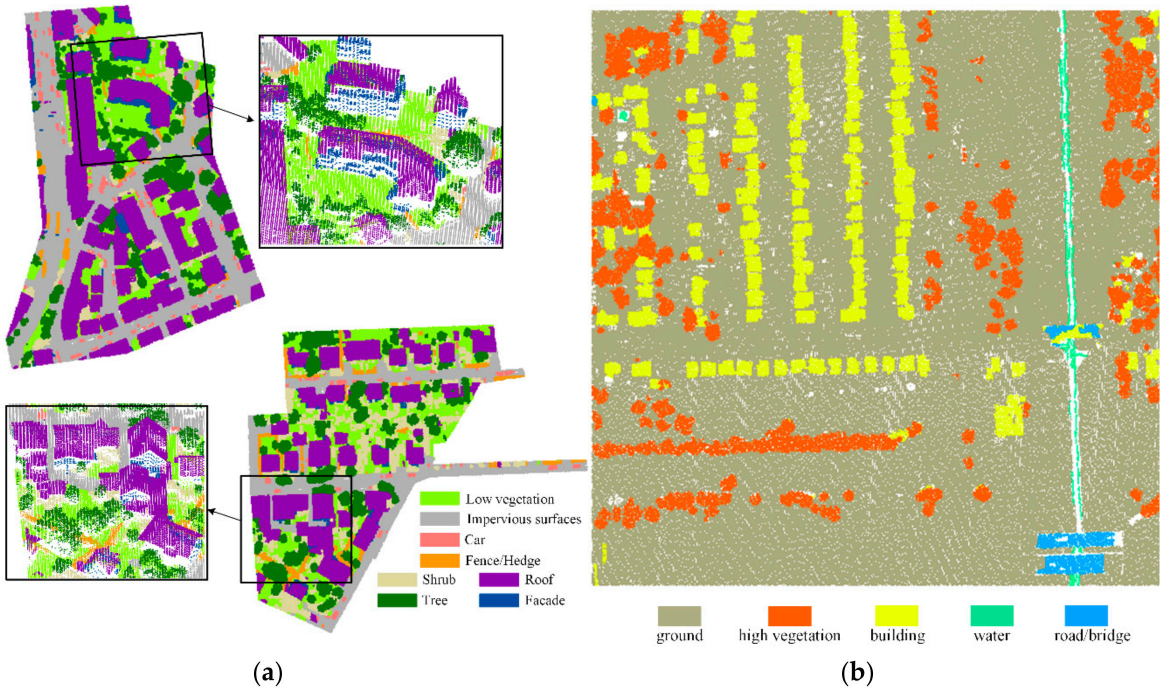

Figure 9. The experimental result: (**a**) the classification results of ISPRS benchmark dataset (the black boxes in the figure represent partial enlargements to show details of the classification result), (**b**) the classification results of the 2019 IEEE GRSS Data Fusion Competition dataset.

In addition to qualitative evaluation, the overall accuracy and F1 score are used to quantitatively evaluate the experimental result. For ISPRS benchmark dataset, the overall accuracy and average F1 score are 89.84% and 83.62%, respectively, which are higher than all the contrast methods. In addition, we also obtain a high classification accuracy for the IEEE dataset. The overall accuracy of the IEEE dataset is 98.31% and the average F1 score is 87.50%. The F1 scores of each category are 99.36% (ground), 97.48% (high vegetation), 92.97% (building), 66.11% (water) and 81.57% (road/bridge), respectively.

4. Discussion

4.1. Impact of Multi-View and Multi-Scale Features

In order to verify the impact of multi-view and multi-scale features on the classification results, we performed experiments using four configurations: (1) using the single scale feature with a grid size of 0.1 m for classification, based on that we project the point cloud into X-Y plane (V1/S1); (2) using multi-scale features with grid sizes of 0.1 m, 0.3 m and 0.5 m for classification, we project the point cloud into X-Y plane (MS); (3) using multi-view features by projecting the point cloud into X-Y, Y-Z and X-Z plane for classification set the grid size as 0.1 m (MV); (4) using the multi-scale and multi-view features for classification (MSMV). Figure 10 shows per-class F1 score of the aforementioned four configurations.

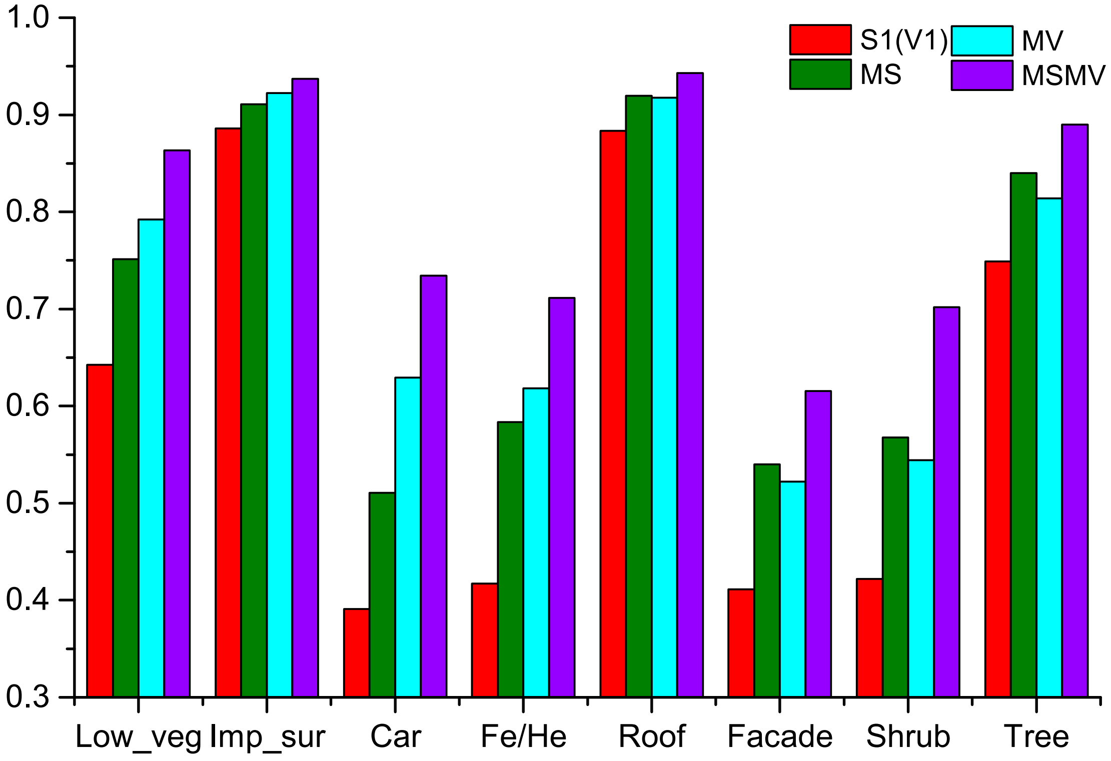

Figure 10. F1 score of each class on ISPRS benchmark dataset under four different configurations.

As shown in Figure 10, the per-class F1 score of S1 (V1) is lower than the configuration using MS features or MV features, and the highest per-class F1 score is obtained by using MSMV features in this experiment. In addition, the overall accuracy of MSMV is 87.96%, which is higher than S1 (73.22%), MS (81.09%) and MV (82.29%). In general, the configuration of using MSMV features achieves the best classification result in this experiment. This can be explained by that the multi-scale and multi-view features can be used to effectively express the inherent features of 3D point cloud. So, we use MSMV features as the input of the subsequent transfer learning.

Note that we voxelize the point cloud by using the grid sizes as the feature maps. By this means, the original point cloud can be reduced by 10.5%, 30.0%, 50.1% in scale of 0.1 m, 0.3 m and 0.5 m, respectively. Experimental results show that the overall accuracy of using voxel is 87.96% and the average F1 score is 79.96%, which are almost the same as the classification accuracy without voxelizing the point cloud, 87.97% (OA) and 80.07% (Avg. F1). It is indicated that the proposed multi-scale voxel projection method can reduce time of calculation while maintaining the classification accuracy.

4.2. Performance of the Improved Fully Convolutional Neural Network

In this paper, we designed an improved fully convolutional neural network with convolutional and pooling layers for point cloud classification. In order to verify the performance of the proposed network, we conducted experiments using three networks for comparison: (1) FC only uses the fully-connected layer; (2) FCN only uses the convolutional layer; and (3) CN-P uses the pooling and the convolutional layers. Table 1 lists the overall accuracy and F1 score of each category by using different networks.

Table 1. F1 score of each class and overall accuracy (OA) using different classification network on ISPRS benchmark dataset. Bold numbers show the highest values (%).

Classifer	**F1 Score**								**OA**
	Low_veg	**Imp_sur**	**Car**	**Fe/He**	**Roof**	**Facade**	**Shrub**	**Tree**	
FC	83.78	92.15	66.45	69.46	92.13	55.07	63.75	85.51	85.09
FCN	84.11	93.1	71.49	65.33	93.67	60.63	64.95	87.53	86.3
CN-P	**86.36**	**93.69**	**73.44**	**71.14**	**94.3**	**61.55**	**70.18**	**89.01**	**87.96**

As shown in Table 1, the classification accuracy of FCN is higher than FC. CN-P achieves the best classification accuracy in this experiment, especially for points of the ground objects, such as low vegetation, fence/hedge, and shrub. For these objects, the F1 score of CN-P is 2.25%, 5.81% and 5.23% higher than that of FCN. From Table 1, we can see that the overall accuracy of CN-P is 87.96%, which indicates that the designed fully convolutional neural network is beneficial for effective feature extraction and can be used to classify the point cloud accurately.

4.3. Effectiveness of Graph-Cuts Optimization

In order to verify the effectiveness of the graph-cuts optimization algorithm, experiments with and without graph-cuts optimization on ISPRS benchmark dataset are performed. The F1 score of each class and overall accuracy of these two experiments are listed in Table 2. In Table 2, N-GC means the result of not using graph-cuts optimization; GC means the result of using graph-cuts optimization.

Table 2. F1 score of each class and overall accuracy (OA) with and without graph-cuts optimization on ISPRS benchmark dataset. Bold numbers show the highest values (%).

Method	**F1 Score**								**OA**
	Low_veg	**Imp_sur**	**Car**	**Fe/He**	**Roof**	**Facade**	**Shrub**	**Tree**	
N-GC	86.36	93.69	73.44	71.14	94.3	61.55	70.18	89.01	87.96
GC	**87.33**	**94.02**	**84.46**	**76.17**	**95.76**	**67.07**	**73.44**	**90.75**	**89.84**

As shown in Table 2, the overall accuracy with graph-cuts optimization is 1.88% higher than N-GC. The F1 scores of all class have been increased, which indicate that the graph-cuts is effective for refining the point cloud classification result. Note that there are noise points in the classification result of transfer learning, as can be seen from Figure 11a,c. In the areas marked by red ellipses in Figure 11a,c, there are scattered misclassification points of shrubs and trees, fences and trees, etc. These misclassification points have been identified as the correct category after graph-cuts optimization, which can be seen in Figure 11b,d.

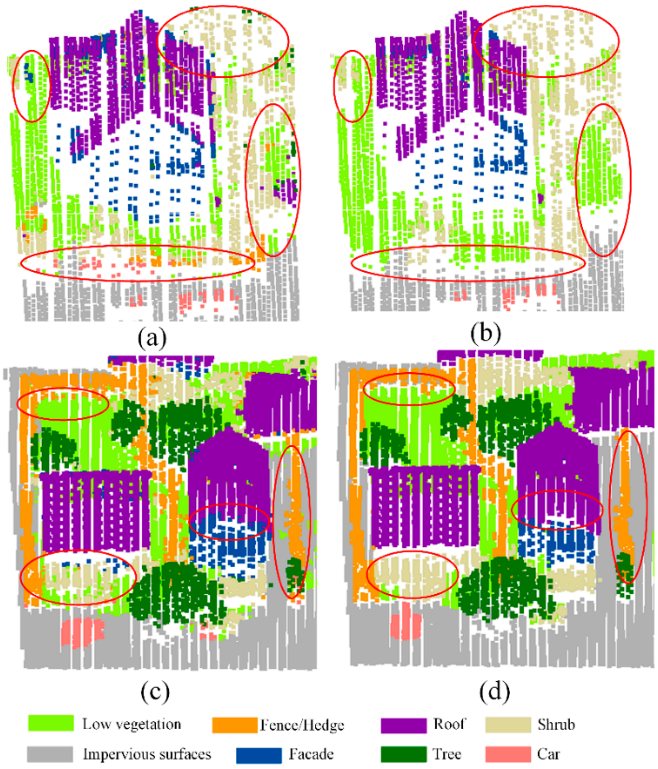

Figure 11. The classification results before and after graph-cuts optimization on ISPRS benchmark dataset: (**a**) the classification results before graph-cuts optimization of area1, (**b**) the classification results after graph-cuts optimization of area1, (**c**) the classification results before graph-cuts optimization of area2, (**d**) the classification results after graph-cuts optimization of area2.

4.4. Comparison with Other Methods

To evaluate the proposed method, we compared it with the top five classification methods submitted on the ISPRS website, including WhuY3 [39], RIT_1 [40], LUH [41], NANJ2 [42] and WhuY4 [43]. To further prove the advantage of our proposed method in this paper, we also compared our method with the recently published point cloud classification methods using deep learning, including D-FCN [16], A-XCRF [30], JGVNet [44], VPNet [45], DRN [31]. Per-class F1 score and overall accuracy of each submission, including ours, are shown in Table 3.

Table 3. F1 scores of each category and overall accuracy (OA) of different classification methods. Bold numbers show the highest values. The underlined numbers are the second-best values for each evaluation metric (%).

Method	F1 Score									OA
	Pow	Low_veg	Imp_sur	Car	Fe/He	Roof	Facade	Shrub	Tree	
WhuY3	37.1	81.4	90.1	63.4	23.9	93.4	47.5	39.9	78.0	82.3
RIT_1	37.5	77.9	91.5	73.4	18.0	94.0	49.3	45.9	82.5	81.6
LUH	59.6	77.5	91.1	73.1	34.0	94.2	56.3	46.6	83.1	81.6
WhuY4	42.5	82.7	91.4	74.7	53.7	94.3	53.1	47.9	82.8	84.9
NANJ2	62.0	**88.8**	91.2	66.7	40.7	93.6	42.6	55.9	82.6	85.2
D-FCN	70.4	80.2	91.4	78.1	37.0	93.0	60.5	46.0	79.4	82.2
A-XCRF	63.0	82.6	91.9	74.9	39.9	94.5	59.3	50.8	82.7	85.0
JGVNet	66.9	82.6	91.6	78.6	37.9	**95.8**	60.6	42.6	83.7	85.0
VPNet	**74.5**	82.1	91.6	83.4	45.9	93.3	60.6	51.1	82.3	84.0
DRN	-	83.3	92.5	52.4	62.5	95.2	62.8	64.9	88.7	86.8
Ours	-	87.3	**94.0**	**84.5**	**76.2**	**95.8**	**67.1**	**73.4**	**90.8**	**89.8**

As shown in Table 3, compared with all the contrast methods, our proposed method shows the best result in overall accuracy and per-class F1 scores except for powerline and vegetation. Note that the proposed method achieves substantially higher performance on the fence/hedge, facade and shrub categories, for which it outperforms the second-best method for the above categories by 13.7%, 4.3%, 8.5%, respectively. The overall accuracy and the average F1 score of our method is 3.0% and 8.3% higher than the overall second-best DRN method. This can be explained by that the multi-scale and multi-view features can be explored to characterize 3D point cloud comprehensively, and these deep features are applicable to the designed a fully convolutional neural network to derive accurate classification.

In addition to compare the classification accuracy, we also compare the number of training samples with other methods. In the proposed method, only 16,000 points are selected as training data However, RIT_1, D-FCN, A-XCRF, JGVNet and VPNet use all original training dataset as training data. The experimental result shows that the proposed method can be used to accurately classify the ALS point cloud with a small number of training samples.

5. Conclusions

In this paper, an ALS point cloud classification method is proposed by integrating an improved fully convolutional network into transfer learning. In this method, the multi-scale and multi-view feature maps are firstly extracted to characterize 3D point cloud comprehensively, and then these feature maps are fed into the pre-trained DenseNet201 model to obtain the deep features with good generalization. Subsequently, a fully convolutional neural network is designed to classify the ALS point cloud. Finally, a graph-cuts algorithm is used to refine the classification results. We test our method on the ISPRS benchmark dataset, the overall accuracy of 89.84% and the average F1 score of 83.62% are achieved, which outperforms other contrast classification method reported in the recent literature. The IEEE dataset experimental results show that, the overall accuracy is 98.31% and the average F1 score is 87.50%.

However, the procedure of feature maps generation is complex in the proposed method. It is because those algorithms in our programs are not integrated efficiently, and a lot of redundant data will be generated during the experiments. In addition, only three kinds of shallow features are extracted to derive deep features, which will affect the classification accuracy. In the future, the algorithms proposed in this paper will be integrated efficiently to reduce data redundancy, and the proposed method will be validated on more datasets.

Author Contributions: Conceptualization, X.L. and H.W.; methodology, X.L.; software, X.L.; validation, X.L., J.M. and P.T.; formal analysis, C.W. and Z.Z.; writing—original draft preparation, X.L.; writing—review and editing, H.W. and C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Henan Province (182300410115), Henan Polytechnic University Doctoral Fund (B2017—13), Key Science and Technology Research Projects of Henan Province (172102310350), the Fundamental Research Funds for the Universities of Henan Province (NSFRF170908).

Acknowledgments: The authors thank the anonymous reviewers for their helpful comments about this article. The ISPRS benchmark dataset was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF): http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kada, M. 3D building generalization based on half-space modeling. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2006**, *36*, 58–64.
2. Axelsson, P. Dem generation from laser scanner data using adaptive tin models. *Int. Arch. Photogramm. Remote Sens.* **2000**, *33*, 110–117.
3. Ene, L.T.; Næsset, E.; Gobakken, T.; Bollandsås, O.M.; Mauya, E.W.; Zahabu, E. Large-scale estimation of change in aboveground biomass in miombo woodlands using airborne laser scanning and national forest inventory data. *Remote Sens. Environ.* **2017**, *188*, 106–117. [CrossRef]

4. Shi, T.; Ma, X.; Han, G.; Xu, H.; Gong, W. Measurement of CO_2 rectifier effect during summer and winter using ground-based differential absorption lidar. *Atmos. Environ.* **2019**, *220*, 117097. [CrossRef]
5. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual classification of LiDAR data and building object detection in urban areas. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 152–165. [CrossRef]
6. Xu, S.; Vosselman, G.; Oude Elberink, S. Detection and classification of changes in buildings from airborne laser scanning data. *Remote Sens.* **2015**, *7*, 17051–17076. [CrossRef]
7. Vosselman, G. Point cloud segmentation for urban scene classification. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *1*, 257–262. [CrossRef]
8. Maltezos, E.; Ioannidis, C. Automatic detection of building points from LiDAR and dense image matching point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *3*, 33–40. [CrossRef]
9. Lin, X.; Zhang, J. Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments. *Remote Sens.* **2014**, *6*, 1294–1326. [CrossRef]
10. Zhang, J.Y.; Zhao, X.L.; Chen, Z. A review of Semantic Segmentation of Point Cloud Based on Deep Learning. *Las. Optoelect. Prog.* **2020**, *57*, 28–46. [CrossRef]
11. Zhang, J.; Lin, X.; Ning, X. SVM-Based Classification of Segmented Airborne LiDAR Point Clouds in Urban Areas. *Remote Sens.* **2013**, *5*, 3749–3775. [CrossRef]
12. Gevaert, C.M.; Persello, C.; Sliuzas, R.; Vosselman, G. Classification of informal settlements through the integration of 2D and 3D features extracted from UAV data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 317–324. [CrossRef]
13. Guo, L.; Chehata, N.; Mallet, C.; Boukir, S. Relevance of airborne LiDAR and multispectral image data for urban scene classification using Random Forests. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 56–66. [CrossRef]
14. Hackel, T.; Wegner, J.D.; Schindler, K. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 177–184. [CrossRef]
15. Ni, H.; Lin, X.; Zhang, J. Classification of ALS point cloud with improved point cloud segmentation and random forests. *Remote Sens.* **2017**, *9*, 288. [CrossRef]
16. Wen, C.; Yang, L.; Li, X.; Peng, L.; Chi, T. Directionally constrained fully convolutional neural network for airborne LiDAR point cloud classification. *ISPRS J. Photogramm.* **2020**, *162*, 50–62. [CrossRef]
17. Wolf, D.; Prankl, J.; Vincze, M. Fast semantic segmentation of 3D point clouds using a dense CRF with learned parameters. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 4867–4873.
18. Lu, Y.; Rasmussen, C. Simplified Markov random fields for efficient semantic labeling of 3D point clouds. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 2690–2697.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
20. Feng, F.; Wang, S.; Wang, C.; Zhang, J. Learning Deep Hierarchical Spatial-Spectral Features for Hyperspectral Image Classification Based on Residual 3D-2D CNN. *Sensors* **2019**, *19*, 5276. [CrossRef]
21. Das, N.; Hussain, E.; Mahanta, L.B. Automated classification of cells into multiple classes in epithelial tissue of oral squamous cell carcinoma using transfer learning and convolutional neural network. *Neural Netw.* **2020**, *128*, 47–60. [CrossRef]
22. Hu, X.; Yuan, Y. Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote Sens.* **2016**, *8*, 730. [CrossRef]
23. Li, Y.; Tong, G.; Li, X.; Zhang, L.; Peng, H. MVF-CNN: Fusion of Multilevel Features for Large-Scale Point Cloud Classification. *IEEE Access* **2019**, *7*, 46522–46537. [CrossRef]
24. Liu, Z.; Tang, H.; Lin, Y.; Han, S. Point-Voxel CNN for Efficient 3D Deep Learning. In Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS), Vancouve, BC, Canada, 12–14 December 2019.
25. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]

26. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the 2017 Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5099–5108.
27. Yizhak, B.; Michael, L.; Anath, F. 3DmFV: Three-dimensional point cloud classification in real-time using convolutional neural networks. *IEEE Robot. Autom. Lett.* **2018**, *25*, 3145–3152. [CrossRef]
28. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN Convolution on X-transformed points. In Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS), Montreal, QU, Canada, 2–8 December 2018.
29. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6410–6419.
30. Arief, H.A.A.; Indahl, U.G.; Strand, G.; Tveite, H. Addressing overfitting on point cloud classification using Atrous XCRF. *ISPRS J. Photogramm.* **2019**, *155*, 90–101. [CrossRef]
31. Zhao, C.; Guo, H.T.; Lu, J.; Yu, D.H.; Li, D.J.; Chen, X.W. ALS Point Cloud Classification with Small Training Dataset Based on Transfer Learning. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1406–1410. [CrossRef]
32. Zhang, W.M.; Qi, J.B.; Wang, P.; Wang, H.T.; Xie, D.H.; Wang, X.Y.; Yan, G.J. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sens.* **2016**, *8*, 501. [CrossRef]
33. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [CrossRef]
34. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Las Vegas, NE, USA, 26 June–1 July 2016. [CrossRef]
35. Kornblith, S.; Shlens, J.; Le, Q.V. Do Better imagenet models transfer better? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Angeles, CA, USA, 16–20 June 2019; pp. 2661–2671.
36. Lin, M.; Chen, Q.; Yan, S. Network in Network. In Proceedings of the IEEE International Conference on Learning Representations (ICLR), Scottsdale, AR, USA, 2–4 May 2013.
37. Boykov, Y.; Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1124–1137. [CrossRef]
38. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the IEEE International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
39. Yang, Z.S.; Jiang, W.S.; Xu, B.; Zhu, Q.S.; Jiang, S.; Huang, W. A Convolutional Neural Network-Based 3D Semantic Labeling Method for ALS Point Clouds. *Remote Sens.* **2017**, *9*, 936. [CrossRef]
40. Yousefhussien, M.; Kelbe, D.J.; Ientilucci, E.J.; Salvaggio, C. A multi-scale fully convolutional network for semantic labeling of 3D point clouds. *ISPRS J. Photogramm.* **2018**, *143*, 191–204. [CrossRef]
41. Niemeyer, J.; Rottensteiner, F.; Sörgel, U.; Heipke, C. Hierarchical higher order crf for the classification of airborne lidar point clouds in urban areas. *Remote Sense.* **2016**, *41*, 655–662. [CrossRef]
42. Zhao, R.; Pang, M.; Wang, J. Classifying airborne LiDAR point clouds via deep features learned by a multi-scale convolutional neural network. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 1–20. [CrossRef]
43. Yang, Z.; Tan, B.; Pei, H.; Jiang, W.S. Segmentation and Multi-Scale Convolutional Neural Network-Based Classification of Airborne Laser Scanner Data. *Sensors* **2018**, *18*, 3347. [CrossRef] [PubMed]
44. Zhang, J.; Hu, X.; Dai, H. A Graph-Voxel Joint Convolution Neural Network for ALS Point Cloud Segmentation. *IEEE Access* **2020**, *8*, 139781–139791. [CrossRef]
45. Qin, N.; Hu, X.; Wang, P.; Shan, J.; Li, Y. Semantic Labeling of ALS Point Cloud via Learning Voxel and Pixel Representations. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 859–863. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.