sensors

MDPI

Article

Point Cloud Semantic Segmentation Network Based on Multi-Scale Feature Fusion

Jing Du [1], Zuning Jiang [1], Shangfeng Huang [1], Zongyue Wang [1], Jinhe Su [1], Songjian Su [2], Yundong Wu [1,3] and Guorong Cai [1,3,*]

[1] Computer Engineering College, Jimei University, Xiamen 361021, China; jingdu@jmu.edu.cn (J.D.); jzn201721121073@gmail.com (Z.J.); shangfenghuang@jmu.edu.cn (S.H.); wangzongyue@jmu.edu.cn (Z.W.); sujh@jmu.edu.cn (J.S.); yundongwu@jmu.edu.cn (Y.W.)
[2] Ropeok Technology Group Co., Ltd., Xiamen 361021, China; songjian.su@ropeok.com
[3] Fujian Collaborative Innovation Center for Big Data Applications in Governments, Fuzhou 350003, China
* Correspondence: guorongcai.jmu@gmail.com; Tel.: +86-18959289198

Abstract: The semantic segmentation of small objects in point clouds is currently one of the most demanding tasks in photogrammetry and remote sensing applications. Multi-resolution feature extraction and fusion can significantly enhance the ability of object classification and segmentation, so it is widely used in the image field. For this motivation, we propose a point cloud semantic segmentation network based on multi-scale feature fusion (MSSCN) to aggregate the feature of a point cloud with different densities and improve the performance of semantic segmentation. In our method, random downsampling is first applied to obtain point clouds of different densities. A Spatial Aggregation Net (SAN) is then employed as the backbone network to extract local features from these point clouds, followed by concatenation of the extracted feature descriptors at different scales. Finally, a loss function is used to combine the different semantic information from point clouds of different densities for network optimization. Experiments were conducted on the S3DIS and ScanNet datasets, and our MSSCN achieved accuracies of 89.80% and 86.3%, respectively, on these datasets. Our method showed better performance than the recent methods PointNet, PointNet++, PointCNN, PointSIFT, and SAN.

Keywords: LIDAR point cloud; semantic segmentation; feature fusion; deep learning; computer vision

Citation: Du, J.; Jiang, Z.; Huang, S.; Wang, Z.; Su, J.; Su, S.; Wu, Y.; Ca, G. Point Cloud Semantic Segmentation Network Based on Multi-Scale Feature Fusion. *Sensors* **2021**, *21*, 1625. https://doi.org/10.3390/s21051625

Academic Editor: Ruofei Zhong

Received: 19 January 2021
Accepted: 23 February 2021
Published: 26 February 2021

1. Introduction

Deep learning algorithms have achieved significant success in many remote sensing image analysis tasks, including object detection, semantic segmentation and classification. On the one hand, the purpose of semantic segmentation is to assign a land cover label to each pixel in an image. Facilitated by deep convolutional neural networks (CNNs), especially end-to-end fully convolutional networks (FCN) [1], interest in the semantic segmentation of remote sensing images has increased in recent years. Furthermore, semantic segmentation focusing on the detection of small objects in remote sensing images [2–6] and in point clouds covering global navigation satellite system (GNSS) indoor and underground environments [7] has become a very attractive research topic.

In the research of a 3D point cloud, semantic segmentation is a hot research topic in the field of autonomous driving and robot localization. Segmentation algorithms that take input in the form of point clouds can be roughly divided into three categories: multiview-based [8–13], voxel-based [14–18], and raw-point-cloud-based algorithms [19–27]. The transformation of point clouds to a regular 3D voxel or images usually leads to serious loss of geometric information and increases the calculation complexity. Therefore, algorithms based on an original point cloud have become a hot research field recently. The original point cloud contains rich geometric and semantic information, so it is easier for algorithms to realize scene perception. Typical algorithms include PointNet [19], PointNet++ [20],

PointCNN [28], and PointSIFT [29]. Although the point-based deep learning models have made remarkable progress in the past three years, they still face difficulties related to the avoidance of information loss in the process of down-sampling. Objects with fewer points will keep fewer points in the final sampled points. Different densities of classes will increase the difficulty of segmentation.

In the field of image, multi-resolution feature extraction and fusion [30] can significantly enhance the ability of object classification and segmentation. Motivated by this phenomenon, we propose a point cloud semantic segmentation network based on multi-scale feature fusion, which can aggregate features of different densities and improve the performance of semantic segmentation. Firstly, point clouds of different densities are obtained by changing the sampling ratio. Low-density point clouds in the proposed network are suitable for extracting global shape features of a target, while high-density point clouds are suitable for extracting detail from local features. Then, features are extracted from point clouds of different scales. Finally, a new feature set is obtained from the extracted features by applying the feature fusion operation. To sum up, there are three main contributions in our work:

Firstly, we propose a multi-scale feature fusion architecture that is suitable for point clouds. The multi-scale point cloud is obtained via stepwise downsampling from the same original point cloud. We set different sampling ratios for different datasets and achieve promising segmentation accuracy compared to state-of-the-art methods on both the ScanNet and Stanford Large-Scale 3D Indoor Spaces (S3DIS) datasets.

Secondly, our MSSCN fuses point features extracted from different network levels through direct mapping and concatenation. This feature fusion method not only allows the advantages of the feature representations extracted at each level to be combined, but also avoids error propagation at each level.

Finally, we design a loss function for MSSCN, which is used to train the network by combining losses at different scales. Experimental results demonstrate that each component of the loss function influences the final segmentation accuracy of MSSCN.

The rest of the paper is organized as follows. In the Section 2, the literature on point cloud segmentation and classification is reviewed. Section 3 introduces the proposed deep learning network structure MSSCN in detail. In the Section 4, the experimental setup is introduced and the results are discussed. The Section 5 is a summary of the paper.

2. Related Work

Point clouds do not have a regular structure, whereas the input data for traditional CNNs [31–34] must have a regular format; consequently, traditional CNNs are not suitable for extracting the features of a point cloud. Previously, researchers have been working to transform 3D point clouds into regular formats that are similar to images or voxels. For example, in the multi-view based methods [35], the original point cloud is projected into the image plane based on its depth or intensity values, and the projection views are generated from a virtual camera posture. A typical example of this approach is MVCNN [8]. Since 2018, projection-based methods have been widely concerned. For instance, the Pointwise Rotation-Invariant Network [36] framework was proposed to achieve rotation invariance in point clouds. RotationNet [37] is effective for real scenes, because it only uses a part of the original multi-view images to perform the inference process. However, due to the loss of local geometry during the compression of 3D data to 2D data, methods based on the projection of point clouds still face some limitations. Ref. [38] uses multiple clues to integrate range and color content, in order to retain local geometric information. In this context, Ref. [39] maps an input point cloud to a scanning pattern grid. Virtual MVFusion [40] provided additional channels to render virtual views, which exceeds the limitations of existing RGB-D sensors. At the same time, Virtual MVFusion designed the backside culling scheme and multi-scale view sensing sampling. Therefore, the occlusion, narrow view and scale invariance problems that plagued most previous multi-view fusion methods were improved.

Voxel, as small units of points set in 3D space, can be used to divide a point cloud into a regular 3D subspace. Most voxel-based deep learning architectures are inspired by 2D CNNs. Generally, 0–1 discrete values are used to confirm whether there are any points in a specific voxel. As a typical method of this type, 3D ShapeNet [15] employed binary voxels for three-dimensional filtering. However, this scheme always leads to an increase in computational complexity. Therefore, researchers are attempting to improve the network structure of voxel CNNs, as in [41,42].

Recently, researchers have paid increasing attention to semantic segmentation networks that take the original point cloud as the initial input. In this case, the input vector for the deep neural network can be composed of coordinates or a combination of coordinates, intensity, and color information. For algorithms based on raw point clouds, it is necessary to solve the problem of achieving invariance in the order of input points. The representative method for settling this conundrum is PointNet [19]. This networks use global feature pooling to make the output vector invariant to the sequence of the input point. However, it is difficult to extract local geometric features for each point since max-pooling layers can be applied only to all points. To effectively overcome this challenge, PointNet++ [20] used a multilevel network structure for the extraction of local features. That being said, the max-pooling operation is also adopted in PointNet++. As a consequence, the network only retains the maximum feature feedback from global and local regions, resulting in a loss of useful geometric information that adversely affects the segmentation task. PointCNN [28] used different levels of representative points to realize the feature extraction schemes proceeding from local regions to the global point cloud. However, this method may introduce a new problem. In most cases, the distribution of the point cloud is uneven, which will lead the selected representative points to gather in a small space. Consequently, after several convolution operations, the reception range will be limited. To handle this problem, PointSIFT [29] selected points adjacent to the representative points in a specified direction. The purpose is to acquire a complete description of the spatial structure features around key points. The disadvantage of PointSIFT is the relatively high time complexity. Recently, [43] designed the novel PointConv operation to achieve network expansion and improve segmentation performance. Ref. [44] proposed a multi-directional convolutional network—called a Spatial Aggregation Network (SAN)—which can utilize local spatial structure information to achieve relatively high efficiency and accuracy.

However, due to the complexity of the point cloud distribution, the features of the chosen alternative points may not be representative of the original features. In this situation, the geometric information for each point will be ignored, which may lead to the loss of local feature information. To extract local geometric and global features synchronously, the authors of TGNet [45] proposed a novel convolution filter that extracts point features in a hierarchical and multiscale manner. Experimental results showed that this strategy effectively combines features from different scales and improves the performance of local region segmentation. The authors of DGCNN [46] proposed an innovative edge convolution that can extract the geometric features of local neighborhoods while maintaining permutation invariance. However, this edge-based convolution obtains neighborhood points based only on distance, which may still lead to local geometric information loss. GeoCNN [47] extracted features based on the angle aggregation between edge vectors and orthogonal bases, so as to keep the geometric structure in the whole feature extraction process. However, it is worth noting that GeoCNN needs to recalculate the K nearest neighbors for all points in each stage, resulting in higher complexity. KPConv [48] can be expanded to deformable convolutions by adapting the kernels to local geometries. Any number of kernel points can be used, giving KPConv more flexibility than grid convolutions. Furthermore, these locations are spatially continuous and can be learned by the network. A new convolution operator learned from relationships in RS-CNN [49] is called relational shape convolution, which can encode the geometric relationship of points and expand the configuration of regular grid CNNs to achieve context-aware learning of point clouds. FPConv [50] proposed the surface style convolution operator. The operator disperses the convolution

weight of each point along the local surface, so it is robust to input data. Finally, points are projected onto the 2D grid by predicting projection weights, and regular 2D convolution can be used for feature learning. However, CNNs may not correctly solve the problem of non-Euclidean data. Therefore, graph convolutional networks (GCNs) were developed to overcome this challenge by creating graphs representing non-Euclidean data. With the help of techniques for increasing the depth of CNNs, DeepGCNs [27] were developed using not only residual/dense connections but also dilated convolutions. Residual/dense connections can solve the problem of gradient disappearance caused by an increase of network depth. By expanding the convolution kernel to increase the receptive field without increasing the number of parameters, the dilated convolutions help to solve the problem of spatial information loss caused by pooling. Finally, a 56-layer GCN was constructed in this way, which offered significantly improved performance in the semantic segmentation of point cloud. Grid-GCN [51] proposed Coverage Aware Grid Query (CAGQ), which samples representative center points and queries adjacent points. CAGQ implements data structuring and makes full use of grid space efficiency, thus increasing space coverage and reducing theoretical time complexity. CAGQ is up to 50 times faster than the most popular sampling methods such as farthest point sampling and spherical query. A Graphic Convolution Module Grid Context Aggregation (GCA) is proposed to integrate context features and coverage information into computation. SPVNAS [52] proposed Sparse Point-Voxel Convolution (SPVConv), which is equipped with a high-resolution point-based branch for sparse convolution. It then introduced the first 3D Neural Architecture Search (3D-NAS) for 3D scene understanding, a framework that searches for the optimal network structure within a given resource constraint. JSENet [53] introduced semantic edge detection into semantic segmentation. Semantic edge detection provides detailed edge location information and can generate accurate edges. At the same time, dual semantic edge loss is proposed to improve the segmentation effect of the edge position.

Algorithms based on raw point clouds [20,28,29,43] typically require several downsampling operations. In this scheme, objects with fewer points will keep fewer points at the final sampled points. Different density of categories will increase the difficulty of segmentation [27,45–49].

3. The Proposed Approach

The network structure of MSSCN is shown in Figure 1. First, downsampling is applied to obtain the point cloud. Then, we extract features from the point clouds at each scale using the network architecture proposed in our previous work, SAN. Finally, feature fusion and optimization are performed on the extracted features.

3.1. Multiscale Point Feature Extraction

To make our method invariant to scale changes, a multi-scale point feature extraction method based on different densities is proposed. We find that features of low-density point clouds are suitable to represent global shape features, while high-density point clouds are appropriate to describe detailed local features. According to the characteristics of the point clouds, features associated with different densities are complementary. To construct the multi-scale feature fusion network, we perform random downsampling operations on the input data. Specifically, three scales are used to construct point clouds via downsampling. Thus, we obtain two point clouds P_1 and P_2 of different densities, representing the point clouds after the first and second downsampling processes, respectively. We record the position of each sampled point in the two down-sampling point clouds, which can be used for feature fusion later. The selection of the sampling ratios for datasets with different characteristics is discussed in Section 4.1.

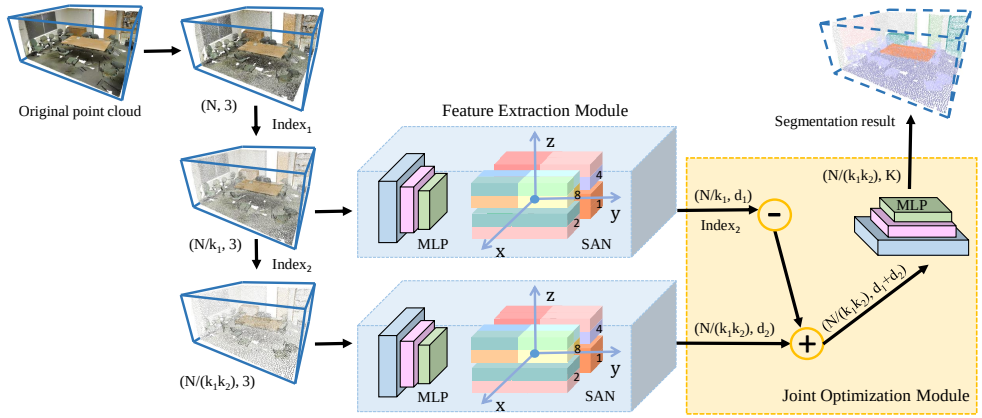


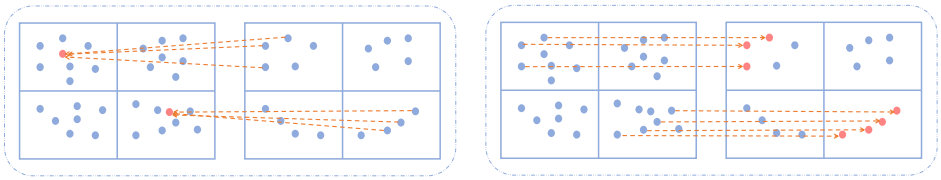
Figure 1. Illustration of the proposed multi-scale feature fusion network (MSSCN). First, downsampling is performed on the original point cloud with sampling proportions of k_1 and k_2. The chosen points are stored in $Index_1$ and $Index_2$ for the first and second downsampling processes, respectively. Then, feature extraction is performed using a Spatial Aggregation Net (SAN) backbone, where d_1 and d_2 are the dimensionalities of the features for each downsampled point cloud. Finally, feature fusion is performed to obtain a relevant set of features, where '$-$' indicates the deletion of descriptors that do not exist in $Index_1$ according to $Index_2$ and '+' indicates feature fusion. Based on the extracted features, a multilayer perceptron (MLP) is used to obtain the score of each point for each of the K object categories.

Then, we extract features of each point from the multi-scale point clouds. Since SAN achieves an effective balance between efficiency and accuracy, it is deemed a good backbone network for abstracting features from P_1 and P_2. In particular, SAN uses a hierarchical structure that combines small area features into semantic features. It contains not only several Directional Spatial Aggregation (DSA) components but also some feature unencoding (FP) modules. The DSA module is the core module of SAN. It is divided into three steps for extracting the features of sampling points. First, point downsampling is performed using the farthest point sampling (FPS) [54]. Secondly, the neighboring points around each sampling point are captured by octant search [29]. Finally, the multi-directional convolution operation is performed on the sampling points. This convolution is followed by max pooling to aggregate features from different directions. The point cloud P_1 contains N/k_1 points. The SAN network extracts features for each point in P_1, and the feature dimension of each point is d_1. The point cloud P_2 contains $N/(k_1k_2)$ points, and the SAN network extracts d_2 dimension features for each point in P_2.

3.2. Feature Fusion and Loss Function

Many methods can be used to fuse features from point clouds of different densities, such as descriptor interpolation based on the distances between adjacent points, as shown in Figure 2a. In this method, the information of one point is combined with information from its neighboring points, and weighted according to distance. However, two points belonging to different classes may be aggregated using this method, which leads to unstable segmentation. Figure 2b shows a direct mapping method for feature fusion, which aims to alleviate this problem. As shown in Equations (1) and (2). F_1 and F_2 represent the feature descriptors of the points of P_1 and P_2, respectively. $Index_1$ is the index set of the points in P_1. $Index_2$ is an index set of points in P_2. Feature fusion combines the point feature F_1 of P_1 with the feature F_2 from P_2. The process of feature fusion is divided into two steps. First, we delete the points not in $Index_2$ from P_1 according to $Index_2$, so that P_1 and P_2 retain the same points. At this time, the indexes of the points of P_1 and P_2 are both $Index_2$. But because SAN extracts features from point clouds of different densities, the features are not the same. Therefore, the feature descriptor corresponding to the point of P_1 is re-expressed as $F_2^{'}$. In the second step, we perform feature fusion through the consistency of the point index. By concating the point features with the same index in P_1 and P_2, we obtain a set of

features with $d_1 + d_2$ dimensions, named F_3. Finally, the score of each point in K categories is obtained through the Multilayer Perception (MLP) operation.



(a) (b)

Figure 2. Feature fusion process: (**a**) feature interpolation based on distance and (**b**) direct mapping.

$$F_2^{'} = F_1[Index_1 - Index_2], F_2^{'} \in R^{d_1} \tag{1}$$

$$F_3 = F_2 \oplus F_2^{'}, F_3 \in R^{d_1+d_2} \tag{2}$$

Because we maintain the point index correlations in the process of downsampling, feature fusion via direct mapping will not lead to redundant calculation. However, the direct mapping scheme has some disadvantages, one of which is that not all points in the high-density set have multi-scale descriptors. To work around this problem, the sampling rate was set to be greater than or equal to $1/2$ in our experiments. Therefore, the lowest density of P_1 was half the original density, as shown in Figure 3b, while the lowest density of P_2 was $1/4$ the original density, as illustrated in Figure 3c. We applied SAN extraction to the point clouds at each scale. The corresponding segmentation results from redlevel 1 and level 2 are presented in Figure 3b,c, respectively. Although the densities of the point clouds are different, the segmentation results are stable. Figure 3d shows the segmentation results of MSSCN.

We also present a loss function that incorporates a different loss for each density to increase the robustness of MSSCN for multiscale point cloud scenes. The loss function is shown in Equation (3). Equations (4)–(6) are annotations for each variable in the loss function. The loss function has four components. $\alpha_1, \alpha_2, \alpha_3$, and α_4 are parameters that determine the trade-off among the four components. The first and second components, respectively, use the cross-entropy loss L_{seg} to penalize the wrong segment labels in the level 1 and level 2 predictions. L_{seg} denotes the cross-entropy classification loss. The third component punishes points with incorrect segmentation labels in the final predictions. pre_1, pre_2, and pre_3 represent prediction results. $label_1$, $label_2$, and $label_3$ represent ground truth labels. In an ideal environment, the predictions obtained for P_2 using F_2 and $F_2^{'}$ should be consistent. Therefore, the fourth component is used to enhance the consistency of the predictions using F_2 and $F_2^{'}$. $Index_1$ represents the indexes of points downsampled from the primitive input data. $Index_2$ denotes the indexes of points obtained via the second downsampling process from P_1. N is the number of points in the original point cloud. S_1 is the ratio of the first down-sampling, and S_2 is the ratio of the second down-sampling. The loss function is shown as follows:

$$\begin{aligned} Loss = {} & \alpha_1 L_{seg}(pre_1, label_1) + \alpha_2 L_{seg}(pre_2, label_2) + \\ & \alpha_3 L_{seg}(pre, label_2) + \alpha_4 \frac{1}{N * S_1 * S_2} \sum (0.5 \\ & * (pre_2! = label_2) + 0.5 * (pre_2^{'}! = label_2)) \end{aligned} \tag{3}$$

$$pre_{2[i]}^{'} = pre_{1[Index_2[i]]}, i = 0, ..., N * S_1 * S_2 \tag{4}$$

$$label_{1[i]} = label_{[Index_1[i]]}, i = 0, ..., N * S_1 \tag{5}$$

$$label_{2[i]} = label_{1[Index_2[i]]}, i = 0, ..., N * S_1 * S_2 \tag{6}$$

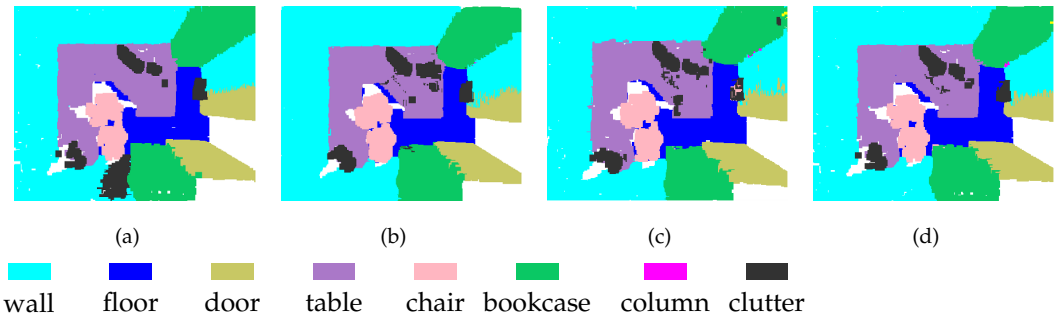


Figure 3. Segmentation results of point clouds with different densities: (**a**) ground truth; (**b**) point cloud at Level-1 (P_1), where the sampling proportion is 1/2; (**c**) point cloud at Level-2 (P_2), where the sampling proportion is 1/4; and (**d**) MSSCN.

3.3. Algorithm Summary

The pipeline of our proposed MSSCN method is shown in Algorithm 1. The input to the network is a point cloud scene, where each point is associated with three-dimensional coordinate information (x, y, z), denoted by P. The other information on the points in P is recorded as the feature set F of those points. N is the number of points in the point cloud. The output of the network is the score of each point for each class. First, we perform downsampling on the input data. Specifically, $N \times S_1$ indexes, denoted by $Index_1$, are randomly generated in the range $[0, N)$ without duplicates. Similarly, $N \times S_1 \times S_2$ indexes are randomly generated in the range $[0, N \times S_1)$, which are denoted by $Index_2$. Here, S_1 and S_2 are the proportions used in the first and second downsampling processes, respectively. In the second step, using $Index_1$ as the indexes of the points in P, a new point cloud P_1 is generated. Similarly, $Index_2$ is used as the indexes to generate a new point cloud P_2. The features of these point clouds of different densities are then extracted by the SAN feature extractor in the third step.

SAN uses the FPS algorithm to obtain a new point cloud P_{new}. For each point P_i in P_{new}, the adjacent 3D space centered on P_i is divided into eight octants. SAN selects the $\frac{k}{8}$ nearest points as the representative points in each octant. In the experiments described in the following section, to ensure that there would be four points in each direction. We set the initial value of k to 32. Then, a feature vector fusion operation is employed for all points in the same direction, using a convolution operator to fuse the feature vectors of the four points into a single vector. Next, we use 2×1 convolution operators to aggregate the points from all eight directions into only four directions. The convolved features representing the spatial structure information of each point are obtained through this multi-directional convolution. An MLP is used to transform the new features, and finally, the seven features are grouped using the max-pooling operation to obtain a new feature set F_{new} corresponding to P_{new}. Then, we repeat the above operation three times to obtain P_{new1} and F_{new1}, P_{new2} and F_{new2}, P_{new3} and F_{new3}, and we weight the corresponding features based on distance. Finally, the newly acquired features are merged with the original features. In detail, the feature set F_{new3} of P_{new3} is mapped to P_{new2}, the feature set F_{new2} of P_{new2} is mapped to P_{new1}, the feature set F_{new1} of P_{new1} is mapped to P_{new}, and the feature set F_{new} of P_{new} is mapped to P_1. Finally, the feature set F_1 corresponding to P_1 is obtained. The feature set F_2 corresponding to P_2 is also obtained in this manner.

Then, feature fusion is performed as shown in steps 4 and 5. We use $Index_2$ to obtain the points in P_1, then obtain the new feature set $F_2^{'}$ of these points and combine the feature set $F_2^{'}$ of P_2 with the new feature set $F_2^{'}$ to obtain feature set F_3 for P_2. Finally, the MLP operation is performed on F_3 to classify every point in P_2, and the result is recorded as P_{seg}. The proposed MSSCN presents many advantages. At first, MSSCN extracts features using SAN. In addition, with the development of new algorithms based on raw point clouds, MSSCN can be further improved. Second, our MSSCN can extract information from different density scales and use the resulting fused features to improve the segmentation results.

Algorithm 1 Multi-Scale Feature Fusion Semantic Segmentation Network

Input: P (N,3)

Output: P_{seg} (N × S_1 × S_2,k)

$Index_1$ = random(N, N × S_1); $Index_2$ = random(N × S_1, N × S_1 × S_2);

P_1=P[$Index_1$]; P_2=P_1[$Index_2$];

F_1=**SAN**(P_1,None); F_2=**SAN**(P_2,None);

$F_2^{'}$ = F_1[$Index_2$];

F_3 = [F_2 , $F_2^{'}$];

$P_{seg} = MLP(F_3)$;

function SAN(P, F):

P_0=P, F_0=F, N = [1024, 256, 64, 32]; // N is the number of down-sampling

for i = 4 to 1 **do**

Index = FPS(P_{i-1}, N_i);

P_i = P_{i-1}[Index], F_i = F_{i-1}[Index]

F_i = Octant_sampling(P_{i-1}, P_i, F_{i-1}, 32);

F_i = Multi_Directional_Conv(Fi);

for i = 1 to 4 **do**

F_interpolate = three_interpolate(F_i, P_i, P_{i-1});

F_{i-1} = [F_interpolate, F_{i-1}];

F_{i-1} = MLP(F_{i-1});

return F_0

4. Results and Discussion

4.1. Experimental Setup

We employed two different datasets to assess the properties of MSSCN: the S3DIS dataset [55] and the ScanNet dataset [56]. The S3DIS dataset is composed of six folders of point cloud data from three different construction projects, including 271 rooms. S3DIS contains 12 semantic classes, including structural elements (ceilings, doors, walls, beams, columns, wooden boards, windows, and floors) and furniture (sofas, bookcases, chairs, and tables). These classes are more fine-grained and challenging than those in many indoor semantic segmentation datasets. Each point is associated with not only XYZ coordinates but also RGB colors, and there is a corresponding space-normalized coordinate for the room where each point is located. Due to this challenge, we chose S3DIS as one of our experimental datasets. In the experiments described below, 16,384 points were randomly selected from each sample. For the first level of the network framework (Level-1), 8192

points were used as input, and for the second level (Level-2), 4096 points were used. The output consists of the classification results for these 4096 points.

ScanNet is a point cloud scene dataset for semantic segmentation that contains 1513 scan scenes and a total of 21 class objects. There are 1201 scenes in the training set, and the remaining 312 scenes are used for testing. We randomly sampled 16,384 points from each sample. For Level-1 of the network framework, 16,384 points were used as input, and for Level-2, 8192 points were used. The output consists of the classification results for these 8192 points.

To ensure the best possible performance, all training samples were divided into two parts. The first part was used to train the SAN feature extractor, and the second part was used to train our proposed MSSCN. Since the tensorflow framework has very efficient computational efficiency, we used the tensorflow framework for encoding. All experiments were run on the Ubuntu operating system. All experiments were performed on an NVIDIA 1080 Ti GPU with 11 GB of memory. All components of the framework were trained by the Adam optimizer. On S3DIS and ScanNet datasets, we trained the models for 400 and 500 epochs, respectively.

To select the most advantageous network structure, several preliminary experiments were performed on the S3DIS and ScanNet datasets. Finally, the SAN model was adopted to extract features at each level of the network framework. At the same time, the network was optimized by adjusting the loss.

4.2. Results on S3DIS

We conducted a comprehensive comparative study on PointNet, PointNet++, PointSIFT, and SAN on S3DIS to assess the properties of MSSCN—the results of which are illustrated in Table 1. The S3DIS dataset is a point cloud dataset, including XYZ coordinate information, RGB color information and label information. In order to verify the robustness of our method, we conducted two versions of experiments on the S3DIS dataset: (a) XYZ coordinate information, RGB color information and label information as the network input, (b) XYZ coordinate information and label information as the network input, RGB color information was not input. It is worth noting that whether RGB information is used or not, the accuracy of MSSCN in the Level-1 and Level-2 is higher than that of the above-mentioned point-based models. Moreover, the accuracy of MSSCN is improved after feature fusion, which shows that our MSSCN framework performs better in terms of feature extraction than the existing models. The accuracy of MSSCN is 87.41% when RGB information is not included, and 89.80% when it is included. The experimental results corroborate the claim that feature fusion can further improve the precision of semantic segmentation.

Table 1. Comparison of the accuracy of different methods on the S3DIS dataset [55].

Method	Accuracy without RGB (%)	Accuracy with RGB (%)
PointNet [19]	70.46	78.62
PointNet++ [20]	75.66	82.23
PointSIFT [29]	76.61	82.33
SPG [24]	-	85.50
SAN [44]	78.39	82.93
DGCNN [46]	-	84.10
ShellNet [57]	-	87.10
RandLA-Net [58]	-	88.00
Level-1	84.64	88.51
Level-2	84.66	87.46
MSSCN	**87.41**	**89.80**

To enable a qualitative assessment of the methods, we present some typical segmentation results in Figures 4 and 5. The scenes include tables, chairs, boards, windows, doors, bookcases, walls, and columns. All methods have achieved satisfactory results

for tables and chairs because these objects exhibit different spatial structures and shapes. However, some areas of boards, windows, doors, bookcases, and columns are very similar to the structure of the walls, which makes these objects difficult to separate. The previous methods have difficulty separating these regions completely, whereas our method shows higher performance in these regions. These results show that because of its multi-scale processing ability, MSSCN can achieve better segmentation performance for objects with similar structures and shapes.

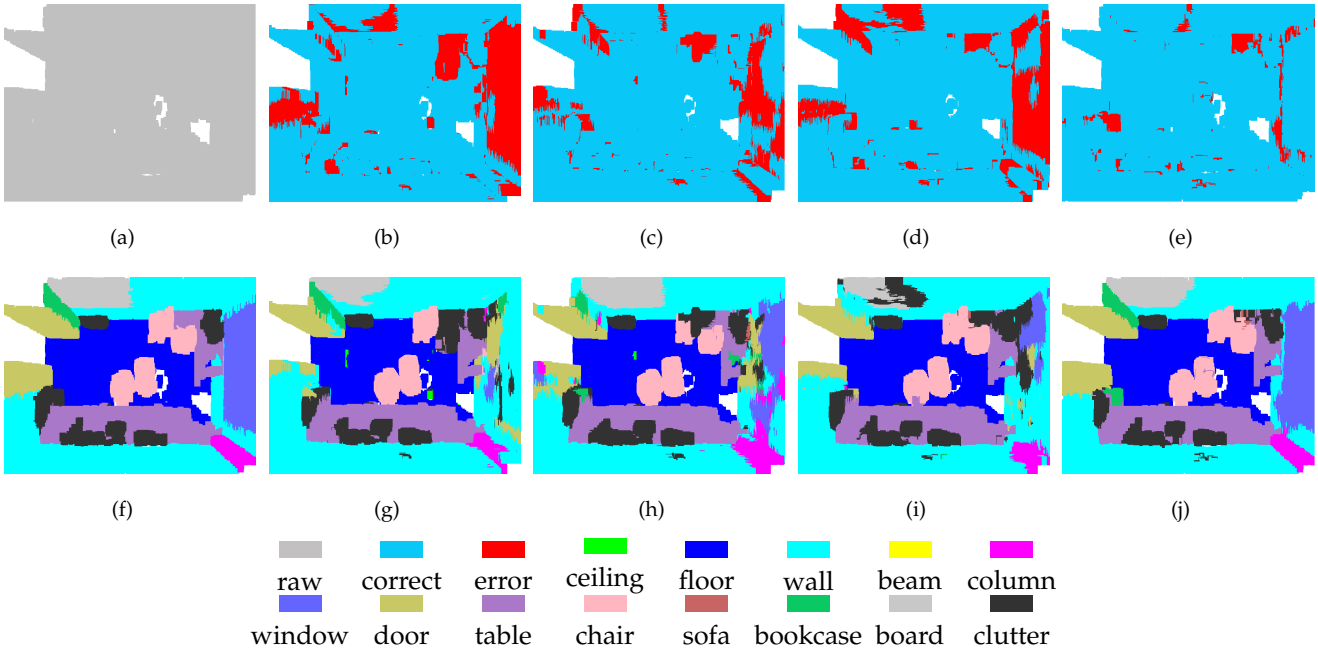


Figure 4. Segmentation results on the S3DIS-1 dataset: (**a**) input, (**f**) ground truth, (**b**,**g**) PointNet++ , (**c**,**h**) PointSIFT, (**d**,**i**) SAN, and (**e**,**j**) MSSCN.

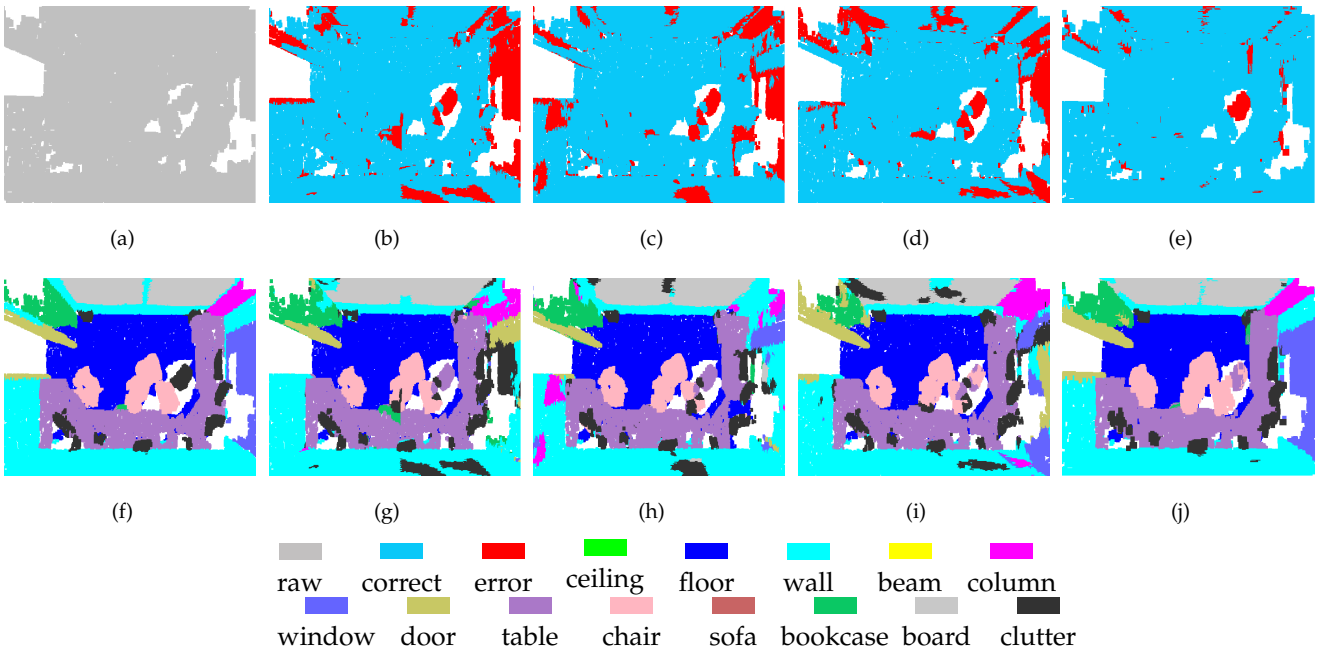


Figure 5. Segmentation results on the S3DIS-2 dataset: (**a**) input, (**f**) ground truth, (**b**,**g**) PointNet++, (**c**,**h**) PointSIFT, (**d**,**i**) SAN, and (**e**,**j**) MSSCN.

As illustrated in Table 2, MSSCN shows good performance for the semantic segmentation of each category in S3DIS. Good results can be obtained not only for objects which are easy to separate (such as ceilings and floors), but also for objects which are difficult to separate (such as beams and columns). Table 2 also shows the effectiveness of feature fusion. Feature fusion can improve the accuracy for most objects (e.g., columns, windows, doors, chairs, and bookcases). These results show that feature fusion can not only combine the advantages of Level-1 and Level-2 feature representation, but also avoid the error propagation of two levels.

Table 2. Comparison of accuracy of each category on the S3DIS dataset [55].

	Level-1 (%)	Level-2 (%)	MSSCN (%)
ceiling	97.65	97.54	**97.77**
floor	**99.20**	98.57	98.86
wall	93.44	92.57	**93.63**
beam	81.05	**85.92**	81.99
column	70.42	74.08	**76.67**
window	80.33	82.15	**89.11**
door	83.30	85.63	**85.86**
table	79.50	80.65	**83.48**
chair	88.42	88.02	**90.19**
sofa	81.30	70.26	**81.58**
bookcase	**84.28**	81.40	84.16
board	75.98	73.21	**77.52**
clutter	**80.37**	79.28	80.16

The experimental results indicate that the feature fusion approach proposed here successfully integrates the representations learned at Level-1 and Level-2. As shown in Figures 6 and 7, separating the board from a wall is challenging task, because these two objects have similar spatial structures. The results show that the board is not completely segmented at either level. By contrast, although there are still some segmentation errors for the board after feature fusion, the error rate is greatly reduced. These results show that the features obtained from point clouds of different densities have their own advantages. Thus, some previously unrecognized objects can be identified by combining these different features.

For the segmentation of some objects, the features of high-density regions are complementary to those from low-density areas. As shown in Figure 8, part of the chair is incorrectly segmented at Level-1, which is not the case in the segmentation obtained at Level-2. On the other hand, these two levels show errors in door segmentation, although these errors occur in different locations. After feature fusion, the segmentation effect of chairs and doors has been greatly improved. This discovery demonstrates that MSSCN can exploit the feature abstraction and representation capabilities of both Level-1 and Level-2 to improve the results.

MSSCN also has disadvantages. The scene shown in Figure 9 contains a table, several chairs, and a black object on the wall. It can be seen that at both Level-1 and Level-2, the black object is completely absorbed into the wall. Therefore, feature fusion cannot yield any additional information about the black object, and it still fails to be distinguished from the wall after feature fusion. Therefore, the performance achieved through feature fusion is limited by the performance of the backbone network to some extent.

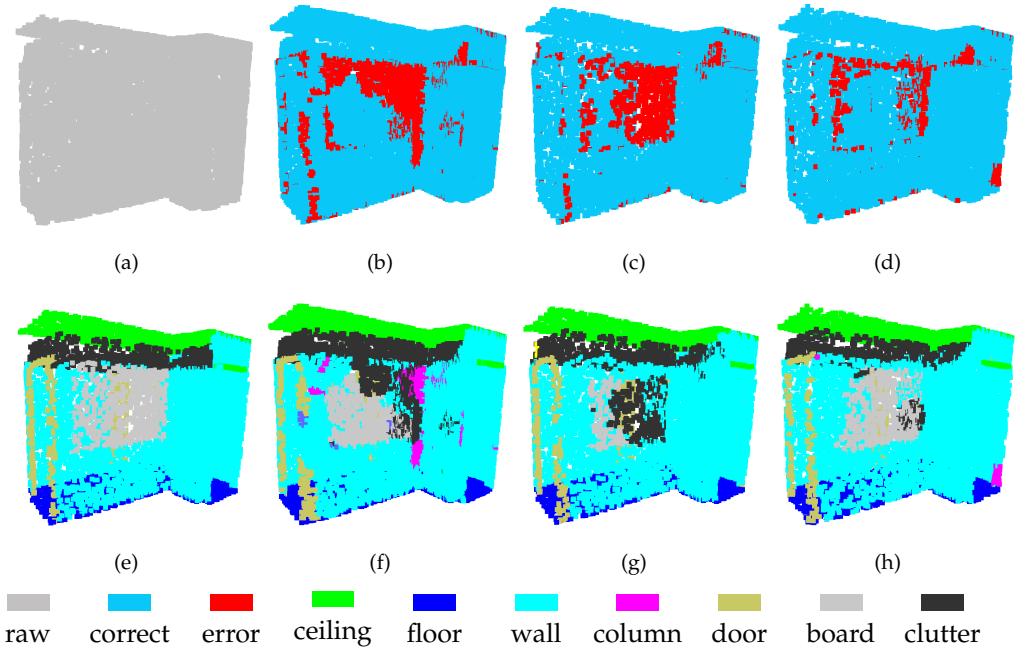


Figure 6. Segmentation results on the S3DIS-3 dataset: (**a**) input, (**e**) ground truth, (**b**,**f**) Level-1, (**c**,**g**) Level-2, and (**d**,**h**) MSSCN.

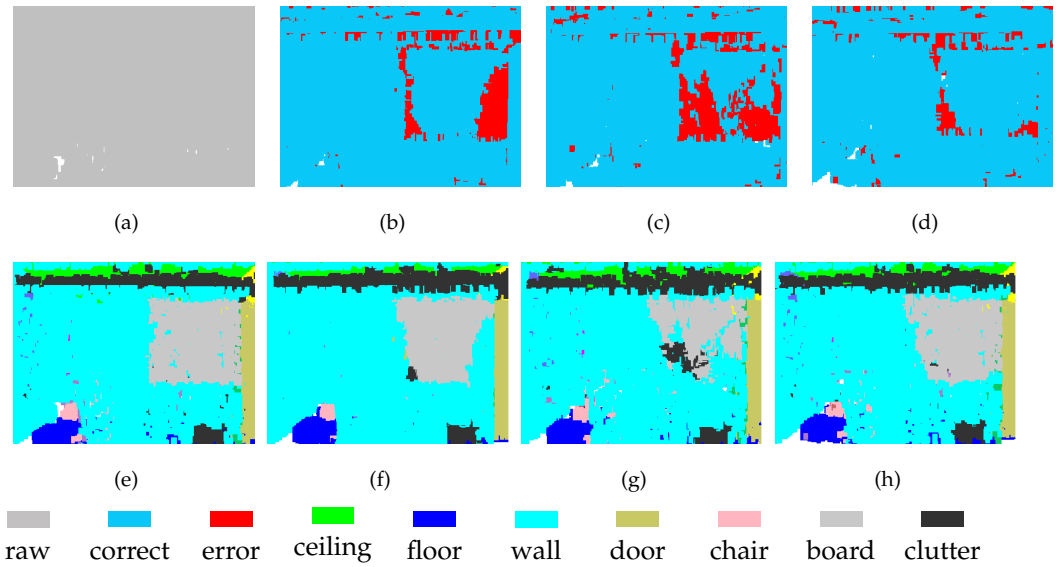


Figure 7. Segmentation results on the S3DIS-4 dataset: (**a**) input, (**e**) ground truth, (**b**,**f**) Level-1, (**c**,**g**) Level-2, and (**d**,**h**) MSSCN.

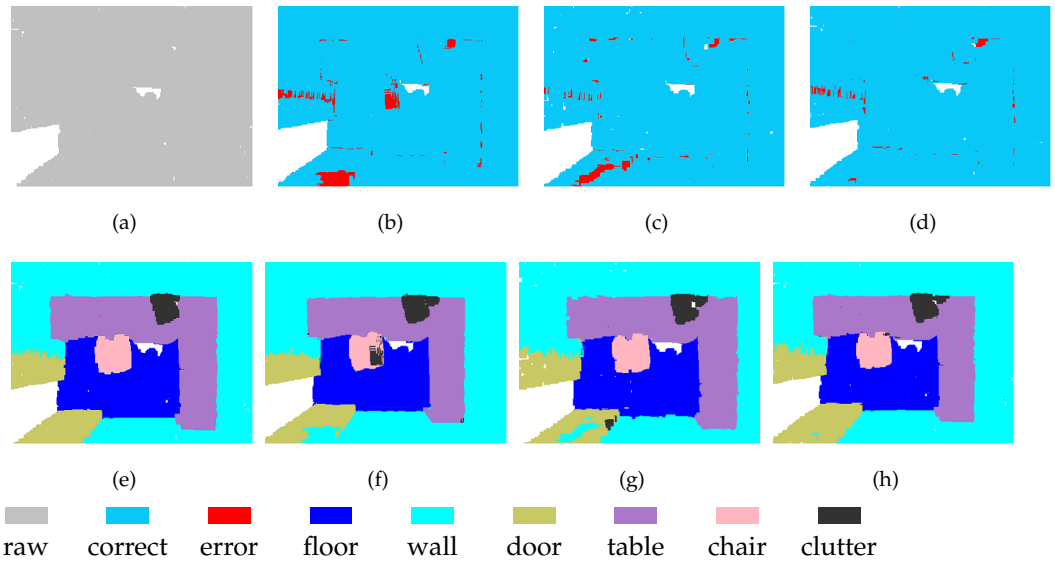


Figure 8. Segmentation results on S3DIS-5 dataset: (**a**) input, (**e**) ground truth, (**b**,**f**) Level-1, (**c**,**g**) Level-2, and (**d**,**h**) MSSCN.

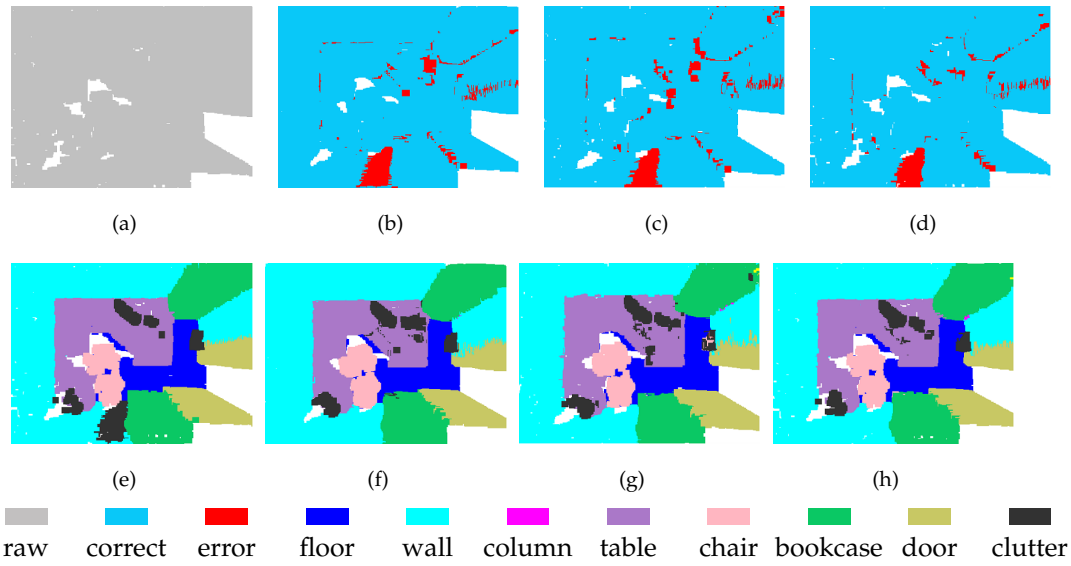


Figure 9. Segmentation results on S3DIS-6 dataset: (**a**) input, (**e**) ground truth, (**b**,**f**) Level-1, (**c**,**g**) Level-2, and (**d**,**h**) MSSCN.

4.3. Results on ScanNet

The comparison of our method with other recent works is presented in Table 3. 3DCNN is a semantic segmentation baseline trained on ScanNet. Our MMSCN has achieved better performance than these methods. Compared with PointNet++ and PointCNN, the segmentation accuracy of MSSCN is improved by more than 1%, and it is also slightly improved compared with PointSIFT. While we use SAN for feature extraction, the segmentation accuracy of MSSCN still achieves a 1.2% improvement over that of SAN alone. These experiments show that the proposed MSSCN architecture has good performance on the ScanNet dataset.

Table 3. Comparison of the accuracy of different methods on ScanNet [56].

Method	Accuracy (%)
3DCNN [56]	73.0
PointNet [19]	73.9
PointNet++ [20]	84.5
PointCNN [28]	85.1
SAN [44]	85.1
PointSIFT[29]	86.0
MSSCN	**86.3**

Figures 10 and 11 show the segmentation results obtained on the ScanNet dataset using our MSSCN method and other methods. As shown in Figure 10, due to the similar appearances of the table and the tea table (which belongs to the other furniture category), PointNet++, PointSIFT, and SAN cannot segment the table and the tea table effectively, whereas MSSCN shows a good segmentation effect. As shown in Figure 11, the wall next to the table is incorrectly segmented to the sofa or bed category by PointNet++, PointSIFT, and SAN due to the similarity of the corresponding spatial structures. It is expected that the presence of the table will interfere with the segmentation of the wall, but the proposed MSSCN method can avoid this interference to some extent and correctly segment the wall. This robust performance can be attributed to the multiscale point feature extraction and feature fusion capabilities of MSSCN.

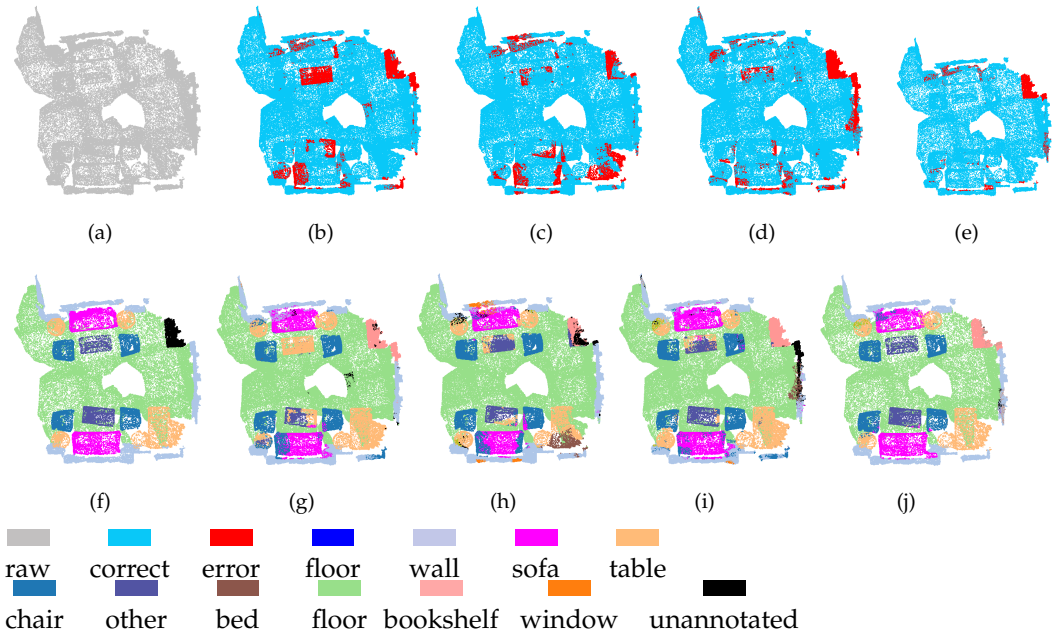


Figure 10. Segmentation results on ScanNet-1 dataset: (**a**) input, (**f**) ground truth, (**b**,**g**) PointNet++, (**c**,**h**) PointSIFT, (**d**,**i**) SAN , and (**e**,**j**) MSSCN.

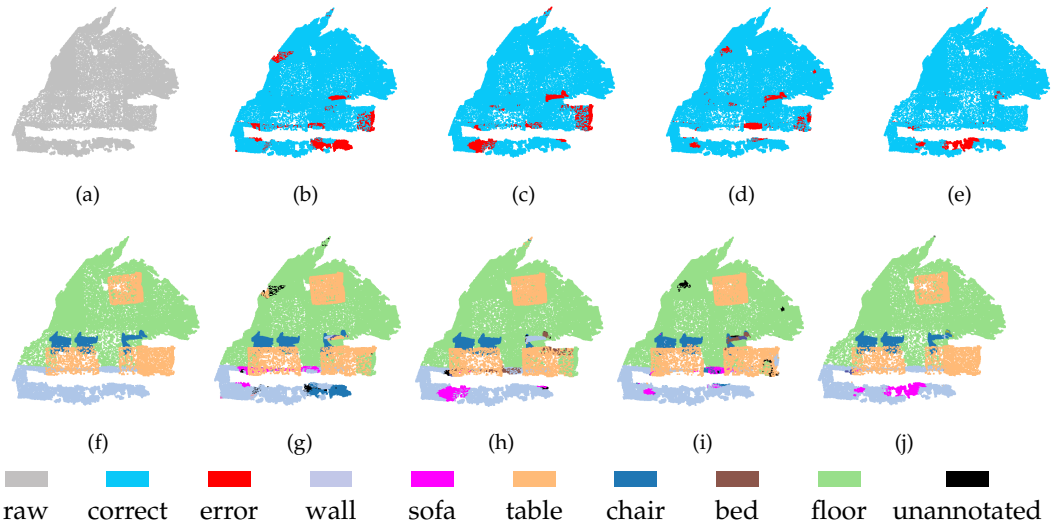


Figure 11. Segmentation results on ScanNet-2 dataset. (**a**) input, (**f**) ground truth, (**b**,**g**) PointNet++, (**c**,**h**) PointSIFT, (**d**,**i**) SAN, and (**e**,**j**) MSSCN.

4.4. Controlled Experiment

To find a suitable backbone network for feature extraction, we chose several lightweight networks to perform experiments, namely, SAN, PointNet and PointNet++. The experimental results show that MSSCN does not achieve good performance with PointNet as the feature extraction network, as shown in Figure 12. The reason is that MSSCN attempts to extract multiscale point cloud features by means of the backbone network, but PointNet employs the max-pooling operation to handle the problem of disordered points. As a result, only the global features of the point cloud scene can be extracted. Qualitative and quantitative experimental results demonstrate that our MSSCN can make good use of the advantages of different backbone networks and achieve better segmentation performance. When SAN or PointNet++ is used as the backbone network for feature extraction in MSSCN, the segmentation accuracy is better than that achieved using only the backbone network, and the performance with SAN is better than that with PointNet++. Therefore, we use SAN as the backbone network in MSSCN.

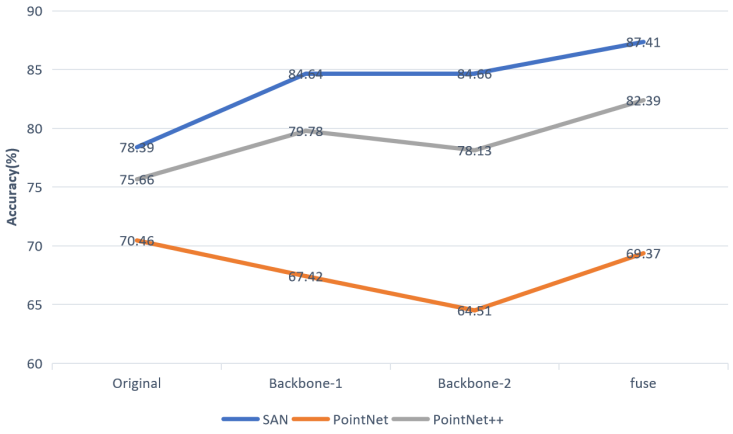


Figure 12. Results by our approach on S3DIS using SAN, PointNet and PointNet++ as backbone network. Where original is the experimental result of directly using the backbone network.

An important part of this experiment was the selection of the optimal combination of (α_1,α_2,α_3,α_4), as mentioned in Section 3.2. As clearly displayed in Table 4, the best parameter set (α_1,α_2,α_3,α_4) is (0.5, 0.4, 0.4, 0.1). The experimental results show that each component of the loss function influences the segmentation accuracy of MSSCN. Therefore,

good segmentation accuracy can be obtained by constantly adjusting the parameters and controlling the weights of features.

Table 4. Results of our approach on S3DIS [55] with different loss functions.

α_1	α_2	α_3	α_4	Accuracy (%)
0.0	0.4	0.4	0.1	86.96
0.1	0.4	0.4	0.1	87.15
0.3	0.4	0.4	0.1	86.67
0.5	**0.4**	**0.4**	**0.1**	**87.41**
0.7	0.4	0.4	0.1	86.96
0.4	0.0	0.4	0.1	86.92
0.4	0.1	0.4	0.1	86.88
0.4	0.3	0.4	0.1	86.94
0.4	0.5	0.4	0.1	87.07
0.4	0.7	0.4	0.1	86.62
0.4	0.4	0.1	0.1	86.91
0.4	0.4	0.3	0.1	86.57
0.4	0.4	0.5	0.1	87.04
0.4	0.4	0.7	0.1	86.67
0.4	0.4	0.4	0.0	87.00
0.4	0.4	0.4	0.1	87.35
0.4	0.4	0.4	0.3	86.86
0.4	0.4	0.4	0.5	86.82
0.4	0.4	0.4	0.7	86.69

5. Conclusions

Point clouds acquired by different sensors have become very popular as a source of representative 3D data. 3D vision research based on 3D point clouds has gradually transitioned from focusing on low-level geometric features to searching for high-level semantic understanding. The semantic segmentation of 3D point clouds is currently a popular research topic, which is undergoing a transition from early multiview-based and voxel-based processing to current point-based deep networks and graph convolution networks.

In this paper, a semantic segmentation network of a point cloud based on multi-scale feature fusion is proposed, which can extract useful feature information from downsampled point clouds of different densities. This is the first contribution of this paper. The second contribution is the use of a direct mapping method to merge features from different levels of the network framework while avoiding error propagation at each level. The third contribution is the proposal of a new loss function for the proposed MMSCN framework. The MSSCN can achieve good segmentation accuracy by controlling the weight of the loss associated with different layers.

Our experimental results show that the overall accuracy of MSSCN reaches 87.41% without RGB information and 89.80% with RGB information on the Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset. Compared with several existing methods, our MSSCN shows remarkable performance on the S3DIS dataset. Our results further show that our feature fusion method can not only combine the advantages of the Level-1 and Level-2 feature representations, but also avoid the error propagation of the two levels. Good segmentation accuracy can be achieved not only for objects that are easy to separate (such as ceilings and floors), but also for objects that are hard to separate (such as beams and columns). Therefore, the feature fusion operation can improve the segmentation accuracy for most objects.

Experiments and evaluations conducted on the ScanNet dataset similarly demonstrate that MSSCN achieves better performance than other recent outstanding methods, with significantly improved segmentation accuracy. Other current algorithms have difficulty segmenting similar objects accurately, whereas our proposed MSSCN shows better results

in this regard. Our results also show that although the existence of a table can interfere with wall segmentation, MSSCN can avoid interference to some extent, and segment walls well. This robust performance can be attributed to multi-scale point feature extraction and fusion.

Although good results have been obtained, we acknowledge that there are still several shortcomings. First, MSSCN relies on the backbone network used for feature extraction. Second, through direct mapping, some of the predicted point information will still be lost during feature fusion. In the future, we will concentrate on proposing new networks to solve the current problems with MSSCN.

Author Contributions: J.D. designed the algorithm, conducted experiments, and led the writing of the manuscript. Z.J. and S.H. designed and conducted experiments, and assisted in writing the manuscript. G.C. managed the project, conceived the experiments, and assisted in writing the manuscript. S.S., Z.W., J.S. and Y.W. took part in designing experiments, and assisted in writing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under grant no. 41971424 and no. 61701191, the key technical project of Xiamen Ocean Bureau under grant no.18CZB033HJ11, the key technical project of Xiamen Science and Technology Bureau under grant nos. 3502Z20191018, 3502Z20201007, 3502Z20191022, 3502Z20203057, the science and technology project of education department of Fujian province under grant nos. JAT190321, JAT190318, JAT190315.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicabl.

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

Abbreviations

The following abbreviations are used in this manuscript:

SAN	Spatial Aggregation Net
MSSCN	Multi-Scale Feature Fusion Semantic Segmentation Network
CNNs	Convolutional Neural Networks
FCN	Fully Convolutional Networks
GNSS	Global Navigation Satellite System
GCNs	Graph Convolutional Networks
CAGQ	Coverage Aware Grid Query
GCA	Grid Context Aggregation
SPVConv	Sparse Point-Voxel Convolution
3D-NAS	3D Neural Architecture Search

References

1. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
2. Kampffmeyer, M.; Salberg, A.B.; Jenssen, R. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1–9.
3. Hamaguchi, R.; Fujita, A.; Nemoto, K.; Imaizumi, T.; Hikosaka, S. Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1442–1450.
4. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Lei, L.; Zou, H. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 3–22. [CrossRef]
5. Ding, P.; Zhang, Y.; Deng, W.J.; Jia, P.; Kuijper, A. A light and faster regional convolutional neural network for object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2018**, *141*, 208–218. [CrossRef]
6. Ren, Y.; Zhu, C.; Xiao, S. Small object detection in optical remote sensing images via modified faster R-CNN. *Appl. Sci.* **2018**, *8*, 813. [CrossRef]

7. Gong, Z.; Lin, H.; Zhang, D.; Luo, Z.; Zelek, J.; Chen, Y.; Nurunnabi, A.; Wang, C.; Li, J. A Frustum-based probabilistic framework for 3D object detection by fusion of LiDAR and camera data. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 90–100. [CrossRef]
8. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 945–953.
9. Lawin, F.J.; Danelljan, M.; Tosteberg, P.; Bhat, G.; Khan, F.S.; Felsberg, M. Deep Projective 3D Semantic Segmentation. In *Computer Analysis of Images and Patterns*; Felsberg, M., Heyden, A., Krüger, N., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 95–107.
10. Feng, Y.; Zhang, Z.; Zhao, X.; Ji, R.; Gao, Y. GVCNN: Group-view convolutional neural networks for 3D shape recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 264–272.
11. Guo, H.; Wang, J.; Gao, Y.; Li, J.; Lu, H. Multi-view 3D object retrieval with deep embedding network. *IEEE Trans. Image Process.* **2016**, *25*, 5526–5537. [CrossRef] [PubMed]
12. Boulch, A.; Le Saux, B.; Audebert, N. Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks. *3DOR* **2017**, *2*, 7.
13. Zhang, R.; Li, G.; Li, M.; Wang, L. Fusion of images and point clouds for the semantic segmentation of large-scale 3D scenes based on deep learning. *ISPRS J. Photogra. Remote Sens.* **2018**, *143*, 85–96. [CrossRef]
14. Maturana, D.; Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
15. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
16. Gadelha, M.; Wang, R.; Maji, S. Multiresolution tree networks for 3d point cloud processing. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 103–118.
17. Qi, C.R.; Su, H.; Nießner, M.; Dai, A.; Yan, M.; Guibas, L.J. Volumetric and multi-view cnns for object classification on 3d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5648–5656.
18. Lin, Y.; Wang, C.; Zhai, D.; Li, W.; Li, J. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS J. Photogra. Remote Sens* **2018**, *143*, 39–47. [CrossRef]
19. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
20. 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 Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5099–5108.
21. Contreras, J.; Denzler, J. Edge-Convolution Point Net for Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 5236–5239.
22. Jia, M.; Li, A.; Wu, Z. A Global Point-Sift Attention Network for 3D Point Cloud Semantic Segmentation. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 5065–5068.
23. Zhao, H.; Jiang, L.; Fu, C.W.; Jia, J. PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5565–5573.
24. Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4558–4567.
25. Pham, Q.H.; Nguyen, T.; Hua, B.S.; Roig, G.; Yeung, S.K. JSIS3D: Joint Semantic-Instance Segmentation of 3D Point Clouds with Multi-Task Pointwise Networks and Multi-Value Conditional Random Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8827–8836.
26. Yi, L.; Zhao, W.; Wang, H.; Sung, M.; Guibas, L.J. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3947–3956.
27. Li, G.; Muller, M.; Thabet, A.; Ghanem, B. Deepgcns: Can gcns go as deep as cnns? In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019; pp. 9267–9276.
28. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 820–830.
29. Jiang, M.; Wu, Y.; Zhao, T.; Zhao, Z.; Lu, C. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv* **2018**, arXiv:1807.00652.
30. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

31. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
32. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
34. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
35. Milz, S.; Simon, M.; Fischer, K.; Pöpperl, M. Points2Pix: 3D Point-Cloud to Image Translation using conditional Generative Adversarial Networks. *arXiv* **2019**, arXiv:1901.09280.
36. You, Y.; Lou, Y.; Liu, Q.; Ma, L.; Wang, W.; Tai, Y.; Lu, C. PRIN: Pointwise Rotation-Invariant Network. *arXiv* **2018**, arXiv:1811.09361.
37. Kanezaki, A.; Matsushita, Y.; Nishida, Y. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5010–5019.
38. Barnea, S.; Filin, S. Segmentation of terrestrial laser scanning data using geometry and image information. *ISPRS J. Photogramm. Remote Sens.* **2013**, *76*, 33–48. [CrossRef]
39. Che, E.; Olsen, M.J. An Efficient Framework for Mobile Lidar Trajectory Reconstruction and Mo-norvana Segmentation. *Remote Sens.* **2019**, *11*, 836. [CrossRef]
40. Kundu, A.; Yin, X.; Fathi, A.; Ross, D.A.; Brewington, B.; Funkhouser, T.A.; Pantofaru, C. Virtual Multi-view Fusion for 3D Semantic Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Volume 12369, pp. 518–535.
41. Li, Y.; Pirk, S.; Su, H.; Qi, C.R.; Guibas, L.J. Fpnn: Field probing neural networks for 3d data. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 307–315.
42. Tatarchenko, M.; Dosovitskiy, A.; Brox, T. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2088–2096.
43. Wu, W.; Qi, Z.; Fuxin, L. Pointconv: Deep convolutional networks on 3d point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9621–9630.
44. Cai, G.; Jiang, Z.; Wang, Z.; Huang, S.; Chen, K.; Ge, X.; Wu, Y. Spatial Aggregation Net: Point Cloud Semantic Segmentation Based on Multi-Directional Convolution. *Sensors* **2019**, *19*, 4329. [CrossRef]
45. Li, Y.; Ma, L.; Zhong, Z.; Cao, D.; Li, J. TGNet: Geometric Graph CNN on 3-D Point Cloud Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3588–3600.
46. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *arXiv* **2018**, arXiv:1801.07829.
47. Lan, S.; Yu, R.; Yu, G.; Davis, L.S. Modeling local geometric structure of 3d point clouds using geo-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 998–1008.
48. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and Deformable Convolution for Point Clouds. *arXiv* **2019**, arXiv:1904.08889.
49. Liu, Y.; Fan, B.; Xiang, S.; Pan, C. Relation-Shape Convolutional Neural Network for Point Cloud Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8895–8904.
50. Lin, Y.; Yan, Z.; Huang, H.; Du, D.; Liu, L.; Cui, S.; Han, X. FPConv: Learning Local Flattening for Point Convolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 4292–4301.
51. Xu, Q.; Sun, X.; Wu, C.; Wang, P.; Neumann, U. Grid-GCN for Fast and Scalable Point Cloud Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 5660–5669.
52. Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; Han, S. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Volume 12373, pp. 685–702.
53. Hu, Z.; Zhen, M.; Bai, X.; Fu, H.; Tai, C. JSENet: Joint Semantic Segmentation and Edge Detection Network for 3D Point Clouds. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Volume 12365, pp. 222–239.
54. Eldar, Y.; Lindenbaum, M.; Porat, M.; Zeevi, Y.Y. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Process.* **1997**, *6*, 1305–1315. [CrossRef] [PubMed]
55. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3d semantic parsing of large-scale indoor spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543.

56. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. Scannet Richlyannotated 3d reconstructions of indoor scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5828–5839.
57. Zhang, Z.; Hua, B.; Yeung, S. ShellNet: Efficient Point Cloud Convolutional Neural Networks Using Concentric Shells Statistics. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019; pp. 1607–1616.
58. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11105–11114.