SPSN: Seed Point Selection Network in Point Cloud Instance Segmentation

1st Fei Sun
School of Software Enginerring
University of Science and Technology
Suzhou, China, 215123
usersunfei@gmail.com

2nd Yangjie Xu
Shenzhen Institutes of Advanced Technology
Chinese Academy of Sciences
Shenzhen, China, 518000
yj.xu@siat.ac.cn

3rd Weidong Sun
College of Computer
National University of Defense Technology
Changsha, China, 410073
weidongsun.nudt@gmail.com

***Abstract*—The current mainstream point cloud instance segmentation methods are mainly divided into two steps. Firstly, the points of each instance are aggregated in the feature space by means of metric learning to make the features of the same instance are as similar as possible, and then the aggregated vector clusters are segmented to construct the proposal of each instance. Much of the previous work has focused on the aggregation of vectors and ignored how the instance is divided after the vector aggregation. In this paper, we propose a seed point selection network. The seed point selection network selects a better seed point generation proposal by judging the "seedness" of each point, and completes the instance-level segmentation of all points. In addition, the speed of instance segmentation effectively improved by the fast processing of the generated instance points and the low "seedness" points. In the experiment, we graft the seed point selection network onto different instance segmentation networks, and the accuracy and efficiency of segmentation are improved in different degrees.**

***Index Terms*—point cloud, seed point, instance segmentation**

I. Introduction

Compared to 2D images, point clouds can describe the real scene more intuitively and accurately. In addition to the information of width and height, which is also contained in the 2D images, the point cloud depicts the depth of the scene. The point clouds have a wide range of applications especially in the scenes where high accuracy is required , such as autonomous driving and AR.

In the early stage, the point clouds are processed by transforming the point cloud into voxels [1] [2] [3], multi-view pictures [3] [4] or other data methods which can realize convolution processing based on "local spatiality" due to the disorder of point clouds [5] [6] and other properties. But without exception, all the processing methods mentioned above make the data and the network used for processing very voluminous. Pointnet [7] realizes the classification of point clouds and semantic segmentation by directly processing the point cloud data for the first time, based on which, SGPN [8] realizes the direct instance segmentation of point cloud data. The majority of papers in recent work related to instance segmentation of point clouds [8] [9] [10], with the help of metric learning [11] [12] [13], realize the feature map according to the principle that the points aggregated are from the same instance and the distance between instances should be ensured, after which, instance segmentation is realized by dividing the aggregated vector space. The focus of all the current papers is on the mapping of points in feature space, but how to perform a more accurate segmentation of the aggregated vector space has not been probed too much yet.

Although the aggregation of vectors is the basis of instance segmentation and directly affects the upper limit of the effect of instance segmentation, in fact, the vectors of the same instance are not always able to be perfectly aggregated together, and a lot of confusion is generated in many cases. For aggregated vector clusters, the segmented instances can be different, depending on the reference points selected. When the selected reference points are more "centered", a better proposal partition can be obtained, and if the reference point is at the junction of two aggregated vector clusters, then it is highly probable that both of the identifiable instances will be misclassified. Based on the reasons above, we expect to find the "center point" of each vector cluster as accurately as possible, so as to achieve a better division effect.

The premier problem we need to solve is how to define the "center point" of the vector cluster, that is, the measure and selection of the "seedness". Inspired by a large amount of previous work of instance segmentation [8] [14], we use the Intersection over Union(IoU) between the proposal and the groudtruth generated by the point to define it. A higher IoU means that the generated proposal contains more points of the instance and fewer noise points, which indicates the higher "seedness" of the point. On the contrary, a lower IoU

Corresponding Author: yj.xu@siat.ac.cn

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

means that there are many noise points and fewer points in the groundtruth in the proposal, indicating that the point is not suitable as a seed point.

Our pipeline first uses SGPN/ASIS [8] [9] to perform feature extraction in both semantic space and instance space. For the extracted features of the instance, we used SPSN to evaluate the seed property of some sampled points. According to the specific score value and semantic features, we completed the partition of the entire instance and each partition instance is attached with a specific semantic label.

In SPSN, since the input is an aggregated feature vector, the points of the same instance converge and the points of different instances are far away from each other. Therefore, for any point in the feature space, when we determine the threshold value, we can get the instance divided based on this point. SPSN is also based on this idea, but the difference is that our threshold adopts an approximate dynamic threshold. For the same point, we try our best to reduce the influence of threshold parameters on instance segmentation by selecting the best value after multiple partition (as shown in Fig. 1).

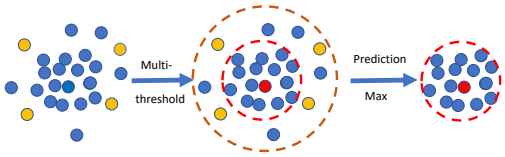

Fig. 1. After dividing the threshold for several times to obtain its optimal value, we get the best proposal corresponding to the seed point.

For each proposal segmented by a threshold, we use 3D-Pooling to pool the feature information and spatial information of the points of the proposal, and in order to obtain better experimental results, we adopt classification loss instead of regression loss to complete the prediction of "seedness" of points. For the distribution of IoU, although it is intuitive to think that regression loss should be adopted, using classification loss can predict the network more accurately and thus get better segmentation results. In summary, our contribution can be attributed into the following three points: (1) We propose a general post-processing method for completing point cloud instance segmentation in a metric learning manner, which can be grafted on any instance features aggregated and thus complete the final instance segmentation. (2) The method we propose can provide confidence for the segmented instance and solve the problem of no label in existing cluster method. (3)We choose multiple instance segmentation networks in the experiment and by comparing the difference between the original network and the grafted one, we announce that our method obtain different degrees of improvement on every network, as well as the improvement of both accuracy and reasoning speed on SGPN.

II. Related works

Instance segmentation from 2D images [15] [16] [17] [18] has made great progress currently. In 2D instance segmentation, the mainstream method performs instance segmentation based on object detection [19] [20] [21] [22] [23] [24] [25] [26], apart from which, some papers also explore the instance segmentation based on metric learning. The masterpiece of instance segmentation based on object detection is Mask R-CNN [27], which is based on Faster RCNN [19], realizing instance segmentation by adding FCN [28] network to each proposal. Instance segmentation based on object detection has high accuracy because of the outstanding performance of RCNNs in object detection. After that, Mask Scoring R-CNN [29] and Hybrid Task Cascade [30] are proposed and further improve the accuracy of instance segmentation. Papers of instance segmentation based on metric learning [14] [31] [32], different from the one based on object detection, adopt a bottom-up model, that is, directly learns the relevance between each pixel and then obtains the result of instance segmentation. Instance segmentation based on metric learning can effectively avoid the impact caused by overlapping objects, which is common in methods based on object detection. Among them, Newell et al. [33] aggregates the points of the same instance. Braban-dere et al. proposes discriminative loss [31] which makes aggregation of vectors more accurate and efficient. Alireza proposes instance segmentation based on metric learning, which uses mean-shift after finding the seed points randomly to complete instance segmentation. Shu Kong proposed a Gaussian Blur Mean Shift (GBMS) [32] to find the center of the instance, which is iterable with a slow rate. After completing the basic segmentation of the hand, Chutisant used the deterministic clustering algorithm [34] to find the reference point on the fingertip more accurately, and realized more reasonable and accurate gesture tracking. Although instance segmentation of images is not exactly the same as point cloud, they share a lot in common. In this paper, we propose a general post-processing framework of point cloud instance segmentation and realize the fast and accurate division of the features of the separately aggregated instances.

a) 3D Instance Segmentation: The current 3D instance segmentation methods mainly depend on metric learning while some papers focus on instance segmentation based on object detection [35]. The reason why methods based on metric learning is still the mainstream is that the accuracy of object detection is still insufficient due to the fact that 3D object detection should also estimate the posture of the object in addition to the functions of positioning and classification which are provided by 2D object detection.Simultaneously, the search space for object detection exponentially rises in 3D space. At present, SGPN realizes instance segmentation directly by using point cloud through learning the similarity matrix under the idea of metric learning. ASIS improves the accuracy of each other through instance segmentation and semantic segmentation. JSIS3D proposes the use of multi-valued conditional random field (MV-CRF) to improve the results of semantic segmentation and instance segmentation. SSCNet [36] proposes a novel loss function and adds GCN network to it to refine the results of instance segmentation. In addition to the methods of instance segmentation based on

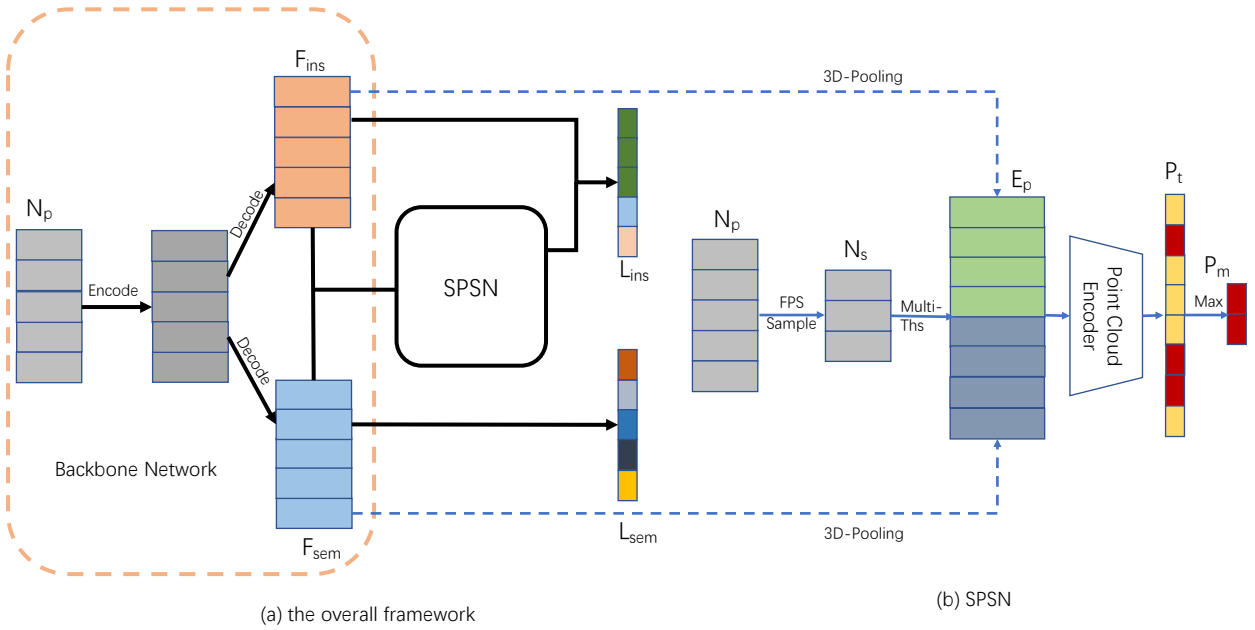

Fig. 2. The overall framework of instance segmentation. (a) Overall Struture. (b) illustration of SPSN module.

metric learning listed above, GSPN [37] first reconstructs the shape of the object and then performs refining segmentation to obtain the result of instance segmentation while 3D-SIS [38] combines the features of both 2D and 3D to enhance the accuracy of the proposal obtained. 3D-BoNet [35] obtains the result of instance segmentation based on the fuzzy region of the object, which is generated by returning to the bounding box first. Majority of papers related to the work based on metric learning as listed focuses on the feature map of the instance space and simply processes the division of features. In this paper, we propose a seed point selection network, which improves the results of instance segmentation by varying degrees when grafted to the papers mentioned above.

b) Learning point cloud features: Different from converting point cloud to voxels or multi-view pictures, Pointnet proposed by Qi realizes the direct feature extraction of point cloud for the first time and achieve good results in classification and segmentation, based on which, Pointnet++ [39], RSNet [40], etc. makes improvement from the perspective of partial information. In addition, PointCNN [41], PointSift [42], etc. further improve the capacity of learning features of point cloud through the combination of Pointnet and other features. Our work achieves accurate classification of seed points by extracting features of point cloud, thereby improving the results of instance segmentation.

III. Methods

In this section, we thoroughly introduce the seed point selection network, a powerful and general post-processing framework for point cloud instance segmentation, the overall structure of which is shown in Fig. 2(a). First of all, we briefly introduce the current general instance segmentation framework as the preface of SPSN and how to generate the final mask by using the "seed score" output from SPSN. Then, we describe the specific structure of the network of SPSN, that is, how to predict the "seed score", and finally we explain the details of training the seed point selection network.

A. The brief introduction of the framework of point cloud instance segmentation

The current framework of point cloud instance segmentation based on metric learning generally consists of a shared encoder used for encoding the semantic features of the point cloud and two parallel decoders, one of which used for predicting the semantic category and the other used for obtain the features of the point in the instance space based on metric learning. Among the features, the ones belonging to the same instance share great similarities while those from different instances are far away, which obviously indicates that labels of instance segmentation of point cloud can not be obtained directly from the features mentioned above. Nevertheless, SPSN is a general framework to complete the segmentation of features in the instance space and therefore obtain the specific instance labels. Specifically, when inputting the point cloud information, the current instance segmentation framework generally extracts features of the point cloud by using Pointnet/Pointnet++, which can perform semantic segmentation itself and thus features of the middle layer of Pointnet/Pointnet++ is generally used to encode. At the same time, the semantic segmentation branch can be obtained through several layers of networks

followed by Pointnet/Pointnet++. For the decoding of features in instance segmentation branches, the current framework usually adopts loss function related to metric learning after adding new neural network layers to complete the features mapping. Specifically, in the double-hinge loss used by SGPN, the distance between points from the same instance should be less than K_1 and tend to 0, otherwise some penalty should be imposed. Simultaneously, distance of points from different instances should be greater than K_2, which is greater than K_1. Double-hinge loss ensures that distance between points from the same instance is small($distance < K_1$) and that from different instances is large ($distance > K_2$) through the methods mentioned above and thereby completes the mapping of the points in the instance space. In the discriminative loss used by ASIS/JSIS3D, the loss function ensures that in one instance, the distance between every point and the center point is smaller than δ_v while the distance between center points of different instances is greater than δ_d, $\delta_d > 4\delta_v$ be ensured, through which, the division of the instance space is completed.

In general, the current framework consists of a shared encoder and two parallel decoder and the encoder completes the extracting of the features of the point cloud while the two decoder complete the prediction of semantic segmentation and the mapping of the points in the instance space respectively.

B. Seed score calculation and generation of masks

When the mapping of the points in the instance space is obtained, we generate a series of masks through the following methods. First we select a seed point and determine a certain threshold for it and then by calculating the distance between the selected point and other points, a point is considered to be within the mask if the distance is less than the threshold. As what is mentioned above, two factors are needed to generate the mask: (1) Select the seed point. (2) Determine the threshold.

We use the approximate dynamic threshold method to determine the threshold. For a certain seed point, if ths is used to represent the threshold, then the range of the is $ths \in [0, \rho]$, ρ is the maximum of the distance between the seed point and all points. If the range is divided into n shares equally, that is, the range can be divided into n intervals with a total of $(n+1)$ endpoints. Excluding the starting and ending points, there are $(n-1)$ intermediate endpoint values, so when the seed generates the mask, its optimal threshold can be expressed as:

$$ths_{best} = \underset{\rho i}{max}\, IoU(seed_{gt}, seed_{\rho i})$$

That is the threshold adopted when the IoU between the generated mask and the instance where the seed point is located is the largest. Among them, ρ_i means the value of the ith endpoint among the $(n-1)$ intermediate points. $seed_{\rho i}$ means the mask generated from the selected seed point when adopting ρ_i as the threshold. IoU represents the Intersection over Union between groundtruth and mask. Through the approximate coverage of the selectable threshold range, we obtain the optimal threshold, under which, the IoU between the mask generated from the seed point and the groundtruth is called seed score, that is

$$seedscore = IoU(seed_{gt}, seed_{ths_{best}})$$

When the network performs inference, because $seed_{gt}$ is unknown, we obtain the seed score of the seed point through predicting the IoU of a selected seed point under different thresholds and then obtain the corresponding threshold value as the optimal one. In the selection of n, it is obvious that the larger the value of n is, the better results will be obtained, but we choose 5 as the current default value because of the limitation of the network convergence and inference speed. We choose the default value through counting the distribution of IoU under different values of n (detailed in the experimental part). So when we choose 5 as the default value it means that the range is divided into 5 equal parts. We choose the 4 intermediate endpoints to predict IoU and then obtain the seed score and the optimal threshold by adopting the maximum value.

In the selection of seed point, we can calculate the seed score of each point and then generate the final instance through NMS ideally, but it costs huge loads of calculation and the number of points in the point cloud is much larger than the number of instances. So we can sample points and calculate the seed score of them to select the final seed point. When sampling points, to ensure the diversity of the sampling points in space, we adopt the farthest point sampling (FPS) algorithm to make the sampling points evenly distributed in the point cloud. When we use the algorithm to sample points ,the default number of points S_g is 64.

C. Seed score prediction model

The overall network results of the seed score predicting model is shown in Fig. 2(b). Firstly, we use the farthest point sampling (FPS) algorithm to sample the points in the point cloud N_p and then predict the seed score of the sampling point set N_s. For each point in N_s, different proposal segmentation results are generated under multiple thresholds selected. We predict the IoU value of each proposal and then adopt the maximum value to obtain the seed score and the corresponding threshold.

With a determined threshold for each proposal, we use 3D-Pooling to randomly sample points of the number of S_p, ensuring that the number of the sampling points in each proposal is the same. After that, the feature matrix E_p which is the result of the feature pooling of points of proposal generated from each seed point with different selected thresholds is obtained through the concatenation of the semantic features and instance features of each point. Its shape is $(n \times S_g) \times S_p \times E(ins + sem)$, in which $E(ins+sem)$ means the feature dimension after the concatenation of semantic features and instance features while S_g is the number of seed points, multiplying which by n, the approximate dynamic threshold parameter , the final number of proposals is obtained.

When predicting the IoU of each proposal, we adopt the structure of vanilla Pointnet which is performing max

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

operation on each proposal after convolution to obtain the maximum value of each proposal in each feature dimension, after which, the classification result is obtained by connecting the fully connected layer. We currently use 0.8 as the threshold and if $IoU < 0.8$, it is 0, otherwise it is 1. Compared with regarding the prediction of IoU as a regression problem, we obtain better results through the above classification method. We use Weighted Softmax Cross Entropy loss as the loss of classification to balance the uneven distribution between the two classes. The loss function is as follows:

$$L(P_s, P_t) = -\sum W_c P_t log P_s$$

The P_s is the predicted value and P_t is the corresponding label. W_c is 2 when it is class 0 and W_c is 1 when it is class 1.

D. Implementation Details

Because the seed point selection network is the judgement of the "position" of the basically fitted features, so we train the seed point selection network at the last 20 epochs or graft the seed point selection network after the network is completely fitted and then perform the training. Currently we adopt the second method, which is more convenient for grafting the seed point selection network into any networks for effect verification and debugging the network. When training the network for seed score predicting, we use ADAM optimizer with initial learning rate 0.001, momentum 0.9 and batch size 4. The learning rate is divided by 2 every 300k iterations. Simultaneously, we compare the two methods and obtain approximate results.

IV. Experiments

A. Experiments Settings

a) Datasets: We evaluate our work on Stanford 3D Indoor Semantics DatasetS3DISand compare the effect of grafting the seed point selecting network into SGPN and ASIS. S3DIS performed 3D scanning on a total of 272 rooms in 6 regions and in the generated point cloud, each point contains semantic labels and instance annotations. The accurate semantic labels and instance labels can be obtained by processing the datasets.

b) Evalution Metrics: We use k-fold cross-validation to compare the results when evaluating the network. Due to the fact that SPSN is aimed at instance segmentation and does not change the results of the semantic segmentation of the original paper, we just evaluate the results of the instance segmentation. We use Cov and WCov in the setting of instance segmentation indicators. Cov is the average value of IoU between each instance generated by prediction and the ground truth, based on which, WCov weights according to the size of the instance.The fact is that the more points the instance has, the greater the weight is. Cov and WCov are defined as follows:

$$Cov(g,p) = \sum_{i=1}^{|g|} \frac{1}{|g|} \underset{j}{max}\, IoU r_i^G, r_j^P \tag{1}$$

$$WCov(g,p) = \sum_{i=1}^{|g|} \omega_i \underset{j}{max}\, IoU r_i^G, r_j^P \tag{2}$$

$$\omega_i = \frac{|r_i^G|}{\sum_k |r_k^G|} \tag{3}$$

Among them, G represents Ground truth, P represents prediction, $|ri^G|$ represents the number of points in the ith instance in the ground truth. Simultaneously, we provide the common index of the accuracy (mPrec) and the recall rate (mRec) which are calculated when the threshold of IoU is 0.5.

B. Overall network results on S3DIS

Table 1 shows the results of grafting SPSN on SGPN and ASIS separately. On SGPN, we achieved growth of Cov by 3.2%, while mCov achieved 3.4% of growth. On ASIS, we achieved an increase of 1.3% in Cov and an increase of 1.2% in mCov. In addition, we also provided common accuracy and recall rate results. On SGPN, we achieved an accuracy increase of 4.4% and a recall rate increase of 3.9%. On ASIS, our accuracy decreased slightly by 0.4%, but our recall rate increased by 0.8%. In the results of instance segmentation, the increase is mainly due to the re-segmentation of the aggregated vectors, which generates the proposal of better quality. On SGPN, we achieved a relatively high improvement by using SPSN instead of the Group Merging algorithm used in the original network, while in ASIS, we obtained a better effect of instance segmentation and provided confidence labels for the segmented instances by using SPSN instead of mean-shift algorithm used in the original network. We also reported per class result of the dataset in table 2.

We also compared the results of each category in Table 2. The accuracy of the segmentation mainly rises in the categories with good aggregation of the original features such as floors, windows, beams, and doors. For example, in the category floor , our Cov has increased 1.5%, Wcov rose by 1.8%, in the category window , the Cov rose by 1.5 %, Wcov rose by 1.5%. While for categories such as columns that may not have been aggregated, our indicator even appeared extremely small drop. It shows that when we generate instances based on confidence, we first consider the examples with good aggregation, and then segment the examples with poor aggregation, which also meets the principle of accurate segmentation.

TABLE I
COMPARISON OF OVERALL PERFORMANCE BETWEEN DIFFERENT NETWORKS.

Method	mCov	mWcov	mPrec	mrec
SGPN	37.9	40.8	38.2	31.2
SGPN+SPSN	40.1	44.2	44.6	35.1
ASIS	51.3	55.2	63.6	47.5
ASIS+SPSN	52.5	56.3	62.5	48.1

C. Ablative Analysis

a) Approximate dynamic threshold: In the experimental setting of hyperparameter selection for approximate dynamic

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE II
PER CLASS RESULTS ON S3DIS DATASET

		mean	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
Cov	ASIS	51.3	74.4	76.5	53.8	56.4	14.3	63.7	50.5	53.2	59.9	32.1	38.8	56.4	37.4
	ASIS+SPSN	52.5	75.6	78.0	55.2	58.6	14.1	65.2	51.9	54.4	61.5	33.7	38.3	58.2	37.3
WCov	ASIS	55.2	80.4	76.6	67.3	58.3	14.6	63.8	51.4	55.3	62.4	34.5	44.3	57.1	51.4
	ASIS+SPSN	56.3	82.3	78.4	68.7	59.4	15.1	65.3	52.5	57.1	64.3	34.0	45.1	58.6	50.5

threshold, we adopt the method described in Section 3. By dividing the range of threshold of each point into n equal parts and adopt the maximum value of IoU between the proposal generated from points in each interval and the groundtruth to obtain the optimal proposal generated from each point. Obviously, the larger the value of n, the more likely it is to approximate the theoretical optimal value, but the larger the number of IoU values of proposals that need to be predicted and the more resources. So we establish balance between the two factors. In the specific experiment, we obtain the final hyperparameter results by calculating the distribution of the maximum IoU value of all points in the training set.

The maximum IoU value distribution is shown in Fig. 3. In Fig. 3, lines of different colors represent different values, and points on the lines represent the proportion of the points that meet the condition when IoU is greater than or equal to the abscissa value, so all the maximum IoU value of each point in the training set is greater than 0 and the overall curve shows a downward trend. We can find in Fig. 3 that when $n = 4$, the distribution of IoU has no huge difference from that when $n = 20$, which simultaneously closely fits the line of IoU distribution when $n = 5$. However, when $n < 4$, the distribution of IoU has great attenuation and the ratio of the best IoU value greater than 0.8 greatly reduced. Therefore, we adopt 4 as the default value. In the above experiment, by comparing the distribution of IoU, other parameters can be prevented from being mixed, and thus a accurate result of the selected parameter can be obtained.

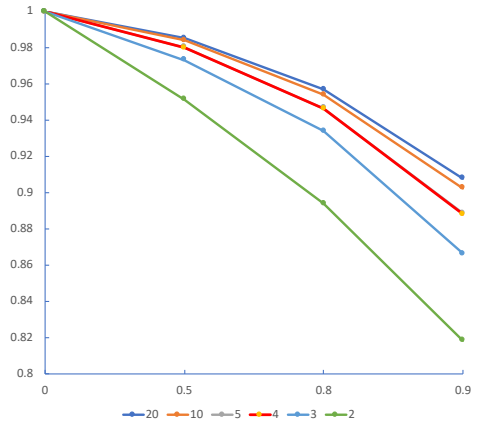

Fig. 3. Multi-threshold parameter selection. The different lines represent the distribution of IoU under different parameters. The points in the graph represent the percentage of points when the IoU is greater than the abscissa.

b) Sampling points of FPS algorithm : Ideally, we can calculate the seed score of all seed points and select the best point to generate proposals by sorting them. But it will generate a huge amount of calculation and occupy a lot of explicit memory during training. At the same time, because the number of points in the point cloud is far greater than the number of instances, we can sample parts of these points and generate proposals based on these points. We use FPS (farthest point sampling) algorithm to sample which can ensure that the sampling points are evenly distributed in the whole point cloud. In the experiment of sampling points, We directly use the groundtruth of the seed scores , which can avoid the impact of the seed score prediction model. We use the result of proposal segmentation as a measure of judgment. At the same time, when all points are taken for segmentation, it represents the upper bound of the segmentation that the backbone network can reach.

The experimental results are shown in Fig. 4. The two lines represent the values of mCov and mWCov respectively. When taking all points as seed points to generate proposals, we get the best segmentation result. At the same time, the fewer the points, the worse the segmentation result. In the results of experiment, we still get great segmentation results when we choose 16 points. It shows that FPS sampling points are evenly distributed in each instance and avoid the degradation of segmentation performance because of no sampling seed point in the instance. In the selection of final sampling points, we use the default value of 128, which makes the training speed of seed scoring prediction model faster. When the memory is large enough, 512 is a better value to choose.

c) Proposed sampling points: In the seed score prediction network, the number of points in each proposal generated by seed points according to the threshold is different, so it is necessary to fix the number of points in each proposal during processing. For the determination of the number of points, we have carried out many experiments and the results of experiments are shown in Fig. 5. At the same time, according to Fig. 5, we report the prediction results of the seed score prediction network. In this experiment, we choose 128 points and our seed scoring network has high accuracy.

d) Qualitative Results: Fig. 6 shows some visualization examples. Different colors represent different instances in instance segmentation and different colors represent different classes in semantic segmentation. It is worth noting that the color itself is meaningless.

e) Computation Time: In Table 3, we measure the inference time of the network. The experiment was completed on a Nvidia2080Ti. Compared with mean-shift used in ASIS,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

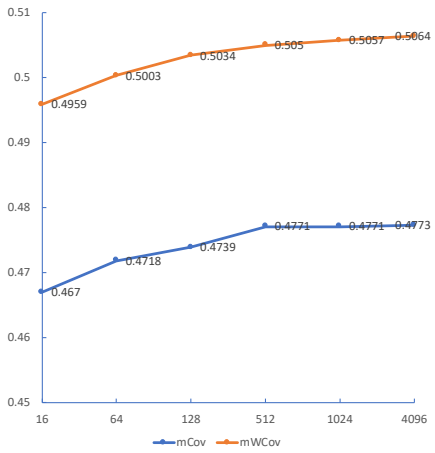

Fig. 4. The result of instance segmentation under different sampling points.

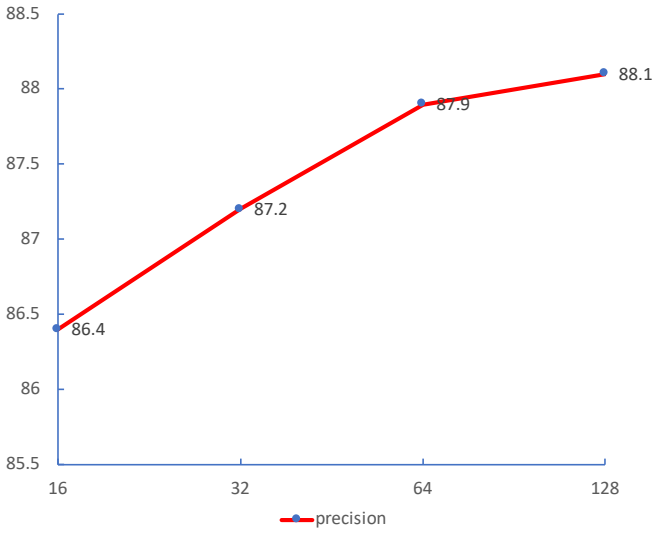

Fig. 5. The accuracy of the seed scoring network in sampling different points for each proposal.

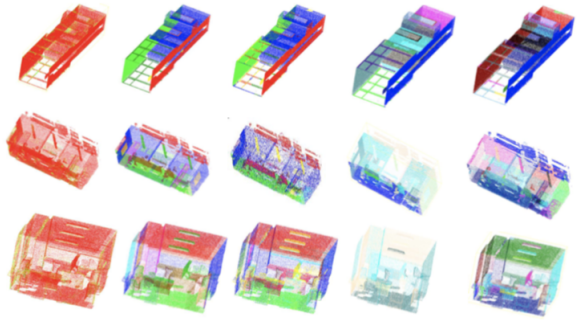

Fig. 6. Qualitative results of SPSN.

we lose some efficiency with an improvement in accuracy, but compared with SGPN and the Group Merging algorithm it uses, We greatly improve the accuracy and speed up the inference.

TABLE III
COMPARISON OF COMPUTATION SPEED BETWEEN DIFFERENT NETWORK.

Method	Overall	Network	Grouping
SGPN	748	40	708
SGPN+SPSN	457	55	402
ASIS	246	35	211
ASIS+SPSN	454	52	402

V. CONCLUSION

In this paper, we present a seed point selection network, a generic method for post-processing point cloud instance segmentation. In the processing, we can complete the instance segmentation more effectively based on the existing algorithm by selecting the seed point. The current instance segmentation of point cloud is still in a relatively early stage. We hope to have more excellent backbone network and get better segmentation results. At the same time, for post-processing, there are still many shortcomings in this paper. We expect more novel networks to improve the result of instance segmentation and broaden our learning horizon.

REFERENCES

[1] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1912–1920, 2015. 2, 5, 11.

[2] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In IEEE/RSJ International Conference on Intelligent Robots and Systems, September 2015. 2, 5, 10, 11.

[3] C.R.Qi,H.Su,M.Nießner,A.Dai,M.Yan,andL.Guibas. Volumetric and multi-view cnns for object classification on 3d data. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2016. 2, 5, 8.

[4] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In Proc. ICCV, to appear, 2015. 2, 5, 6, 8.

[5] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203, 2013. 2.

[6] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 37–45, 2015. 2.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 1(2):4, 2017. 1, 2, 3.

[8] W. Wang, R. Yu, Q. Huang, and U. Neumann. SGPN: Similarity group proposal network for 3d point cloud instance segmentation. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2018. 2, 5, 8.

[9] Wang X, Liu S, Shen X, et al. Associatively segmenting instances and semantics in point clouds[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 4096-4105.

[10] Pham Q H, Nguyen T, Hua B S, et al. JSIS3D: joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 8827-8836.

[11] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-convolutional siamese networks for object tracking. In ECCV Workshops, 2016. 2.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

[12] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In CVPR, 2005. 2.
[13] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In ICCV, 2015. 2.
[14] Fathi A, Wojna Z, Rathod V, et al. Semantic instance segmentation via deep metric learning[J]. arXiv preprint arXiv:1703.10277, 2017.
[15] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In CVPR, 2016. 1, 2.
[16] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In ECCV, 2016. 2.
[17] Pinheiro P O, Lin T Y, Collobert R, et al. Learning to refine object segments[C]//European Conference on Computer Vision. Springer, Cham, 2016: 75-91.
[18] P. O. Pinheiro, R. Collobert, and P. Dolla r. Learning to segment object candidates. In NIPS, 2015. 1, 2.
[19] S.Ren,K.He,R.Girshick,andJ.Sun.Fasterr-cnn:Towards real-time object detection with region proposal networks. In NIPS, 2015. 2.
[20] R. Girshick. Fast r-cnn. In ICCV, 2015. 2.
[21] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection.
[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016. 2.
[23] J.RedmonandA.Farhadi.Yolo9000:Better,faster,stronger. In CVPR, 2017. 2.
[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In ECCV, 2016. 2.
[25] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017. 2.
[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In CVPR, 2017. 2.
[27] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In ICCV, 2017. 1, 2.
[28] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.
[29] Huang Z, Huang L, Gong Y, et al. Mask scoring r-cnn[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 6409-6418.
[30] Chen K, Pang J, Wang J, et al. Hybrid task cascade for instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2019: 4974-4983.
[31] De Brabandere B, Neven D, Van Gool L. Semantic instance segmentation with a discriminative loss function[J]. arXiv preprint arXiv:1708.02551, 2017.
[32] Kong S, Fowlkes C C. Recurrent pixel embedding for instance grouping[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 9018-9028.
[33] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In Proc. Advances in Neural Inf. Process. Syst. 2017. 2.
[34] Kerdvibulvech C. A methodology for hand and finger motion analysis using adaptive probabilistic models[J]. EURASIP Journal on Embedded Systems, 2014, 2014(1): 18.
[35] Yang B, Wang J, Clark R, et al. Learning object bounding boxes for 3d instance segmentation on point clouds[C]//Advances in Neural Information Processing Systems. 2019: 6737-6746.
[36] Doi K, Iwasaki A. SSCNET: Spectral-Spatial Consistency Optimization of CNN for Pansharpening[C]//IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2019: 3141-3144.
[37] Yi L , Zhao W , Wang H , et al. GSPN: Generative Shape Proposal Network for 3D Instance Segmentation in Point Cloud[J]. 2018.
[38] Hou J , Dai A , Nießner, Matthias. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans[J]. 2018.
[39] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proc. Advances in Neural Inf. Process. Syst., 2017. 2, 3, 8.
[40] Q. Huang, W. Wang, and U. Neumann. Recurrent slice net- works for 3d segmentation of point clouds. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2018. 3.
[41] Y. Li, R. Bu, M. Sun, and B. Chen. Pointcnn. arXiv: Comp. Res. Repository, 2018. 3.
[42] Jiang M, Wu Y, Zhao T, et al. Pointsift: A sift-like network module for 3d point cloud semantic segmentation[J]. arXiv preprint arXiv:1807.00652, 2018.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE