Point Cloud based Object Recognition Method Report

Siming Fan[1]
[1] UESTC

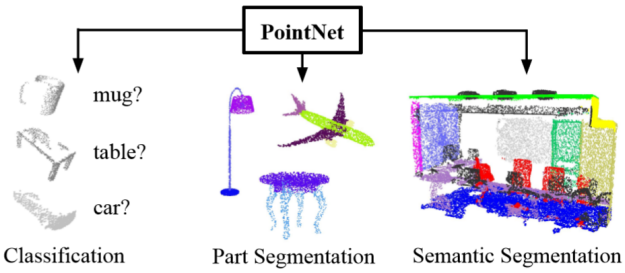

Figure 1. Applications of PointNet. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks

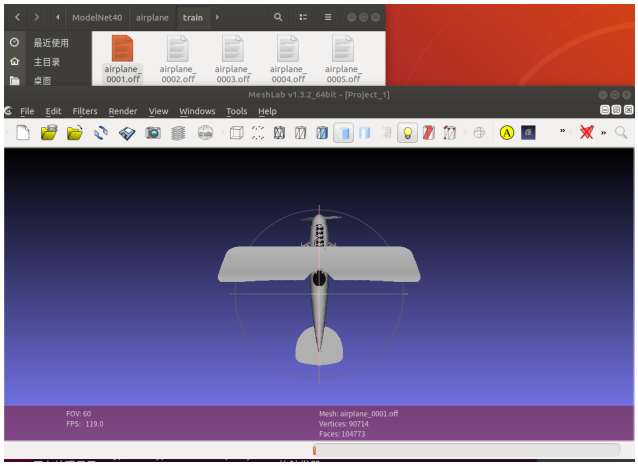

Figure 2. 'aircraft' category of ModelNet40, which contains 90,714 points and 104,733 triangular meshes.

Abstract

This article mainly focuses on three parts, one is the recognition task of the 3D model, the other is the recognition task of the 3D scene, and finally it is based on my own understanding to propose improvements. This article mainly analyzes the method that takes raw points cloud as input.

1. Introduction

The price of RGB-D cameras has been decreasing year by year, the point cloud data has become easier, and there are already dozens of RGB-D datasets. From the perspective of tasks, such as target detection tasks, 2D bounding boxes are far from sufficiency in the areas of autonomous driving, augmented reality, and robotics. They need 3D bounding boxes. From a feature point of view, objects are naturally separated in the 3D world, so problems such as occlusion are naturally solved in the 3D feature representation. At the same time, point clouds can provide rich shape information. How to extract point cloud shape information end-to-end with deep learning is a hot topic.

2. Tasks, Datasets,Preprocessing and Methods

2.1. 3D Model Pointcloud Recognition

2.1.1 3D model Classification Task

ModelNet [9] is a large-scale 3D CAD dataset whose task is a 3D model classification task. The original intention was to learn a 3D representation that can well capture intra-class differences. It is 22 times larger than the latest data set at the time, and contains 151,128 3D CAD models and 660 different classes. In the benchmark, the training set has 9843 samples and the test set has 2468 samples. Its 3D model consists of a series of points and triangular meshes, as shown in Figure 2.

2.1.2 3D model Part Segmentation Task

ShapeNet [9] is also a large 3D CAD dataset containing 3Million+ models and 4K+ categories. The goal is to use a data-driven approach to learn complex shape distributions from raw 3D data for various object categories and poses (such as seeing the side of a cup to predict that it is hollow inside), and automatically discover layered combinations Part representation. The

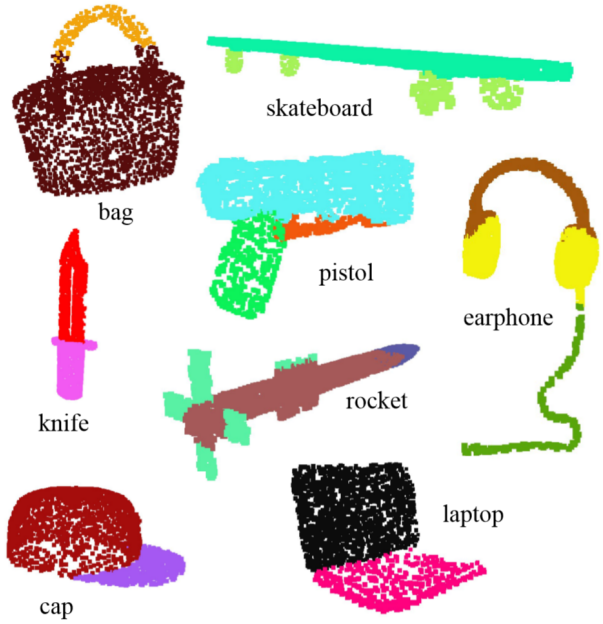

Figure 3. ShapenetPart

tasks included are: object category recognition, shape completion, part segmentation, Next-Best-View Prediction, 3D Mesh Retrieval. Its subset, ShapeNet Part, is used for component segmentation in this article, including 16881 shapes, 16 categories, and 50 Parts, as shown in Figure 3.

2.1.3 Representations

3D model datasets usually provide mesh, Volumetric, depth map, and point cloud representations.

Mesh representation Grid representations commonly used in computer graphics include N points and M faces, where N points can be represented by a $Nx3$ matrix, if M faces are all triangular Can be represented by a $Mx3$ matrix. The grid representation has a lot of choices, such as choosing a triangle or quadrilateral, pentagon, and the size of each patch. Due to different choices, migration of different data sets can become difficult.

Volumetric representation This representation uses a three-dimensional Boolean array to indicate whether there is an object at that location. Selecting a grid representation means extracting and selecting the size of each grid. If the grid is too large, it means that the resolution is low and quantization errors will occur. If the grid is too small, it means that most of the area is blank, and the storage and computational overhead

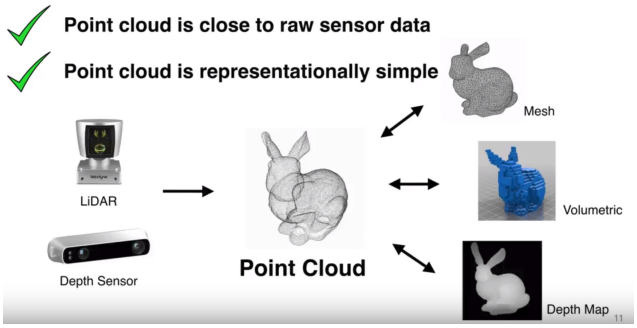

Figure 4. Representation of 3D models

will be large. The methods using raster representation are Subvolume [5]

Depth Map Representation The depth map is a representation that can be directly spliced with RGB maps and 2D convolved. The advantage is simplicity. However, the depth map representation has three disadvantages. First, for a 3D model dataset, a single depth map is not comprehensive for 3D, so multi-perspective depth maps are often used. However, the direction of the optimal viewing angle is difficult to determine. Second, the depth map cannot easily solve problems such as occlusion. In contrast, objects are naturally separated on other 3D representations. Third, there is no evidence that CNN can make good use of the features of the depth map, instead of just treating the depth map as another image with rich edge and texture information. And the fact that the depth map is used as the fourth channel of RGB for object recognition does not perform well on many tasks. Methods using depth map representation are MVCNN [7]

Point Cloud Representation The point cloud is the direct output of LIDAR and Depth Sensor, which is represented by a $Nx3$ -dimensional matrix. The advantage is that it is simple and natural, and there are no hyperparameters to choose, so it is very suitable for use as an input to an end-to-end neural network. The disadvantage is that the point cloud is out of order, which means that when the i and j points are exchanged, the matrix of the point cloud has changed, but the output of the neural network should not change. For this, Pointnet provides a good solution, first map the point cloud to a high-dimensional space, and then perform the maxpool operation to obtain the global characteristics of the point cloud with replacement invariance. The point cloud representation method is Pointnet [2, 4, 6]

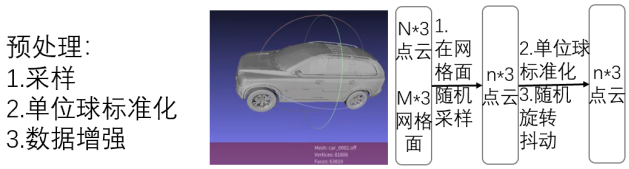

Figure 5. ModelNet preprocessing: point sampling, unit sphere normalization, and data augmentation.

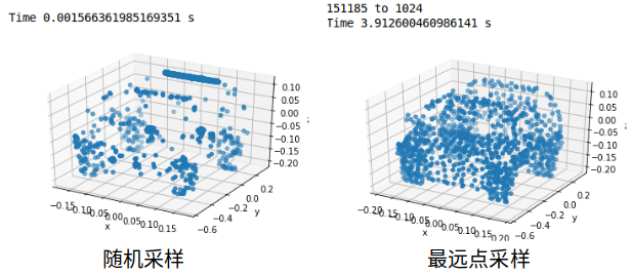

Figure 6. Comparison of random sampling and Farthest Point Sampling(FPS).

2.1.4 Poinc Cloud based recognition method

Because there are many classic methods for different representations, such as MVCNN with multi-view representation as input, and Subvolumn with raster representation as input, this article focuses on the method using point cloud as input, and focuses on Pointnet. Because Pointnet is a pioneering work, taking the original point cloud format as input and using deep neural networks to learn the shape characteristics of the point cloud end-to-end, and perform classification and segmentation. Before Pointnet, point clouds were often used as input for manual features.

Data Preprocessing As Figure 5 shows First perform point sampling, then unit sphere normalization, and finally data augmentation.

Point Sample There are many ways to sample points. First, random point sampling, that is, randomly select 2048 points from the 3D model points, or 2048 points from the face of the 3D model; second, Furthest Point Sampling(FPS), that is, selecting points one by one, requiring the current The distance between the point selection and the previous point is the largest, which can sample the points with more uniform spatial distribution, but the speed is also slower. As shown in Figure 6.

Unit Sphere Normalization The unit ball normalization makes the maximum distance between any two points normalized to 1. It is worth noting that this

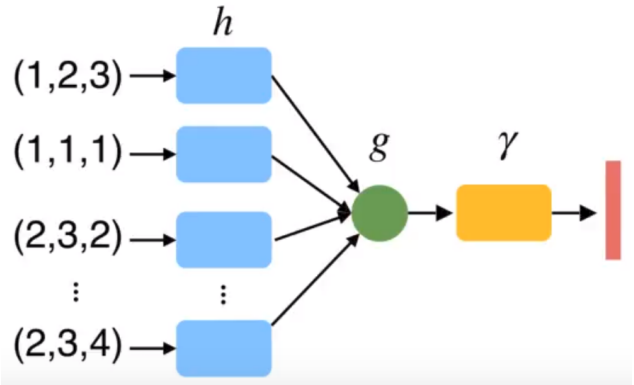

Figure 7. Pointnet simplified version.

standardization should be edge-preserving, not the traditional min-max normalization, as shown in the figure. The method is also very simple, just find the distance between any two points, take the maximum value, and then divide the coordinates of all points by the maximum value.

Data Augmentation There are only two steps for data enhancement: random rotation around the vertical axis, and random jitter of points. (Strangely, the accuracy rate has decreased after using data augmentation when recurring. Some people also have a similar situation in the official github issue.)

Model Architecture The data obtained after preprocessing is a point cloud matrix of $nx3$, which can be directly input into Pointnet.

Pointnet simplified version The simplest version of Pointnet is shown in the figure, learning a classifier function γ and a compound function for feature extraction

$$f(\{x_1,\ldots,x_n\}) \approx g(h(x_1),\ldots,h(x_n)) \qquad (1)$$

where f has Permutation invariance

$$f(x_1,x_2,\ldots,x_n) \equiv f(x_{\pi_1},x_{\pi_2},\ldots,x_{\pi_n}) \qquad (2)$$

It can be implemented with max function or sum function (experiments show that max function is better). The g maps the point cloud from the 3 dimension space to the 1024 dimension space, which increases the information redundancy, thereby retaining more information after maxpool.

Transform	accuracy
none	87.1
input (3x3)	87.9
feature (64x64)	86.9
feature (64x64) + reg.	87.4
both	**89.2**

Figure 8. T-Net loss, the rotation matrix of forced regression is an orthogonal matrix.

T-Net From the experimental results, T-Net will greatly increase the running time, and the effect is not particularly obvious (such as Figure ??), so it is not necessary. T-Net is actually a three-dimensional version of Spatial Transformer Networks (STN). The general idea is that the simplified version of Pointnet can implicitly learn attitude/rotation invariance, but implicitly is not as good as explicit. Therefore, the initial $nx3$ point cloud is input into another simplified version of Pointnet and an orthogonal rotation of $3x3$ is output. A matrix, multiplied with the original point cloud. This operation can also be extended to the point cloud feature of $nx64$ to predict an orthogonal rotation matrix of $64x64$. However, because there are too many $64x64$, it is difficult to predict accurately, so an auxiliary loss is added:

$$L_{reg} = \left\| I - AA^T \right\|_F^2 \tag{3}$$

Pointnet full version Pointnet takes $nx3$ point cloud as input, first uses T-Net to perform attitude calibration, and then uses mlp (actually implemented with $1x1$ convolution) to map the point cloud to high-dimensional space, and then uses maxpool to obtain global features Finally, use mlp as a classifier for classification. Part segmentation is actually treated as point-by-point classification. In addition, point-by-point features and global features need to be stitched. Such stitching is equivalent to searching the position of point-by-point features in the global.

critical point sets and upper bound point sets At maxpool, the feature contributions of each point are different. Find all the points that do not contribute to maxpool and delete, the rest is the set of key points; similarly, add points to the unit ball space infinitely, so that the global characteristics of maxpool will not change. Boundaries set. Official implementations such as Figure 10. Since there is no official method, I performed my reproduction Figure 11. Methods as below: For the key point set, just return the index when pointnet does maxpool. For the upper bound point set, first find a non-critical point in the original point cloud (this is to facilitate the addition of a new point), and then traverse the normalized space in steps of 0.02 ($[-1,1]x[-1,1]x[-1,1]$), and replace the non-critical points just found, then find the global features, and if the global features change, add upper bound set

Pointnet++ Pointnet lacks hierarchical feature extraction from local to global, which has been implemented in Pointnet ++ [6]. The means of implementation include three steps (Figure ??): Sampling, Grouping, Pointnet. Sampling the $nx3$ point cloud first to get m points, then draw m balls around these m points, and convolve the points in the ball to get local aggregation features This step is Grouping. Finally, the local features of mxc are obtained. Such an operation can also be repeated multiple times. Finally, enter it into Pointnet.

2.2. 3D Scenes Pointcloud Recognition

2.2.1 datasets

There are many datasets for 3D models. Here we only introduce the data sets used in this article.

Semantic segmentation Stanford 3D semantic parsing data set is an indoor scene semantic segmentation dataset with a point cloud of $Nx9$ matrix and 9 channels are X, Y, Z, RGB, normalized location.

3D Object Detection The KITTI dataset is captured by a binocular color camera and a 64-line LIDAR. Its 3D target detection benchmark contains categories including car, van, truck, pedestrian, pedestrian(sitting), cyclist, tram, misc. It contains 7481 training samples and 7518 tests Sample, about 80,000 3D tags, as shown in Figure 13

2.2.2 Representations

For 3D object detection tasks, different schemes have different feature representations.

Depth Map Representation The simplest one is to use the depth map as the fourth channel of RGB, and use 2D convolution to directly return to the 3D bounding box, but this method has lower accuracy.

Bird's Eye View Representation In 2017, Xiaozhi Chen first proposed to encode LIDAR point cloud data into a bird's eye view(Figure ??), and proposed an end-to-end neural network MV3D [7] that returns 2D

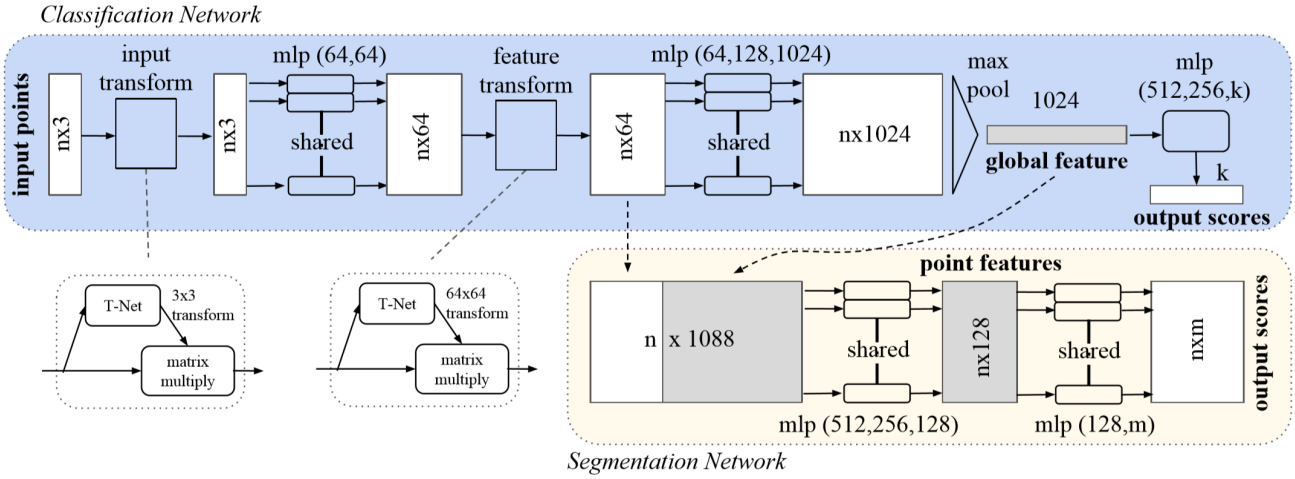

Figure 9. PointNet Architecture. The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by maxpooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

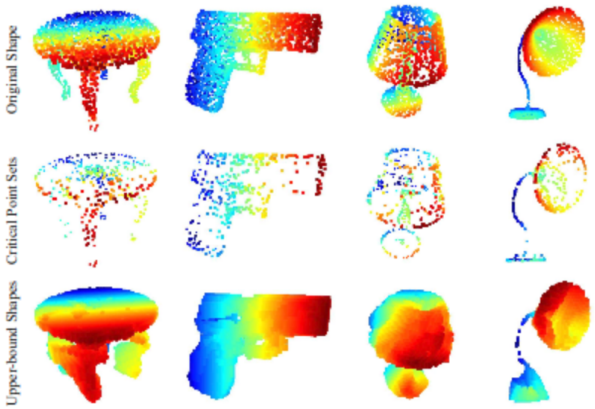

Figure 10. critical point sets and upper bound point sets (official implementation).

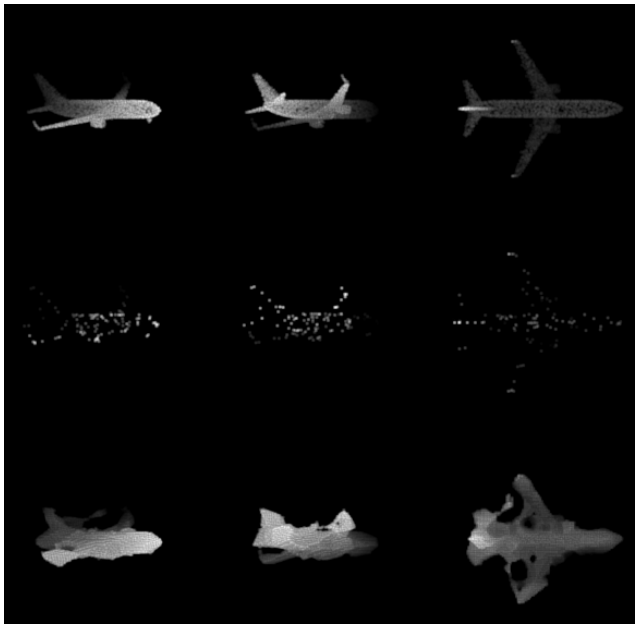

Figure 11. critical point sets and upper bound point sets (personally implementation).

bounding boxes from the bird's-eye view with 2D convolution. The method is still widely used up to now, but it is a manual feature and the quantization resolution needs to be selected in advance.

Point Cloud Representation In 2018 Charles Qi promotes Pointnet as Frustum-Pointnets [3] to solve 3D object detection problems. This method uses mature 2D detectors to obtain RoIs, then back-projects 2D RoIs using camera internal parameters to point cloud frustums, and finally uses Pointnet for segmentation and 3D target regression. The advantage of this method is that it is simple and suitable for both indoor and outdoor use.

Voxel Representation Unlike the Volumetric representation, although the concept of voxels is used here, it is actually a promotion of Frustum-Pointnets. Similar to Pointnet++ using balls to aggregate local features, VoxelNet [12] in 2018 Use voxels to aggregate local features. This method has very high speed and accuracy after applying the sparse convolution method [11], and it is a widely studied method now.

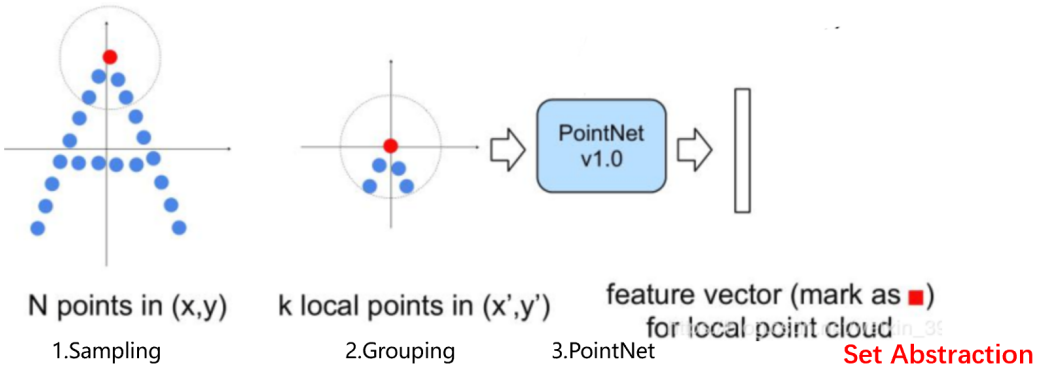

Figure 12. PointNet++ ideas.

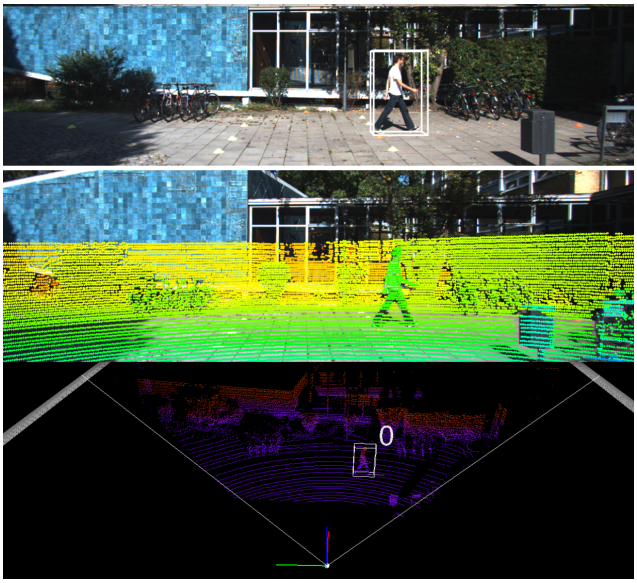

Figure 13. an example of KITTI data and label.

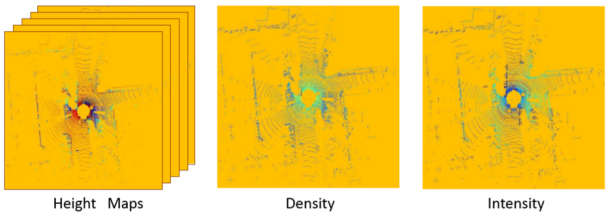

Figure 14. Bird's Eye View Representation

2.2.3 Frustum-Pointnets

Frustum-Pointnets is a direct promotion of Pointnets, and its architecture is shown in in Figure 15 and Figure 16.

Normalization Standardization is especially important in the training of F-Pointnets. Unlike spherical standardization of 3D models, Frustum-Pointnets is normalized as shown in Figure 17.

3. Ideas

pre-train Pre-training can provide a better initial value for the neural network, make the neural network converge faster, and often get better accuracy. Therefore, most 2D object recognition methods are pre-trained using ImageNet. However, the related work of 3D object detection based on point cloud so far has been to train from strach [3, 7, 11, 12].

Transfer from Model to Reality There are already dozens of 3D models and 3D scene datasets, and there are many types of tasks, but most of the related work is to design neural network architectures for specific datasets and tasks, regardless of the differences between Links between tasks. For example, how can the prior knowledge trained on the 3D model datasets ModelNet and ShapeNet be used reasonably for tasks in real scenarios.

I come up with two simple ideas at present, one is through pre-training. When training in Shapenet, make sure the preprocessing and normalization are consistent with Frustum-Pointnets, then the weights obtained from training can be used as the initial values of Frustum-Pointnets. Second, through data augmentation, Randomly place Shapenet's vehicle model in the real scene, simulate the real laser reflection effect, and render real lighting and texture and so on.

Transfer from Virtuality to Reality The game can provide a very rich 3D datasets(such as GTA5-LIDAR [8]) , and can save the cost of labeling. Therefore, there have been related works to study the field adaptation from virtual to real scenes, but these works are limited to the recognition of RGB images. In fact, the accuracy of the 3D object detection method based

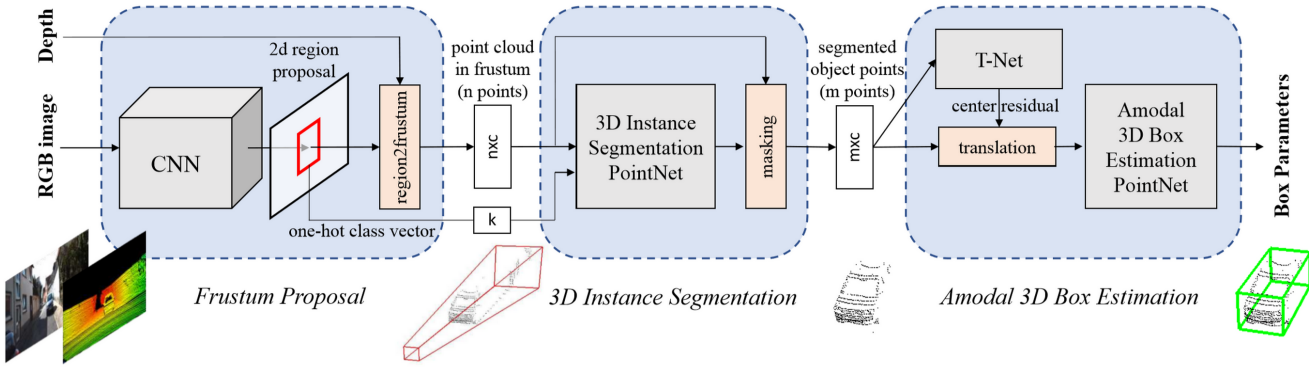

Figure 15. Frustum Pointnets Architecture.

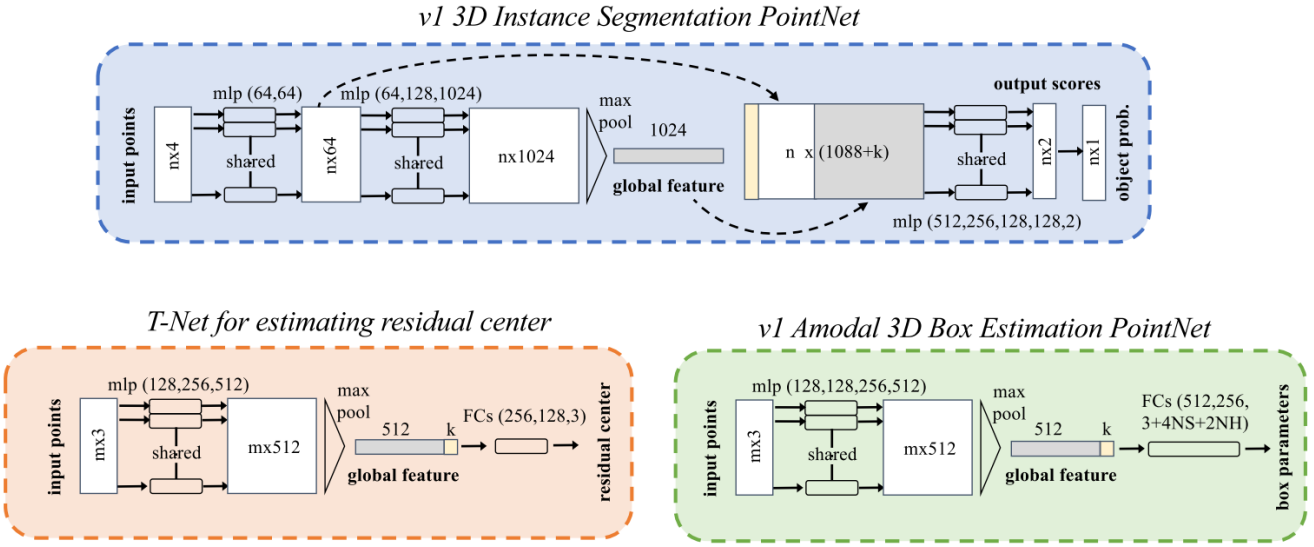

Figure 16. Frustum Pointnets Detailed Architecture.

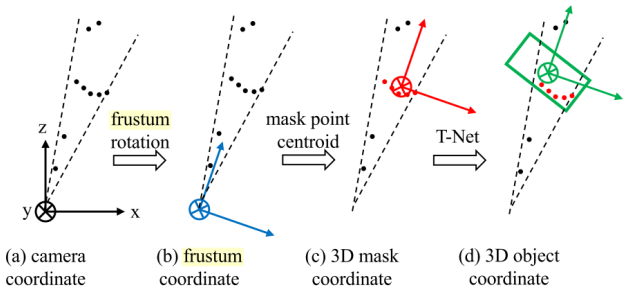

Figure 17. Frustum-Pointnets Normalization

on point cloud will also decrease sharply when moving from virtual to real.

Multi-scale Architecture In ModelNet training, I found that the maximum number of point clouds in the training set should be consistent with the maximum number of point clouds in the test set, otherwise the effect will be poor. However, in real scenes, the distance will cause a large difference in the number of point clouds, and different lidars (64 lines, 32 lines) will also cause a difference in the number of points. How to design a multi-scale module is also a problem.

Fusion of RGB and pointcloud Different feature representations require different fusion methods. For example, the depth map and bird's-eye view representations can be naturally merged with the image, although their fusion efficiency needs to be improved. For Pointnet using raw point cloud representation, we can try to integrate image features as point features into Pointnet. The papers that have been tried are PointFusion [10] and DenseFusion [1]

References

[1] S. Alikhanov, I. Konkashbaev, and P. Z. Chebotaev. The energy balance in a dense fusion plasma contained

by walls. Nuclear Fusion, 10(1):13, 1970. 7

[2] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. arXiv preprint arXiv:1807.00652, 2018. 2

[3] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 918–927, 2018. 5, 6

[4] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 652–660, 2017. 2

[5] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5648–5656, 2016. 2

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in neural information processing systems, pages 5099–5108, 2017. 2, 4

[7] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE international conference on computer vision, pages 945–953, 2015. 2, 4, 6

[8] B. Wu, A. Wan, X. Yue, and K. Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1887–1893. IEEE, 2018. 6

[9] 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. 1

[10] D. Xu, D. Anguelov, and A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 244–253, 2018. 7

[11] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. Sensors, 18(10):3337, 2018. 5, 6

[12] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4490–4499, 2018. 5, 6