Light bottle transformer based large scale point cloud classification*

XIE En[1], ZHANG Zhiyong[2], ZHANG Guodao[3], CHEN Pingkuo[4], and GE Yisu[5]**

1. College of Computer Information and Engineering, Nanchang Institute of Technology, Nanchang 330044, China

2. School of Information Engineering, East China University of Technology, Nanchang 330000, China

3. Department of Digital Media Technology, Hangzhou Dianzi University, Hangzhou 310023, China

4. Great Bay University, Dongguan 523000, China

5. College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325100, China

(Received 10 November 2022; Revised 9 February 2023)

With the rapid development of computer vision, point clouds technique was widely used in practical applications, such as obstacle detection, roadside detection, smart city construction, etc. However, how to efficiently identify the large scale point clouds is still an open challenge. For relieving the large computation consumption and low accuracy problem in point cloud classification, a large scale point cloud classification framework based on light bottle transformer (light-BotNet) is proposed. Firstly, the two-dimensional (2D) and three-dimensional (3D) feature values of large scale point cloud were extracted for constructing point cloud feature images, which employed the prior knowledge to normalize the point cloud features. Then, the feature images are input to the classification network, and the light-BotNet network is applied for point cloud classification. It is an interesting attempt to combine the traditional image features with the transformer network. For proving the performance of the proposed method, the large scale point cloud benchmark Oakland 3D is utilized. In the experiments, the proposed method achieved 98.1% accuracy on the Oakland 3D dataset. Compared with the other methods, it can both reduce the memory consumption and improve the classification accuracy in large scale point cloud classification.

Document code: A **Article ID:** 1673-1905(2023)06-0377-8

DOI https://doi.org/10.1007/s11801-023-2190-2

With the continuous advancement of laser scanning technology and computer vision, general two-dimensional (2D) images no longer satisfied the application requirements, and three-dimensional (3D) point cloud data is attracting more and more attention from researchers. The 3D point cloud is formed from a series of irregular and disordered 3D points, which contains the geometric information. It can better represent the geometric feature of objects and is more accurately describe the spatial distribution information of objects. The 3D point cloud information is employed to digitalize real-world objects and scenes on computer and can be widely utilized in many fields, such as robotics, autonomous driving, and 3D city modeling. LiDAR technology provides fantastic convenience for obtaining point cloud data[1]. The collected point cloud data both contains the spatial information of points and material features of the target object. For further utilizing the collected data for autonomous driving and city digitization, different data points in point cloud should be classified. With the enlargement of the measurement environment, the amount of point cloud is increased rapidly, which amplifies the computational burden. Furthermore, the wider environment expands the problem of the laser scanning distance variation, which makes the uneven distribution of cloud points. Especially in the urban transport scene, the target size is quite different.

For accelerating the point cloud classification, uniform sampling is a fundamental method to reduce the processing data. But uniform sampling of the large scale point cloud may miss vital information, which will affect the further analysis of point cloud processing, and may cause a serious accident in practical application, such as missing obstacles in autonomous driving. In addition, one of the important requirements of the practical applications is real-time, which means the large scale point cloud classification method should be efficient as well. How to design a framework for large scale point cloud classification, which both achieves high accuracy and efficiency? It is a valuable problem.

* This work has been supported by the National Natural Science Foundation of China (No.71872131), the STU Scientific Research Initiation Grant (No.20007), the Wenzhou Science and Technology Plan Project (No.G20220035), and the General Scientific Research Fund of Zhejiang Provincial Education Department (No.Y202248776).

** E-mail: ysg@wzu.edu.cn

At present, point cloud classification methods[2] can be roughly divided into four categories, handcrafted feature-based methods, projection-based deep learning methods, voxel-based deep learning methods, and point-based deep learning methods.

Before the notable success of deep learning technique, the handcraft feature-based method is the most useful way for point cloud classification. The local geometric features are extracted from the neighborhoods of point cloud data, and a traditional machine learning classifier is employed for further processing. For example, WEINMANN et al[1] demonstrated that the selection of optimal neighborhoods for individual 3D points significantly improves the results of 3D scene analysis, and a detailed evaluation involving 7 neighborhood definitions, 21 geometric features, 7 approaches for feature selection, and 10 classifiers are compared. The machine learning-based classification methods construct a model to classify the point cloud by training. The random forest (RF) and support vector machine (SVM) are the most common methods for point cloud classification[3]. Moreover, conditional random field (CRF) is also used in point cloud classification, such as NIEMEYER et al[4] established a two-layer framework of space and semantic context by CRF, and improved the classification performance through point clouds segments random generating an iterative calculation. The handcraft features are reasonable and useful, but the limitation of classifiers affects the performance of point cloud classification.

After the deep neural network was proposed, the point cloud classification method based on deep learning has attracted wide attention in academia. There are three different ways to combine the point cloud with the deep neural network, projection-based deep learning methods, voxel-based deep learning methods, and point-based deep learning methods. Projection-based methods convert 3D disordered point cloud data into 2D images and classifies 2D images by convolutional neural network (CNN), such as ROVERI et al[5]. The projection-based methods benefit from the well-established 2D CNN, but the projection operation may cause the redundant computation or miss parts of geometric information of point cloud. Voxel-based methods divide the point cloud as the 3D regular cubes, and make use of 3D CNN to process each cube, such as VoxNet[6]. In the large scale point cloud, the sizes of object are various, which masks the uneven distribution of point cloud. The voxel-based methods partitioning the big problem into small one speed up the efficiency, but the voxels of point cloud loss the global information. For extracting the feature of point cloud automatically, point-based methods were studied. QI et al[7] advanced PointNet framework, which uses symmetric functions to solve the disorder problem, and combines T-Net matrix to integrate point cloud features for keeping the rotation invariance. Multiple-feature extraction strategy was applied in a latter version named PointNet++[8]. Differently, LI el al[9] utilized multi-layer perceptron (MLP) and X-transform matrix to deal with the point cloud disorder problem. Point-based methods both obtain the local and global features in point cloud, but processing each point is a high time complexities work.

The point cloud classification frameworks have made great progress in recent years, but there is still a certain distance from practical requirement on performance and efficiency. The traditional handcraft feature is an interpretable and efficient way to extract the point cloud feature, and CNN as a classifier shows the great potential of post process. Therefore, a fusion framework is proposed in this paper to solve the large scale point cloud classification problem. The network input feature images consist of the 2D and 3D geometric information in point cloud. Then, a light weight neural network combining the transformer and residual network is utilized to classify the point cloud. The fusion framework both contains the advantages of handcraft feature-based methods and deep learning-based methods.

With the rise of artificial intelligence, deep learning has been widely used in image-based scene analysis and has obtained great successes. The core problem of point cloud classification is to design features with strong distinguishability. Before the rise of deep learning, the processing of point cloud data was mainly based on artificial features designed by humans, while the existing geometric features and physical attribute features have been saturated stage. The point cloud classification method based on deep learning assigns the task of feature design to the computer, instead of manually selecting the feature description, the 3D coordinates of the point are directly used as input, and the feature is also defined as an unknown parameter, calculate these parameters together with the classifier parameters. The goal of deep learning is to find the parameter value that minimizes the loss function. Therefore, deep learning is guided by the classification results, and simultaneously solves the optimal features and the optimal classifier. Although deep learning algorithms have achieved great success in the fields of image segmentation and object recognition, there is no mature solution for applying deep learning to the single-point classification of point clouds. There are still problems worthy of further discussions, such as how to make the network model take into account high precision and high efficiency, how to use the connection between points more fully and flexibly, and so on.

Most point cloud classification methods based on deep learning are formed alternately by the convolution layer and pool layer. The neurons in the convolution layer are only connected with part of the upper layer and partial features are learned, which may lose part of the information in feature extraction, and leads to an accuracy decline. The transformer brings a new way of feature extraction, which links the global features in each layer. The transformer was first presented in the field of natural language processing (NLP) and achieved great success,

which applied a self-attention mechanism to extract intrinsic features. Inspired by the function of transformers in NLP, researchers began to apply transformer in computer vision. For example, stand-alone self-attention (SASA)[10] is a transformer model that replaces the convolution layer in ResNet bottleneck with different forms of self-attention (local, global, vector, axial, etc). By training sequence transformer to auto-regressively predict pixels, CHEN et al[11] achieved comparable results to CNN in image classification tasks. VASWANI et al[12] advanced to stack transform blocks based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.

In the task of large scale 3D point cloud data classification, handcraft features are computational friendly, and deep learning methods are good at potential feature extraction. In the deep learning methods, CNNs are adept in extracting local features, while the attention mechanism in transformer is expert in extracting the global information. BotNet is a combination of CNN and transformer by SRINIVAS et al[13]. It combines the local feature extraction ability of CNN and the self-attention mechanism of transformer in a hybrid way, and the performance is both better than CNN and transformer. Hence, for balancing the performance with computational burden, a light-BotNet is proposed to extract and aggregate features in this paper. The proposed framework fuses the traditional handcraft features, CNNs and transformer to achieve the better result of point cloud classification. The contribution of this paper is as follows.

The geometric information of 3D point cloud is used to construct the feature image as the input of network, which combines the traditional features with deep learning methods to make the solution more interpretability.

For improving the efficiency of framework, a lightweight network model light-BotNet is proposed based on the BotNet, which gathers the benefit both on CNNs and transformer.

A framework of large scale point cloud classification is constructed by the 32×32 point cloud feature matrix image and light-BotNet network, which achieves 98.1% accuracy on Oakland 3D dataset.

Tab.1 Point based features

Type	Components
Geometric 3D feature	L_λ, N_x, N_y, N_z, P_λ, S_λ, M_x, M_y, M_z, O_λ, A_λ, E_λ, T_λ, C_λ, D, Q, V
Geometric 2D feature	r_k, D_{2D}, $R_{\lambda,2D}$, *Evratio*, S_2

The geometric information is an important property of point cloud, and the geometric information relays on the neighborhood of each point in the point cloud. So, the geometric information of 3D point cloud is applied as the input features of framework in this work. The 2D and 3D geometric features are both support the point cloud classification. Therefore, there are 17 3D features and 15 2D features combined into a 2D feature image as shown in Fig.1. For further promoting the feature extraction, multi-level feature combination is adopted in preprocessing. The features of each point are calculated by the *K-D* tree (K=50), as listed in Tab.1.

The 3D structure tensor $\boldsymbol{S}\in\mathbb{R}^{3\times3}$ is constructed for a given 3D point $\boldsymbol{X}=\boldsymbol{X}_0$ by involving its K closest neighbors $\boldsymbol{X}_i$ with i=1,2,3,…, K,

$$\boldsymbol{S}=\frac{1}{K+1}\sum_{i=0}^{k}\left(\boldsymbol{X}_i-\bar{\boldsymbol{X}}\right)\left(\boldsymbol{X}_i-\bar{\boldsymbol{X}}\right)^{\mathrm{T}}, \tag{1}$$

$$\bar{\boldsymbol{X}}=\frac{1}{K+1}\sum_{i=0}^{K}\boldsymbol{X}_i, \tag{2}$$

where $\boldsymbol{S}$ is a non-negative matrix, whose rank is 3 in general. Therefore, the feature values λ_1, λ_2, $\lambda_3\geq0$, and the feature values determine a set of orthogonal vectors. Sort the feature values so that $\lambda_1\geq\lambda_2\geq\lambda_3$. The feature values are used to further extract the 3D geometric feature of 3D point cloud data, as shown in Tab.1. There are 22 different geometric feature elements in geometric feature vector, and each feature is listed as follows.

Features based on the local surface variation C_λ are as follows

$$C_\lambda=\frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3}, \tag{3}$$

where λ_1, λ_2, and λ_3 can derive dimensional features, such as linearity L_λ, planarity P_λ and scattering S_λ, which represent 1D, 2D, 3D features:

$$L_\lambda=\frac{\lambda_1-\lambda_2}{\lambda_1}, \tag{4}$$

$$P_\lambda=\frac{\lambda_2-\lambda_3}{\lambda_1}, \tag{5}$$

$$S_\lambda=\frac{\lambda_3}{\lambda_1}. \tag{6}$$

The 3D local point density D is

$$D=\frac{K+1}{\frac{4}{3}\pi r_{K\text{-NN}}^3}, \tag{7}$$

where $r_{K\text{-NN}}$ denotes the radius of the local 3D neighborhood encapsulating the K closest neighbors.

For estimating the order/disorder of 3D points within the local 3D neighborhood, the three feature values λ_1, λ_2, and λ_3 are normalized by their sum $\Sigma\lambda$, and the normalized feature values $e_i=\lambda_i/\Sigma\lambda$ with $i\in\{1,2,3\}$ and $e_i\in[0,1]$ thus sum up to 1. In the calculation of the 3D feature vector, some local 3D shape features such as omnivariance O_λ, verticality V (n_z is from e_3, which is the vertical component of the normalized vector), anisotropy A_λ and eigenentropy E_λ are

$$O_\lambda=\sqrt[3]{e_1e_2e_3}, \tag{8}$$

$$V=1-\left|n_z\right|, \tag{9}$$

$$A_\lambda=\frac{e_1-e_3}{e_1}, \tag{10}$$

$$E_\lambda = -\sum_{i=1}^{3} e_i \ln(e_i). \tag{11}$$

In the 2D geometric features, the 2D feature values are obtained by the point cloud projecting on three different coordinate planes, and $\lambda_{1,2D}$ and $\lambda_{2,2D}$ are the feature values of 2D structure tensor. $r_{K\text{-NN},2D}$ is the radius of the circular neighborhood defined by 2D point and its K closest neighbors, and D_{2D} represents 2D local point density, which is the analogy to the 3D case.

$$D_{2D} = \frac{K+1}{\pi r_{K-\mathrm{NN},2D}^2}. \tag{12}$$

$R_{\lambda,2D}$ means the ratio of eigenvalues in a 2D covariance matrix.

$$R_{\lambda,2D} = \frac{\lambda_{2,2D}}{\lambda_{1,2D}}. \tag{13}$$

Evratio is the ratio of eigenentropy on different coordinate axes (*axisa*=*x*, *y*, *z*, *axisb*=*y*, *z*, *x*), and $E_{\lambda 2D}$ is the 2D eigenentropy.

$$Evratio = \frac{E_{\lambda 2D}(axisa)}{E_{\lambda 2D}(axisb)}. \tag{14}$$

S_2 implies sum of the feature values.

$$S_2 = \lambda_{1,2D} + \lambda_{2,2D}. \tag{15}$$

T_λ is the trajectory.

$$T_\lambda = \frac{2}{\pi}\arctan(\lambda_1 + \lambda_2 + \lambda_3). \tag{16}$$

Then, the features are grouped together and the feature image $\boldsymbol{f}$ is constructed. The feature image is a diagonal matrix as presented in Eqs.(17)—(19). $\boldsymbol{f}_{3D}$ denotes the 3D features. $\boldsymbol{f}_{2D}(x)$, $\boldsymbol{f}_{2D}(y)$, and $\boldsymbol{f}_{2D}(z)$ are the features obtained by the point cloud projecting on three different coordinate planes, respectively. $\boldsymbol{I}_{32}$ is a 32 identity matrix.

$$\boldsymbol{f}_{3D} = [L_\lambda, N_x, N_y, N_z, P_\lambda, S_\lambda, M_x, M_y, M_z, O_\lambda, A_\lambda, E_\lambda, T_\lambda, C_\lambda, D, Q, V], \tag{17}$$

$$\boldsymbol{f}_{2D} = [r_k, D_2, R_{\lambda,2D}, Evratio, S_2], \tag{18}$$

$$\boldsymbol{f} = \mathrm{concat}[\boldsymbol{f}_{3D}, \boldsymbol{f}_{2D}(x), \boldsymbol{f}_{2D}(y), \boldsymbol{f}_{2D}(z)] \cdot \boldsymbol{I}_{32}. \tag{19}$$

The process of feature image generating is illustrated in Fig.1. The feature image bridges the handcraft feature and neural network, which has simple, interpretable and efficient advantages. The 2D CNNs are well-established and mutated, and the feature image is a flexible way to get the benefit from CNNs.

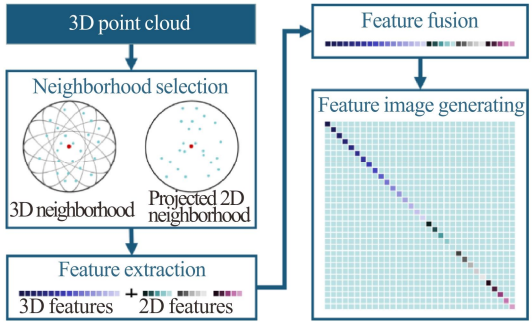


Fig.1 Process of feature images generation

Tab.2 Comparison of three different neural networks

Layer name	ResNet50	BotNet50	Light-BotNet
*Conv*1_*x*	3×3, 64, stride 1 3×3 max pool, stride 2		
*Conv*2_*x*	$\begin{bmatrix}1\times1 & 64\\ 3\times3 & 64\\ 1\times1 & 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 64\\ 3\times3 & 64\\ 1\times1 & 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 64\\ 3\times3 & 64\\ 1\times1 & 64\end{bmatrix}\times1$
*Conv*3_*x*	$\begin{bmatrix}1\times1 & 128\\ 3\times3 & 128\\ 1\times1 & 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1 & 128\\ 3\times3 & 128\\ 1\times1 & 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1 & 64\\ 3\times3 & 64\\ 1\times1 & 128\end{bmatrix}\times3$
*Conv*4_*x*	$\begin{bmatrix}1\times1 & 256\\ 3\times3 & 256\\ 1\times1 & 1\,024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1 & 256\\ 3\times3 & 256\\ 1\times1 & 1\,024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1 & 64\\ 3\times3 & 64\\ 1\times1 & 256\end{bmatrix}\times1$
*Conv*5_*x*	$\begin{bmatrix}1\times1 & 512\\ 3\times3 & 512\\ 1\times1 & 2\,048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 512\\ \mathrm{MSHA} & 512\\ 1\times1 & 2\,048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 128\\ \mathrm{MSHA} & 128\\ 1\times1 & 512\end{bmatrix}\times1$

BotNet is a fusion model which makes up the superiorities of CNN and transformer, and the performance is both better than CNN and transformer. Light-BotNet is a modified network based on BotNet[13] with light-weight

design, which reduces calculations and parameter amount of the BotNet. A hybrid model of convolution and self-attention mechanisms is applied in the network, which further combines the advantages from CNN and transformer. The CNN network layers can effectively learn abstract and low-level features in large scale images, and the global self-attention of transformer can obtain the information contained in the whole feature maps. However, the self-attention block in transformer consumes too much computing resources, and the multi-head self-attention (MHSA) layer is applied to instead the original self-attention block with 3×3 convolution layer. The light-BotNet uses 2×2 average pooling for down sampling, because MHSA module does not support stride operation, and down sampling can reduce the calculation of network. The positional encoding strategy is equipped in the light-BotNet network for effectively associating information between objects with location perception, which is also suitable for computer vision tasks[14]. The network of light-BotNet refers to transformer block, self-attention with relative position and none local structure. The construction of different network structures is presented in Fig.2.

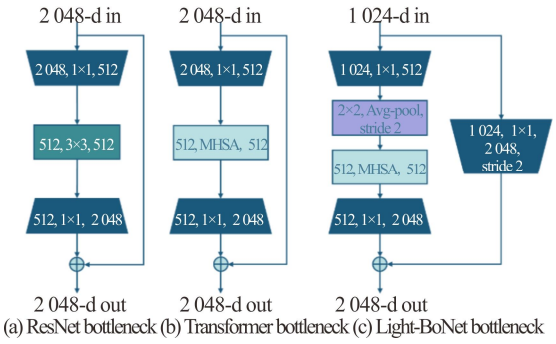


Fig.2 Contrast of bottleneck in different networks

In Fig.2, (a) shows the bottleneck in ResNet, (b) presents the transformer bottleneck with MHSA, and (c) illustrates the bottleneck structure in light-BotNet. The difference is that the 3×3 convolutional layer is replaced with MHSA and an average pooling layer is supplemented for reducing the calculation of network features and maintaining the feature learning ability. The residual link avoids the gradient disappearance, and helps the network converge stably.

The specific network of light-BotNet is compared with the ResNet and BotNet50 in Tab.2. Due to the input size of the feature images, the layer *Conv*1_*x* is modified for suiting the input structure, the 7×7 convolution layer with stride 2 is replaced by the 3×3 convolution layer with stride 1. The structure of light-BotNet is similar to the other two networks, but the repeated block amount is smaller for reducing the computation consumption. The latter block of light-BotNet uses the special bottleneck as presented in Fig.2.

The proposed framework utilizes the handcraft geometric feature as the preprocessing, and employs the light-BotNet as the classifier for post processing. The network of large scale point cloud classification based on light-BotNet is illustrated in Fig.3. First, the neighborhood of 3D *K-D* tree (K=50) and projected 2D neighborhood are selected respectively, and the feature values of structure tensor is obtained. Then, the features of 2D and 3D are extracted for feature images generating. Next, the feature images are inputted into the light-BotNet network for point cloud classification, and the network adaptively selects useful information from the point cloud features to classify the 3D point clouds.

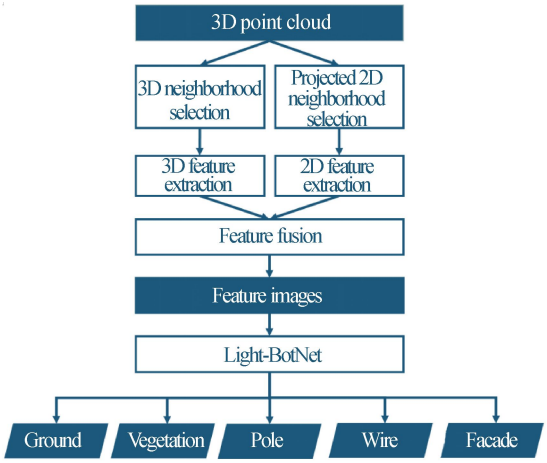


Fig.3 Flowchart of light-BotNet

The fusion structure is a valid way to balance the efficacy and performance of large scale point cloud classification. The geometric feature selection not only avoids the information missing like voxel-based methods, but also decreases the calculation of point cloud, unlike the point-based deep learning methods. One of the shortcoming of neural network is the high computational complexity, and if the framework should be popular to apply in the practical application, the network should be light-weight. Therefore, the light-BotNet is designed to select and classify the point cloud in the post processing of the framework.

This work provides experiments on the Oakland 3D large scale point cloud dataset to verify the effectiveness and robustness of the proposed framework, and conducts the analysis of experimental results. The experiments are implemented in Python 3.7 and run on an Intel i7-4790 processor with 3.6 GHz central processing unit (CPU), a NVIDIA TITAN RTX graph processing unit (GPU) and 32 GB random access memory (RAM). The framework is built in Pytorch 0.6, CUDA 10.0, and CUDNN7.6.4 in Window10. The Adam optimizer is applied in the network training, and the initial learning rate is set to 10^{-3} which is gradually decreased by a factor of 10 every 40 epochs.

The Oakland 3D large scale point cloud dataset is derived from the campus surrounding of Carnegie Mellon

University in Oakland by the most widely used mobile laser scanning (MLS). The dataset mainly covers various objects in the large-scale urban environment, including wires, poles, facade, ground, and vegetation. The sample amount of each category is listed in Tab.3. As we can see, vegetation and facade are the major part in train dataset, while the vast majority data of test dataset is ground.

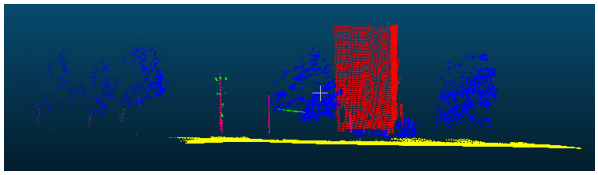
Tab.3 Subsample data of Oakland 3D dataset

Category	Train dataset	Test dataset
Vegetation	14 441	9 278
Wire	2 571	481
Pole	1 086	368
Ground	4 713	71 863
Facade	14 121	7 821
Total	36 932	89 811

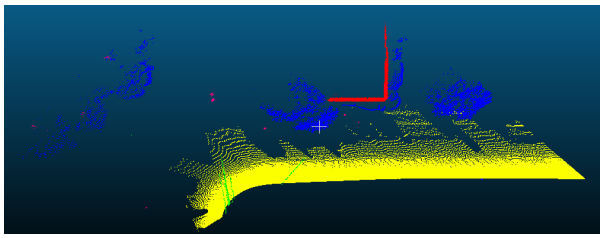
In this paper, the performance of the proposed framework is compared with different methods on Oakland dataset and the results are listed in Tab.4, where OA represents the overall classification accuracy. Comparing with the other methods[1,3,15-18], the framework gets the best performance in overall accuracy, of which the accuracy reaches 98.1%. As presented in Tab.4, the facade classification result of the proposed framework is much better than other methods, and the classification performance in vegetation and ground are similar to the best results, but the results of pole and wire are disappointing. Because the number of cloud point in facade, ground and vegetation categories are much more than the number in pole and wire, the accuracy of overall is significantly affected by the point cloud with large number. When the 2D feature of small number point cloud is obtained, the points are projected in x, y, z directions, and different points may overlap on the same plane. It will influence the classification in the objects with small number point cloud, but benefit for large objects. The visualization results are depicted in Fig.4.

Tab.4 Comparison with other methods on Oakland dataset

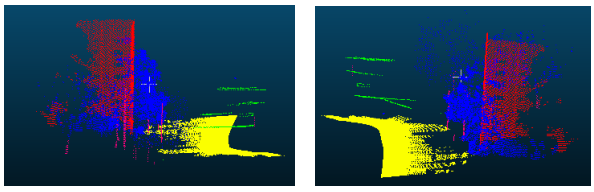
Method	Pole (%)	Vegetation (%)	Wire (%)	Ground (%)	Facade (%)	OA (%)
WANG[1]	70.11	80.55	**93.08**	98.22	70.95	94.68
RF[1]	79.99	84.41	86.05	98.48	67.01	92.25
SVM[2]	78.1	79.46	82.99	94.92	64.39	89.1
MUNOZ[3]	28.7	97.4	12.5	98.2	90.8	91.66
MRF[5]	68.0	95.5	51.3	98.4	92.9	97.0
CCM[6]	**82.67**	**97.83**	30.26	99.17	90.33	97.59
Our method	20.7	93.0	18.0	99.6	**98.7**	**98.1**



(a) Front view



(b) Top view

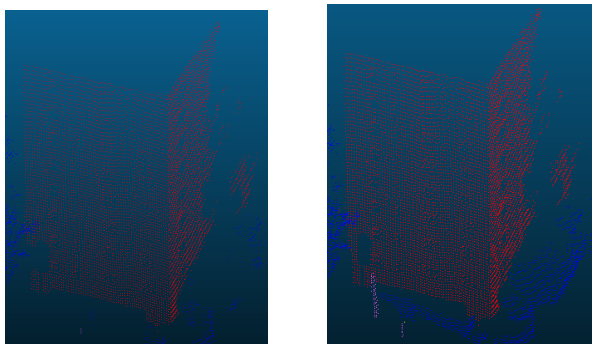


(c) Left view (d) Right view

Fig.4 The visualization results on Oakland dataset

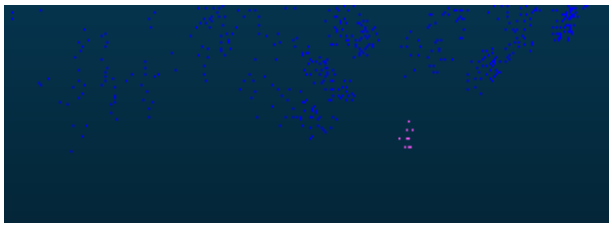
Fig.4(a), (b), (c) and (d) are the front view, top view, left view and right view of results, respectively, and the yellow, green, purple, red, blue points mean ground, wire, pole, facade, and vegetation, respectively.

Fig.5 shows the visual comparison of results and ground truth on facade point cloud, where (a) is the result obtained by the proposed framework, and is the ground truth. The facade classification is pretty good. Fig.6 is the visual comparison on wire point cloud, where (a) is the result, and (b) is the ground truth.

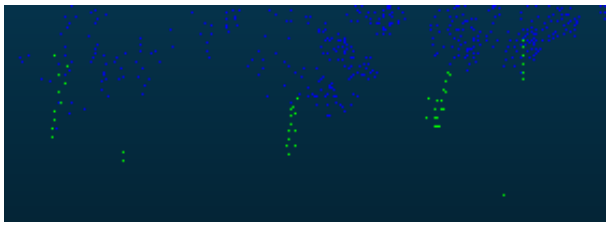


(a) Facade label result (b) Ground truth

Fig.5 Comparison of facade point cloud



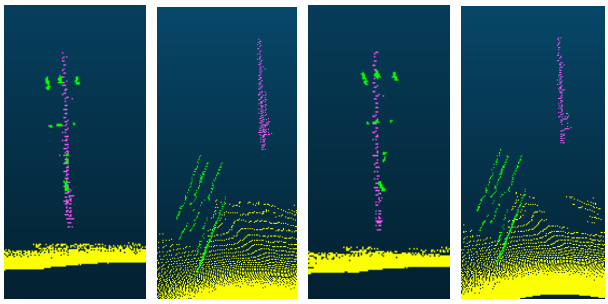
(a) Wire label result



(b) Ground truth

Fig.6 Comparison of wire point cloud

Fig.7(a) and (b) are the visualization of the results, and Fig.7(c) and (d) are the ground truth of the test set.



(a) (b) (c) (d)

Fig.7 Comparison of other label point clouds: (a) (b) Visualization of the results; (c) (d) Ground truth of the test set

The result of experiments show that the proposed method can achieve good result to classify the big object in point cloud, especially the objects are relatively large in multiple dimensions, such as vegetation, ground, and facade. And the object is small in one dimension like wire and pole, of which the feature might be vanished in 2D projection, and the classification results are unsatisfied. The big object is the main part in the dataset, so the overall accuracy of our method can be the best.

In order to prove the performance of light-BotNet based on channel attention mechanism, the performance and memory consumption of three networks are evaluated, and the results are presented in Tab.5. By comparing the performance and memory consumption of the three networks in Tab.5, the networks ResNet50 and BotNet50 have lower accuracy than light-BotNet, because the large scale point cloud has already been preprocessed in the proposed framework, and the deeper network models provide no advantage. Although the ResNet50 backbone can be applied to multiple models to extract effective features, it may not be suitable for this framework. Comparing the consumption of all networks, light-BotNet reduces 50% network layers, and forms a light-weight transform network model. There is little difference between BotNet50 and ResNet50 in the calculation flops and network parameters, but light-BotNet gets the best result with nearly 50% consumption reduction.

Tab.5 Comparison of network performance and memory consumption

Network	Flops (byte)	Params (byte)	Accuracy
Light-BotNet	**23 068 672 000**	**10 808 773**	**98.1%**
BotNet50	41 799 385 088	23 513 285	90.5%
ResNet50	40 596 406 272	18 799 301	75.1%

Aiming at the problems of the large scale 3D point cloud classification, a fusion framework based on handcraft features, CNN, and transformer is proposed. The 2D and 3D geometric features of large scale point clouds were extracted for constructing point cloud feature images as the preprocessing. The light-weight network named light-BotNet is provided for point cloud classification, which reduces the framework parameters and improves the performance of large scale point cloud classification. In the experiment, a variety of point cloud classification methods are compared in the Oakland 3D large scale point cloud dataset, and the proposed framework gets better performance than other methods in overall classification accuracy.

In the process of feature image construction, only the spatial features are applied to deal with the point cloud disorder and rotation invariance problem, and some other information about the point cloud such as object color is not used. How to combine the multi-level features in large scale 3D point clouds, and which is the most useful feature for point cloud classification are our future works.

Ethics declarations

Conflicts of interest

The authors declare no conflict of interest.

References

[1] WEINMANN M, SCHMIDT A, MALLET C, et al. Contextual classification of point cloud data by exploiting individual 3D neigbourhoods[C]//Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, March 25-27, 2015, Munich, Germany. Munich：Copernicus Publications, 2015, 2：271-278.

[2] GUO Y, WANG H, HU Q, et al. Deep learning for 3D point clouds：a survey[J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 43(12)：4338-4364.

[3] WANG L, MENG W, XI R, et al. 3D point cloud analysis and classification in large-scale scene based on deep learning[J]. IEEE access, 2019, 7：55649-55658.

[4] NIEMEYER J, ROTTENSTEINER F, SÖRGEL U, et al. Hierarchical higher order CRF for the classification of airborne lidar point clouds in urban areas[C]//Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information, July 12-19, 2016, Prague, Czech Republic. Hannover：Leibniz Universität Hannover, 2016, 41：655-662.

[5] ROVERI R, RAHMANN L, OZTIRELI C, et al. A network architecture for point cloud classification via automatic depth images generation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 19-21, 2018, Salt Lake City, USA. New York：IEEE, 2018：4176-4184.

[6] MATURANA D, SCHERER S. Voxnet：a 3D convolutional neural network for real-time object recognition[C]//Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 28-October 03, 2015, Hamburg, Germany. New York：IEEE, 2015：922-928.

[7] QI C R, SU H, MO K, et al. Pointnet：deep learning on point sets for 3D classification and segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, USA. New York：IEEE, 2017：652-660.

[8] QI C R, YI L, SU H, et al. Pointnet++：deep hierarchical feature learning on point sets in a metric space[C]//Proceedings of the Advances in Neural Information Processing Systems, December 4-9, 2017, Long Beach, CA, USA. New York：Curran Associates Inc., 2018：30.

[9] LI Y, BU R, SUN M, et al. Pointcnn：convolution on x-transformed points[C]//Proceedings of the Advances in Neural Information Processing Systems, December 3-8, 2018, Montreal, Canada. New York：Curran Associates Inc., 2019, 31：1-11.

[10] RAMACHANDRAN P, PARMAR N, VASWANI A, et al. Stand-alone self-attention in vision models[C]//Proceedings of the Advances in Neural Information Processing Systems, December 8-14, 2019, Vancouver, Canada. New York：Curran Associates Inc., 2020, 32：1-12.

[11] CHEN M, RADFORD A, CHILD R, et al. Generative pretraining from pixels[C]//Proceedings of the International Conference on Machine Learning, July 12-18, 2020, online. New York：PMLR, 2020：1691-1703.

[12] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the Advances in Neural Information Processing Systems, December 4-9, 2017, Long Beach, CA, USA. New York：Curran Associates Inc., 2018：30.

[13] SRINIVAS A, LIN T-Y, PARMAR N, et al. Bottleneck transformers for visual recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 19-25, 2021, online. New York：IEEE, 2021：16519-16529.

[14] BELLO I, ZOPH B, VASWANI A, et al. Attention augmented convolutional networks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, June 16-20, 2019, Long Bench, CA, USA. New York：IEEE, 2019：3286-3295.

[15] MUNOZ D, VANDAPEL N, HEBERT M. Directional associative markov network for 3-D point cloud classification[J]. Carnegie Mellon University, 2008.

[16] LI Y. Research on building classification and structure line extraction for urban vehicle borne laser point cloud data[D]. Nanjing：Nanjing Normal University, 2019. (in Chinese)

[17] ELONG H, HONGPING W, QI C, et al. An improved contextual classification method of point cloud[J]. Acta geodaetica et cartographica sinica, 2017, 46(3)：362-370.

[18] WEINMANN M. Feature relevance assessment for the semantic interpretation of 3D point cloud data[C]//Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, November 27-29, 2013, Istanbul, Turkey. Munich：Copernicus Publications, 2013：2.