Slice-sampling based 3D Object Classification

XiangWen Zhao zhaoxw@mail.sdu.edu.cn
School of Computer Science and Technology, Shandong University, Qingdao, China
State Key Laboratory of Astronautic Dynamics, Xi'an, China

Yi-Jun Yang yangyijun@xjtu.edu.cn
School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China

Wei Zeng wz@xjtu.edu.cn
School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China

Liqun Yang lyang028@fiu.edu
School of Computing and Information Science, Florida International University, Miami, U.S.

Yao Wang wangyaosdu@mail.sdu.edu.cn
School of Computer Science and Technology, Shandong University, Qingdao, China

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Multiview-based 3D object detection achieved great success in the past years. However, for some complex models with complex inner structures, the performances of these methods are not satisfactory. This paper provides a method based on slide sampling for 3D object classification. First, we slice and sample the model from the different depths and directions to get the model's features. Then, a deep neural network designed based on the attention mechanism is used to classify the input data. The experiments show that the performance of our method is competitive on ModelNet. Moreover, for some special models with simple surfaces and complex inner structures, the performance of our method is outstanding and stable.

Keywords: 3D object classification, slice-sampling, attention mechanism

1. Introduction

With the development of 3D shape retrieval, shape recommendation, computer-aided design, and Game AI, the role of 3D shape classification is becoming more and more significant. Although deep learning has been successful in many fields, there are still many challenges in using neural networks to recognize 3D shapes. As a collection of 3D shape vertices, edges, and faces, the mesh provides a very efficient and non-uniform representation for 3D shapes to store and render computer graphics tasks. But 3D shape represented as mesh is different from Euclidean data such as images and natural language. Thus it is difficult to apply deep neural networks to 3D shape classification directly. Since the face number of the 3D shape and the adjacency relationship is not fixed, the number of input neurons cannot be determined. The convolution operation cannot be directly used like in images. In order to solve these problems, researchers have proposed many methods based on voxel [Wu et al. (2015a);Maturana and Scherer (2015)], mesh [Hanocka et al. (2019);Feng et al. (2019)] and image [Su et al. (2015);Han et al. (2019;2018;)]. These works generally focus on providing descriptors for 3D shapes suitable for deep neural network input and defining convolution operations based on their representation.

The image-based method has excellent classification accuracy, such as 90.1% in MV-CNN[Su et al. (2015)] on dataset ModelNet40[Wu et al. (2015a)]. It fully uses the achievements of deep learning in image classification and solves the problems mentioned above in 3D recognition through multi-view rendering. Image-based methods usually render images from multiple viewpoints surrounding a 3D shape and then use these rendered images as the input of a deep neural network to complete the classification of the model[Su et al. (2015)]. Nevertheless, the image-based 3D shape classification still has some problems. Firstly, the inner structure of the model cannot be rendered into the image, so that important features of the model may be lost. Secondly, it is impossible to obtain 360 surrounding the model in some occluded cases, resulting in a lack of complete or valid input for the classification network.

Recently, with the development of deep neural networks, especially the application of attention mechanisms and graph neural networks, the image-based classification of 3D shapes has achieved rapid development. They focused on multi-view feature aggregation. Attention mechanisms assigns different weights to different views[Ma et al. (2018);Han et al. (2019;2018;)]. Multiple views can be viewed as nodes of a graph so that we can use the feature diffusion or view selection to generate shape descriptors, which are usually more effective in global recognition, and improved classification accuracy significantly[Wei et al. (2020)]. But these improvements did not overcome the problems mentioned above. The internal structure of the shape is not represented in the image, and the virtual camera still needs to surround or enclose the shape, which limits the application scenarios of the method.

We analyzed the shortcomings of multiple views and proposed the slice-sampling-based 3D object classification method. This method is inspired by the contours of topographic maps and CT slices. Our approach stacks the intersection lines of the 3D model with slicing planes in different directions and different heights onto the image to generate the input of the deep neural network. In the process of stacking intersecting lines, we give different colors to the lines at different heights. Thus different from the multi-view rendering method, the images we get are 2.5D depth images. In the design of the recognition network, we use the attention mechanism to break the locality of convolution operations in CNN. In more detail, we use the attention mechanism from DANet[Fu et al. (2020)] in the feature extraction network, which generates attention value for position and filter, and use the attention mechanism from SENet[Jie et al. (2017)] in the recognition network to generate attention value for views. Experiments on the ModelNet10 and ModelNet40[Wu et al. (2015a)] verify that the slice-sampling-based approach can also achieve good classification accuracy.

1. Using slice-sampling to generate images for multi-view CNN to retain more internal information and depth information of the model on the images.

2. Providing the corresponding network that shows competitive performance on current models and outstanding performance models with meaningful inner structure.

2. Related Works

More and more research focuses on applying deep neural networks to 3D shape classification and segmentation as the great success of deep learning in computer vision and natural language processing. To tackle this challenge mentioned above, the researchers constructed inputs for deep neural networks from methods based on voxel, mesh, and image.

Voxel-based methods. The voxel-based method discretizes the 3D bounding box into a 3D occupancy grid. Each element inside is called a voxel. The number and size of voxels are fixed, so that three-dimensional convolution can be used to construct CNNs on the occupancy grid, just like 2D convolutions on images. Wu et al. (2015b) proposed 3D ShapeNets for object recognition and shape.This work proposed a convolutional deep belief network to represent a geometric 3D shape as a probability distribution of binary variables on a 3D voxel grid. Moreover, ModelNet-a large-scale dataset of a 3D CAD model - was constructed in their work, which provides a data benchmark for subsequent research. Maturana and Scherer (2015) applied a 3D CNN based on a voxel grid for point cloud and CAD data recognition, which is called VoxNet. Qi et al. (2016) explored both view-based and volumetric approaches and observed the superiority of the first compared to the methods available at that time. Wu et al. (2016) generates 3D objects by volumetric convolutional networks and generative adversarial nets from a probabilistic space. But the resolution of occupancy grids only can reach $30 \times 30 \times 30$, both 3D ShapeNets and VoxNet. As a classical method of graphical encoding [Meagher (1982)], octree has entered the field of vision of researchers. Octrees are used in voxel sampling to improve the voxel resolution. Both Wang et al. (2017a) and Riegler et al. (2017) focus on optimizing the structure of voxel grids with an octree. Le and Ye (2018) proposed the PointGrid, which samples a constant number of points within each grid and then classified and segmented point cloud with CNNs.

Mesh-based methods. Mesh is most widely used in the construction and rendering of CAD models and other computer graphics scenes. However, because the mesh data structure contains the vertices, edges, and faces of the model, it is irregular and non-uniform for a fixed number of neural input units of a deep neural network. Despite many challenges, researchers are still obsessed with using CNN on mesh. MeshCNN[Hanocka et al. (2019)] simplified the number of edges to the same number and extracted feature for each edge. MeshCNN defined convolution operator and pooling operation for edges and then constructed CNN for 3D shapes. Similarly, Feng et al. (2019) proposed MeshNet, which regarded face as the basic unit for convolution and pooling layers of convolutional neural network. Recently, researchers have tried to apply RNN to mesh to avoid the shortcomings of CNN that require fixed input in MeshNet or MeshCNN. Lahav and Tal (2020) applied random walks on surfaces to represent the mesh, and it fed the random walks into a Recurrent Neural Network (RNN) to utilizing the representation. The advantage of using RNN is that simplifying the model to a fixed number of faces or edges is no longer necessary. The mesh-based method can complete the model's classification and segment the model based on the deep neural network. However, relatively few researches are based on mesh and cannot reach a high classification accuracy.

Image-based methods. Image-based methods are the most studied method in the field of 3D shape classification because it can leverage the research results of deep learning in computer vision and has achieved high classification accuracy. In 2015, Su et al. (2015) proposed multi-view CNN (MVCNN), which is a CNN architecture with multi-view images rendered surround the model from different viewpoints. MVCNN uses pre-trained CNNs on ImageNet[Jia et al. (2009)], such as VGG[Simonyan and Zisserman (2014)], to extract features for view. Features were aggregated at a view pooling layer and then fed into another CNN to classify models. Song et al. (2016) realized a 3D shape search engine based on images. Since then, the main development of image-based methods has two focal points. The first is how to optimize the design of the recognition network, including the aggregation of multi-view features and applying the latest developments in deep neural networks. Another point is to obtain more effective input images. Qi et al. (2016) used an elongated anisotropic kernel to capture the global structure of the 3D shape. Kanezaki et al. (2018) jointly estimates object category and viewpoint from each single-view image and aggregates classified results get from a subset of multi-view images, significantly outperformed the classification accuracy of 3D shapes. Yavartanoo et al. (2018) generated images using stereographic projection and achieve high performance. In 2018, Han et al. (2019;2018;) and Ma et al. (2018) applied Recurrent Neural Network(RNN) and long short-term memory(LSTM) to obtain aggregated information from multi-view images. Furthermore, an attention mechanism is used to perform weighted aggregation of view features in both these two papers and Gao et al. (2020). Wei et al. (2020) viewed the relationship of virtual cameras as a graph and applied a Graph Convolutional Network to learn descriptors for the shape. On the other hand, using different methods to generate images is also a dimension of this method. Huang et al. (2015) employed depth image of the shape, and Chu et al. (2017) rendered surface normals with an RGB colormap additionally. Johns et al. (2016) decomposed a view sequence into a set of image pairs and classified them independently before learning the contribution of each pair. Depth images also play a role in other image-based tasks, such as 3D shape synthesize[Soltani et al. (2017)] and point cloud recognition[Gao et al. (2020)].

Attention mechanism in vision. Since its remarkable success in the field of natural language processing[Vaswani et al. (2017)], the attention mechanism has attracted great interest in recent years. Since it is biasing the allocation of available computational resources towards the most informative components of a signal, employing the attention block can improve feature extraction and understanding of CNNs. Xu et al. (2015) applied attention in vision task, although it is still closely related to natural language processing at that time. Jie et al. (2017) proposed Squeeze-and-Excitation block(SENet), which tries to enhance the channel-wise relationships in CNNs. Fei et al. (2017) proposed a Residual Attention Network which stacks multiple Attention Modules inspired by the ResNet [He et al. (2016)]. The module calculated spatial, channel, and mixed attention corresponding to three types of activation functions. In the same year, Wang et al. (2017b) applied a non-local block to capture long-range dependencies both in videos and static images. Fu et al. (2020) developed a network that contains a dual attention module named as DANet for semantic segmentation of images. The attention module in DANet consists of position attention and channel attention module. We applied attention from DANet in the feature extraction network and attention module derived from SENet in the classification network in our method. We modified SE Block to make it suitable for calculating the correlation between 1D feature vectors.

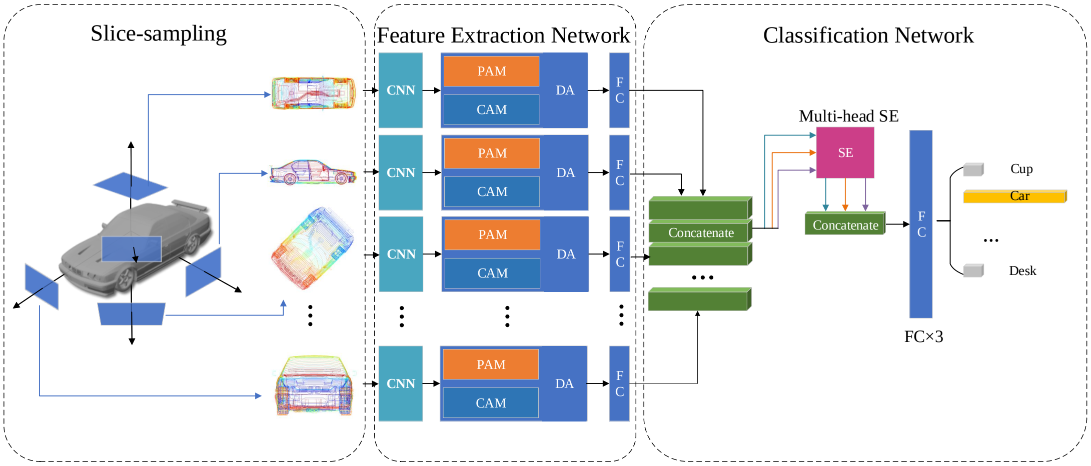


Figure 1: The pipeline of our method. We slice-sampling the shape from multiple views, and each view generates an input image for the feature extraction network. The feature extraction network applies a CNN and a Dual Attention block to get the features of each image. At the third part of the pipeline, we concatenate 1D feature vectors from multiple views and pass the tensor to a multi-head SE module. The classification result can be obtained after 3 FC layers (see Sec.3.2).

Based on the above work, because of the insufficiency of using the rendered image or the depth buffer image to carry out shape classification, we stacked the intersection lines of the model and a set of parallel planes to generate a 2.5D image. These images can show more details of the internal structure and retain the depth information of the model in different directions. Thus, the approach is more suitable for situations where enclosing rendering is not possible. We also adopted the idea of independent design of feature extraction network and classification network. Furthermore, we used the attention mechanism to improve the network's performance in feature extraction and aggregation.

3. Approach

Our method will be described in detail in the following two sections. In the first section, we will show the process of slice sampling. In the second section, we introduce the architecture of our network, including the feature extraction network and the classification network.

3.1. Slice-sampling With 3D Shape

The main idea of slice sampling is to calculate the intersection lines between a set of planes and a 3D shape. Then we project these intersection lines onto the image plane as the first part shown in Fig. 1. For one sampling, the normal vectors of these planes are parallel to each other, and the distances between these planes along the normal vectors are equal. Color codes are generated for the line according to the plane's height in projecting the intersection line to the image plane, so lines at the same height have the same color on the

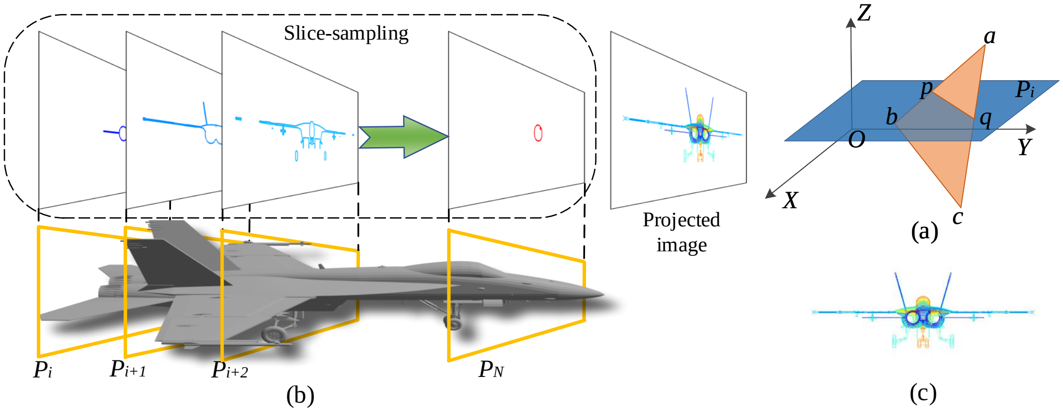


Figure 2: The Pipeline of slice-sampling. (a) shows that we get the line segment l_{pq} for a triangle by intersecting a non-parallel plane P_i with it in the Cartesian coordinate system $O-XYZ$. We traverse triangles of the shape and apply a group of parallel planes $\{P_1, ..P_N\}$ to slice-sample the shape from different heights as (b) shows. We get the output image (c) by projecting the lines to an image plane (see Sec.3.1).

image. We do not regard the 3D shape as a complex topological structure but only as a set of triangles to maintain the internal structure information of the shape and simplify the calculation process.

Before beginning slice-sampling, we normalized all the shapes' vertices to ensure that different shapes have scale consistency. If there is a necessity to consider the possible interference caused by the shape's rotation, we can also use principal component analysis(PCA) to transform shapes.

We denote a shape as $\mathbb{S}$, and $N = \{\mathbf{n}_0, \mathbf{n}_1, ..., \mathbf{n}_m\}$ is the set of sampling directions for S, where m is the count of directions. For every $\mathbf{n}_j$ in N, there are a group of planes taken $\mathbf{n}_j$ as their unit normal vector. As Fig. 2(a) shows, suppose the three vertices of the triangle T is a, b and c. The line segment obtained by intersecting the non-parallel plane P_i as Fig. 2 (b), which belong to the planes group, is l_{pq}, and l_{pq} is located inside the triangle. In Cartesian coordinate system $O-XYZ$, vertice a,b,c are point (x_a, y_a, z_a), (x_b, y_b, z_b) and(x_c, y_c, z_c) respectively. Unit normal vector of P_i is $\mathbf{n}_j = (n_1^j, n_2^j, n_3^j)$, that is $|\mathbf{n}_j| = 1$. Then we can construct the equation of P_i as

$$n_1^j x + n_2^j y + n_3^j z + d_i^j = 0, i = 0, 1, 2...k_n \tag{1}$$

where k_n is the count of planes parallel to each other. Since the shape has been normalized, to simplify the problem, we assume the change interval of the shape along the normal vector direction of the plane is $[0, 1]$. Thus, distance between plane P_i and P_{i+1} is

$$dist = \frac{\left|d_{i+1}^j - d_i^j\right|}{|\mathbf{n}_j|} = \frac{1}{k_n}, k_n = 1, 2, 3, ... \tag{2}$$

and

$$d_i^j = i \times \frac{1}{k_n}, i = 0, 1, 2, ..., k_n \tag{3}$$

We construct the equation of the line segment l_{ab} as

$$\frac{x - x_a}{x_b - x_a} = \frac{y - y_a}{y_b - y_a} = \frac{z - z_a}{z_b - z_a} = t_{ab} \in [0, 1) \tag{4}$$

where t_{ab} is a parameter, and $t_{ab} \in [0, 1)$ ensures that (x, y, z) is on the line segment. Eaquations of line l_{bc} and l_{ca} could be obtained similarly.

We could get point p with coordinate (x_p, y_p, z_p) by solving Eq. (1) and (4). Respectively, t_{ab} for p is

$$t_{ab} = -\frac{n_1^j x + n_2^j y + n_3^j z + d_i^j}{n_1^j(x_a - x_b) + n_2^j(y_a - y_b) + n_3^j(z_b - z_a)} \in [0, 1) \tag{5}$$

We could get coordinate of point p after fed t_{ab} back to Eq. (4). Coordinate of point q can be calculated similarly. l_{pq} is the line segment obtained by the intersection of the plane and the triangle. As projecting l_{pq} to the image plane I, which also has a unit normal vector $\mathbf{n}_j$, we generate color from a color bar with D_i

$$Color_i = ColorBar(D_j^i) \tag{6}$$

We traverse D_j^i for all triangles in the S to complete one slice-sampling. All the directions in N need to be traversed.

3.2. Network Architecture

The network structure used in this article is the classic structure of multi-view, that is, the use of feature extraction network and feature aggregation network to classify shapes in series. The task of the feature extraction network (see the second part of Fig. 1) is to extract a feature vector from a single image. The task of the feature aggregation network is to aggregate the feature vectors of multiple images projected from different sampling directions and give the final shape classification (see the third part of Fig. 1).

The following description will first show the Squeeze-and-Excitation attention block and dual attention block used in network design. Then we will describe the design ideas of the network and the application of the attention mechanism in it.

Squeeze-and-Excitation Attention Block. This attention block was constructed by Jie et al. (2017), which is very simple but effective. As its name implies, SE attention calculation mainly includes two parts: squeeze and excitation. For feature maps $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_C]$, where $\mathbf{u}_C \in \mathbb{R}^{H \times W}$ and H is height and W is the width of feature maps, C is the count of feature maps, the squeeze of $\mathbf{u}_C$ is that

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \tag{7}$$

Eq.(7) denoted squeeze operation is calculating the average value z_c of a feature map. We get the attention value from a neural network with two fully connected layers for different channels, an excitation operation called. The first layer used Relu as an activation function and sigmoid for the last layer. Output feature map $\mathbf{u}_c^{se}$ is obtained by multiplying $\mathbf{u}_c$ with the excitation value z_c

$$\mathbf{u}_c^{se} = z_c \times \mathbf{u}_c$$

SE block was used in the aggregation network in our approach, as mentioned above. We modified the squeeze process to 1-dimensional operate as

$$z_c = \mathbf{F}_{sq}\left(\mathbf{u}_c\right) = \frac{1}{L}\sum_{i=1}^{L} u_c(i)$$

where $\mathbf{u}_c$ is the feature vector extracted from a depth image of the shape, and L is the length of vector $\mathbf{u}_c$. We use multiple SE attention blocks to improve the network's performance, which is the so-called multi-head attention mechanism. The structure of each SE attention module is the same, and their output will be concatenated and feed into the following fully connected layer.

Dual Attention Block. The output of a 2D convolution layer in CNN is 3-dimensional, a channel dimension and two-position dimension. Attention mechanism derived from DANet[Fu et al. (2020)] provided two attention modules for feature maps of CNNs, position attention module(PAM) and channel attention module(CAM). For $\mathbf{U}$ noted as above, PAM generated two new feature maps $\mathbf{G}$ and $\mathbf{M}$ by feeding $\mathbf{U}$ into a convolution layers respectively, where $\{\mathbf{G}, \mathbf{M}\} \in \mathbb{R}^{C\times H\times W}$. Then reshaped $\mathbf{G}$ and $\mathbf{M}$ to $\mathbb{R}^{C\times T\}}$, $T = H \times W$. PAM employed a softmax layer after performed a matrix multiplication between the transpose of $\mathbf{G}$ and $\mathbf{M}$. That is

$$s_{ji} = \frac{\exp\left(G_i \cdot M_j\right)}{\sum_{i=1}^{T} \exp\left(G_i \cdot M_j\right)}. \tag{8}$$

Here, Eq. (8) measures the i^{th} position's impact on j^{th} position. We could get the final output $\mathbf{E} \in \mathbb{R}^{C\times H\times W}$ of PAM as follows:

$$E_j^{PAM} = \alpha \sum_{i=1}^{T} \left(s_{ji} Q_i\right) + U_j \tag{9}$$

where $\mathbf{Q} \in \mathbb{R}^{C\times H\times W}$ is a new feature map generated by a convolution layer fed with $\mathbf{U}$, and α is initialized as 0 and gradually learns to assign more weight.

CAM generate channel attention for $\mathbf{U}$. Similar to PAM, CAM applied a softmax layer to generate attention after reshape and transpose of $\mathbf{U}$.

$$r_{ji} = \frac{\exp\left(U_i \cdot U_j\right)}{\sum_{i=1}^{C} \exp\left(U_i \cdot U_j\right)} \tag{10}$$

Here, Eq. (10) measures the i^{th} channel's impact on the j^{th} channel. The final output E_j^{CAM} of CAM could get by

$$E_j^{CAM} = \beta \sum_{i=1}^{C} \left(r_{ji} U_i\right) + U_j \tag{11}$$

where β gradually learns a weight from 0.

The outputs of PAM(see Eq. (9)) and CAM (see Eq. (11)) are used as the input of two 2D convolution layers to extract features. Then, we use their aggregation as the feature map. Convolution layers of these two branches have the same count of filters, respectively. In our network, we employed a dual attention module behind the last convolution layer of ResNet50 in the feature extraction network, which attempts to pay different attention to different positions and channels.

Network Architecture. Our network is similar to the classical architecture of multi-view CNN derived from Qi et al. (2016) with two main parts. As Fig. 1 shows, there are two main parts in our network, feature extraction network, and classification network. The feature extraction network was designed to extract a 1D feature vector from each image of the shape. The classification network aggregates feature extracted from images generated in different directions of a shape in sampling.

We designed the feature extraction network based on ResNet50[He et al. (2016)] without its top layer (the CNN in Fig.1) and add one dual attention module and two fully connected layers behind the last convolution layer behind. This network shares the same parameters for images from different directions and shapes. As a set of intersection lines projects our image, the feature may be sparse, and ResNet allows us to construct a deeper network to learning features in the image but without worry about the sparsity. We trained the feature extraction network by fed all images into it and classify them without direction aggregate.

We constructed it as a four fully connected layer with SE attention in front of them for the classification network. SE attention module generates weights for different feature vectors of a shape generated by the feature extraction network. After the flattened output of SE block to 1D, these feature vectors were fed into fully connected layers and output the classification result at the network's end. The architecture of this part is quite simple, so it is very fast in both training and prediction.

4. Experiments

This section will introduce the experiments used to verify and analyze the method in this paper, including the parameter setting of these experiments. Our method was implemented with TensorFlow[Abadi et al. (2015)] framework in Python. All experiments were conducted on a computer equipped with 4 GeForce RTX3090 GPUs, 2 Intel Xeon 6266C CPUs, and 1TB RAM.

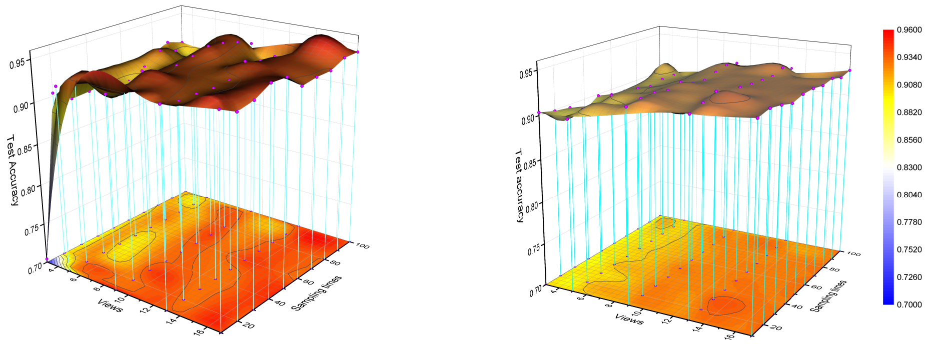
4.1. Classification on ModelNet

ModelNet[Wu et al. (2015b)] is a large scale 3D shape dateset which contains $15,128$ 3D shapes categorized into 660 classes. ModelNet40 consists of $12,311$ 3D shapes from 40 classes, while ModeNet10 is formed by $4,899$ 3D shapes from 10 categories. There are $9,843$ shapes consisted in the train set of ModelNet40 and $2,468$ shapes in test set. For ModelNet10, there are $3,991$ shapes in train set and 908 shapes in test set.

We slice-sample the shape to generate images as described above. Since the intersection line on the image is generated by a set of planes and triangles in shape, it can penetrate the shape's surface without worrying about the occlusion of the surface. Hence, hemispherical enclosing or fewer directions can be used. We choose five groups directions above the

$O-XY$ plane of the Cartesian coordinate system as normal plane vectors in experiments. The angle between these normal vectors is 45°. Size of images is $224 \times 224 \times 3$, because we encode D_j^i with RGB in Eq. (6). In 3D shape classification experiments, we employed *average instance accuracy* and *average class accuracy* as metrics of performance.

We train the network in two terms with Adam as parameter optimizer. The first step is to train the feature extraction network, which uses ResNet50[Fei et al. (2017)]as the backbone. All images obtained by slice sampling are classified according to the shape label during the process. The training enabled the feature extraction network to extract effective features of images from different directions. The training at this stage adopts a learning rate reduce strategy with an initial learning rate of 0.0001 and a minimum learning rate of 0.000001. If the result does not improve for 5 consecutive epochs, the learning rate will be reduced by a factor of 0.5. The second step is to train the classification network. In this term, the parameters of the feature extraction network are frozen, and the classification network is trained. The training of the classification network aggregates the features of different views generated by the same shape and inputs them into the classification network. The output of the classification network is the final result. At this stage, we also use the learning rate reduction strategy. Initial learning rate is 0.000004, the minimum learning rate is 0.000000001, the reduction factor is 0.2, and the patience is 3. Since our images are generated by slice-sampling, they are different from rendered images or natural images. Therefore, the feature extraction network should be retrained based on the pre-trained parameters from the ImageNet[Jia et al. (2009)] dataset. As Tab. 2 shows, on ModelNet40, the performance of our method is competitive.



(a) Test accuracy on ModelNet10

(b) Test accuracy on ModelNet40

Figure 3: A reference for hyperparameter setting of slice-sampling. (a) is the test accuracy on ModelNet10 and (b) is the performance on ModelNet40. The x-axis in the figure represents the number of views, from 3 to 17, the y-axis represents the number of samples under each view, from 10 to 100, and the z-axis is the accuracy of classification, from 0.7 to 0.96. The purple dots are actual results obtained from our experiments, and the surface is obtained by linear interpolation of these results.

Methods	Input	Views	Class Acc.(%)	Ins. Acc.(%)
MeshNet Feng et al. (2019)	Mesh	-	-	91.9
MeshWalker Lahav and Tal (2020)	Mesh	-	-	92.3
3DShapeNets Wu et al. (2015b)	Voxel	12	77.32	-
VoxNet Maturana and Scherer (2015)	Voxel	-	83.0	-
VRN Brock et al. (2016)	Voxel	24	-	91.33
3DGAN Wu et al. (2016)	Voxel	-	83.3	-
MVCNN-Sphere Qi et al. (2016)	Voxel	20	86.6	89.5
Octree Wang et al. (2017a)	Voxel	12	90.6	-
PointNet Qi et al. (2017b)	Point	1	86.2	89.2
PointNet++ Qi et al. (2017a)	Point	1	-	91.9
SONet Li et al. (2018)	Point	1	87.3	90.0
FoldingNet Yang et al. (2017)	Point	1	-	88.4
PANORAMA Sfikas et al. (2018)	Image	6	90.7	-
MVCNN Qi et al. (2016)	Image	20	89.7	92.0
Su-MVCNN Su et al. (2015)	Image	80	90.1	-
Spherical projection Ca O et al. (2017)	Image	36	-	93.1
RotationNet Kanezaki et al. (2018)	Image	12	-	90.65
Pairwise Johns et al. (2016)	Image	12	90.7	-
GIFT Song et al. (2016)	Image	64	89.5	-
3D2SeqViews Han et al. (2019)	Image	12	91.4	93.4
Views2Labels Han et al. (2019;2018;)	Image	12	91.12	93.31
CNN+LSTM Ma et al. (2018)	Image	12	-	91.05
Dominant Set Chu et al. (2017)	Image	12	-	92.2
LFD	Image	10	75.47	-
Ours	Image	13	**90.35**	**93.68**

Table 1: Classification comparison under ModelNet40

Moreover, as a reference to the parameters set in other applications, we test the performance of our method with different sampling rates and views as Fig. 3. The x axis is the views, and the y axis is the sampling times. These two parameters can be set based on the complexity of the task. Generally, we need more dense sampling and more views for a more complex task to provide enough information. The best accuracy we achieved on the ModelNet40 test set is 93.68%, using 13 views and 30 sampling times, which are evenly distributed above the $O-XY$ plane. As showed in Fig.3.(a), the classification accuracy of the test set can reach 92.6% under 3 views(from x, y, z axis) on ModelNet10, while on ModelNet40, this accuracy can reach 90.8%. As the number of views and sampling times increase, the accuracy gradually increases. At the time of 9 views, the accuracy on these two data sets was increased to 94.24% and 92.75% respectively, and they generally reached the best accuracy (94.74% on ModelNet10) at the time of 13 views, although the average accuracy is still slowly increasing with the views increasing. The best accuracy of classification can usually achieved when the slicing-sampling times is less than 60, which proves that the network has extracted the sparsely distributed structural features for classification tasks. As this number is greater than 60, the lines on the input image begin to obscure

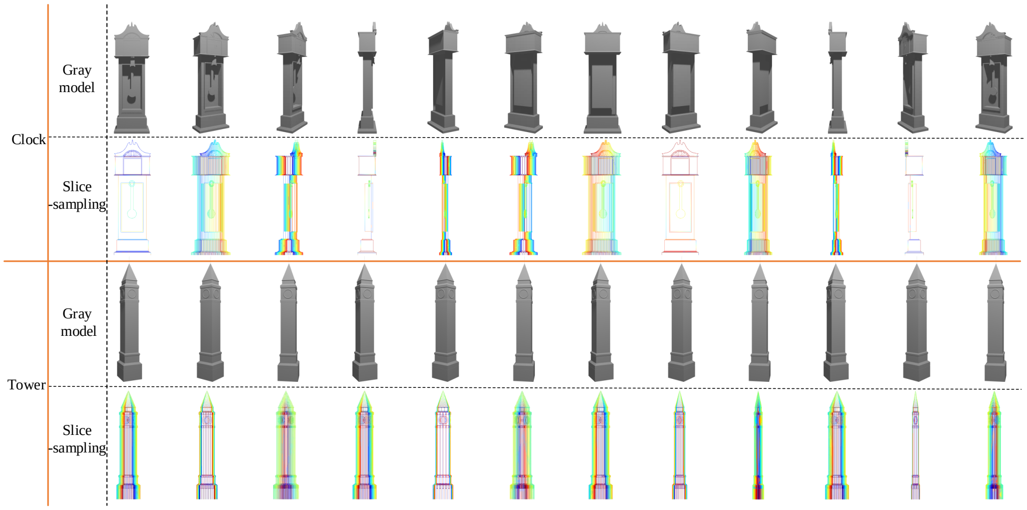


Figure 4: These images are generated by 12 views evenly distributed around the z-axis. Because of lack of sense of size, classifying the model shaped like a clock tower is onerous if the inner structure cannot be represent appropriately. Compared with the gray model, distinguishable features are more obvious on the slicing-sampling images.

each other, making some important feature lost and the accuracy dose not increase. On the other hand, it can be seen that as the number of samples increases by comparing the performance of classification accuracy on ModelNet10 and ModelNet40, the fluctuation of classification accuracy becomes smaller and the result is more stable.

4.2. Classification for models with complex inner structure

The advantage of this slice-sampling method is in analyzing the meshes with complex inner structure ambiguous appearance. It can reflect their feature effectively, which is fundamental to make the correct classification. For example, because of lack of sense of size, the difference between the bracket clock and the clock tower is ambiguous if we just render their surfaces as Fig. 4 shows. However, the sampling result based on our method can reflect the difference explicitly. Specifically, the bracket clock usually contains a special mechanical structure, like the pendulum, but the clock tower does not have a similar feature. Similar cases widely exist in practice. For example, the mechanical clock contains big gears inside, but the quartz watch does not. Most of these cases can be classified easily by their inner feature. And these inside iconic features can be presented by our method appropriately. To show the advantages of our method, we collect meshes of bracket clock and clock tower and use the same DNN model to do binary classification with different sampling strategies. As the Tab. 2 shows, the precise and recall of samples with label ”tower” are both 0, which means the model based on gray model cannot distinguish these two kinds of samples. In contrast, the model trained by our data shows higher accuracy and stronger discrimination.

	gray model		slice-sampling	
	precise	recall	precise	recall
clock	0.67	1.00	0.89	1.00
tower	0.00	0.00	1.00	0.75
test accuracy	0.666		0.916	

Table 2: Binary classification results.

5. Conclusion

In this paper, we present a novel solution for 3D model recognition and classification. Firstly, we provide a slice-sampling method to extract the feature of the model from different depths and directions. Then, a deep neural network designed based on the attention mechanism is used to classify the input data. The experiments on ModelNet show that the performance of our method is competitive on common models. Moreover, for some special models with simple surfaces and complex inner structures, the performance of our method is outstanding.

Acknowledgments

Wei Zeng is the corresponding author. This work was supported by the National Key R&D Program of China (2020YFB1708900), the China National Natural Science Foundation (61872224,12090021,12090020), Shandong Provincial Natural Science Foundation of China (ZR2017MF003).

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Generative and discriminative voxel modeling with convolutional neural networks. *Computer Science*, 2016.

Z. Ca O, Q. Huang, and K. Ramani. 3d object classification via spherical projections. In *2017 International Conference on 3D Vision (3DV)*, 2017.

W. Chu, M. Pelillo, and K. Siddiqi. Dominant set clustering and pooling for multi-view 3d object recognition. In *British Machine Vision Conference*, 2017.

W. Fei, M. Jiang, Q. Chen, S. Yang, and X. Tang. Residual attention network for image classification. *IEEE*, 2017.

Yutong Feng, Yifan Feng, Haoxuan You, Xibin Zhao, and Yue Gao. Meshnet: Mesh neural network for 3d shape representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8279–8286, 2019.

J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Zhongpai Gao, Guangtao Zhai, Juyong Zhang, Junchi Yan, Yiyan Yang, and Xiaokang Yang. Learning local neighboring structure for robust 3d shape representation. *arXiv e-prints*, pages arXiv–2004, 2020.

Z. Han, H. Lu, Z. Liu, Chi Man Vong, Yu Shen Liua, Matthias Zwicker, Junwei Han, and C. L. Philip Chen. 3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, pages 1–1, 2019.

Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and C. L. P. Chen. Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention. *IEEE transactions on image processing*, 28(2):658–672, 2019;2018;.

R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. Meshcnn: A network with an edge. *ACM Transactions on Graphics*, 38(4CD):90.1–90.12, 2019.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE*, 2016.

Huang, Hui, Xiong, Yueshan, Shan, Wen, Xu, Kai, Liu, and Ligang. Projective feature learning for 3d shapes with multi-view depth images. *Computer Graphics Forum Journal of the European Association for Computer Graphics*, 2015.

D. Jia, D. Wei, R. Socher, L. J. Li, L. Kai, and F. F. Li. Imagenet: A large-scale hierarchical image database. *Proc of IEEE Computer Vision & Pattern Recognition*, pages 248–255, 2009.

H. Jie, S. Li, S. Gang, and S. Albanie. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), 2017.

E. Johns, S. Leutenegger, and A. J. Davison. Pairwise decomposition of image sequences for active multi-view recognition. *IEEE*, 2016.

A. Kanezaki, Y. Matsushita, and Y. Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Alon Lahav and Ayellet Tal. Meshwalker: deep mesh understanding by random walks. *ACM Transactions on Graphics (TOG)*, 39(6):1–13, 2020.

T. Le and D Ye. Pointgrid: A deep network for 3d shape understanding. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

J. Li, B. M. Chen, and G. H. Lee. So-net: Self-organizing network for point cloud analysis. *IEEE*, 2018.

C. Ma, Y. Guo, J. Yang, and W. An. Learning multi-view representation with lstm for 3-d shape recognition and retrieval. *IEEE Transactions on Multimedia*, 2018.

D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

D. Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 1982.

C. R. Qi, H. Su, M. Niebner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

C. R. Qi, Y. Li, S. Hao, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. 2017a.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017b.

G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Konstantinos Sfikas, Ioannis Pratikakis, and Theoharis Theoharis. Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval. *Computers & Graphics*, 71:208–218, 2018.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014.

A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum. Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2017.

B. Song, B. Xiang, Z. Zhou, Z. Zhang, and L. J. Latecki. Gift: A real-time and scalable 3d shape search engine. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. *IEEE*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

P. S. Wang, Y. Liu, Y. X. Guo, C. Y. Sun, and X. Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *Acm Transactions on Graphics*, 36(4):72, 2017a.

X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. 2017b.

X. Wei, R. Yu, and J. Sun. View-gcn: View-based graph convolutional network for 3d shape analysis. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. 2016.

Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. *IEEE*, 2015a.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015b.

K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *Computer Science*, pages 2048–2057, 2015.

Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: Interpretable unsupervised learning on 3d point clouds. 2017.

M. Yavartanoo, E. Y. Kim, and K. M. Lee. Spnet: Deep 3d object classification and retrieval using stereographic projection. 2018.