Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 6467957, 11 pages
https://doi.org/10.1155/2018/6467957

Research Article

An Improved 3D Shape Recognition Method Based on Panoramic View

Qiang Zheng,[1,2] **Jian Sun**,[1,2] **Le Zhang**,[1,2] **Wei Chen**,[1,2] **and Huanhuan Fan**[1,2]

[1]*State Key Laboratory for Strength & Vibration, School of Aerospace, Xi'an Jiaotong University, Xi'an, 710049, China*
[2]*Shaanxi Engineering Laboratory for Vibration Control of Aerospace Structures, Xi'an Jiaotong University, Xi'an, 710049, China*

Correspondence should be addressed to Jian Sun; sunjian10@xjtu.edu.cn

Received 4 June 2018; Revised 29 July 2018; Accepted 1 August 2018; Published 7 August 2018

Academic Editor: Ioannis Kostavelis

Recognition of three-dimensional (3D) shape is a remarkable subject in computer vision systems, because of the lack of excellent shape representations. With the development of 2.5D depth sensors, shape recognition is becoming more important in practical applications. Many methods have been proposed to preprocess 3D shapes, in order to get available input data. A common approach employs convolutional neural networks (CNNs), which have become a powerful tool to solve many problems in the field of computer vision. DeepPano, a variant of CNN, converts each 3D shape into a panoramic view and shows excellent performance. It is worth paying attention to the fact that both serious information loss and redundancy exist in the processing of DeepPano, which limits further improvement of its performance. In this work, we propose a more effective method to preprocess 3D shapes also based on a panoramic view, similar to DeepPano. We introduce a novel method to expand the training set and optimize the architecture of the network. The experimental results show that our approach outperforms DeepPano and can deal with more complex 3D shape recognition problems with a higher diversity of target orientation.

1. Introduction

Three-dimensional (3D) shapes contain information about real-world objects, and scientists in the field of computer vision have focused on how to fully use 3D shapes to solve problems such as object classification and object recognition. Convolutional neural networks (CNNs) have been widely applied to 2D images and show brilliant performance, and hence it is natural to introduce CNNs to 3D object recognition. One of the most important challenges in generalizing CNNs from 2D to 3D images is that while 2D images have a regular data structure, 3D shapes such as point clouds or meshes are too irregular to act as direct input for CNNs. Hence, it is clear that the performance of object recognition is greatly affected by whether good representations can be obtained for 3D shapes. According to the theoretical basis, previous works on 3D object recognition can be roughly summed up in two categories: hand-designed features based methods and deep learning based methods [1]. The primary difference between the two categories is in how to choose the features of 3D shapes: the former depends on artificially designing features, while the latter can automatically learn more abstract features by employing multilayer neural networks. Hand-designed methods mainly use features such as histogram of oriented gradient (HOG), scale-invariant feature transform (SIFT), signature of histograms of orientations (SHOT), viewpoint feature histogram (VFH), and point feature histogram (PFH), which are applied to various machine learning algorithms, for instance, random forests [2], support vector machines (SVMs) [3], or perception machines [4]. The recognition process is hence summarized by the following steps: feature extraction, feature coding, feature combination, and object recognition. Although this method has been popular and achieved success in 3D recognition, the features are low-level. Additionally, designing an appropriate feature usually requires domain expertise and experience. Even then, it requires much research and time and it remains difficult to achieve a satisfactory level of recognition. The rise of deep learning greatly changed the state of image recognition. Methods based on deep learning not only map from features to output but also extract the features themselves, and often result in much better performance than can be obtained with hand-designed features. Deep learning can make the recognition algorithm quickly adapt to variable tasks, with less human intervention.

Previous work on 3D recognition based on deep learning can be broadly categorized into two styles, and different types of methods have different advantages and disadvantages. Volumetric CNNs[5–7] convert 3D shapes to voxelized shapes in a similar approach to the representation of 2D data. However, volumetric representation is subject to the limits of resolution. In order to obtain proper resolution comparable to 2D images, the number of voxels is large and the data tends to be sparser. This poses a challenge to storage capacity and operational performance of algorithms. Li et al. [8] and Wang et al. [9] proposed a special method to alleviate this problem; however, it does not fundamentally solve the problem. Multiview CNNs render a 3D shape into a group of 2D images by projecting the 3D model onto a plane from various perspectives [6, 10]. Thus, 3D recognition subjects can be converted to large 2D recognition problems, which have been solved with excellent performance. With available network architecture, this method has achieved dominating performance on shape classification and retrieval tasks. However, when this method converts 3D shapes to 2D images, part of the spatial information is inevitably lost, and this results in a reduction of spatial discrimination. Additionally, this method is difficult to extend to other 3D tasks such as scene understanding and shape completion. For feature-based CNNs, the main idea is that 3D data is converted into a vector by this method, and traditional shape features act as the input for networks [11, 12]. The use of traditional features additionally implies that this method does not take full advantage of the ability of deep neural networks to automatically learn features.

In this work, we propose a new machine learning technique for 3D shape recognition, related to the previous method DeepPano [13], which transforms 3D shapes into 2D panoramic views. Different from multiview convolutional neural networks[10], DeepPano renders a 3D shape into only one image. The left and right sides of the rendered image are divided; however, they are generally connected in the original 3D shape. Hence, there is a loss of information in the conversion process. In order to avoid any loss, DeepPano adds to one side of the map an extra padded area cloned from the other side. This means that some of the pixels are repeated, and redundant information is included. The convolutional feature map extracted from the panoramic views shifts as the 3D shape rotates, so in order to obtain rotational invariance, another creative method of DeepPano, row-wise max-pooling layer (RWMP), is introduced. Unlike the typical pooling layer in CNN, RWMP transforms a long line of elements in a feature map into only one element, which causes an additional large information loss. We propose a new technique to avoid the loss of information and the redundancy that occur in DeepPano. In our approach, for panoramic views, when the 3D shape rotates, pixels of the corresponding 2D rendered image move in parallel. When rendered images are obtained, we do not directly treat them as input data, but instead transform the pixels in each image by certain, or even random, step sizes to generate a series of copies of each image; the expanded training set included a series of rotational copies of the original 3D shape. CNNs trained by the expanded training set can obtain excellent rotational invariance. We called this method rotation expansion.

In short, the key contribution of this paper is the rotation expansion method, a novel processing method on training sets for 3D shape recognition. In addition to achieving higher accuracy on original test sets, we also design comparative trials to study the recognition performance on more complex test sets. Our results show that rotation expansion is a more effective method and can preserve more original information to higher accuracy when compared to DeepPano.

This paper is structured as follows. Section 2 reviews the related work on the application of deep learning to 3D object recognition. Section 3 describes the details of panoramic view rotation expansion. Section 4 shows the experimental results. Finally, Section 5 draws some conclusions about our work and sets future research lines to improve the proposed method.

2. Related Work

Deep learning has achieved outstanding performance in 2D object class recognition. There are many representative network architectures, and convolutional neural networks are some of the most significant approaches. Krizhevsky et al. apply a deep learning model, which was developed based on CNN in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [14] 2012 and obtained excellent results compared with previous work.

Due to the distinguished performance of CNNs on problems related to 2D images, it is natural and reasonable to introduce them for similar applications on 2.5D and 3D data. RGB-D (R: red, G: green, B: blue, D: depth) data is typically a kind of 2.5D data. Methods based on CNNs simply add the depth as a forth channel of input to the previous RGB image, which has three channels of input. This approach is essentially equivalent to the old method based on the RGB data and, hence, does not fully use the geometric information provided by the depth.

It is a notable fact that 3D data has a much more complex structure than 2D data, which is in a more regular format. This makes it quiet difficult to directly apply CNN methods to 3D shape recognition. The input data of CNN is generally in tensor form. In order to make use of CNN in 3D shape recognition, the first issue to be solved is how to transform 3D shapes into an available format that can act as the input for a CNN. Different methods to represent 3D shapes have led to a series of solutions.

Voxelization based methods: voxelization is based on volumetric representation, which has played an important role in the computer vision community since the 1980s. It provides a simple and robust description and can be regarded as an extension of the pixels in 2D images; the 3D models have regular data structure, which facilitates digital processing. With the successful application of CNNs in the field of 2D images, many researchers have begun to apply 3D CNNs to volumetric shapes. Maturana et al. [5] trained a real-time 3D supervised learning architecture with volumetric 3D shapes.

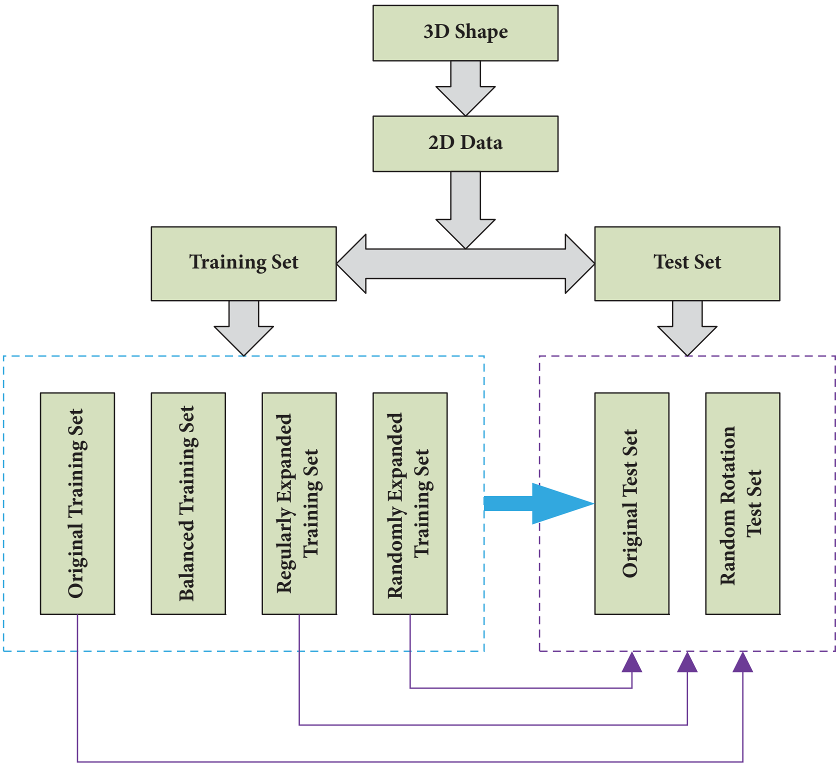


Figure 1: Flowchart used to illustrate the proposed method. The blue arrow represents the pipeline corresponding to Section 4.3, used to study classification; the purple arrows represent the pipeline related to Section 4.4, used to study robustness to rotation.

Wu et al. [7] transformed each 3D shape into a 3D grid, and volumetric models were applied for 3D shape recognition with the use of a convolutional deep belief network. In [15], features from a probabilistic space were learned by leveraging advances in volumetric convolutional networks and generative adversarial nets, which can be widely applied to 3D recognition. In [8], a 3D convolutional auto-encoder was applied to recognize 3D shapes. While volumetric shapes were not directly considered as an input for 3D CNNs in [16], multilayer dense representations of the volumetric shapes were extracted using a feature generator consisting of a convolutional neural network and recurrent neural network (RNN) and were fed into a classifier to recognize 3D shapes.

Projection based methods, which are different from direct 3D representations, project 3D shapes into various 2D spaces. In [10], a multiview CNN was applied to extract visual features from projected images from different views, and max-pooling technology was applied to combine information from multiple views of a 3D shape into a single shape descriptor and achieved higher recognition performance than single image recognition architectures. Similarly, coding technology was proposed in [17] which constructs a compact descriptor based on a series of 2D views, in the format of depth rendering from 3D objects. A light field descriptor (LFD) [18] was used to extract features from light fields rendered from cameras located on a sphere in order to improve robustness to rotation. Papadakis et al. [19] used a set of panoramic views of a 3D shape to generate a 3D shape descriptor named PANORAMA, which describe the position and orientation of the shape in 3D space. The 2D discrete Fourier Transform and the 2D Discrete Wavelet Transform are computed for each view. Shi et al. [13] proposed DeepPano. The main idea of DeepPano is to render 3D shapes into 2D panoramic views by projecting the 3D shape onto a cylindrical surface whose axis is parallel to the principal axis of the 3D shape. This approach is based on the assumption that models in ModelNet-10 and ModelNet-40 are oriented upright. Thus, an image containing spatial information is obtained and can be transmitted to a variant of CNN for object recognition. The most creative part of this method is an additional layer, the row-wise max-polling layer (RWMP), which is inserted between the convolutional and fully connected layers, to enhance the robustness of the learned representations to rotation.

3. Methodology

In this section, we outline the proposed method for 3D shape recognition. We firstly render the 3D shapes of both the training and test sets into 2D panoramic views; in this way we convert 3D shapes into 2D datasets. Thus, the 2D training and test sets we use are obtained. Next, we process the training set through some reasonable methods, including oversampling and rotation expansion, to produce an expanded training set. To improve the performance of the network for object recognition, we feed both the original training set and the expanded training set into the CNN architecture. An overall flowchart to illustrate the proposed method is shown in Figure 1.

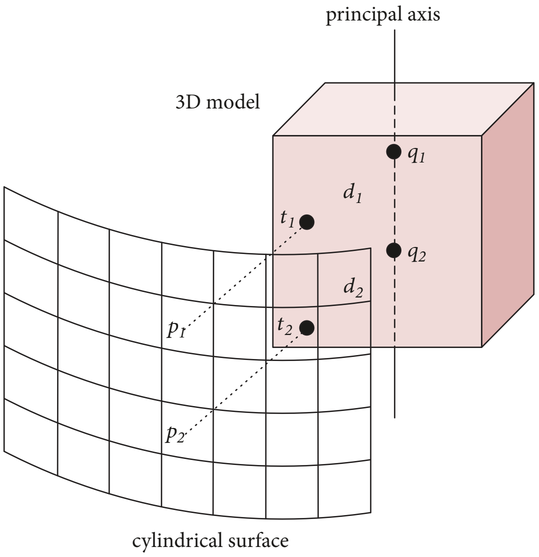


Figure 2: Illustration of the panoramic view construction process. In this figure, p_1 is the center point of the corresponding grid, q_1 is the point on the z-axis at the same height as p_1, t_1 is the intersection of the ray drawn from q_1 and the model surface, and d_1 represents the distance between q_1 and t_1. The geometric meanings of q_2, p_2, t_2, *and* d_2 are similar to q_1, p_1, t_1, *and* d_1, respectively.

3.1. Converting a 3D Shape into a 2D Panoramic View. In order to construct the panoramic view, the 3D model is surrounded by a cylindrical surface whose central axis coincides with the principal axis of the model. A 3D mesh model is composed of a series of small triangles, and the position of each triangle vertex is represented by Cartesian coordinates. In this case, it is convenient to represent the spatial location by cylindrical coordinates. We define the principal axis in the vertical upward direction as the positive z-direction, and the z value of the lowest vertex as the origin (zero). Since objects in the same category may have different sizes, which can affect the performance of object classification algorithms, we automatically match the height of the cylindrical surface to the height of the object.

The process of generating a 2D panoramic view from a certain 3D shape is shown in Figure 2. The cylindrical surface is divided into a dense grid structure, and each grid square is represented by the cylindrical coordinate (r, θ, h) of its center point p, where θ is the polar angle, h is the z value, and r is the value of radial parameter. For each grid point $p = (r_i, \theta_i, h_i)$, there is a corresponding point $q = (0, 0, h_i)$ of the same height on the z-axis. A ray drawn from an axis point q toward the corresponding grid point p may or may not intersect with the surface of the 3D shape. Considering the complexity of 3D shapes, there may even be more than one intersection. We are measuring the distances between the starting point q and the intersections t, and for each grid, we choose the maximum distance as the corresponding mapped distance. Additionally, if no intersection occurs, we set the distance to zero. Each square on the grid thus obtains a certain distance value. We discretize these distance values to 256 levels and assign a corresponding grayscale. Finally, the cylindrical surface is unfolded from a certain angle θ (here we choose θ=0) to form the final 2D image. Figure 3 shows some examples of panoramic view.

3.2. Oversampling and Rotation Expansion. In this paper, in addition to directly feeding the generated 2D images to the network, we also preprocess the 2D data in two ways: oversampling and rotation expansion.

Oversampling is proposed based on a study of the impact of imbalanced training data on CNN performance in image classification [20]. In certain datasets, some categories contain large amounts of samples, while other categories may include only a few samples. A set of this kind presents an imbalanced distribution between classes. Results show that imbalanced training data can potentially have a severely negative impact on the performance of CNN. A method named oversampling is proposed to overcome the negative effects. Here, we apply oversampling for 3D recognition by randomly selecting and duplicating samples in all, except the largest, classes until they all have the same number of samples.

During the process of converting 3D shapes to 2D panoramic views, after projecting the 3D shape to the cylindrical surface, we unfold the surface from a certain angle and thus obtain a 2D image. It is obvious that pixels on either side of the unfolding angel are initially interlinked and represent an integral part of the surface of the 3D model. The left and right boundaries of the 2D image hence lead to a loss of information which may have a negative influence on the performance of CNN. To avoid the artifacts caused by boundaries, DeepPano pads the panoramic view on one side, and the padded area is cloned from other side of the 2D image. Although the purpose is to avoid boundary artifacts, the padded area means that some of the repeated pixels are artificially added to the image, and this can be regarded as another type of artifact. The padded area is not just a matter of information redundancy, but also a great distortion of the original information. It also introduces a negative factor that influences the network performance.

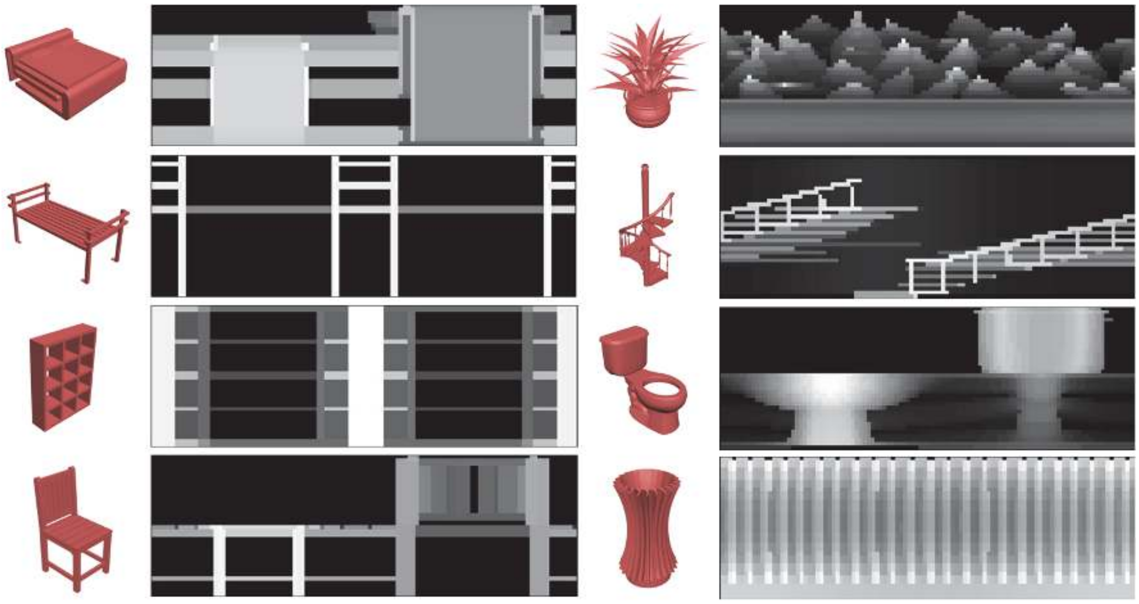


Figure 3: 3D shapes and their corresponding panoramic views.

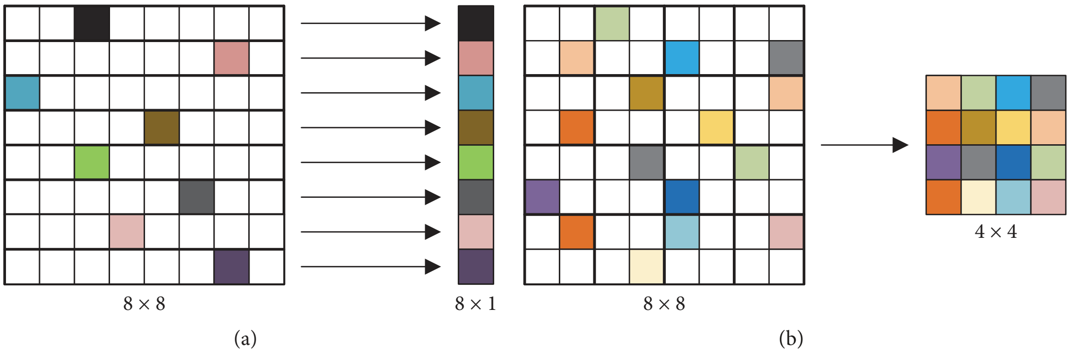


Figure 4: Comparison of two pooling methods: (a) row-wise max-pooling (RWMP), the colored grids represent the maximum elements in each row; (b) typical max-pooling, the colored grids represent the maximum elements in a local area of 2 × 2.

Although the 3D models are oriented upright, the convolutional features extracted from the panoramic views change with rotation of the 3D shape. In order to get robustness for rotation, a novel layer named row-wise max-pooling (RWMP) is introduced in DeepPano. Unlike the typical pooling layer in CNN, RWMP simply transforms row elements in feature maps to a single feature, which takes the maximum value of each row in the input map of this layer and concatenate them into the output vector (Figure 4(a)). In this way, the output of the RWMP layer is not affected by the shift of the input map; thus its output is invariant to the rotation of the 3D shape. However, this also means that a large number of elements, which are obtained by massive computing between multiple convolutional layers, in the feature maps are discarded. And, during the conversion from a matrix to a column vector, only spatial information along the vertical direction is preserved and the spatial distribution of elements inside each row is lost. Moreover, the typical pooling layer of CNN acts on a fairly local area, and after the pooling operation, the spatial distribution is largely retained (Figure 4(b)).

In order to retain the advantages of DeepPano but also try to overcome the shortcomings mentioned above, a novel method named rotation expansion (Figure 5) is proposed. Instead of a padded area cloned from one side and spliced with the other side, the circle is divided into n equal angles, and a certain 3D shape is rotated n times according to this set of angles. Thus we obtain n copies of every 3D shape. In fact, considering the correspondence between the 3D shape and the 2D panoramic view, it is more convenient to directly apply the circulation translation to the 2D image according to a series of uniformly spaced steps, and this operation can be called “rotation” for convenience. This process is equivalent to unfolding the cylinder at a series of different reference angles. Although each copy has boundary artifacts, the combination of a series of copies retains complete information without any boundary artifacts, and no repeated pixels are introduced into the images as in the padded area

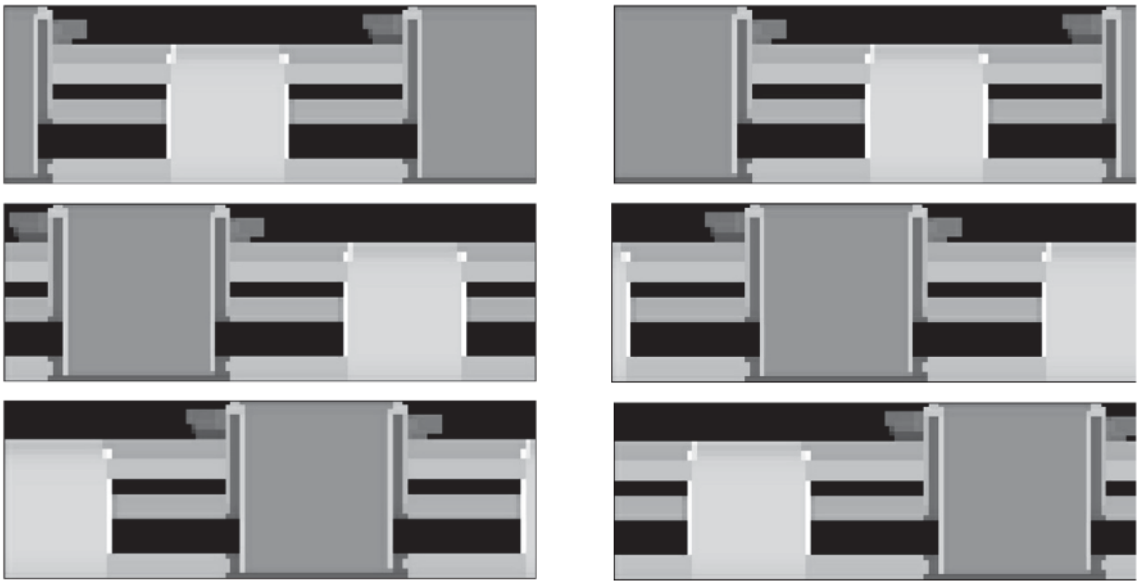


Figure 5: Illustration of rotation expansion. The top left image is the original view, and the others are variants, corresponding to rotation angles of 60°, 120°, 180°, 240°, and 300°, read from top to bottom and left to right.

approach [13]. If we apply rotation expansion to the training set and feed it into the neural network, features that are robust to rotation will be fully extracted and the network will obtain rotational invariance, similar to the effect of RWMP in DeepPano, but without the loss of spatial information inside a row.

As an extension, for the purpose to further strengthening the rotational robustness of features extracted from training data, we generate several copies for each 3D shape by rotating randomly rather than according to uniformly spaced angles. In order to test the strength of the robustness of the recognition results to rotation, we also generate a variation of the test set by randomly rotating each sample in the original test set. This is a reasonable and necessary verification, because the samples in ModelNet-10 and ModelNet-40 that we used have been artificially adjusted, more or less, based on observation and information from the Princeton ModelNet website. In fact, results of comparative experiments later in this paper prove this point. Details of ModelNet data and the experimental results are represented in Experiments.

4. Experiments

4.1. Datasets. One of the most widely used 3D shape datasets is the Princeton ModelNet, which contains 127,915 CAD (computer aided design) models in 662 object categories. Two subsets of it are often selected to train and test networks: (1) ModeleNet-10 is a subset composed of 4899 CAD models classified into 10 categories and divided into a training set and a test set. The training set includes 3991 models and the test set includes 908 models. All the CAD models in ModelNet-10 are manually cleaned and the orientation of each one is aligned. (2) Modelnet-40 is a larger set than ModeleNet-10, featuring 12,311 models classified into 40 categories; 9843 models are used for training and the 2468 are used for testing. It is claimed that the orientations of these models were not aligned. However, most models in ModelNet-40 in fact satisfy the upright assumption based on the observation of a considerable part of the models through 3D graphics software.

4.2. Implementation Details. In our implementation, each 3D model is projected into a 36 × 108 panoramic image, and the training set composed of these panoramic views is regarded as the original training set. In order to improve the performance of CNN networks, oversampling and rotation expansion are applied to the original data. We call the oversampled dataset balanced data. Rotation expansion has two different implementation options: expanding the data according to either n fixed or n random steps. We call the former regularly expanded data and the later randomly expanded data. The architecture of the network is illustrated in Figure 6. For the convolutional layers (conv1–conv4), there are 64, 80, 160, and 320 feature maps, and for each convolutional layer, the size of filters is 1, 2, 4, and 6, respectively. A 2 × 2 max-pooling layer is inserted after every convolutional layer. For the fully connected layers (fc1–fc2), there are 512 and 1,024 hidden units. The softmax layer output class probabilities, and the class with the highest probability is regarded as the prediction. Table 1 illustrates the comparison of dimensions between our architecture and other typical methods. It is obvious that the input size of our architecture is much smaller compared with other architectures, also the numbers of kernels in convolutional layers and the numbers of weights in fully connected layers. In addition, except conv1, the strides of conv2, conv3, and conv4 are all 2. All these facts result in the fact that the dimension of our architecture is much smaller than other architectures. The network is trained using the stochastic gradient descent (SGD), with rms-decay, weight-decay, and dropout techniques.

Table 1: Comparison of the parameters of architectures used in the three methods. In this table, "conv" represents convolutional layer and "fc" represents fully connected layer. For each convolutional layer, the first number in brackets is the kernel size, and the second is the number of kernels. For each fully connected layer, the number represents the number of hidden units in this layer.

Method	3D ShapeNets [7]	DeepPano [21]	Our Method
Input Size	30 × 30 × 30	160 × 64	108 × 36
Conv1	(6, 48)	(5, 96)	(1, 64)
Conv2	(5, 160)	(5, 256)	(2, 80)
Conv3	(4, 512)	(3, 384)	(4, 160)
Conv4	None	(3, 512)	(6, 320)
Fc1	1200	Unknown	512
Fc2	4000	Unknown	1024
Output Size	10 or 40	10 or 40	10 or 40

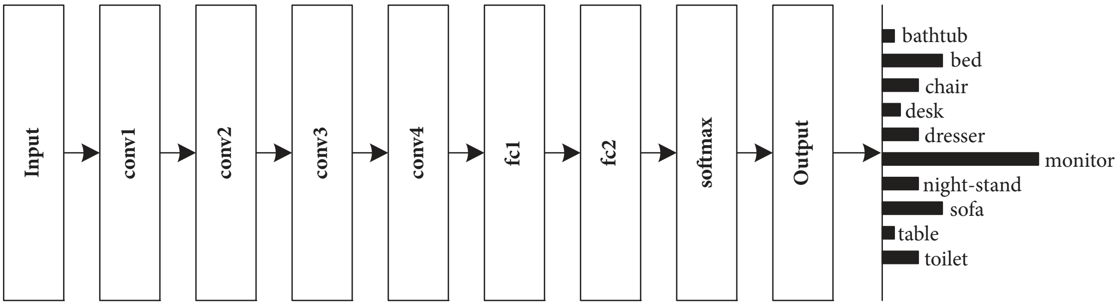


Figure 6: Illustration of network structure. In this figure, "conv" represents convolutional layer (each layer contains the corresponding max-pooling layer), "fc" represents fully connected layer, and "softmax" represents softmax layer.

The construction of panoramic views is implemented separately in C++, rendering the panoramic view for each 3D shape with an unoptimized CPU implementation. The GPU-accelerated network is implemented within the TensorFlow framework, running on a machine with Intel Core-i7 CPU, NVIDIA TITAN-XP GPU, and 16 GB RAM. The training process takes from 20 min to 4 h, depending on which type of training set is selected.

4.3. 3D Shape Classification. To evaluate the proposed method on 3D shape recognition, we trained the classification network with the various training sets (original, oversampled, regular expansion, and random expansion). The pipeline is illustrated by the blue arrow in Figure 1. The performance is evaluated by the average category accuracy. We compare our method with the light field descriptor [23] (LFD, 4700 dimensions), spherical harmonics descriptor [22] (SPH, 544 dimensions), 3D ShapeNets [7], and DeepPano [21] methods.

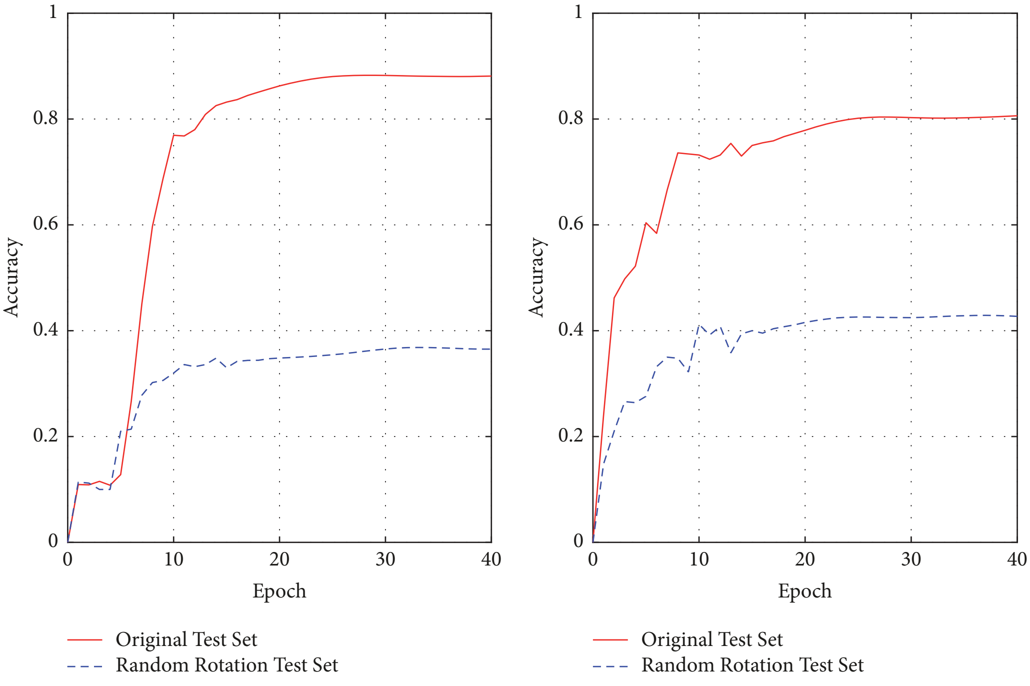
Table 2 summarizes the results. Based on the comparison of parameters of architectures and classification results, our method outperforms all the other methods to varying degrees. The hand-crafted LFD, SPH, PANORAMA and DeepPano are all designed to be rotationally invariant, and the deeply learned representation our method obtained was effective. This is mainly because we designed an available architecture and required the network learn a rotationally invariant representation by feeding it a training set processed by oversampling and rotation expansion.

For ModeNet-10, the highest accuracy is 89.80%, corresponding to the balanced training set. The balanced training set did not get too much better result than the original training set. It is well known that training a network with an unbalanced dataset tends to harm those classes with the least number of examples and benefit those with the most [24], and it is not clear how the imbalanced attribute affects the experimental results. Sometimes balancing the training set improves the accuracy of the classes with fewer examples but harms the success rate for classes with more samples [7]. For ModelNet-10, the overall effect of the balanced training set is to maintain a slightly higher accuracy than the original one. For balanced training sets processed by rotation expansion, both the regular and random expansion caused the accuracy to slightly drop. The reason is that ModelNet-10 is well aligned and the 3D orientation of the models is highly consistent. This makes the complexity of the original training set and test set exactly match; representation learned from the original or balanced training set is sufficient to recognize the models in the test set with quite good performance. Representation learned from the training set processed by rotation expansion has a high degree of rotational invariance that is not necessary for the original test set. That is, for a test set that has less complexity, rotation expansion in fact introduces noise in the representation and has a negative impact on the results.

ModelNet-40 is not as well aligned as ModelNet-10. This means that the 3D orientation of models in ModelNet-40 has much higher complexity, and the experimental results are quite different from ModelNet-10. The balanced training

Table 2: Classification accuracy of various methods on the ModelNet-10 and ModelNet-40 datasets. Best results of this method are marked in bold font. The expanded training sets are expanded based on the balanced training set.

Method	ModelNet-10	ModelNet-40
Spherical harmonics descriptor [22]	79.79%	68.23%
Lightfield descriptor [23]	79.87%	75.47%
3D ShapeNets [7]	83.54%	77.32%
DeepPano [13]	88.66%	82.54%
Original training set	89.04%	80.68%
Balanced training set	**89.80**%	81.85%
Regularly expanded training set	87.54%	**82.47**%
Randomly expanded training set	85.62%	80.55%



(a)

(b)

Figure 7: Accuracy results on different test sets corresponding to the original training set: (a) ModelNet-10; (b) ModelNet-40. During each epoch, all samples in the training set were used once for training.

set still maintains slightly higher accuracy and the highest accuracy is 82.47%, corresponding to the regularly expanded training set, not the original training set. This is primarily because the test set is more complex in 3D orientation, and so proper representation requires equivalent or a little higher complexity. While the accuracy corresponding to the original training set is lower than 82.47%, the representation learned from the regularly expanded training set obtains suitable complexity compared with the test set. Accuracy corresponding to the randomly expanded training set is almost equal to the original training set. The reason is that randomly rotating models in the original training set lead to learned features with higher complexity than the test set. Hence, it does not help improve recognition accuracy.

4.4. Robustness and Generalization. As mentioned in Section 3, rotation expansion can enhance the rotational invariance of recognition. It is also known that models in ModelNet-10 are well aligned and the 3D orientation of the models is highly consistent, whereas models in ModelNet-40 are not manually aligned but still have a certain regularity. Hence, whether the representation learned by the network has a high degree of rotational robustness cannot be ascertained by the experimental results of the original test set. A viable solution is to randomly rotate the models in the test set and compare the results of the original test set and the test set processed by random rotation, corresponding to the same training set (pipeline shown by the purple arrows in Figure 1). If the accuracy is equivalent, we can confirm that the representation learned from the expanded training set has a high degree of robustness and that the network can recognize more complex shapes, which also means good generalization performance.

Results corresponding to the original training set are illustrated in Figure 7. It is obvious that the difference in accuracy between the original test set and the random

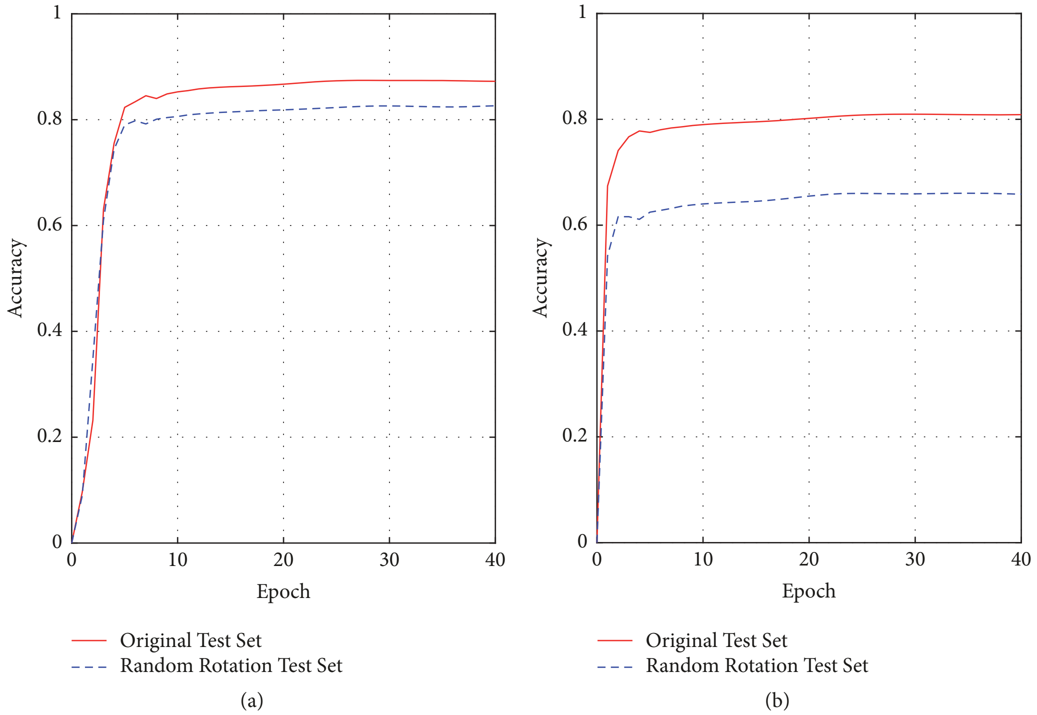


Figure 8: Accuracy results on different test sets corresponding to the regularly expanded training set: (a) ModelNet-10; (b) ModelNet-40. During each epoch, all samples in the training set were used once for training.

rotation test set is large. This proves that the representation learned by training with the original training set is not robust. This also indirectly proves that models in Modelnet-10 are well aligned, and although every model in ModelNet-40 is not well aligned, the 3D orientations have a certain regularity. With this kind of training set, the learned representation does not have sufficient rotational invariance to handle recognition on a more complex test set.

Figure 8 displays the results of the regularly expanded training set. A difference in accuracy between the two kinds of test sets still exists, but the gap is significantly narrower. Accuracy on the random rotation test set is still lower. Although the network trained by the training set obtains high rotational invariance though regular expansion, the complexity still does not match that of the random rotation test set.

Figure 9 shows the result of experiments on the randomly expanded training set. For both ModelNet-10 and ModelNet-40, the accuracy curves of the two kinds of test sets are almost coincident and the convergence value is close to the best recognition accuracy illustrated in Table 2. This means that representation obtained by training the network with the randomly expanded training set has a higher degree of rotational invariance; the network is able to identify shapes that have a more complex 3D orientation. Therefore this approach has the best generalization performance.

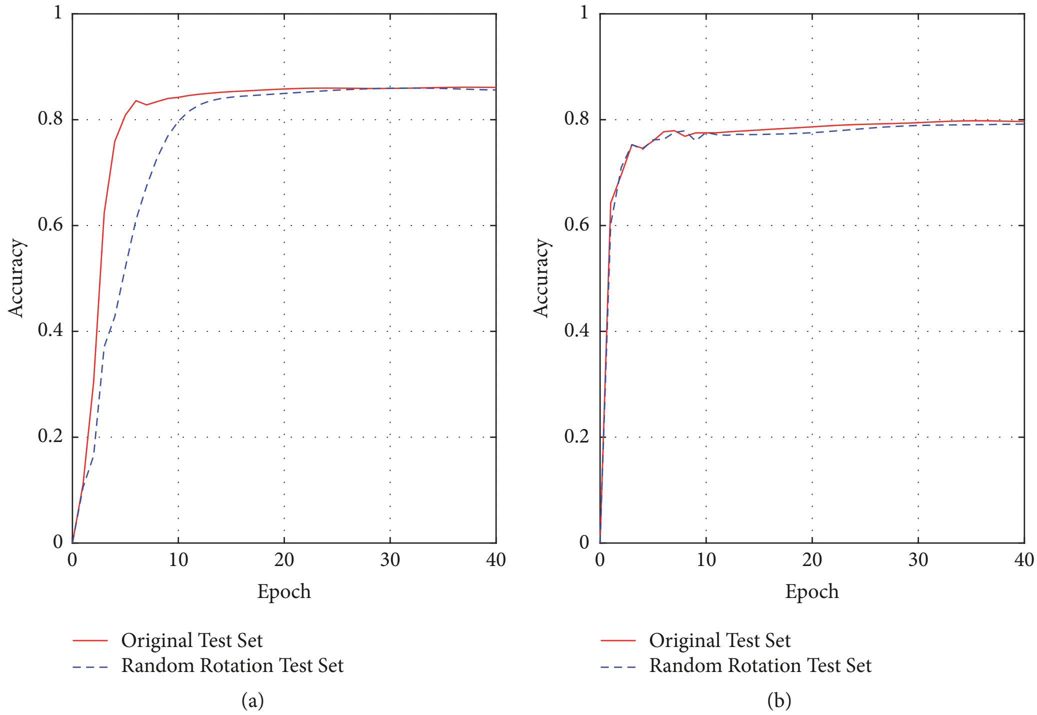
5. Conclusions

In this paper, we propose a method to process datasets based on panoramic view to improve the performance of 3D recognition. The expansion method can potentially lead to a higher accuracy, substantially increased robustness to more complex 3D shapes, and better generalization performance. We also discuss the relationship between the complexities of the training set and the test set; we conclude that when the complexities of the training and test sets match, the network will show best performance. These results suggest a promising future for real-time 3D recognition tasks.

Following on this work, there are a number of directions to explore in the future. During the experiment stage, we realized that the improvement in accuracy had an upper limit. This means that the method we applied has natural restrictions, and this may be due to the fact that projecting a 3D shape onto a cylindrical surface cannot completely retain the original 3D information. We will explore new methods to transform 3D shape into 2D data. Also, the databases, including the expanded ones we used, do not have a complexity that can be compared with the real world. The generalization performance of the network we trained is still far from meeting real-world needs. We need to discover methods of achieving higher robustness for various 3D gestures. In addition, instead of panoramic view, recognition based on a single perspective is worthy of study.

Data Availability

The ModeleNet-10 and ModeleNet-40 datasets used to support the findings of this study have been deposited in the Princeton ModelNet website (available from http://modelnet.cs.princeton.edu/).



(a)

(b)

Figure 9: Accuracy results on different test sets corresponding to the randomly expanded training set: (a) ModelNet-10; (b) ModelNet-40. During each epoch, all samples in the training set were used once for training.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported in part by Equipment Development Foundation of China (61404150303).

References

[1] J. Wang, J. Lu, W. Chen, and X. Wu, "Convolutional neural network for 3D object recognition based on RGB-D dataset," in *Proceedings of the 10th IEEE Conference on Industrial Electronics and Applications, ICIEA 2015*, pp. 34–39, New Zealand, June 2015.

[2] A. Tejani, D. Tang, R. Kouskouridas, and T. Kim, "Latent-Class Hough Forests for 3D Object Detection and Pose Estimation," in *Computer Vision – ECCV 2014*, vol. 8694 of *Lecture Notes in Computer Science*, pp. 462–477, Springer International Publishing, 2014.

[3] M. Pontil and A. Verri, "Support vector machines for 3D object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 637–646, 1998.

[4] S. Azzakhnini, L. Ballihi, and D. Aboutajdine, "Machine perception in gender recognition using RGB-D sensors," in *Proceedings of the 13th IEEE/ACS International Conference of Computer Systems and Applications, AICCSA 2016*, Morocco, 2017.

[5] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015*, pp. 922–928, Hamburg, Germany, October 2015.

[6] 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 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 5648–5656, USA, July 2016.

[7] Z. Wu, S. Song, A. Khosla et al., "3D ShapeNets: a deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1912–1920, Boston, Mass, USA, 2014.

[8] A. Sharma, O. Grau, and M. Fritz, "VConv-DAE: Deep Volumetric Shape Learning Without Object Labels," in *Proceedings of the in European Conference on Computer Vision*, 2016.

[9] D. Z. Wang and I. Posner, "Voting for voting in online point cloud object detection," in *Proceedings of the 2015 Robotics: Science and Systems Conference, RSS 2015*, Italy, July 2015.

[10] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 945–953, Santiago, Chile, December 2015.

[11] Y. Fang and etal., "3D deep shape descriptor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[12] K. Guo, D. Zou, and X. Chen, "3D mesh labeling via deep convolutional neural networks," *ACM Transactions on Graphics*, vol. 35, no. 1, pp. 1–12, 2015.

[13] B. Shi, S. Bai, Z. Zhou, and X. Bai, "DeepPano: Deep Panoramic Representation for 3-D Shape Recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[14] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[15] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, "Deep Learning Representation using Autoencoder for 3D Shape Retrieval," *Neurocomputing*, vol. 204, pp. 41–50, 2016.

[16] M. Ren, L. Niu, and Y. Fang, *3D-A-Nets: 3D Deep Dense Descriptor for Volumetric Shapes with Adversarial Networks*, 2017.

[17] X. Bai, S. Bai, Z. Zhu, and L. J. Latecki, "3D Shape Matching via Two Layer Coding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2361–2373, 2015.

[18] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Computer Graphics Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[19] P. Papadakis, I. Pratikakis, T. Theoharis, and S. Perantonis, "Panorama: a 3D shape descriptor based on panoramic views for unsupervised 3d object retrieval," *International Journal of Computer Vision*, vol. 89, no. 2-3, pp. 177–192, 2010.

[20] F. J. Pulgar, A. J. Rivera, F. Charte, and M. J. del Jesus, "On the Impact of Imbalanced Data in Convolutional Neural Networks Performance," in *Hybrid Artificial Intelligent Systems*, vol. 10334 of *Lecture Notes in Computer Science*, pp. 220–232, Springer International Publishing, 2017.

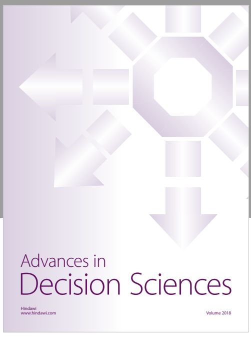
[21] S. Bai, X. Bai, W. Liu, and F. Roli, "Neural shape codes for 3D model retrieval," *Pattern Recognition Letters*, vol. 65, pp. 15–21, 2015.

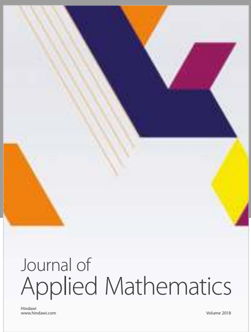
[22] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proceedings of the Eurographics acm SIGGRAPH Symposium on Geometry Processing*, 2003.

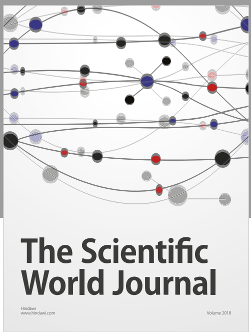
[23] D. Y. Chen et al., "On visual similarity based 3D model retrieval," in *Computer Graphics Forum*, 2003.

[24] M. Buda, A. Maki, and M. A. Mazurowski, *A systematic study of the class imbalance problem in convolutional neural networks*, 2017.

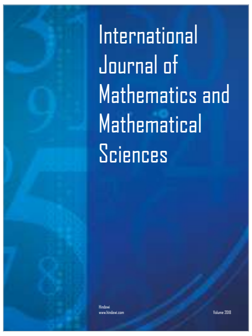
Advances in
Operations Research


Advances in
Decision Sciences

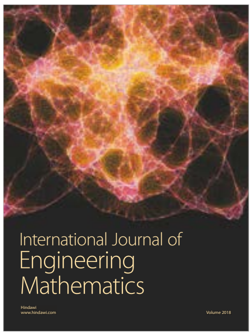

Journal of
Applied Mathematics


The Scientific
World Journal

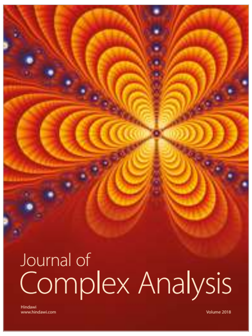
Journal of
Probability and Statistics

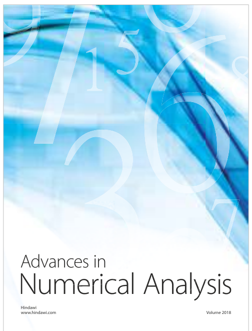

International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Optimization

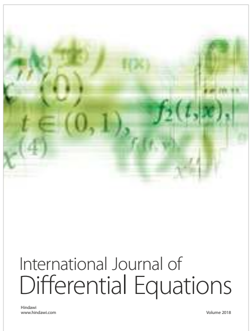

International Journal of
Engineering
Mathematics

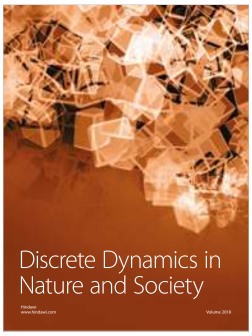
International Journal of
Analysis

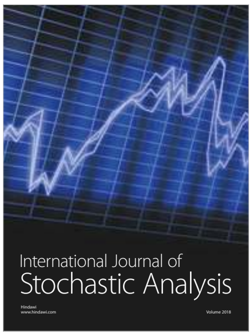

Journal of
Complex Analysis


Advances in
Numerical Analysis

Mathematical Problems
in Engineering

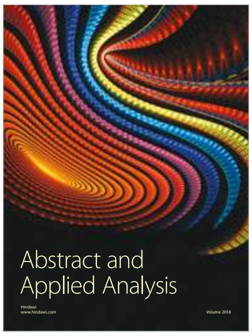

International Journal of
Differential Equations

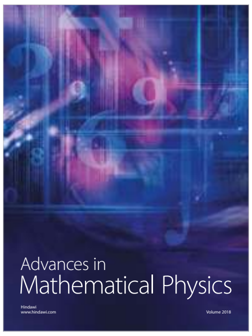

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics