J Inf Process Syst, Vol.15, No.5, pp.1179~1191, October 2019
https://doi.org/10.3745/JIPS.02.0120

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Learning-Based Multiple Pooling Fusion in Multi-View Convolutional Neural Network for 3D Model Classification and Retrieval

Hui Zeng*, Qi Wang*, Chen Li**, and Wei Song**

Abstract

We design an ingenious view-pooling method named learning-based multiple pooling fusion (LMPF), and apply it to multi-view convolutional neural network (MVCNN) for 3D model classification or retrieval. By this means, multi-view feature maps projected from a 3D model can be compiled as a simple and effective feature descriptor. The LMPF method fuses the max pooling method and the mean pooling method by learning a set of optimal weights. Compared with the hand-crafted approaches such as max pooling and mean pooling, the LMPF method can decrease the information loss effectively because of its "learning" ability. Experiments on ModelNet40 dataset and McGill dataset are presented and the results verify that LMPF can outperform those previous methods to a great extent.

Keywords

Learning-Based Multiple Pooling Fusion, Multi-View Convolutional Neural Network, 3D Model Classification, 3D Model Retrieval

1. Introduction

Due to the advances of computer technology, the objects of the information processing system have gradually changed from single text information to multimedia information which including 2D images, 3D models, 3D scenes, and so on. With 3D devices (such as Core3D, 3Dmax, AutoCAD, etc.) being constantly updated, the acquisition of 3D models becomes more and more simple. At the same time, the demands and requirements for 3D model classification and retrieval technology are getting higher and higher. We live in a 3D world, and all objects exist in 3D forms. Our vision system not only can perceive 2D information, but also has 3D stereo characteristics. Compared with 2D images, 3D models come into greater alignment with the cognitive characteristics of human vision, and they can provide more detailed discriminative information. So the research on 3D model classification and retrieval is attracting more researchers' attentions.

Usually, feature extraction is the most key part in 3D model classification/retrieval system. It aims to

Manuscript received June 9, 2017; first revision August 1, 2017; second revision September 11, 2017; third revision September 27, 2017; accepted January 28, 2018.
Corresponding Author: Chen Li (lichen@ncut.edu.cn)
* Beijing Engineering Research Center of Industrial Spectrum Imaging, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China (hzeng@ustb.edu.cn, ustb_wangqi@163.com)
** School of Computer Science and Technology, North China University of Technology, Beijing, China (lichen@ncut.edu.cn, sw@ncut.edu.cn)

extract the efficient and discriminative characteristics from 3D models and it determines the performance of classification/retrieval directly. The existing methods for feature extraction can be sorted into the following two categories:

1) **3D Model Based Feature Extraction**: This kind of method works on the native 3D models directly to represent the shape information according to the grid structure or the voxel structure of 3D models. For example, the features of 3D models can be described as shape histogram [1], point density [2], shape distribution [3], bag-of-features models [4], and spherical harmonic representation [5], and so on. These aforementioned methods describe the shape features of the 3D model as a whole, and the similarity distances between two 3D models are usually used for classification/retrieval. These methods are more aligned with human visual recognition because they usually ignore the differences in details but focus on the whole structure. In spite of this, there are a certain number of problems about these methods. At first, the discriminative ability is weakened because of the information loss. Therefore they are only suitable for rough classification. Secondly, the dimensions of these descriptors are often very high, and they cause the classifier to be overfitting easily.

2) **Multi-view Based Feature Extraction**: This kind of method describes a 3D model through a group of images. By placing different projectors in different directions, 3D models can be projected into a series of images. And then we can achieve the purpose of classification and retrieval by comparing the similarity of those projected images. Compared with 3D model based feature extraction methods, this kind of method has some advantages. For example, plenty of mature 2D image classification/retrieval techniques can be fully applied to 3D field. The descriptors of these projected images are relatively low-dimensions, and that makes it get rid of the problem of "curse of dimensionality". In addition, the classification/retrieval of the 3D models doesn't need the original 3D models [6]. At the early stage, most methods mainly focuse on "shallow features", including color characteristics, texture features, or bag-of-words (BoW) models established by some descriptors such as SIFT [7] and SURF [8]. Later different algorithms from machine learning [9] are applied to this field. Although these aforementioned methods have been applied in 3D model classification and retrieval successfully, there remains much room for improvement. How to extract effective features from multiple views is also a technological challenge to be solved. For the past few years, convolutional neural network (CNN) [10] has caused widespread concern in academia. The CNN based feature extraction methods work a lot better in classification/retrieval tasks than traditional methods which are "hand-crafted".

Recently, researchers have started studying the application of CNN in 3D model classification/retrieval. For example, Su et al. [11] proposed a network named multi-view convolutional neural network (MVCNN), which achieved a successful combination of 3D models and CNN. At first, they presented a basic CNN model which is trained to classify the multiple 2D projected images independently. Their experiments have shown that even a single-view based method has higher accuracy than the other existing methods mentioned in the paper. Then an ingenious CNN structure was proposed to improve retrieval accuracy. The principle is that multi-view feature maps of 3D models are transformed into a single and efficient descriptor. The experimental results on ModelNet40 dataset have validated the effectiveness of the MVCNN architecture. The view-pooling layer uses element-wise max pooling strategy to combine the discriminative information of multiple views and increase the computational efficiency. The view-pooling method is similar to traditional max-pooling operation, which may result in some information loss. What's more, the traditional pooling operation can't ensure the error minimized in the training

phase. To improve the performance, this paper presents a modified view-pooling method called learning-based multiple pooling fusion (LMPF). It minimizes the training error by learning a set of optimal weights for the best fusion of multiple different view-pooling methods. Experiments can prove that our LMPF method acts excellent on two popular 3D shape datasets and is more efficient than the method proposed in [11].

The structure of our work is as below. Section 2 involves an overview of CNN and some pooling methods. Section 3 introduces the architecture and implementation of MVCNN. Meanwhile, the proposed LMPF method is also elaborated in detail. Section 4 gives the procedures and analysis about our experiments. Section 5 provides the final conclusion about our work at last.

2. Related Work

This section first reviews the origin and development of CNN, and then gives a brief summary about the common methods of pooling.

2.1 Convolutional Neural Network

Since the concept of 'deep learning' was first raised by Hinton and Salakhutdinov [12] in 2006, it has exerted a profound impact on academia and industry. Through a multi-layer structure, all of these types of methods can transform the underlying features into a high-level feature which is more compact. So they have excellent ability of feature learning. CNN is a type of representational deep model and has achieved breakthrough performance in computer vision. LeCun et al. [10] proposed the first CNN in 1998 with the purpose of recognizing hand-written letters. In 2012, Krizhevsky et al. [13] won the championship in ILSVRC by applying CNN model. The second team used the SIFT + FVs, which is the best traditional method of the image field, and was 11 percentage points worse than the CNN based method. This was a milestone for the development of CNN. In the 2013's ILSVRC, Zeiler and Fergus [14], as well as He et al. [15], Simonyan and Zisserman [16], Szegedy et al. [17] in the 2014's competition, perfected Hinton's CNN model and achieved higher retrieval accuracy. So far, the image classification ability of CNN model has gone beyond the ability of human identification. Although the CNN model has successfully applied in the domain of image classification/retrieval, how to apply it in the 3D model domain is still lack of systematic investigation.

A typical CNN network, in general, constitutes by convolutional layer and pooling layer. The former is designed to refine features of input images within local regions. The latter aims to refine the feature maps that obtained from convolutional layer by preserving important information and abandoning irrelevant information. To some extent, the feature maps after pooling layer will be more robust because of the information aggregation within each local region by pooling operator. In addition, the pooling operation can enhance the distortion invariance of inputs. So in short, CNN model can select compact and robust features, which remains the most important discriminative information while decreasing the computational complexity.

2.2 Pooling Method

As we all know the pooling layer acts a particular role in CNN model. Common pooling methods are

usually divided into *hand-crafted pooling* and *learning-based pooling*.

Hand-crafted pooling: This category contains most of the well-known pooling approaches, such as max pooling and mean pooling. The former takes the maximum of the region as output while the latter takes the average. Although they have been widely used in CNN model, there are some disadvantages about them. For example, the max pooling can be easily falling into overfitting, and the generalization ability of the network is poor. The mean pooling cannot reflect the characteristics of the pooling area because the smaller elements weaken the contribution of larger activation values. To settle the aforementioned problems, the stochastic pooling was proposed by Zeiler and Fergus [18]. The selection of the activation value is random, which is beneficial to avoid overfitting in some degree. The selection probability is obtained by normalizing all the activation values, and one of activation values is selected as the output according to the multinomial distribution of selection probability. Furthermore, the spatial pyramid pooling method proposed by He et al. [15] and multi-pooling operation applied by Zhong et al. [19] all belong to hand-crafted pooling. These methods have been successfully used in CNN model, but there are still some limitations.

Learning-based pooling: This category joins the "learning" strategy into the traditional pooling. It aims to train a set of weight parameters which act on the activation value in pooling region. These weight parameters are learned during the end-to-end network training. Lee et al. [20] proposed a learning operation that mixes various traditional pooling methods to replace the original pooling layer in the CNN model. This kind of methods usually reach better effect than the hand-crafted pooling methods, but the model complexity becomes higher due to the increase of parameters. Furthermore, although several learning-based pooling methods have been proposed, we haven't found related reports about learning-based view-pooling. So the learning-based pooling methods for multiple views and its application in CNN need to be deeply studied.

3. Learning-based Multiple Pooling Fusion in MVCNN

This section begins with an introduction of MVCNN model, followed by a detailed analysis of the proposed LMPF method and its application in MVCNN. At last, we describe the implementation procedures of our experiments.

3.1 Architecture of MVCNN

The MVCNN model proposed in [11] is adopted as the baseline model in this paper. The structure of MVCNN is designed based on the basic CNNs by adding a new layer named view-pooling. As shown in Fig. 1, the first part of MVCNN is CNN1, which is formed by five convolutional layers (Conv1–Conv5) to deal with multiple views. For each branch of CNN1, its input is a projected image of the 3D model and its output is a feature map of the corresponding projected image. The quantity of branches is equal to the quantity of views. The second part of MVCNN is the view-pooling layer, which aggregates the feature maps of multiple views into one feature map. The third part of MVCNN is CNN2, which is comprised of three fully-connected layers (Fc6–Fc8). CNN1 is connected with CNN2 by the view-pooling layer. In fact, we can put the view-pooling layer in any location of the network. Our experimental results have shown that when it is set after the Conv5 layer, the classification/retrieval results can reach optimums.

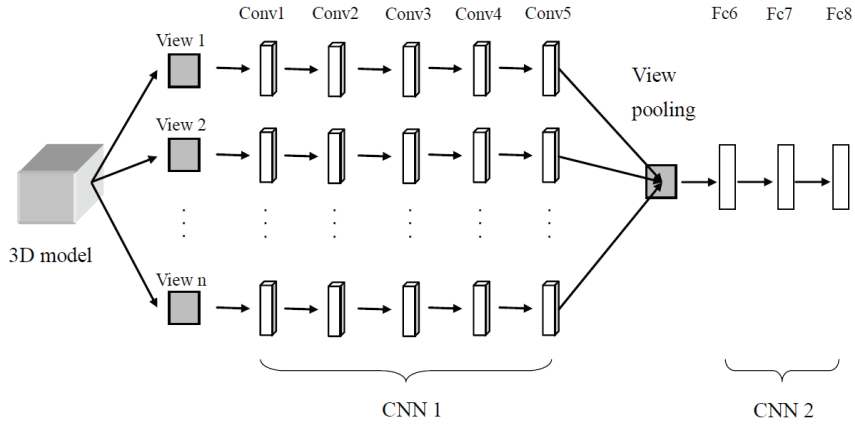


Fig. 1. Architecture of MVCNN model.

The detailed parameter settings of basic CNN structure are organized in Table 1. In the first row of Conv layers, " $num \times size \times size$ " represents the filter numbers and their receptive field size in convolution layers. The forth row gives the patch size of max pooling. In the last three columns (Fc6-Fc8), we give the information of their dimensionality and whether the dropout strategy is applied. The last layer is used as a softmax classifier. And ReLU is positioned after each layer (except for Fc8) as activation function. All branches of CNN1 have the same parameters, and the training phase is accomplished by Stochastic Gradient Descent (SGD) algorithm.

Table 1. Detailed parameters setting of the basic CNN architecture

Conv1	Conv2	Conv3	Conv4	Conv5	Fc6	Fc7	Fc8
96×11×11	256×5×5	512×3×3	512×3×3	512×3×3	4096	4096	classes
stride=2	stride=2	stride=1	stride=1	stride=1	dropout	dropout	softmax
pad=1	pad=1	pad=1	pad=1	pad=1	-	-	-
2×2 pool	2×2 pool	-	-	2×2 pool	-	-	-

3.2 Learning-based Multiple Pooling Fusion

In this paper, the purpose of view-pooling layer is to convert the multi-view feature maps to a single and effective descriptor. From [11], we can conclude that the method with element-wise maximum pooling across the multiple views works better than the method without view-pooling layer. And the element-wise maximum pooling is more effective than the element-wise mean pooling. However, neither of the above two kinds of pooling methods may be qualified to be optimal. Selecting the maximum or average as activation of view-pooling layer can result in a loss of significant information. And the model will fall into overfitting to a great extent. In allusion to these issues, we put forward the LMPF method. It can aggregate the features of multiple views by learning a set of optimal weights and can fuse different view-pooling methods effectively. In other words, it introduces the "learning" ability into the previous hand-crafted view-pooling method, ensuring that the training error is minimized throughout the whole training phase. Fig. 2 presents the illustrations of three kinds of view-pooling methods. Fig. 2(a) shows

the max pooling method in view-pooling layer, which perform the maximum operation. The red rectangle represents the max value in the corresponding area of all n views. Fig. 2(b) shows the mean pooling method in view-pooling layer, which perform the average operation. The red rectangles indicate that all elements of the same pooled area are involved in the operation. Fig. 2(c) shows the LMPF method, which is a combination of maximum operation and average operation by setting a set of learnable weights.

As is shown in Fig. 2, we design our view-pooling method based on the max pooling method and the mean pooling method. These two kinds of methods act similarly with the traditional max or average operation in the pooling layers, and the only difference is that the pooling region is changed from an area in one feature map to a set of corresponding elements across multi-views (the sub-rectangles of each view in Fig. 2). Suppose that the last feature maps for view-pooling are $[m_1, m_2, ..., m_n]$, and n is defined as the quantity of views. The value of a certain point on feature map m_k can be written as $\alpha_k(p,q)$, in which p represents the abscissa and q represents the ordinate. Then for the max pooling method in view-pooling layer, the corresponding output $o_{\max}(p,q)$ of the location (p,q) is the maximum of $[\alpha_1(p,q), \alpha_2(p,q), ..., \alpha_n(p,q)]$. Then the expression for $o_{\max}(p,q)$ appears as shown below:

$$o_{max}(p,q) = max\{\alpha_1(p,q), \alpha_2(p,q), ..., \alpha_n(p,q)\} \tag{1}$$

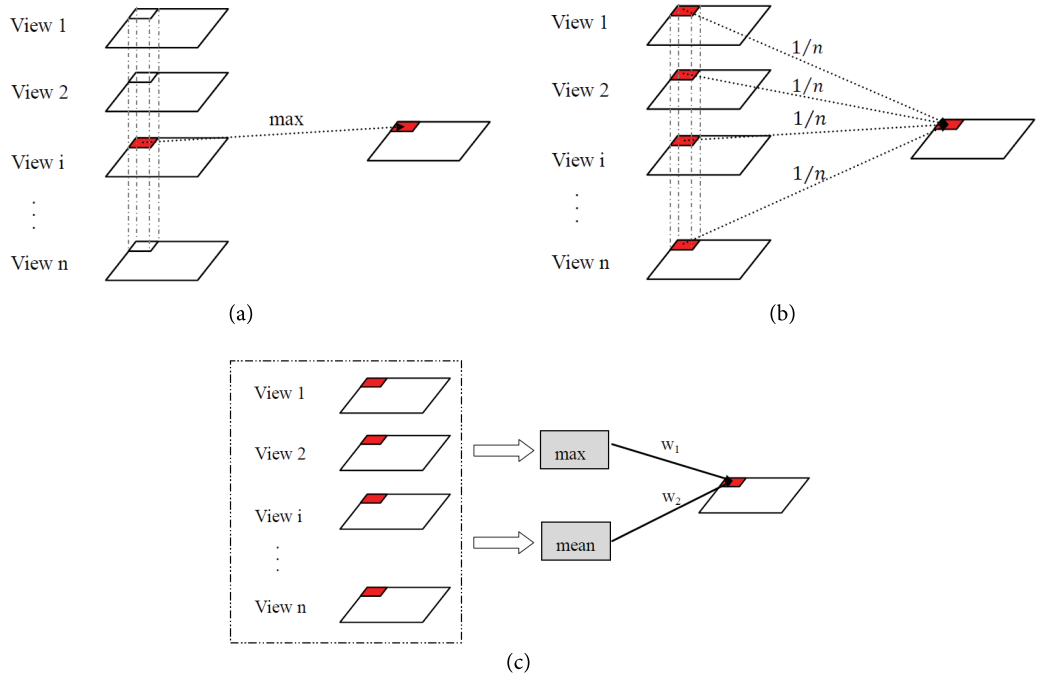


Fig. 2. The illustrations of three kinds of view-pooling methods: (a) the max pooling method in view-pooling layer, (b) the mean pooling method in view-pooling layer, and (c) the LMPF method.

For the mean pooling method in view-pooling layer, the corresponding output $o_{mean}(p,q)$ at location (p,q) is the mean value of $[\alpha_1(p,q), \alpha_2(p,q), ..., \alpha_n(p,q)]$. Similarly, the expression for $o_{mean}(p,q)$ is as below:

$$o_{mean}(p,q) = mean\{\alpha_1(p,q),\alpha_2(p,q),...,\alpha_n(p,q)\} \quad (2)$$

For our proposed learning based view-pooling method, the output can be expressed by weighted sum of the results obtained from max pooling method and mean pooling method in view-pooling layer.

$$o_{LMPF}(p,q) = w_1 * o_{max}(p,q) + w_2 * o_{mean}(p,q) \quad (3)$$

where w_1 and w_2 are the weights of max pooling and mean pooling. The weights are initialized by small random values, as long as their sum is guaranteed to be 1. And the standard back propagation (BP) algorithm is applied to search optimization weights in the whole training phase. From Eq. (3) we can conclude that our method is a fusion of the max pooling strategy and the mean pooling strategy for multiple feature maps, and its purpose is to select a set of optimum values for w_1 and w_2 by learning in the end-to-end training phase. So the method that we proposed can combine max pooling and mean pooling effectively, and it can reduce information loss in the view-pooling stage.

3.3 Implementation Procedures

In summary, the implementation procedures of our experiments can be recapitulated as follows:

1) **Input:** Generate multiple projected images of each 3D model in dataset as the input of the MVCNN model.
2) **Initialization:** Initialize each convolutional layer of MVCNN randomly, and set proper values to the related parameters, such as learning rate, momentum, the weights $[\omega_1,\omega_2]$ of LMPF and so on.
3) **Training phase:** Choose the SGD algorithm to fine-tune the MVCNN model on the training dataset. Then we can obtain the optimal values of the weights.
4) **Classification/Retrieval:** Use the linear SVMs method [21] to classify and the L_2 distance [6] to retrieve on testing dataset.

4. Experimental Results

To validate the accuracy and feasibility of LMPF objectively, two datasets named ModelNet40 [22] and McGill [23] are used in our experiments. All CNN models in our experiments are built by MatConvNet toolbox [24]. Our experimental environment is MATLAB R2014a based on i7-6700 CPU 3.40 GHz 12.0G memory Lenovo computer. To analyze the experimental results comprehensively, we choose the following indicators to measure the performance of classification and retrieval: accuracy for classification, mean average precision (mAP), nearest neighbor (NN), the first tier (FT), the second tier (ST) and discounted cumulative gain (DCG) for retrieval.

In our experiments, we first obtained the multiple projected images of 3D models in different views. Then we used the LMPF based MVCNN to perform 3D classification/retrieval experiments. The detailed steps have been listed in Section 3.3. Finally, we testified the effective of LMPF method, and compared it with other methods. The momentum of MVCNN is set to 0.5, and the initial values of $[\omega_1,\omega_2]$ is initialized as $[1,0]$. In the process of learning, the view-pooling layer is placed after Conv5. And we adopt the SGD algorithm to satisfy the update of parameters in training phase.

4.1 ModelNet40 Dataset

ModelNet40 is a subset of ModelNet which is published on the Princeton ModelNet website [22]. This dataset contains a total of 12,311 well-annotated shapes from 40 common categories. We construct the training dataset and testing dataset of ModelNet40 in accordance with the study [11]. Fig. 3 shows some sample models of ModelNet40.

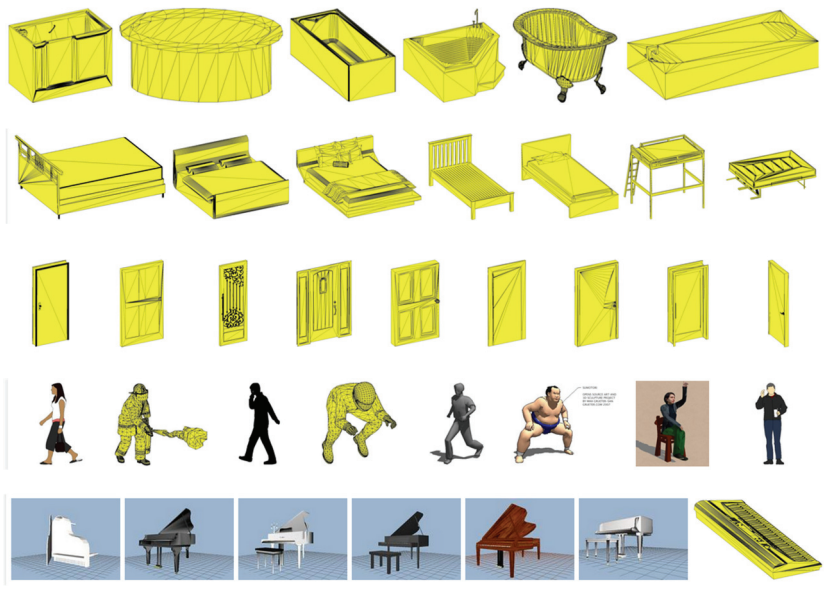


Fig. 3. Some example models from the ModelNet40 dataset.

We use the 1st camera setup mentioned in [11] to obtain the multiple projected images as the inputs of the MVCNN model. The 1st camera setup requires that the input 3D model is placed vertically according to a constant axis (most 3D model datasets conform to this assumption, including ModelNet40). For each 3D model, there are 12 virtual cameras in an interval of 30° placed around it. And each of the cameras aims at the center of the model with a 30° angle to the horizontal. In this case, we can capture 12 views of each 3D model. An illustration is provided in Fig. 4.

At first, we make a contrast between our proposed LMPF method and the other two hand-crafted approaches (max pooling and mean pooling). Table 2 summarizes our 3D model classification/retrieval results on ModelNet40 dataset with three kinds of view-pooling methods. Obviously, our LMPF method in Table 2 shows the best performance both in classification and retrieval. It outperforms mean pooling by nearly 4% in mAP and max pooling by nearly 1%. All in all, we can conclude that LMPF can decrease the information loss effectively on account of its 'learning" ability in training phase.

Then we perform experiments on ModelNet40 dataset to make the contrast of our method with other classification methods. Table 3 gives the comparative classification/retrieval results on ModelNet40 dataset. It is clear that our method outperforms the others by nearly 10%–20% in classification accuracy and 20%–40% in mAP. These results further verify the conclusion that LMPF can boost the performance effectively.

Table 2. Classification/retrieval results on ModelNet40 dataset with three kinds of view-pooling methods

Pooling method	Classification	Retrieval				
	Accuracy (%)	mAP (%)	NN (%)	FT (%)	ST (%)	DCG (%)
Mean pooling	88.00	64.40	86.38	64.42	75.25	87.63
Max pooling	89.90	70.10	88.13	70.81	79.62	90.11
LMPF	89.90	71.00	88.88	71.61	81.08	90.54

Table 3. Comparative classification/retrieval results on ModelNet40

Pooling method	Classification	Retrieval				
	Accuracy (%)	mAP (%)	NN (%)	FT (%)	ST (%)	DCG (%)
SPH [5]	68.20	33.30	-	-	-	-
LFD [25]	75.50	40.90	-	-	-	-
3D ShapeNets [26]	77.30	49.20	-	-	-	-
LMPF	89.90	71.00	88.88	71.61	81.08	90.54

4.2 McGill Dataset

The McGill dataset is provided on the McGill 3D Shape Benchmark website, which involves a variety of 3D models [23]. In our experiments, the McGill dataset we used is formed from a set number of non-rigid 3D models selected from the above website. There are 255 models in this dataset, and they are divided into 10 classes. Each class has nearly 25 models in different posture and appearance. Fig. 4 shows a series of sample models of this dataset.

In our experiments, 24 virtual cameras are placed on the surface of the sphere surrounding the model to produce multi-view projection images. The model center coincides with the sphere center and all cameras aim at the centre of the 3D model. The location of cameras can be obtained through Isocube Spherical Map method [27] which typically contains two steps. Firstly, the sphere is divided into six equal areas. We divide the sphere into equatorial region and two polar crowns with two parallel circles, and then the equatorial region is divided into four symmetrical regions. It is shown that these six regions are equal in size (refer to Ref. [27] for mathematical proof). Secondly, we subdivide each area with different accuracy of N to generate many smaller areas of equal size. Then the cameras are placed in the center of each small area. In this paper, we choose N = 2 as the segmentation accuracy, yielding total 24 views per model.

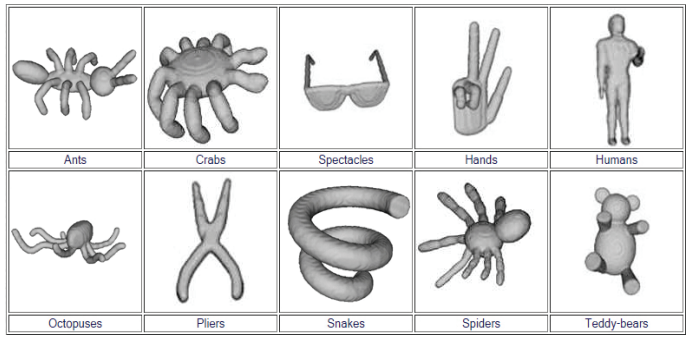


Fig. 4. Sample 3D shapes of McGill dataset [23].

Similar to the experiments on ModelNet40 dataset, firstly, a series of contrastive experiments are performed among the above three view-pooling methods. All the outcomes of experiments are recorded in Table 4. According to Table 4, it is obvious that LMPF achieves the best results whether in classification or retrieval, and outperforms the max pooling method and the mean pooling method by nearly 1%–4% in classification accuracy and 2%–5% in retrieval measures. Then we make a comparison between LMPF and other retrieval methods of McGill dataset, and summarize the retrieval indicators in Table 5. Through the anatomization of Table 5, we can verify the conclusion that LMPF method really outperforms other methods in the domain of 3D model retrieval. In summary, it is obvious that LMPF strategy is a relatively better method for MVCNN compared to those hand-crafted methods.

Table 4. Classification/retrieval results on McGill dataset with three kinds of view-pooling methods

Pooling method	Classification	Retrieval				
	Accuracy (%)	mAP (%)	NN (%)	FT (%)	ST (%)	DCG (%)
Mean pooling	93.30	85.20	95.24	80.70	93.71	93.47
Max pooling	96.20	86.10	95.24	81.46	93.90	94.37
LMPF	97.10	88.20	98.10	86.10	95.94	95.79

Table 5. Comparative classification/retrieval results on McGill

Pooling method	Classification	Retrieval				
	Accuracy (%)	mAP (%)	NN (%)	FT (%)	ST (%)	DCG (%)
Covariance [28]	-	-	97.70	73.20	81.80	93.70
Graph-based [29]	-	-	97.60	74.10	91.10	93.30
PCA-based VLAT [30]	-	-	96.90	65.80	78.10	89.40
Hybrid BOW [31]	-	-	95.70	63.50	79.00	88.60
LMPF	97.10	88.20	98.10	86.10	95.94	95.79

5. Conclusions

In summary, we have proposed an ingenious view-pooling method named Learning-based Multiple Pooling Fusion (LMPF) in our work. And on the basis of multiple experiments, it is verified that this method can be successfully applied to the MVCNN model. At first, we generate multiple projected images of 3D models and use them as the inputs of the MVCNN model. Secondly, initialize each convolutional layer of MVCNN randomly, and set proper values to the related parameters. Then fine-tune the network by SGD algorithm, so that we can get a group of optimal weights for MVCNN model. Finally, the linear SVM is used for classifying and the L_2 distance is used for retrieving. The results show that LMPF has more efficient performance than traditional hand-crafted view-pooling methods. So in general, the LMPF method that we proposed in this paper combines the learning-based pooling method and the hand-crafted pooling method, and can decrease the information loss effectively. In the future, we will further optimize the architecture of the MVCNN and investigate more effective view-pooling methods.

Acknowledgement

This work couldn't have been finished successfully without the great support of the National Natural

Science Foundation of China (No. 61375010 and No. 61503224) and Beijing Key Discipline Development Program (No. XK100080537).

References

[1] M. Ankerst, G. Kastenmuller, H. P. Kriegel, and T. Seidl, "3D shape histograms for similarity search and classification in spatial databases," in *Advances in Spatial Databases*. Heidelberg: Springer, 1999, pp. 207-226.

[2] M. T. Suzuki, T. Kato, and N. Otsu, "A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors," in *Proceedings of 2000 IEEE International Conference on Systems, Man and Cybernetics*, Nashville, TN, 2000, pp. 2946-2952.

[3] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 4, pp. 807-832, 2002.

[4] B. K. P. Horn, "Extended Gaussian images," *Proceedings of the IEEE*, vol. 72, no. 12, pp. 1671-1686, 1984.

[5] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proceedings of the 2003 Eurographics Symposium on Geometry Processing*, Aachen, Germany, 2003, pp. 156-164.

[6] S. K. Vipparthi and S. K. Nagar, "Color directional local quinary patterns for content based indexing and retrieval," *Human-centric Computing and Information Sciences*, vol. 4, article no. 6, 2014.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints,"*International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," in *Computer Vision-ECCV 2006*. Heidelberg: Springer, 2006, pp. 404-417.

[9] J. Zhu, R. San-Segundo, and J. M. Pardo, "Feature extraction for robust physical activity recognition," *Human-centric Computing and Information Sciences*, vol. 7, article no. 16, 2017.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[11] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 945-953.

[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504-507, 2006.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012.

[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision-ECCV 2014*. Cham: Springer, pp. 818-833.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision-ECCV 2014*. Cham: Springer, pp. 346-361.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014 [Online]. Available: https://arxiv.org/abs/1409.1556.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, 2015, pp. 1-9.

[18] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," 2013 [Online]. https://arxiv.org/abs/1301.3557.

[19] Z. Zhong, L. Jin, and Z. Feng, "Multi-font printed Chinese character recognition using multi-pooling convolutional neural network," in *Proceedings of 2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015, pp. 96-100.

[20] C. Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: mixed, gated, and tree," *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Cadiz, Spain, 2016, pp. 464-472.

[21] M. Zouina and B. Outtaj, "A novel lightweight URL phishing detection system using SVM and similarity index," *Human-centric Computing and Information Sciences*, vol. 7, article no. 17, 2017.

[22] The Princeton ModelNet [Online]. Available: http://modelnet.cs.princeton.edu.

[23] McGill 3D Shape Benchmark [Online]. Available: http://www.cim.mcgill.ca/~shape/benchMark.

[24] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM International Conference on Multimedia*, Brisbane, Australia, 2015, pp. 689-692.

[25] 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.

[26] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: a deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, 2015, pp. 1912-1920.

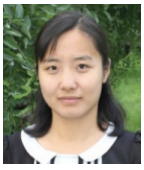
[27] L. Wan, T. T. Wong, C. S. Leung, "Isocube spherical mapping," *Journal of Computer-Aided Design & Computer Graphics*, vol. 20, no. 8, pp. 978-985, 2008.

[28] H. Tabia, H. Laga, D. Picard, and P. H. Gosselin, "Covariance descriptors for 3D shape matching and retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 4185-4192.

[29] A. Agathos, I. Pratikakis, P. Papadakis, S. J. Perantonis, P. N. Azariadis, and N. S. Sapidis, "Retrieval of 3D articulated objects using a graph-based representation," in *Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR)*, Munich, Germany, 2009, pp. 29-36.

[30] H. Tabia, D. Picard, H. Laga, and P. H. Gosselin, "Compact vectors of locally aggregated tensors for 3D shape retrieval," in *Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR)*, Girona, Spain, 2013, pp. 17-24.

[31] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, and S. Perantonis, "3D object retrieval using an efficient and compact hybrid shape descriptor," in *Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR)*, Crete, Greece, 2008, pp. 9-16.

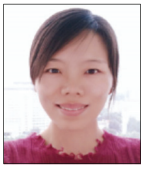


Hui Zeng https://orcid.org/0000-0002-4137-7424

She received B.S. and M.S. degrees from Shandong University in 2001 and 2004, respectively, and received the Ph.D. degree from National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences in 2007. She is currently an associate professor at School of Automation and Electrical Engineering, University of Science and Technology Beijing, China. Her main research interests include computer vision, pattern recognition and machine learning.

Qi Wang https://orcid.org/0000-0002-1256-4327

She received B.S. degree from University of Science & Technology Beijing in 2016. Now she is currently a graduate student at School of Automation and Electrical Engineering, University of Science and Technology Beijing, China. Her current research direction is computer vision and pattern recognition.



Chen Li https://orcid.org/0000-0001-5983-5895

She received the Ph.D. in Control Science and Control Engineering from the University of Science and Technology Beijing, China, in 2013. She has been an associate professor at North China University of Technology, China, since 2017. She has long been engaged in the research and development and teaching work of image processing, pattern recognition, and information hiding.

Wei Song https://orcid.org/0000-0002-5909-9661

He received his B.Eng. degree in Software Engineering from Northeastern University, Shenyang, China, in 2005 and his M.Eng. and Dr.Eng. degrees in the Department of Multimedia from Dongguk University, Seoul, Korea, in 2008 and 2013, respectively. Since September 2013, he has been an Associate Professor at the department of Digital Media Technology of North China University of Technology. His current research interests are focused on IoT, virtual reality, and multimedia technologies.