Multi-view Rank Pooling for 3D Object Recognition**

Chaoda Zheng*, Yong Xu*†‡, Ruotao Xu*, Hongyu Chi† and Yuhui Quan*§
* School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
†Peng Cheng Laboratory, Shenzhen, China
‡Communication and Computer Network Laboratory of Guangdong, China
§Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, China

***Abstract*—3D shape recognition via deep learning is drawing more and more attention due to huge industry interests. As 3D deep learning methods emerged, the view-based approaches have gained considerable success in object classification. Most of these methods focus on designing a pooling scheme to aggregate CNN features of multi-view images into a single compact one. However, these view-wise pooling techniques suffer from loss of visual information. To deal with this issue, an adaptive rank pooling layer is introduced in this paper. Unlike max-pooling which only considers the maximum or mean-pooling that treats each element indiscriminately, the proposed pooling layer takes all the elements into account and dynamically adjusts their importances during the training. Experiments conducted on ModelNet40 and ModelNet10 shows both efficiency and accuracy gain when inserting such a layer into a baseline CNN architecture.**

I. Introduction

3D shape recognition task is a fundamental challenge in computer vision domain. Compared with 2D images, 3D data contain more geometric information and can encode more informative real world structures. However, 3D object analyzing is a still challenging task due to the irregularity of data representation. Unlike 2D images which are in the form of matrices, 3D data can be stored and represented in various format (polygon meshes, point clouds, voxels, multi-view images, etc). With the recent progress in large scale 3D shape datasets, 3D deep learning has shown its great potential in the 3D domain [1]–[14]. And those methods can be divided into two trends in terms of their input formats: shape-based methods and view-based methods. Shape-based methods directly consume the native formats of 3D models such as voxels and point clouds. Voxel-based methods partition the 3D space into regular grids and thus they can use a 3D tensor to represent a 3D shape. Due to the regularity of voxels, it's intuitive to directly apply 3D convolutional neural networks on voxelized 3D shapes [1]–[3], [5], [15]. However, the computational and memory cost of these methods grow cubically as the data resolution increases. Differently, a point cloud represents a 3D shape with a set of surface points, which are permutation invariant. The pioneer of point-based methods is PointNet [16], which utilizes a per-point MLP followed by a max-pooling to extract the global feature of a given point cloud. And PointNet++ [17] further improved the PointNet architecture using a multi-scale grouping scheme, so that the geometric distribution of local regions can be considered. Besides processing point clouds using PointNets as basic blocks, graph convolutional networks can also be used for point clouds analysis, since a point cloud can be easily converted to a graph [6]. Though shape-based methods directly operate on native 3D data, they are sensitive to geometric representation artifacts (e.g. non-manifold geometry, polygon soups, no interior), which are usually present in most of the datasets. View-based methods, however, can effectively avoid such artifacts. And this is one of the main reason why view-based methods are generally better than shape-based methods in object recognition tasks.

View-based methods represents a 3D object via multiple 2D images, which are captured from different viewpoints. MVCNN [8] is one of the very first view-based methods utilizing 2D convolutional neural networks. The network use a shared CNN to extract features from multiple 2D images of a 3D object and use a view-pooling layer to aggregate the view-level features into a global shape-level descriptor. This pipeline is then widely adopted in other view-based approaches [9]–[12]. In object recognition task, the main goal is to design discriminative shape descriptors which can be used to distinguish between objects from different categories. And a well designed shape descriptor must consider the intrinsic properties of the input data. Thus for the view-based deep learning methods, the fundamental challenge is to design a effective view aggregation module which fully considers the intrinsic properties of the multi-view representations. MVCNN [8] simply use a view-wise max-pooling to combine the view-level features. Nevertheless, only keeping the maximums leads to loss of visual information. In order to prevent the loss of visual details, it's intuitive to replace the max-pooling layer with the average-pooling layer. However, the average-pooling was proven to be less effective than the max-pooling in the experiments of [8]. We think this is due to the useful features being contaminated by a large amount of redundant features. Recently, a number of methods were presented to cope with such an issue. For example, [10], [11] used grouping techniques to exploit the similarity among views. [9] presented a harmonized bilinear pooling to consider patch-to-patch similarities. In this paper, we mainly focus on improving the effectiveness of the view aggregation module in the view-based deep learning pipeline. We propose a rank-pooling layer to deal with the shortcomings of the max-pooling layer and the average-pooling layer. The idea of the rank-pooling layer, though is quite intuitive, can be divided into two parts:

**This work was supported in part by the National Natural Science Foundation of China (61672241, 61602184, 61872151, and U1611461), in part by the Natural Science Foundation of Guangdong Province (2016A030308013 and 2017A030313376), in part by the Science and Technology Program of Guangzhou(201707010147 and 20180201005).

978-1-7281-3723-0/19/$31.00 ©2019 IEEE

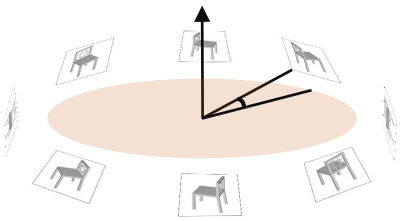

Fig. 2: M virtual cameras are placed evenly around the rotation axis, elevated by angle ϕ from the ground plane. The cameras are pointing to the centroid of the 3D shape when capturing images.

1) taking all the view-level features into consideration,
2) assigning weight to each view-level feature according to their importance.

The first part is to prevent information loss which is caused by max-pooling, and the second part is to prevent the useful information being contaminated which may happen when using average-pooling. Fig. 3 illustrates the workflow of the rank-pooling layer. The rank-pooling layer takes multiple view-level features as input and ranks them along the view axis. After that, the feature aggregation is conducted by combining the ranked view-level features with a set of learnable weights which are dynamically adjusted in an end-to-end manner. Comprehensive experiments in Section III demonstrate that the proposed rank-pooling layer, when being inserted into a standard convolutional neural network, achieves competitive results compared with the state-of-the-art methods.

The structure of the rest this paper is described as follow. Firstly, detailed discussion about our view aggregation module is given in Section II. After that, experiments and result analysis are provided in Section III. Finally, a summary of this paper is presented in Section IV.

II. Methods

The architecture of our network is illustrated in Fig. 1. The input of the network is a multi-view representation of a 3D shape, denoted as $\{I_i\}_{i=1}^m$. Firstly, each image I_i in the set is fed into a CNN to extract corresponding view based feature X_i. We use VGG11 with batch normalization from [18] as our base CNN architecture, which is mainly composed of eight 3×3 convolutional layers $conv_{1,\dots,8}$ and three fully connected layers $fc_{9,\dots,11}$. The view-level features $\{X_i\}_{i=1}^m$ are then passed through a designed view aggregation module to obtain a compact shape descriptor. The view aggregation module, as we can see, plays an important role in the whole architecture, because the performance of the network on shape recognition tasks highly depends on its output descriptor. In this paper, we proposed the rank pooling layer, which focuses on the differences between views, to improve the effectiveness of the view aggregation module. We will discussed its details in Section II-B.

A. The Multi-view Representation

The multi-view representation of a given 3D object (which is usually stored as a polygon mesh) is created by capturing images from different virtual cameras placed at different viewpoints. In a sense, viewpoints selection determines how well a multi-view representation can depict the original 3D shape. Here we use the camera setup illustrated in Fig. 2 to generate multi-view representations, because all the models we use to train our network satisfy the upright orientation assumption. M is set to 12 and ϕ is set to 30° by default.

B. View Aggregation via the Rank-Pooling Layer

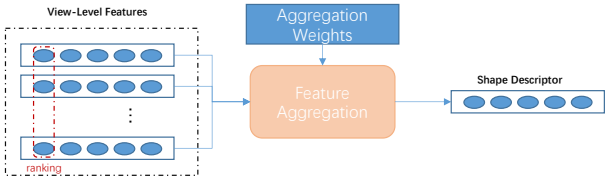

Fig. 3: The framework of the rank-pooling layer.

Given the view-level features $\{X_i\}_{i=1}^m$ of m multi-view images representing a 3D shape, the view-pooling layer aggregates them into a global feature vector which is then fed into a classifier to make decisions. Before diving into the detail of our proposed rank-pooling layer, let's firstly consider the view-wise max-pooling operation employed by MVCNN [8], which only keep the maximal element at each feature dimension over views and discard the others. Apparently, the intention of using max-pooling is to find out the most distinguishable view contributing to each feature dimension. However, in many cases, especially where objects to be classified are from a large number of possible categories, using just one view per feature dimension is far from enough. For example, although one can easily distinguish between bowls and dressers by only looking at the top, it's hard to tell the differences between bowls and cups since their top views are similar. A straight forward idea of taking more views into consideration is to simply replace max-pooling by average-pooling. But the experiments in [8] shows that average-pooling is even worse. Our conjecture on such a performance degradation is that average-pooling doesn't enable inter-view differences. Back to the earlier example, the top view of a bowl is more distinguishable than its side view, which means people can get more visual information from the top. When using average-pooling, each view is weighted the same, resulting in the most distinguishable views being contaminated by the others.

To address the issue of previous pooling scheme, we propose the rank-pooling(illustrated in Fig. 3). The idea of this pooling scheme is of two parts. One is to take more views into account, and the other is to dynamically weight the views according to their importances. The view-level features $\{X_i\}_{i=1}^m$ extracted by the previous layer can be regarded as a matrix $X \in \mathbb{R}^{m \times d}$, where d is the length of each feature vector. Given the matrix $X \in \mathbb{R}^{m \times d}$, the rank-pooling layer firstly sorts them along the view dimension by the magnitude of each element in a ascending manner. We denote the resulting feature set as $\{\hat{X}_i\}_{i=1}^m$ where $\hat{X}_i$ is the feature vector ranking at the position i. In fact, $\hat{X}_m$ is actually the final result of the max-pooling operation. But as we need to involve more views at each feature dimension, a corresponding weight w_i is assigned to each $\hat{X}_i$. So the output $F \in \mathbb{R}^d$ of the rank-pooling

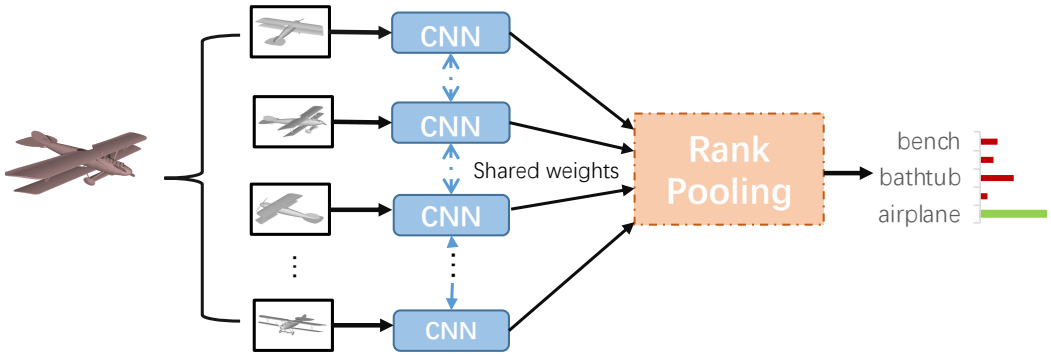

Fig. 1: A shared CNN is used to extract features from the multi-view images of a 3D shape, followed by a view aggregation module which processes the view-level features and output the final classification result.

layer can be written as the following:

$$F = \sum_{i=1}^{m} w_i \hat{X}_i \qquad \text{s.t. } w_i \geq 0, \sum_{i=1}^{m} w_i = 1. \tag{1}$$

Nevertheless, it's a non-trivial task to manually design a fixed weighted scheme. On the one hand, people don't usually know the exact data distribution of a given dataset. On the other hand, the weights that work well on a specific dataset may not work well on another dataset. For these reasons, the weights are preferred to be a set of learnable parameters which can be dynamically adjusted during the back-propagation pass of the training. However, directly using the learned parameters to weight the features will violate the restriction in (1) where the summation of all the weights equals to one and each weight is not less than 0. To meet our need, we calculate the weights $\mathbf{w} \in \mathbb{R}^{1\times m}$ by applying the softmax function to the learned parameter vector $\mathbf{p} = \langle p_1, p_2, ..., p_{m-1}, p_m \rangle \in \mathbb{R}^{1\times m}$:

$$\mathbf{w} = \langle w_1, w_2, ..., w_{m-1}, w_m \rangle = Softmax(\mathbf{p}), \tag{2}$$

$$w_i = Softmax(\mathbf{p})_i = \frac{e^{p_i}}{\sum_{k=1}^{m} e^{p_k}}. \tag{3}$$

Given the matrix $\hat{X}^{m\times d} \in \mathbb{R}$ corresponding to $\{\hat{X}_i\}_{i=1}^{m}$, equation (1) can be rewritten as the following:

$$F = Softmax(\mathbf{p}) \times \hat{X}. \tag{4}$$

For shape recognition, a fully connected layer is used to calculate the final category label after getting the final shape descriptor F.

III. Experiments

We evaluate our proposed view aggregation modules on two 3D benchmark datasets ModelNet10 and ModelNet40 from [1]. The ModelNet40 consists of 12,311 CAD models from 40 categories and the ModelNet10 contains 4,899 CAD models belonging to 10 categories. For fair comparison, we follow the training and test split of the dataset as in [1]. Transfer learning technique is applied to train our network, which fine-tunes the weights pre-trained on the ImageNet ILSVRC 2012 dataset [19] using SGD with momentum set to 0.9. The view-level features consumed by our view aggregation module are obtained from fc_{10} of VGG11.

A. 3D Shape Classification

Firstly, in order to find out how the multi-view representation influences the performances of our modules, we evaluate the modules using various multi-view representations under different camera setups, where different numbers of views and elevation angles are used. The results are shown in Table I. Just as expected, the performances drop when changing the camera setups with elevation angle from 30° to 0°. This phenomenon is reasonable because the multi-view representation with $\phi = 30°$ contains more visual information than that of the multi-view representation with $\phi = 0°$ under the upright orientation assumption. However, when it comes to the number of views, things are not as we expected. Although more views contain more information, our rank-pooling module performs better on 8-view representation than on 12-view representation. We think this is due to the aggregation weights being harder to adjust when the number of views increases.

Finally, we compare our methods to those existing 3D shape recognition approaches including 8 view-based methods, 4 pointset-based methods as well as 5 voxel-based methods. The experiments results are summarized in Table II. As we can see, voxel-based and pointset-based methods are not as good as view-based methods, except the VRN-Ensemble [4]. However, as written in [4], the result of VRN-Ensemble achieved by ensembling with a mix of predictions is not general. The main result of VRN [4] is 91.3% on ModelNet40, which is lower than both our rank-pooling module. Compared with MVCNN which uses max-pooling [8], our rank-pooling module improves more than 3% on ModelNet40. When put together with other methods using view aggregation techniques [9]–[11], our rank-pooling module still achieves comparable results. Noticed that [9] and [10] used additional feature modalities such as surface normals and depth values to further improve the classification accuracy. Using RGB-only data, the performance of [10] is inferior to ours. In short, our rank-pooling module is superior to most of the view-based methods and all of the pointset-based and voxel-based methods except the VRN-Ensemble, achieving comparable result.

IV. Conclusion

In this paper, we proposed an effective view-based approach for 3D shape recognition. Our approach is built upon a CNN based view feature extraction module and a rank-pooling

TABLE I: Comparison of classification accuracy on ModelNet40 under different camera setups

Camera Setups(M, ϕ)	8 views,		12 views,	
	30°	0°	30°	0°
Accuracy(%)	93.60%	91.90%	93.40%	89.50%

TABLE II: Comparison of classification accuracy on ModelNet10 and ModelNet40(%)

Method	ModelNet40	ModelNet10
MHBN(RGB + Depth) [9]	94.7%	95.0%
MHBN(RGB) [9]	94.1%	94.9%
Dominant(RGB + Depth + Surf) [10]	93.8%	-
Dominant(RGB) [10]	92.2%	-
GVCNN [11]	93.1%	
MVCNN-MultiRes [20]	91.4%	-
PANORAMA-NN [21]	90.7%	91.1%
Pairwise [12]	90.7%	92.8%
MVCNN [8]	90.1%	-
GIFT [22]	83.1%	92.35%
PointNet++ [17]	91.9%	-
KCNet [23]	91.0%	94.4%
ECC [6]	83.2%	90.0%
PointNet [16]	89.2%	-
VRN-Ensemble [4]	95.5%	97.1%
VRN [4]	91.3%	93.6%
FusionNet [24]	90.8%	93.11%
3D-GAN [5]	83.3%	91.0%
VoxNet [2]	83.0%	92%
3DShapeNets [1]	77%	83.5%
Ours	93.6%	92.73%

s

view aggregation module. The rank-pooling layer ranks view features according to their importances, and aggregates them via an attention mechanism. Shape features generated by the rank-pooling doesn't lose any visual information and can focus on more informative details. When applied to 3D shape classification, our method showed comparable performance with respect to state-of-the-art methods. In future, we would like to investigate the exploitation of other stochastic relations among views, as well as develop more advanced view aggregation techniques, to improve the performance of 3D shape recognition.

ACKNOWLEDGMENT

The authors would like to express the special thanks of gratitude to the chairs of VCIP as well as the reviewers.

REFERENCES

[1] 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*, 2015, pp. 1912–1920.

[2] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 922–928.

[3] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, "Orientation-boosted voxel nets for 3d object recognition," *arXiv preprint arXiv:1604.03351*, 2016.

[4] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," *arXiv preprint arXiv:1608.04236*, 2016.

[5] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *Advances in Neural Information Processing Systems*, 2016, pp. 82–90.

[6] M. Simonovsky and N. Komodakis, "Dynamic edgeconditioned filters in convolutional neural networks on graphs," in *Proc. CVPR*, 2017.

[7] J. Li, B. M. Chen, and G. H. Lee, "So-net: Self-organizing network for point cloud analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9397–9406.

[8] 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*, 2015, pp. 945–953.

[9] T. Yu, J. Meng, and J. Yuan, "Multi-view harmonized bilinear network for 3d object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 186–194.

[10] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3d object recognition." in *Proceedings of British Machine Vision Conference (BMVC)*, 2017.

[11] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "Gvcnn: Group-view convolutional neural networks for 3d shape recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[12] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016, pp. 3813–3822.

[13] A. Kanezaki, Y. Matsushita, and Y. Nishida, "Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," *arXiv preprint arXiv:1603.06208*, 2016.

[14] K. Sfikas, I. Pratikakis, and T. Theoharis, "Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval," *Computers & Graphics*, 2017.

[15] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, vol. 1, no. 2, p. 4, 2017.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5105–5114.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

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

[21] K. Sfikas, T. Theoharis, and I. Pratikakis, "Exploiting the panorama representation for convolutional neural network classification and retrieval," in *Eurographics Workshop on 3D Object Retrieval*, vol. 8. The Eurographics Association, 2017.

[22] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. Jan Latecki, "Gift: A real-time and scalable 3d shape search engine," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5023–5032.

[23] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 4, 2018.

[24] V. Hegde and R. Zadeh, "Fusionnet: 3d object classification using multiple data representations," *arXiv preprint arXiv:1607.05695*, 2016.