Learning Discriminative 3D Shape Representations by View Discerning Networks Supplementary Materials

Biao Leng, Cheng Zhang, Xiaocheng Zhou, Cheng Xu, Kai Xu*

I. The Accuracy of 3D Shape Classification on ModelNet Datasets

TABLE I
Accuracy of classification on ModelNet datasets

Method	ModelNet 40	ModelNet 10
SPH [1]	68.20%	%-
LFD [2]	75.50%	-%
ShapeNets [3]	77.00%	83.50%
PointNet [4]	86.20%	-%
PointNet++ [5]	91.90%	-%
MVCNN [6]	90.10%	-%
Pairwise [7]	90.70%	92.40%
CNN_MAX	88.62%	90.50%
CNN_AVE	89.00%	91.00%
VDN_Channel	90.37%	93.50%
VDN_Part	90.25%	93.00%

This section examines the classification accuracy of our methods. We firstly make the comparison with the State-of-the-Art methods on the original ModelNet 40, which is shown in Table I. Then we evaluate the classification accuracy of each category in ModelNet 40. The experiments and parameter settings here are the same as previous examples.

We make a comparison on the classification accuracy with the State-of-the-Art methods, including SPH, LFD, ShapeNets, PointNet, MVCNN, and Pariwise. We use the same training and testing split of ModelNet40 and ModelNet10 as in [3]. And then we averaged per-class accuracy to make a detailed comparison with other methods. Since our network is a view-based method, we mainly focus on its comparison with view-based state-of-the-art like MVCNN and pairwise. As we can see, pairwise achieves best classification performance around different methods. Compared with pairwise adopting depth images and greyscale images, our methods only used depth images as input. On ModelNet40, although the proposed view discerning network outperforms MVCNN in terms of retrieval task, the performance of MVCNN is better than VDN on classification. We think the reason is that MVCNN adopted additional post-processing that a one-vs-rest linear SVMs was trained to classify shapes using extracted image features. However, taking into account the goal of end-to-end training, our method got the classification results directly from the softmax layer in the network.

Fig. 1 shows the classification accuracy of all categories. As can be seen, most of the categories achieve high classification accuracy, while the accuracy of some categories may not be satisfying, like plant pots. This deficiency results from the limitation of our methods, which is presented in section Limitation and Failure Cases in the paper.

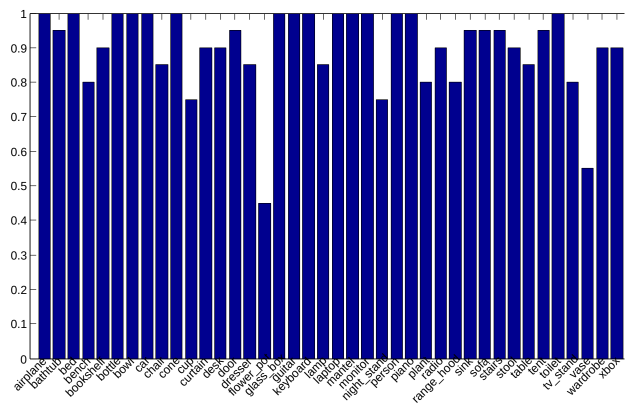


Fig. 1. Accuracy of classification on ModelNet 40

There is an interesting discovery that the improvement of our method on classification task is lower than the gain on the retrieval task, compared with baseline method CNN_AVE. Therefore, we visualize some results of classification and retrieval tasks in Fig. 2. As we can see, for the same query model, both our method and baseline method give the correct classification. However, the baseline method has some mistakes in later search results and our method retrieves objects more accurately.

II. The Processing of Generating the Occlusion and Background Clutter Datasets

To evaluate the robustness of our methods when encountering the noise in practical applications, we generate the noisy datasets with ModelNet 40, where models are influenced by object occlusion and background clutter. Fig. 3 shows the process of generating the object occlusion and background clutter datasets. For object occlusion, the original 3D shape is placed in the center and another irrelative 3D model as the occlusion is set beside it. Then we render the images from different angles for multiple views. The models as occlusion are randomly selected from ModelNet 40 and the size of them vary for training and testing phases. Specifically, we set the size of the occlusion 1.2 times the original one for the training phase and the scale of occlusion changes from 0.3 to 2.1 in the testing phase. For background clutter, we first render the

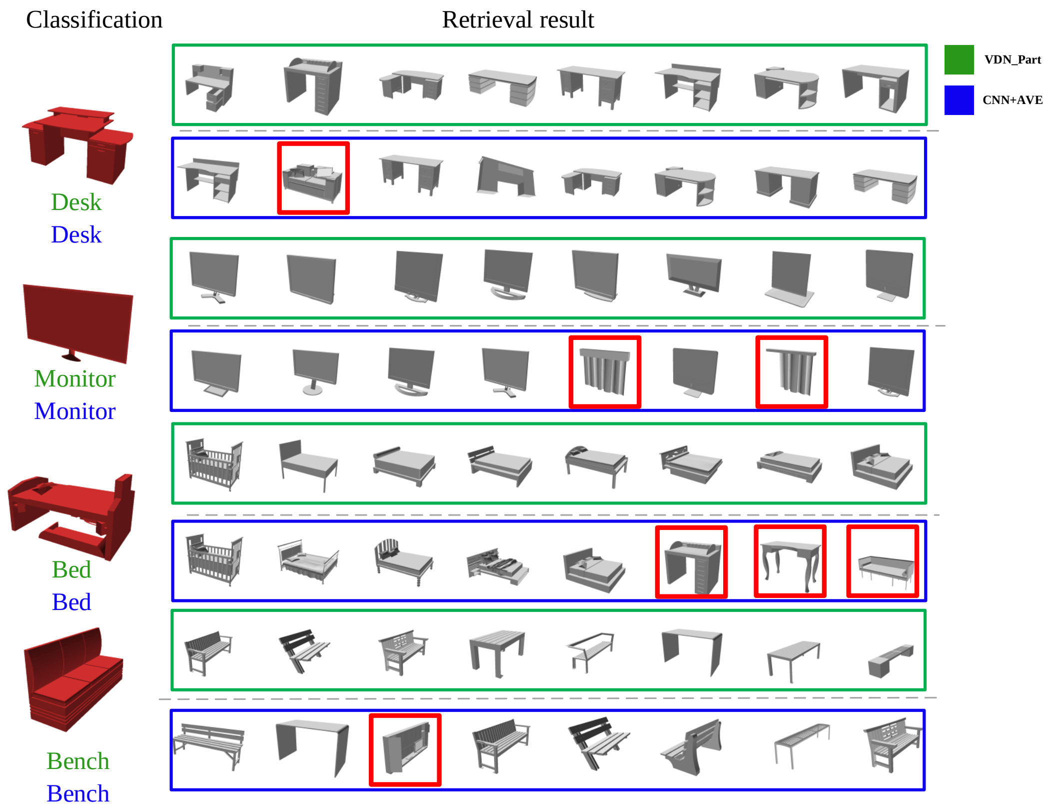


Fig. 2. Retrieval results and classification results of our proposed method(VDN_Part) and the baseline method(CNN_AVE). Green represents the retrieval result and classification result for each query model of VDN_Part. And blue represents the results from CNN_AVE. The red box denotes the wrong searched object. This figure is best viewed in color.

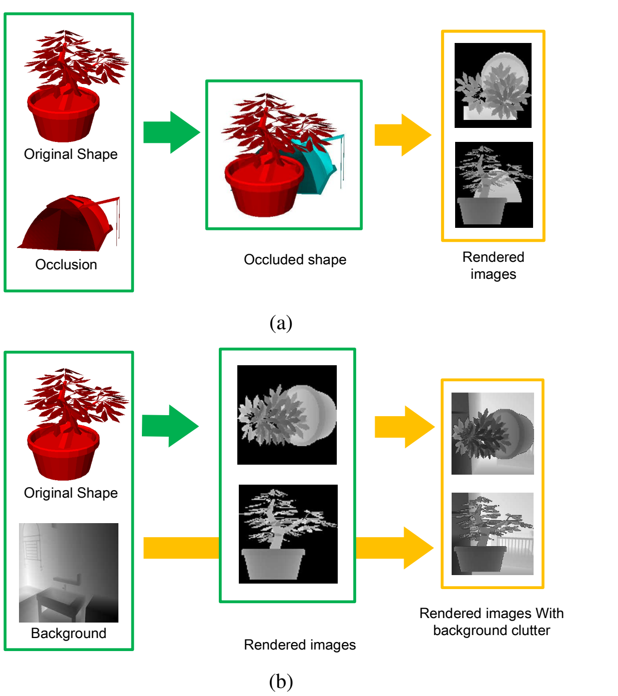


Fig. 3. Generating the noise of object occlusion and background clutter. (a) The occlusion is set next to the model and images of the concatenated 3D shape are rendered from different angles. (b) 2D images are rendered from original shape and concatenated with the background clutter.

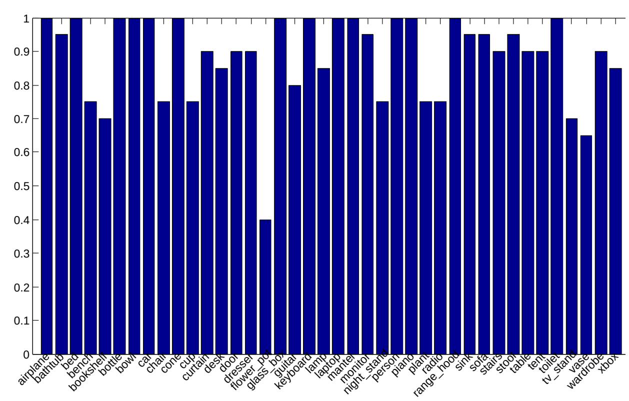
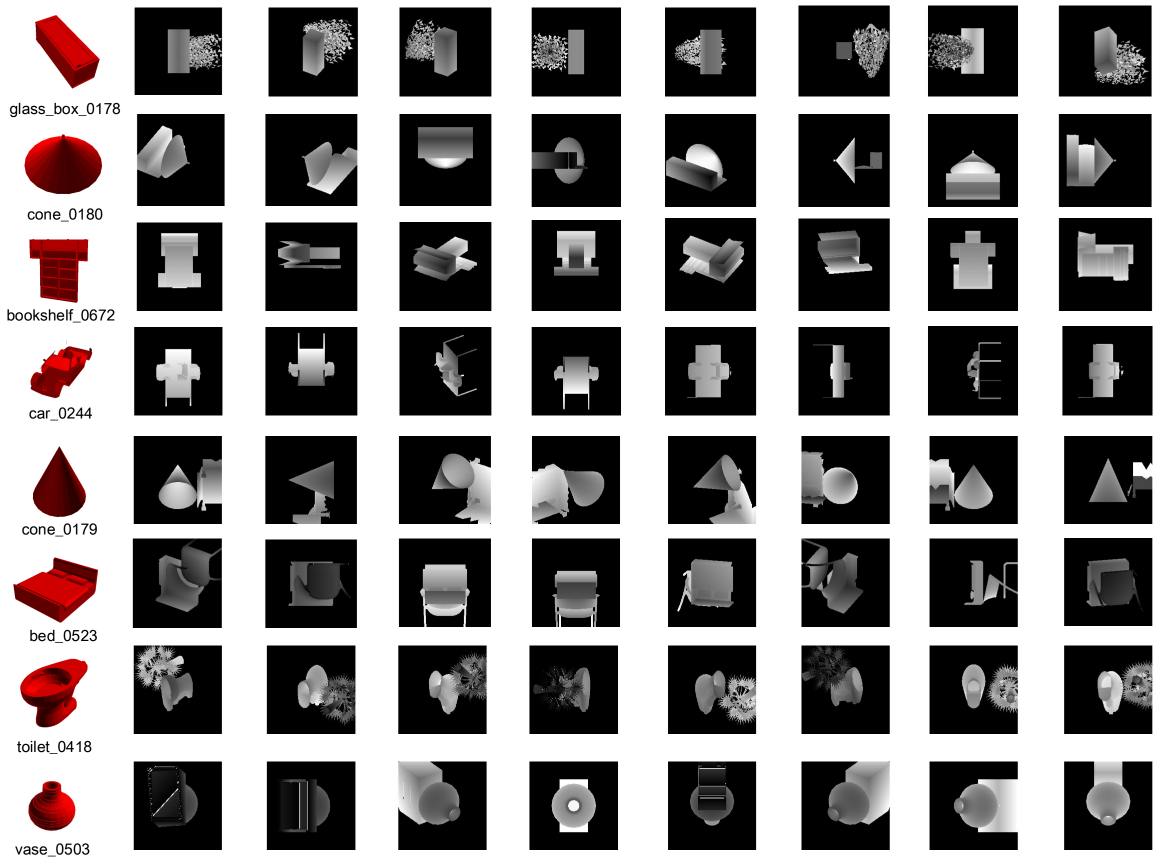
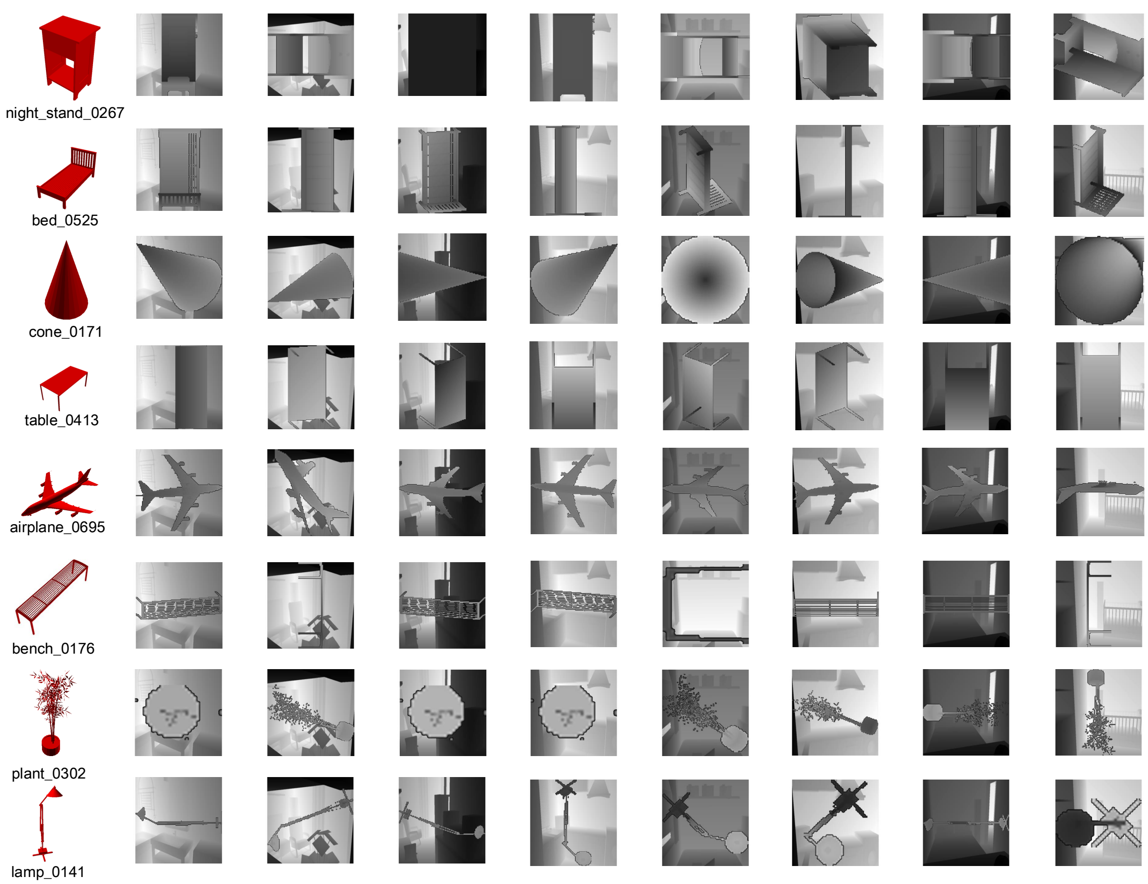


Fig. 4. Accuracy of classification on ModelNet 40 with occlusion

images from original 3D shapes. Then we randomly combine the 2D images with the 3D scene from SUNCG [8]. 112 3D scenes from SUNCG are used to generate the background clutter images. Fig. 4 presents the classification accuracy of all categories on ModelNet 40 with object occlusion and Fig. 5 shows the examples of noisy datasets. Compared with the performance on original dataset, the classification accuracy on occlusion dataset still achieves satisfying results, which proves the robustness of our method when dealing with the occlusion effect.



(a)



(b)

Fig. 5. Examples of datasets with object occlusion and background clutter. (a) Models with object occlusion (b) Models with background clutter.

III. Backward Propagation of the Loss Function

In the paper, we have introduced the loss function of View Discerning Network. Here, we will discuss the optimization of the shape feature $\mathbf{F}$ and the classification layer under the loss L. The loss function is fomulated as below:

$$L = \frac{1}{2M}\{\sum_{j=1}^{2M} L_{S_j} + \sum_{i=1}^{M}[s_i E_i + (1-s_i)max(\mathcal{M} - E_i, 0)]\} \tag{1}$$

where

$$E_i = \|\mathbf{N_{2i-1}} - \mathbf{N_{2i}}\|_2^2 \tag{2}$$

L_{S_j} refers to the Softmax loss value for the j-th pair of models. $\mathbf{N_{2i-1}}$ and $\mathbf{N_{2i}}$ are obtained via L2-normalization of the shape features $\mathbf{F_{2i-1}}$ and $\mathbf{F_{2i}}$, which compose a shape pair. s provides similarity information between them. If they are from the same category, s is set to 1, otherwise set to 0. $\mathcal{M}$ denotes the desired distance between shape features of different categories, which is manually adjusted based on specific cases.

Based on Equation 1, we can obtain the derivative of L with respect to $\mathbf{F}$:

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{F_{2i-1}}} = \widetilde{s}_i \cdot \dfrac{\partial E_i}{\partial \mathbf{F_{2i-1}}} + \dfrac{\partial L_{S_{2i-1}}}{\partial \mathbf{F_{2i-1}}} \\ \dfrac{\partial L}{\partial \mathbf{F_{2i}}} = \widetilde{s}_i \cdot \dfrac{\partial E_i}{\partial \mathbf{F_{2i}}} + \dfrac{\partial L_{S_{2i}}}{\partial \mathbf{F_{2i}}} \end{cases} \tag{3}$$

where

$$\widetilde{s}_i = \begin{cases} 2s_i - 1, & \mathcal{M} \geq E_i \\ s_i, & \mathcal{M} < E_i \end{cases} \tag{4}$$

According to Equation 2, we can derive the derivative of E with respect to $\mathbf{F}$ as below:

$$\begin{cases} \dfrac{\partial E_i}{\partial \mathbf{F_{2i-1}}} = 2(\mathbf{N_{2i-1}} - \mathbf{N_{2i}}) \cdot \dfrac{\partial \mathbf{N_{2i-1}}}{\partial \mathbf{F_{2i-1}}} \\ \dfrac{\partial E_i}{\partial \mathbf{F_{2i}}} = 2(\mathbf{N_{2i}} - \mathbf{N_{2i-1}}) \cdot \dfrac{\partial \mathbf{N_{2i}}}{\partial \mathbf{F_{2i}}} \end{cases} \tag{5}$$

Now a detailed version of Equation 3 can be derived by combining it with Equation 5:

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{F_{2i-1}}} = 2\widetilde{s}_i \cdot (\mathbf{N_{2i-1}} - \mathbf{N_{2i}}) \cdot \dfrac{\partial \mathbf{N_{2i-1}}}{\partial \mathbf{F_{2i-1}}} + \dfrac{\partial L_{S_{2i-1}}}{\partial \mathbf{F_{2i-1}}} \\ \dfrac{\partial L}{\partial \mathbf{F_{2i}}} = 2\widetilde{s}_i \cdot (\mathbf{N_{2i}} - \mathbf{N_{2i-1}}) \cdot \dfrac{\partial \mathbf{N_{2i}}}{\partial \mathbf{F_{2i}}} + \dfrac{\partial L_{S_{2i}}}{\partial \mathbf{F_{2i}}} \end{cases} \tag{6}$$

We also pay attention to the learning of the classification layer, which generates the classification vector out of the shape feature. The parameter vector of it is called $\mathbf{P_C}$. According to the chain rule of network back-propagation, the derivative of L with respect to P_C is represented as:

$$\frac{\partial L}{\partial \mathbf{P_C}} = \alpha \mathbf{P_C} + \sum_{j=0}^{2M} \frac{\partial L_{S_j}}{\partial \mathbf{P_C}} \tag{7}$$

where α is the weight decay coefficient.

IV. The Runtime Analysis of View Discerning Network

Compared with MVCNN, VDN introduces an additional score generation unit, leading to extra computational cost and runtime cost. In Table II, we give a comprehensive analysis of the time cost of our method. "CNN" includes feature extraction for views, "CNN_MAX" includes feature extraction and max-pooling aggregation and "VDN" includes feature extraction, score generation and weighted aggregation. Note that all methods use the same base network architecture and we use 10 views per 3D shape.

As the table shows, the least time-consuming method is CNN_MAX. Compared with CNN method, the aggregated feature in CNN_MAX leads to less computation than learning to extract each visual feature. The time cost of VDN is 1.15 times the time cost of CNN_MAX while processing a shape and this extra time cost mainly because of the generation of scores. It is shown that the score generation unit can bring an extra 15% ($\frac{112-97}{97}$) time cost compared to the original MVCNN.

TABLE II
Runtime comparison on ModelNet40

Method	test time (per shape)
CNN	103.3 ms
CNN_MAX	97.0 ms
VDN	112.1 ms

References

[1] M. Kazhdan, T. Funkhouser and S. Rusinkiewicz. *Rotation invariant spherical harmonic representation of 3d shape descriptors*, in *Symposium on geometry processing*, volume 6, pages 156-164, 2003.

[2] D.-Y. Chen, X.-P. Tian, Y.-T. Shen and M. Ouhyoung. *On visual similarity based 3d model retrieval*, in *Computer graphics forum*, volume 22, pages 223-232. Wiley Online Library, 2003.

[3] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang and J. Xiao, *3d shapenets: A deep representation for volumetric shapes*, in *CVPR*, pages 19121920, 2015.

[4] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo and Leonidas J. Guibas, *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*, in *CVPR*, 2017, pp.77-85

[5] Charles Ruizhongtai Qi, Li Yi, Hao Su and Leonidas J. Guibas, *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*, in *NIPS*, 2017, pp.5105-5114

[6] H. Su, S. Maji, E. Kalogerakis and E. Learned-Miller, *Multi-view Convolutional Neural Networks for 3D Shape Recognition*, in *CVPR*, 2015, pp. 945-953.

[7] Johns, E.; Leutenegger, S.; and Davison, A. J. *Pairwise decomposition of image sequences for active multi-view recognition* In *CVPR*, 2016, pp.3813-3822.

[8] Song, Shuran and Yu, Fisher and Zeng, Andy and Chang, Angel X and Savva, Manolis and Funkhouser, Thomas, *Semantic Scene Completion from a Single Depth Image*, in *CVPR*, 2017.