DEEP LEARNING FOR 3D SHAPE CLASSIFICATION FROM MULTIPLE DEPTH MAPS

Pietro Zanuttigh and Ludovico Minto

Department of Information Engineering, University of Padova, Italy

ABSTRACT

This paper proposes a novel approach for the classification of 3D shapes exploiting deep learning techniques. The proposed algorithm starts by constructing a set of depth maps by rendering the input 3D shape from different viewpoints. Then the depth maps are fed to a multi-branch Convolutional Neural Network. Each branch of the network takes in input one of the depth maps and produces a classification vector by using 5 convolutional layers of progressively reduced resolution. The various classification vectors are finally fed to a linear classifier that combines the outputs of the various branches and produces the final classification. Experimental results on the Princeton ModelNet database show how the proposed approach allows to obtain a high classification accuracy and outperforms several state-of-the-art approaches.

***Index Terms*—** 3D Shape Classification, Deep Learning, Convolutional Neural Networks, Depth Map

1. INTRODUCTION

Classification of 3D shapes is a long term research field but until recently there has been a limited interest on this task, specially if compared with the large efforts on the image classification problem. The recent widespread diffusion of consumer depth cameras has made the acquisition of 3D shapes a simple task available to everyone and this has largely increased the interest on the considered task. Furthermore the diffusion of advanced machine learning techniques, in particular Deep Learning techniques has allowed to obtain performances that were not possible with previous approaches.

This work proposes a novel deep learning architecture for the classification of 3D shapes exploiting a multi-branch Convolutional Neural Network (CNN). In order to use the 3D data into the CNN structure, first of all a set of different depth maps is built from the input shape. The depth maps are used as input for the CNN: each branch analyzes one of the depths by using 5 different layers of smaller and smaller resolution until a description vector is obtained. The rationale behind this strategy is to turn a pixel-wise classifier into an image-wise one by progressively reducing the resolution up to a single pixel representation where a single classification output is provided. We also propose a variant with the weights shared across two groups of branches, one for the side views and one for the top and bottom views, an approach never exploited before. Finally, the 6 vectors are concatenated into a single vector which is fed to a linear classifier in order to produce the shape classification.

We acknowledge NVIDIA for the donation of the GPU used for the training of the CNN. Thanks also to G. Pagnutti for some preliminary ideas.

2. RELATED WORKS

The retrieval and classification of 3D shapes has been the subject of a large amount of research works exploiting both global representation and local shape descriptors. An overview of the field can be found in review papers like [1, 2, 3] or by looking at the various editions of the SHREC 3D retrieval contest [4]. The widespread diffusion of deep learning approaches has allowed large improvements also in this field. Various recent works have exploited deep learning techniques and in particular Convolutional Neural Networks (CNN) for 3D object classification. In order to feed the 3D representation to a CNN it is firstly necessary to turn it into a representation suitable for the network structure. For this task two main strategies have been developed.

The first, used also in the proposed work is to render the 3D model from a set of viewpoints and then use the obtained silhouettes, images or depth maps as input to the deep learning framework. An example of this approach is the work of [5] that exploits a spherical parametrization to represent the mesh in a geometry image containing curvature information that is fed to the CNN. The work of [6] exploits the idea of representing the 3D object with a panoramic view and uses an ad-hoc CNN structure for this kind of images. In the work of [7] pairs of views of the object are used together with a second CNN for the selection of the best viewpoints. Another approach exploiting this strategy is [8] that extracts a set of color views from the 3D model and combines the information into a single shape descriptor using a CNN architecture.

The second strategy is instead based on the use of volumetric representations together with 3 dimensional CNNs applied on the voxel structure. The approach of [9] exploits a Convolutional Deep Belief Network to represent the input shapes as probability distributions on a 3D voxel grid. The approach of [10] jointly exploits Volumetric Convolutional Networks and Generative Adversarial Networks. The *PointNet* approach [11] uses density occupancy grids as input for a

CNN that performs the classification. Volumetric occupancy grids are used by [12] together with a 3D CNN.

3. EXTRACTION OF THE INPUT DATA

The proposed algorithm works in two stages: a pre-processing stage where a set of 6 depth maps is extracted from the 3D model followed by a multi-branch Convolutional Neural Network (CNN) that performs the classification. The pre-processing stage is described in this section while the proposed CNN architecture will be the subject of next section. First of all, the bounding box of the input 3D model is computed. Then the 3D model is rendered from 6 different viewpoints, each corresponding to one of the 6 faces of the bounding box. For each of the 6 views we extracted the depth information from the z-buffer, thus obtaining 6 different depth maps for each object (see Fig. 1). The output depths have a resolution of 320×240, that in our experiments proved to be a reasonable trade-off between the accuracy of the representation and the computational effort required to train the neural network. The 6 depth maps represent the input for the proposed classifier. This representation allows to capture a complete description of the 3D shape with a smaller amount of data compared to volumetric schemes used in some competing approaches [12, 10]. Furthermore, the depth map has a greater information content if compared with the silhouettes of the shape. Notice that, assuming that the object is lying on the ground, there are 4 side views, a bottom view and a top view (we will exploit this fact in the next section). This corresponds to the fact that the object can rotate around the vertical axis but it is lying on the ground, a reasonable assumption for many real world objects. In order to improve the robustness of the proposed approach with respect to rotations we also augmented the training dataset by creating randomly rotated copies of the 3D models, but in our experiments this did not lead to improvements in the accuracy (this feature has been disabled in the presented results, however it can be useful in more generic situations). Finally local contrast normalization is applied to each input depth map independently.

Fig. 1. Example of the 6 depth maps used for the analysis of a *chair* 3D model.

4. DEEP NETWORK ARCHITECTURE

The proposed classifier takes in input the 6 depth maps and gives in output a semantic label for each scene. We developed an ad-hoc Convolutional Neural Network (CNN) structure for this task and its architecture is shown in Fig. 2.

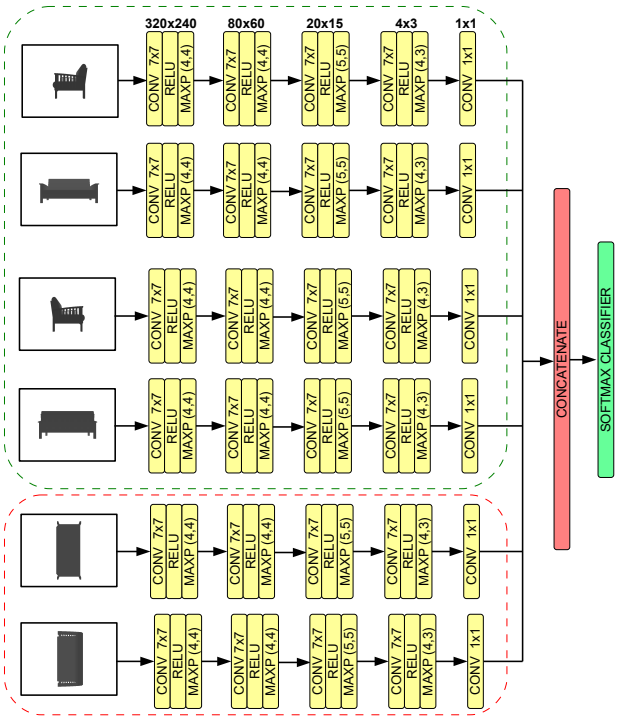


Fig. 2. Layout of the proposed Convolutional Neural Network

As common in many approaches [13, 14, 15] the network is made of two main parts, namely a set of convolutional layers followed by a linear classification stage.

In the first part there are 6 different branches, one for each of the input depth maps, that extract a local representation of the input by applying a sequence of convolutional layers. More in detail, each branch has 5 convolutional layers (CONV), each followed by a rectified linear unit activation function (RELU) and by a max-pooling layer (MAXP), except for the last layer. The first 4 convolutional layers have 64 filters all being 7x7 pixels wide. The max-pooling stages subsample the data by a factor of 4 in each dimension in the first 2 stages and of 5 in the third one. The last pooling operation uses a factor of 4 in the horizontal direction and a factor of 3 in the vertical one. The rationale of these values is to progressively reduce the resolution of the data until a single descriptor vector is obtained for each image. In this way, in the first layer the full resolution is exploited while at the end we obtain a single classification hypothesis for the whole image that is the required output. This approach also allows to limit the computational resources required for the training since in

the inner layers the resolution is strongly reduced. The last convolutional layer has a single pixel input (in fact there is no spatial convolution) and 128 filters. It does not have any activation function and it produces a 128 elements descriptor for the considered depth map.

For what concerns the weights of the convolutional filters we decided to explore two variants of the proposed approach. In the first one each branch has its own set of weights, differently from other approaches where weights are shared [13, 14]. This allows to exploit the fact that different views can capture different features of the 3D objects but also requires a larger number of parameters and thus a higher computational effort for the training,

In the second variant we exploited the previous observation that the top and bottom views typically captures different features than the side ones and we used a shared set of weights for the 4 side views and a different set for the top and bottom ones. This proved to be a good trade-off allowing to obtain results similar to the previous one with a reduced training time and is more coherent with the idea of obtaining rotation invariance at least around the vertical axis.

The 128-elements descriptor vectors of the 6 depth maps are then concatenated in a 768-elements vector and fed to a final softmax classifier with weight matrix of size $768 \times n_c$ and no bias, where n_c is the considered number of classes (10 or 40 in the results dataset).

The network is trained end-to-end to produce a labeling of each 3D shape by assigning it one out of the n_c different categories. To this aim, a multi-class cross-entropy loss function is minimized throughout the training process. We set a limit of 100 epochs, even if the optimal solution is typically reached earlier.

5. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed approach we used the Princeton ModelNet database [9]. Two different subsets of this database are typically used for the 3D shape classification task.

The *ModelNet10* subset contains 4899 3D models divided into 10 different categories. The training subset is made of 3991 samples while 908 are used for testing. The second column of Table 1 reports the results of our approach compared with some very recent state-of-the-art approaches on this subset (all the compared approaches are from the last two years). Our approach has a very high average accuracy of 91.6% outperforming most of the compared approaches except for [12] and [7] that obtain better results but with very limited improvements of 0.4% and 1.2% with respect to our method. Furthermore, if the weights are shared between the side and top views as proposed in Section 4 the performance remains roughly the same (91.5%) even if the number of parameters in the network is reduced by almost a factor of 3.

The confusion matrix for the proposed approach on the *ModelNet10* database is shown in Table 2 while some examples of correctly and wrongly classified objects are shown in Fig. 3 and Fig. 4 respectively. The average accuracy is very high on most classes. Some of them are almost perfectly recognized, e.g., the chair or monitor classes. On the other side some critical situations exist, for example the confusion between the *night stand* and the *dresser* classes, an expected issue since these two classes have very similar shapes (see the first example in Fig. 4). Another example of challenging recognition is the distinction between the *table* and *desk* classes. In several instances these classes share the basic structure of a flat surface with four legs supporting it, as in the example of Fig. 4. However most samples in these classes are correctly recognized even if some errors are present. Table 3 contains the confusion matrix for the case where the weights are shared. Results are similar with a small improvement in the most critical classes at the expenses of a slightly lower accuracy on the well recognized ones.

Table 1. Average accuracies on the *ModelNet10* and *ModelNet40* datasest for some state-of-the-art methods from the literature and for the proposed method.

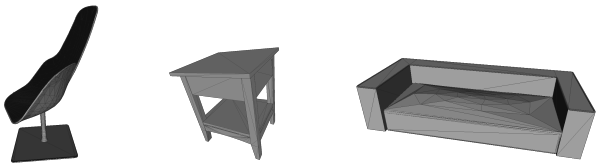
Approach	*ModelNet10*	*ModelNet40*
3DShapeNets [9]	83.5%	77.0%
DeepPano [6]	85.5%	77.6%
VoxNet [12]	92.0%	83.0%
Pairwise [7]	92.8%	90.7%
3D-GAN [10]	91.0%	83.3%
Geometry Image [5]	88.4%	83.9%
Proposed Method	91.6%	87.6%
Proposed Method (shared w.)	91.5%	87.8%

Table 2. Confusion matrix for the proposed approach (with independent weights) applied on the *ModelNet10* dataset.

Class	BA	BE	CH	DE	DR	MO	NS	SO	TA	TO	N.S.	Acc.
BA	46	2	0	0	0	0	0	1	1	0	50	92%
BE	1	97	1	0	0	0	0	1	0	0	100	97%
CH	0	0	100	0	0	0	0	0	0	0	100	100%
DE	0	1	1	73	0	0	1	2	8	0	86	85%
DR	0	1	0	1	69	1	14	0	0	0	86	80%
MO	0	0	0	0	0	100	0	0	0	0	100	100%
NS	0	0	1	1	10	0	67	0	7	0	86	78%
SO	0	1	0	0	1	1	0	97	0	0	100	97%
TA	0	0	0	16	0	0	1	0	83	0	100	83%
TO	0	0	0	0	0	0	0	0	0	100	100	100%

BA:Bathub, BE:Bed, CH:Chair, DE:Desk, DR:Dresser, MO:Monitor, NS:Night stand, SO:Sofa, TA:Table, TO:Toilet, N.S.: Number of samples in the class, Acc.:Average accuracy for the class

The larger *ModelNet40* subset contains 12311 models



Chair 893 Night stand 470 Sofa 692

Fig. 3. Examples of objects correctly recognized by the proposed approach.

Night stand 221 (*Dresser*) Table 470 (*Desk*) Dresser 241 (*Monitor*)

Fig. 4. Examples of objects wrongly recognized by the proposed approach (the last row reports the recognition from the proposed approach).

divided in 40 different classes. In this case 9843 samples are used for training and the test set is made of 2468 models. Results on this dataset are in the last column of Table 1. As expected the average accuracy on this more challenging dataset is lower, i.e., 87.6% and 87.8% with independent and shared weights respectively. However the accuracy remains very good with a drop of just around 5%, even if this time there are 4 times more classes. Furthermore, as in the previous case, the two versions of the proposed approach have very similar performance. If compared with the competing approaches, the results in this case are even better and only the approach of [7] is able to outperform the proposed one. Notice that the approach of [7] uses a quite complex strategy for the selection of the rendering viewpoints instead of the simpler approach in this work, that is probably the reason for the performance gap. The average accuracy for each single class is shown in Table 4: it is high on most classes for both versions of the approach, however for a few of them (e.g., the *flower pot* and *radio* classes) the results are low. These correspond to classes with a very limited amount of training samples and a large variability between the samples for which the algorithm is not able to properly learn the structure.

Finally concerning the training time, it is about 8 hours on the *ModelNet10* dataset and 22 hours on the larger *ModelNet40* subset (the employed system has an i7-970 CPU at 3,2Ghz and an NVIDIA Tesla K40 GPU).

6. CONCLUSIONS

In this paper we proposed a deep network architecture for 3D objects recognition. We used different branches to capture multiple depth maps extracted from the object and we progressively analyzed the data at smaller and smaller resolution until a single classification output is produced for each view. A final classification layer combines the outputs and produces the classification. The proposed approach is fast and requires a relatively small training effort, specially if the proposed weights sharing approach is employed, however it outperforms several recent state-of-the-art approaches.

Future work will focus on achieving a better invariance to the object pose by using a more advanced strategy for the selection of the input views. We will also consider the combined usage of different clues including color information and surface orientation properties.

Table 3. Confusion matrix for the proposed approach with the CNN weights shared across side and top/bottom views applied on the *ModelNet10* dataset.

Class	BA	BE	CH	DE	DR	MO	NS	SO	TA	TO	N.S.	Acc.
BA	47	1	0	0	0	0	0	1	1	0	50	94%
BE	3	97	0	0	0	0	0	0	0	0	100	97%
CH	0	1	99	0	0	0	0	0	0	0	100	99%
DE	0	0	0	67	1	1	4	1	12	0	86	78%
DR	0	0	0	0	76	1	9	0	0	0	86	88%
MO	0	1	0	0	0	99	0	0	0	0	100	99%
NS	0	0	0	1	16	0	64	0	5	0	86	74%
SO	0	1	0	1	0	0	0	98	0	0	100	98%
TA	0	0	0	13	0	0	1	1	85	0	100	85%
TO	1	0	0	0	0	0	0	0	0	99	100	99%

Table 4. Accuracy of the proposed approach on the various classes of the *ModelNet40* dataset with independent weights (IW, *second column*) and shared weights (SW, *third column*). Notice that the mean accuracy is not the average of these values since the number of models in the various classes is different.

Class	Acc. (IW)	Acc. (SW)	Class	Acc. (IW)	Acc. (SW)	Class	Acc. (IW)	Acc. (SW)
airplane	100%	100%	dresser	81%	81%	radio	50%	50%
bathtub	94%	84%	fl. pot	30%	15%	r.hood	88%	92%
bed	98%	96%	gl. box	96%	97%	sink	90%	80%
bench	80%	75%	guitar	95%	94%	sofa	98%	97%
booksh.	96%	95%	keyb.	100%	100%	stairs	70%	70%
bottle	95%	95%	lamp	80%	85%	stool	75%	65%
bowl	85%	90%	laptop	100%	95%	table	74%	79%
car	99%	99%	mantel	90%	92%	tent	95%	95%
chair	95%	94%	monitor	98%	96%	toilet	99%	98%
cone	100%	95%	night st.	70%	78%	tv st.	72%	80%
cup	70%	45%	person	80%	100%	vase	77%	81%
curtain	80%	80%	piano	84%	79%	wardr.	80%	70%
desk	80%	84%	plant	79%	79%	xbox	55%	55%
door	90%	100%						

7. REFERENCES

[1] Johan W.H. Tangelder and Remco C. Veltkamp, "A survey of content based 3d shape retrieval methods," in *Proceedings of IEEE Int. Conference on Shape Modeling Applications*, 2004, pp. 145–156.

[2] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan, "3d object recognition in cluttered scenes with local surface features: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2270–2287, 2014.

[3] Bo Li, Yijuan Lu, Chunyuan Li, Afzal Godil, Tobias Schreck, Masaki Aono, Martin Burtscher, Qiang Chen, Nihad Karim Chowdhury, Bin Fang, et al., "A comparison of 3d shape retrieval methods based on a large-scale benchmark supporting multimodal queries," *Computer Vision and Image Understanding*, vol. 131, pp. 1–27, 2015.

[4] "Shrec contest," http://www.shrec.net, Accessed: 5-2-2017.

[5] Ayan Sinha, Jing Bai, and Karthik Ramani, "Deep learning 3d shape surfaces using geometry images," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 223–240.

[6] Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai, "Deeppano: Deep panoramic representation for 3-d shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

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

[8] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015, pp. 945–953.

[9] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1912–1920.

[10] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh 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.

[11] Alberto Garcia-Garcia, Francisco Gomez-Donoso, Jose Garcia-Rodriguez, S Orts-Escolano, M Cazorla, and J Azorin-Lopez, "Pointnet: A 3d convolutional neural network for real-time object class recognition," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 1578–1584.

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

[13] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[14] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun, "Indoor semantic segmentation using depth information," in *International Conference on Learning Representations*, 2013.

[15] L. Minto, G. Pagnutti, and P. Zanuttigh, "Scene segmentation driven by deep learning and surface fitting," in *Geometry Meets Deep Learning ECCV Workshop*, 2016.