3D ResNets for 3D object classification

Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris

Information Technologies Institute, Centre for Research and Technology Hellas, Thermi 57001, Greece {ioananas, ehatzi, nikolopo, ikom}@iti.gr
http://mklab.iti.gr/

Abstract. During the last few years, deeper and deeper networks have been constantly proposed for addressing computer vision tasks. Residual Networks (ResNets) are the latest advancement in the field of deep learning that led to remarkable results in several image recognition and detection tasks. In this work, we modify two variants of the original ResNets, i.e. Wide Residual Networks (WRNs) and Residual of Residual Networks (RoRs), to work on 3D data and investigate for the first time, to our knowledge, their performance in the task of 3D object classification. We use a dataset containing volumetric representations of 3D models so as to fully exploit the underlying 3D information and present evidence that '3D ResNets' constitute a valuable tool for classifying objects on 3D data as well.

Keywords: 3D object classification · 3D object recognition · Deep Learning · Residual Networks.

1 Introduction

During the last few years, Deep Neural Networks (DNNs) have achieved state-of-the-art performance in almost every computer vision task. Initially, they were successfully adopted to applications such as speech recognition, object tracking and image classification, but today they are also used to tackle more complicated problems, e.g. video classification, 3D segmentation and 3D object recognition. Convolutional Neural Networks (CNNs), in particular, have shown excellent performance in scenarios involving large datasets. A detailed review on CNNs and their various applications can be found in [4]. As expected, CNNs' outstanding performance later attracted the attention of researchers working towards 3D data analysis and understanding as well.

Experimental results indicate that deeper networks provide more representational power and higher accuracy. One of the latest trends in designing efficient deep networks is adding residual connections. Residual Networks (ResNets) were initially introduced in [6] and later extended in [7] achieving remarkable performance on the tasks of image classification, segmentation, object detection and localization. ResNets address one of the biggest DNNs' challenges, i.e. exploding/vanishing gradients, by adding shortcut (or skip) connections to the network

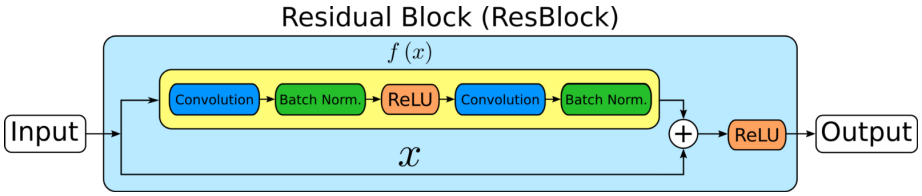

Fig. 1. Original Residual Block (image from [9])

that allow to better update the weights in the early layers of a deep network. This development allowed the training of very deep networks (up to 1K layers [7]) that led to performances even beyond the human-level ones [5]. The basic residual block is shown in Fig.1. As it can be seen, input x passes through two stacked weight layers (i.e. convolutional layers) and the output is added to the initial input which skips the stacked layers through the employed identity function. In the original version of ResNets, after each convolutional layer, Batch Normalization (BN) [11] and the ReLU activation function were applied. In the improved version of ResNets (Pre-ResNets), though, it was shown that pre-activation, i.e. applying BN and ReLU before the convolution layer, led to better results.

Concurrently with ResNets, Highway Networks [20] were proposed employing shortcut connections as well, but with gating functions whose weights needed to be learned. Recently, several variations of the original ResNets have also been proposed. In [9], a novel training algorithm that allows training deep residual networks with Stochastic Depth (SD) was introduced. The authors explored the scenario of randomly removing layers during the training phase while exploiting full network depth at test time. Experimental results showed that stochastic depth can lead to reduced training time and test error. "ResNet in ResNet" (RiR) [22] presented an extension of the standard resnet blocks by adding more convolutional layers. The new RiR block has two stacked layers each of which is composed of two parallel streams, a residual and a non-residual one. Improved results were reported on CIFAR-10 and CIFAR-100 datasets. The authors of Wide Residual Networks (WRNs) [25] proposed the widening of ResNet blocks by adding more feature planes in the convolutional layers and argued that 'wide' networks are faster to train and perform better than 'thin' ResNet models with approximately the same number of parameters. Residual Networks of Residual Networks (RoRs) [26] from the other side is a novel architecture that introduces level-wise shortcut connections that can also be incorporated to other residual networks for increasing their performance. RoRs achieved state-of-the-art results with the most popular image datasets used for classification. Long Short-Term Memory (LSTM) networks are variants of Recurrent Neural Networks (RNNs) proposed for tackling the problem of vanishing gradients in recurrent networks. Interestingly, the authors of [16] proposed an architecture, referred to as Convolutional Residual Memory Networks (CRMNs), where a LSTM is placed on top of a ResNet leading to promising results.

Despite their success though, residual networks have not been tested yet in tasks utilizing 3D data. In this work, we modify two recently proposed variations of residual networks, namely (1) Wide Residual Networks (WRNs) [25] and (2) Residual of Residual Networks (RoRs) i.e. Multilevel Residual Networks [26], and use them to perform 3D object classification. We test the adapted architectures on one of the most popular 3D datasets, i.e. Princeton's ModelNet [24] consisting of 3D CAD models from common object categories, and present comparable experimental results with the state-of-the-art.

The remainder of the paper is organized as follows. Section 2 briefly reviews the DNN-based state-of-the-art works for 3D object classification. Section 3 presents the residual architectures studied in this work, while Section 4 describes all experimental details and results. Finally, Section 5 discusses conclusions and future work.

2 Related Work

Due to increased availability of 3D data, a need for efficient and reliable 3D object recognition and classification methods has emerged. The popular DNNs are primarily designed to work with 1D and/or 2D data, hence their adaptation to the 3D case is not trivial. A review on how 3D data can be employed in DNNs can be found in [10].

Towards 3D object classification, several works addressing the task using a deep architecture are already available. One of the first approaches is 3D ShapeNets [24], i.e. a Convolutional Deep Belief Network (CDBN) with five layers accepting as input binary 3D voxel grids. Along with the proposed network, the authors of this work released a large-scale 3D dataset with CAD models from 662 unique categories, named *ModelNet*, that is used in the experimental evaluation of almost every related method ever since. A voxelized representation of the 3D data is also used in [15]. A CNN with two convolutional, one pooling and one fully-connected layer is employed in this work leading to better classification results compared to 3D ShapeNets. Also working on the voxelized 3D point cloud, in [18], the authors propose a convolutional network (ORION) that not only produces the labels of the 3D objects, but also their pose. The authors of [14] proposed the Kd-Nets, working directly on the unstructured point clouds without requiring that the point clouds are voxelized. This is accomplished since there are no convolutional layers in their architecture and as a result they avoid any problems that might occur during the voxelization due to poor scaling.

Approaches where multiple views of the 3D objects are provided to the network can be found in [21, 12]. Multi-View CNN (MVCNN) [21] learns to combine any number of input views of an object without any particular order through a view pooling layer. Setups with 12 and 80 views were tested increasing the classification accuracy significantly compared to other DNNs like [24]. Qi et al. [17] managed to introduce improvements to MVCNN's performance by using enhanced data augmentation and multi-resolution 3D filtering in order to exploit information from multiple scales. Multiple views organized in pairs were

used in [12]. The authors employed a known CNN, that is VGG-M [2], and concatenated the outputs of the convolutional layers from the two images before providing them to the first fully-connected layer. The introduced model surpassed the performance of voxel-based 3D ShapeNets [24] and the MVCNN approach of Su et al. [21] on the ModelNet dataset.

Recently, ensemble architectures have become popular. A work that attempts to combine the advantages of different modalities of the 3D models can be found in [8]. Two volumetric neural networks were combined with a multi-view network after the final fully-connected layer. A linear combination of class scores was then taken with the predicted class being the one with the highest score. An ensemble of 6 volumetric models was proposed in [1] achieving the current state-of-the-art classification accuracy on both ModelNet10 (i.e. 97.14%) and ModelNet40 (i.e. 95.54%) datasets. The final result was computed by summing the predictions from all 6 models. The proposed architecture led to excellent performance, however, is significantly more complex compared to most existing networks from the relevant literature requiring 6 days of training on a Titan X.

Despite the existing significant works on 3D object classification, computational cost is still a bottleneck, especially when working on pure 3D representations. Networks including sophisticated modules or ensembles of large topologies require increased training time and hardware resources that are not always available. In this work, we extend two of the most recent variants of residual networks in 2D-image classification, adapt them to the 3D domain keeping complexity in mind and investigate the efficiency of these '3D ResNets' on classifying volumetric 3D shapes.

3 3D Classification with Residual Networks

The authors of [25] have recently investigated several architectures of ResNet blocks and ended up proposing 'widening' by adding more feature planes in the convolutional layers. More specifically, WRNs consist of an initial convolutional layer followed by 3 groups of residual blocks. Additionally, an average pooling layer and a classifier completes the architecture, while dropout [19] was used for regularization. Experimental evaluation showed that widening boosts the performance compared to that of 'thin' ResNet models with approximately the same number of parameters and at the same time, accelerates training mostly due to the strong parallelization that can be applied in the convolutional layers.

In [26], level-wise shortcut connections were introduced to enhance the performance of ResNets. 'Residual Networks of Residual Networks' (RoRs) is a novel architecture with 3 shortcut levels (i.e. root, middle and final level) that allow information to flow directly from the upper layers to lower layers. Except for their original RoR architecture, the authors also incorporated the RoR concept to other residual networks, in particular Pre-ResNets and WRNs (denoted as Pre-RoR-3 and RoR-3-WRN respectively in [26]). Extensive experiments on the most popular image datasets used for classification indicated that RoRs can improve performance without bringing additional computational cost.

group	output size	[3D filter size, #filters]	
conv1	32x32x32	$[3 \times 3 \times 3,\ 16]$	
conv2	32x32x32	$[3 \times 3 \times 3,\ 16 \times k]$ $[3 \times 3 \times 3,\ 16 \times k]$	$\times N$
conv3	16x16x16	$[3 \times 3 \times 3,\ 32 \times k]$ $[3 \times 3 \times 3,\ 32 \times k]$	$\times N$
conv4	8x8x8	$[3 \times 3 \times 3,\ 64 \times k]$ $[3 \times 3 \times 3,\ 64 \times k]$	$\times N$
avg-pool	1x1x1	$[8 \times 8 \times 8]$	

Table 1. Structure of adapted WRNs for 3D object classification

Network	#params	Train Accuracy	Test Accuracy
WRN-16-2-modified	∼0.5M	98.70%	92.18%
WRN-22-2-modified	∼0.7M	99.57%	92.95%
Pre-RoR (N=2, k=1)	∼0.5M	99.8%	92.84%
RoR-WRN-16-2	∼2M	99.8%	94.00%

Table 2. Classification Results on ModelNet10 using Wide Residual Networks & Residual of Residual Networks

In this paper, our goal is to study the performance of residual networks on the task of 3D object classification. Towards this direction, we explored several variations of the original ResNets and trained a variety of models with different network and training parameters in order to get insights and identify best strategies. We tested networks of varying depth and width and explored suitable values for the learning rate, dropout, weight decay and activation functions. Except from the classification accuracy, the computational cost was also taken into account during our experimentation. We focused on networks with a relatively small number of parameters (up to 2.5 M) requiring a reasonable time to train.

Starting from *Wide Residual Networks*, we initially explored different values for the width (denoted with k) and the number of convolutional layers denoted with n. The depth of the network denoted with N is computed as $N = (n - 4)/6$. Due to memory limitations, we were able to train networks with k=2, i.e. networks that are two times wider than the original ResNets. With respect to the number of convolutional layers, we tested values between 10 and 22, therefore N was in the range [1...3]. In addition, the notation WRN-n-k is used to describe a wide residual network with n convolutional layers and width k. The adapted WRN structure incorporates 3D convolutions and is depicted in Table 1. Regarding multilevel residual networks, we tested in our experiments Pre-RoR and RoR-WRN with 16 convolutional layers, i.e. N=2, k=1 (for Pre-RoR) and k=2 (for WRNs), on ModelNet10. The multilevel structure is demonstrated in Figure 2.

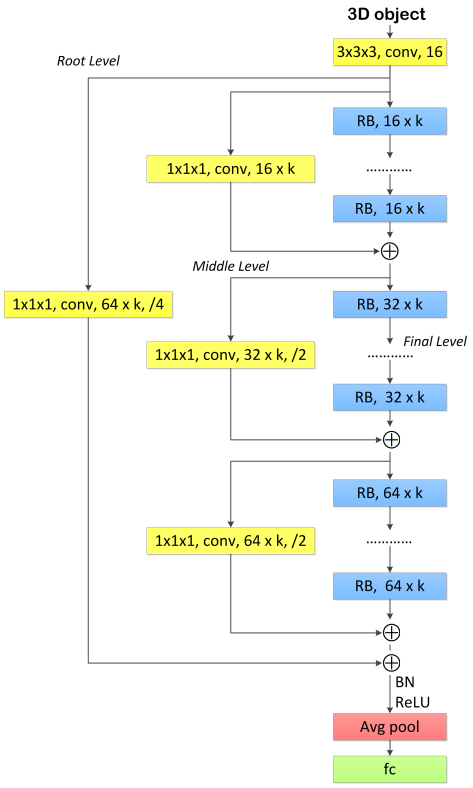

Fig. 2. Adapted Pre-RoR-3 (if k=1) and RoR-3-WRN (if k>1) architectures

4 Experimental Results

4.1 Dataset and implementation

ModelNet is a large 3D dataset containing more than 120K CAD models of objects from 662 categories. The dataset was released in 2015, and thereafter its two publicly available subsets, i.e. ModelNet10 and ModelNet40, are commonly used in works related to 3D object recognition and classification. To perform our experimental evaluation, we employ ModelNet10 that consists of 4899 models (3991 for training and 908 for testing) each manually aligned by the authors of the dataset. Binary voxelized versions of the 3D models are provided to our network. The resolution of the occupancy grid affects the classification accuracy, since it determines in which extent the 3D object's details will be apparent, as depicted in Fig.3. Obviously, a larger volume size leads to a better representation but also to an increased computational cost, hence a compromise needs to be made. In this work, the employed grid size is 32x32x32. As a pre-processing step, the voxels were transformed from {0, 1} to {-1, 1}. In addition, the dataset is augmented by 12 copies (i.e. rotations around the z axis) of each model. Inspired

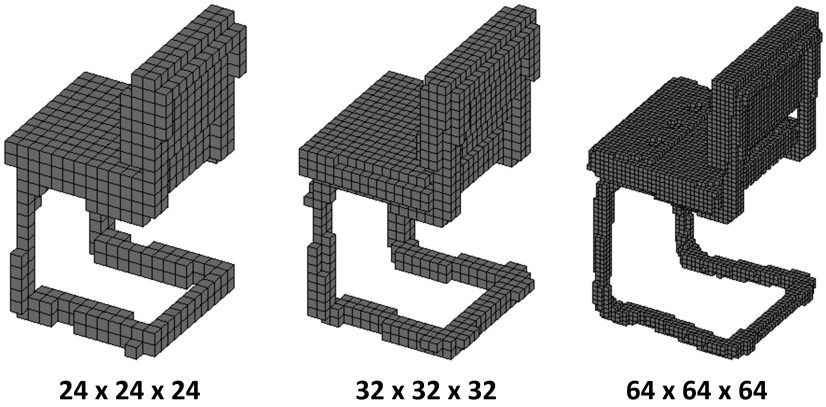

Fig. 3. 3D object from ModelNet voxelized in 3 different resolutions

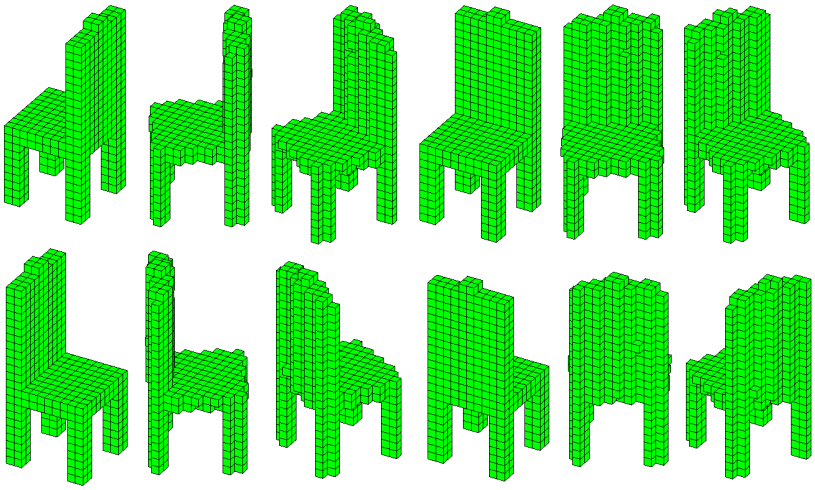

Fig. 4. 12 voxelized rotations of a 'chair' train sample from ModelNet10

by [15], one randomly mirrored and shifted instance of each object is also added in the dataset. An indicative 3D object and its 12 voxelized rotations are depicted in Fig.4.

All of our experiments were conducted on a Linux machine with 128GB RAM and a NVIDIA GeForce GTX 1070 GPU. The deep learning framework that was used was Keras [3] running on top of Theano [23].

4.2 Training

During training, we split the original training set randomly into a 'train' set, containing 75% of the 3D models, and a 'validation' set, containing the re-

maining 25% of models. The tested networks were trained from scratch using Adam optimizer [13] for fast convergence. We used fixed learning rates, such as 0.001 or 0.0001, since larger values reduced the performance. Categorical cross-entropy was used as the objective. All convolutional layers were initialized with the method of [5]. During training, every copy of a 3D model was considered as a separate train sample. At inference time, the predictions of all copies of a 3D model were summed up in order to make the final label assignment to it, i.e. pick the argmax on the sum.

4.3 Results on ModelNet10

Our initial experimentation with large wide networks, e.g. WRN-22-2 containing approximately 3.2M parameters, led to relatively low performance ($\sim$ 90%) in comparison to the state-of-the-art (97.1% [1]). Aiming to keep the computational cost as low as possible, we changed the structure of WRNs by removing the final group of convolutions, i.e. *conv4*. Hence, WRN-22-2 in our setting actually contains 15 convolutional layers and not 22 as the original WRN would have. We denote these networks as *WRN-n-k-modified*. Additionally, inspired by works like [15], we investigated using Leaky ReLU as the activation function in the trained networks instead of the original ReLU and found this to lead to a slight boost of approximately 0.5% in the classification accuracy. In these experiments, a dropout keep rate of 0.7 was used, while batch size was set to 32. Moreover, L2 regularization to the weights by a factor of 0.0001 was applied. Some of the results we obtained with WRNs after training for 50 epochs are provided in Table 2. As shown, an accuracy of over 92% can be yielded by a 'wide' network of less than 500K parameters. In contrast, VoxNet [15], for example, achieves the same accuracy with a network containing twice the parameters. By adding more (convolutional) layers leading to a network of approximately 700K parameters, a slight improvement in performance is observed (0.77%).

For training Pre-RoRs, no dropout or regularization of the weights was applied. Additionally, ReLU was used as the activation function as originally proposed by the authors. The classification results for ModelNet10 are included in Table 2. It can be seen that a Residual of Residual Network of approximately 500K parameters leads to better performance in comparison to a Wide network with the same number of parameters. In addition, a 'wide' Residual of Residual Network containing around 2M parameters achieves an accuracy of 94%.

In Table 3, recent classification results on ModelNet10 from relevant works are reported. As it can be seen, the state-of-the-art performance on this dataset is 97.1% achieved with an ensemble of 6 networks, though, containing 90M parameters. The next best performing networks have an accuracy in the range of 93.3%-94% achieved from networks containing several million parameters. Our best model, i.e. RoR-WRN-16-2 with only 2M parameters, after approximately 18 hours of training achieves a classification accuracy equal to the best performance reported so far from a single model architecture. In contrast, the equally performing Kd-Net with depth 15 (94%), requires 5 days to train on the faster Titan GPU, while its slimmer version (depth 10) performing 93.3% requires 16

Model	Type	# params	ModelNet10
VoxNet [15]	single	0.92M	92
FusionNet [8]	ensemble	118M	93.1
VRN Single [1]	single	18M	93.6
ORION [18]	single	4M	93.9
Kd-Net (depth=10) [14]	single	-	93.3
Kd-Net (depth=15) [14]	single	-	94
VRN Ensemble [1]	ensemble	90M	97.1
RoR-WRN-16-2	single	2M	**94**

- "not reported in the original paper"

Table 3. Classification Accuracy (%) on ModelNet10 of our best performing model in comparison with other models from the literature

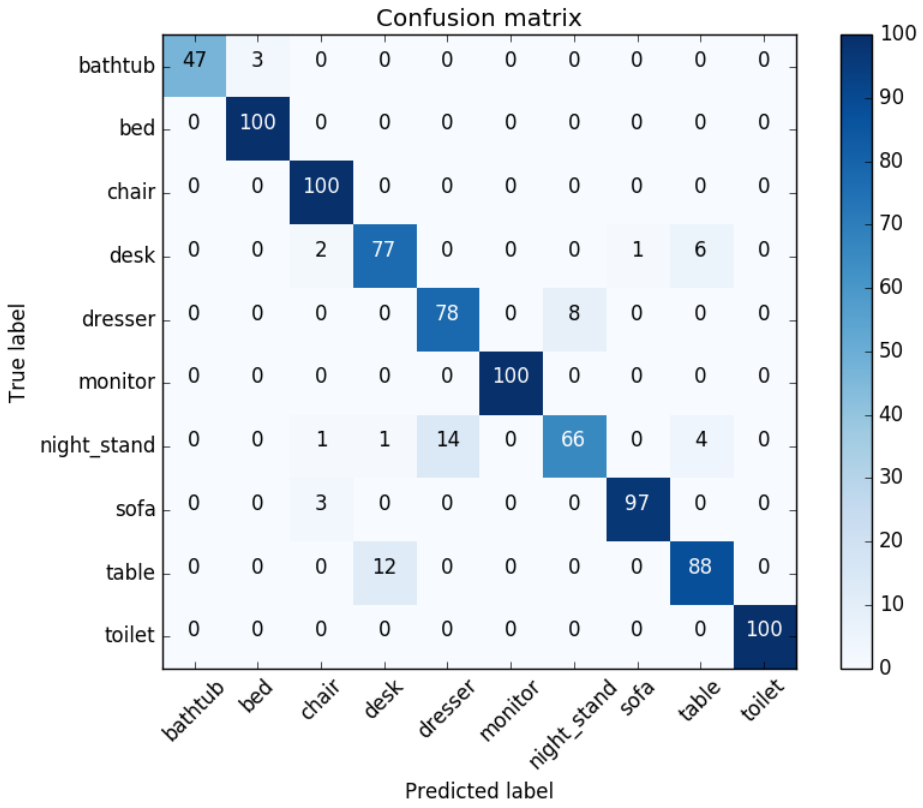

Fig. 5. Confusion matrix of our best performing model on ModelNet10

hours. In Figure 5, the confusion matrix of this model on ModelNet10 is depicted. As it can be seen, most of the misclassified 3D models were assigned a label of a similar category compared to the ground truth.

5 Conclusions

We have explored the extension of residual networks in the 3D domain for addressing the task of 3D object classification. In particular, we used volumetric representations as they provide a rich and powerful representation of 3D shapes. Our experiments have validated the effectiveness of residual architectures and have shown that the combination of multilevel and wide residual connections can result in competitive performance. More specifically, we managed to achieve equivalent or better classification accuracy than bigger and more complicated networks on a well-known dataset. In future work, we would like to investigate other variants of the original ResNets and test different training configurations in order to gain more insights considering the effectiveness of 3D residual networks on classifying and recognizing 3D shapes.

Acknowledgements

The research leading to these results has received funding from the European Union H2020 Horizon Programme (2014-2020) under grant agreement 665066, project DigiArt (The Internet Of Historical Things And Building New 3D Cultural Worlds).

References

1. Brock, A., Lim, T., Ritchie, J., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks. CoRR **abs/1608.04236** (2016), http://arxiv.org/abs/1608.04236
2. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: British Machine Vision Conference (BMVC) (2014)
3. Chollet, F., et al.: Keras. https://github.com/fchollet/keras (2015)
4. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, ., Wang, G., Cai, J., et al.: Recent advances in convolutional neural networks. Pattern Recognition (2017)
5. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: IEEE International Conference on Computer Vision (ICCV). pp. 1026–1034 (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
7. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Proceedings of the 14th European Conference on Computer Vision (ECCV) - Part IV. pp. 630–645 (2016)
8. Hegde, V., Zadeh, R.: FusionNet: 3d object classification using multiple data representations. CoRR **abs/1607.05695** (2016), http://arxiv.org/abs/1607.05695
9. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.: Deep networks with stochastic depth. In: Proceedings of the 14th European Conference on Computer Vision (ECCV), Part IV. pp. 646–661 (2016)

10. Ioannidou, A., Chatzilari, E., Nikolopoulos, S., Kompatsiaris, I.: Deep learning advances in computer vision with 3d data: A survey. ACM Computing Surveys **50**(2), 20:1–20:38 (2017). https://doi.org/10.1145/3042064, http://doi.acm.org/10.1145/3042064
11. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning (ICML). pp. 448–456 (2015), http://jmlr.org/proceedings/papers/v37/ioffe15.html
12. Johns, E., Leutenegger, S., Davison, A.: Pairwise decomposition of image sequences for active multi-view recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3813–3822 (2016)
13. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014), http://arxiv.org/abs/1412.6980
14. Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. CoRR **abs/1704.01222** (2017), http://arxiv.org/abs/1704.01222
15. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 922–928 (2015)
16. Moniz, J., Pal, C.: Convolutional residual memory networks. CoRR **abs/1606.05262** (2016), http://arxiv.org/abs/1606.05262
17. Qi, C., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.: Volumetric and multi-view cnns for object classification on 3d data. CoRR **abs/1604.03265** (2016), http://arxiv.org/abs/1604.03265
18. Sedaghat, N., Zolfaghari, M., Brox, T.: Orientation-boosted voxel nets for 3d object recognition. CoRR **abs/1604.03351** (2016), http://arxiv.org/abs/1604.03351
19. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**, 1929–1958 (2014)
20. Srivastava, R., Greff, K., Schmidhuber, J.: Highway networks. CoRR **abs/1505.00387** (2015), http://arxiv.org/abs/1505.00387
21. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 945–953 (2015)
22. Targ, S., Almeida, D., Lyman, K.: Resnet in resnet: Generalizing residual architectures. CoRR **abs/1603.08029** (2016), http://arxiv.org/abs/1603.08029
23. Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints **abs/1605.02688** (May 2016), http://arxiv.org/abs/1605.02688
24. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d ShapeNets: A deep representation for volumetric shapes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
25. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: BMVC (2016)
26. Zhang, K., Sun, M., Han, X., Yuan, X., Guo, L., Liu, T.: Residual networks of residual networks: Multilevel residual networks. IEEE Transactions on Circuits and Systems for Video Technology **PP**(99), 1–1 (2017)