Multi-level 3D CNN for Learning Multi-scale Spatial Features

Sambit Ghadai Xian Yeow Lee Aditya Balu Soumik Sarkar Adarsh Krishnamurthy
[sambitg | xylee | baditya | soumiks | adarsh]@iastate.edu
Iowa State University, Ames, IA

Abstract

3D object recognition accuracy can be improved by learning the multi-scale spatial features from 3D spatial geometric representations of objects such as point clouds, 3D models, surfaces, and RGB-D data. Current deep learning approaches learn such features either using structured data representations (voxel grids and octrees) or from unstructured representations (graphs and point clouds). Learning features from such structured representations is limited by the restriction on resolution and tree depth while unstructured representations creates a challenge due to non-uniformity among data samples. In this paper, we propose an end-to-end multi-level learning approach on a multi-level voxel grid to overcome these drawbacks. To demonstrate the utility of the proposed multi-level learning, we use a multi-level voxel representation of 3D objects to perform object recognition. The multi-level voxel representation consists of a coarse voxel grid that contains volumetric information of the 3D object. In addition, each voxel in the coarse grid that contains a portion of the object boundary is subdivided into multiple fine-level voxel grids. The performance of our multi-level learning algorithm for object recognition is comparable to dense voxel representations while using significantly lower memory.

1. Introduction

A three dimensional object comprises of a different multi-scale features inherent to its geometry and its overall shape. Deep Neural Networks have been used to extract meaningful information from spatial data and perform object recognition. Several works have made substantial efforts to perform object recognition from 3D data by extending image recognition principles such as projection of the 3D information to 2D or 2.5D (depth inclusion) images [18, 15] and multiple 2D views of the 3D object [7, 5, 14, 11]. Though this is effective in many applications including 3D reconstruction, some spatial relationships among the features get lost and this makes it infeasible for certain problems such as graphics rendering [16], point cloud labeling [12], design and manufacturing [3]. However, a major limitation in learning directly from 3D data is the high memory requirement. The presence of abundant information in spatial data coupled with the large data requirement for efficient training of deep learning algorithms render this task impractical for high-resolution 3D data.

Convolutional Neural Networks (CNNs) are natural candidates for this task as they have been proven to be effective for learning features from 3D spatial data [4, 9]. However, training CNNs using uniform data representations (such as voxels) become inefficient when spatial features exist on different physical scales since uniform data representation cannot effectively accommodate this non-uniformity [1]. Hence, efficient and scalable deep learning techniques that exploit sparse and hierarchical data representations are necessary to deal with large 3D data sets. The most common high resolution voxel representation of 3D geometries is Octree [10], which is a structured representation that recursively divides each voxel into 8 sub-voxels and stores them in a tree structure. Octree based learning frameworks like OctNet [13] require custom convolution operations specific for the octree data structure. This approach facilitates learning from high-resolution structured data.

In this paper, we present a novel approach to enable hierarchical learning of features from a flexible multi-level unstructured voxel representation of spatial data. We achieve this by adopting the multi-level voxelization framework developed by Young et. al [20]. A multi-level voxel grid is defined as a binary occupancy grid at two levels to represent a 3D object with two independent user-defined resolutions of voxel grids. We developed a multi-level CNN that can effectively learn features despite the unstructured nature of the multi-level data representation.

2. Multi-level Voxelization

In this section, we briefly describe the GPU-accelerated algorithm [20] we used to generate the multi-level voxelization from boundary representation(B-rep) of a 3D model. The multi-level voxelization is a binary occupancy grid having two major components namely, *coarse-level voxelization* and *fine-level voxelization*. The *coarse-level* voxel grid represents the whole 3D CAD model at a coarse resolution

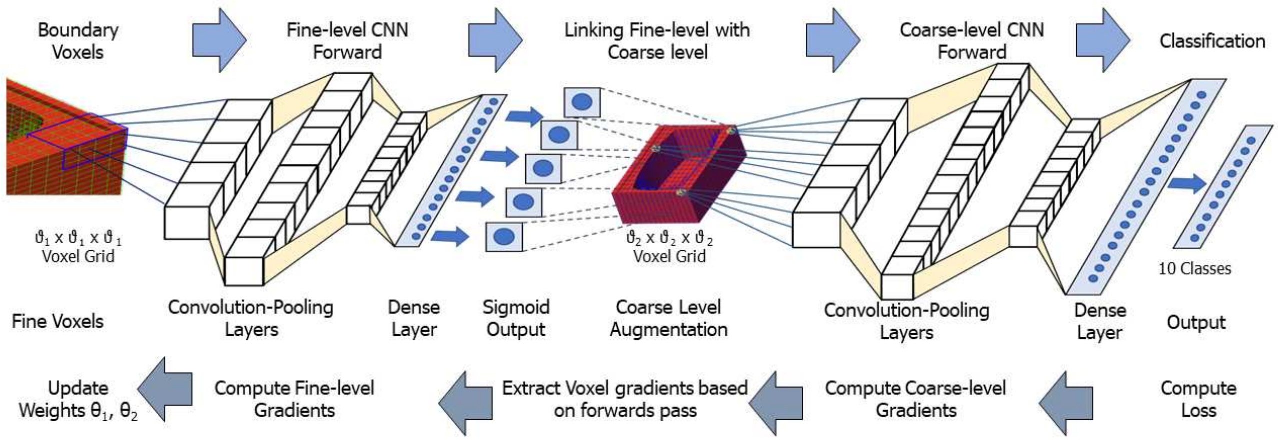


Figure 1: Multi-Resolution Convolutional Neural Network (MRCNN). Our proposed network can learn from a hierarchical data representation with a coarse level of information and information of boundary voxels which connects to the information from fine level voxels. For a forward pass (left to right) the information learnt from selected fine level voxels using the *fine-level* CNN is embedded in the coarse level input to *coarse-level* CNN and then the final prediction is obtained. The backward pass follows the reverse order of the forward pass (right to left).

and the *fine-level* voxel grid represents the boundary of the *coarse-level* voxel grid at a finer resolution in a hierarchical manner. The two levels of voxel grids are mapped to each other using a prefix-sum array mapping. For example, a CAD model can be represented at the *coarse-level* with a voxel resolution of $32 \times 32 \times 32$ and each of the coarse boundary voxels can be further voxelized at a resolution of $4 \times 4 \times 4$ (see Figure 2). This makes the CAD model to be represented with an effective resolution of $128 \times 128 \times 128$ using the multi-level voxelization. We use a multi-level voxel data structure to store information pertaining to the geometry of an object in two hierarchical levels, thus exploiting the sparse nature of the data.

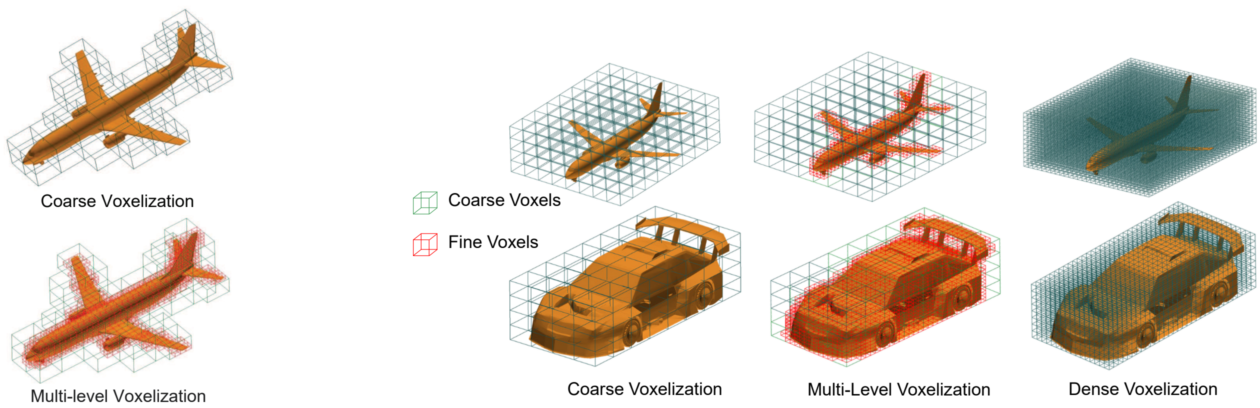
3. Multi-resolution CNN

The multi-resolution convolutional neural network (MRCNN) consists of two 3DCNNs, with each CNN kernels performing 3D convolution operations, to learn the features in each level of the voxel grid. One of these 3DCNNs, named as *Coarse-level CNN*, takes in the coarse level voxels as input while the other 3DCNN called *Fine-level CNN* takes the fine level voxels as input. These two neural networks are intelligently combined to work together as a single unit in both forward pass and backward pass of the algorithm. This facilitates optimal learning from a multi-level data representation.

The forward computation of MRCNN starts by learning from the fine-level voxel grids by randomly sampling a subset, ϕ, of the total boundary voxels, Φ, in a 3D voxelized model. Each of these ϕ boundary voxels, with individual fine voxel grid ϑ_2, are used as input to *Fine-level CNN*. The *Fine-level CNN* consists of blocks of *convolution - max pooling* layer pairs and *fully connected* layers connected sequentially, each with a ReLU activation function associated with it. *Fine-Level CNN* outputs a single real numbered value η_b for each of the selected boundary voxels Φ. We replace the original coarse voxel grid values with η_b at the corresponding voxel positions. This is performed with the help of the prefix sum based index arrays of the multi-level voxel grid as explained in [20].

In the next phase of the MRCNN forward computation, the coarse-level voxel grid with selective embedding of the fine level voxel information η_b, is used as an input to the *Coarse-level CNN*. The architecture of *Coarse-level CNN* network comprises of different set of *convolution - max pooling* layers. The end of the network has multiple *fully connected* layers and the output is the class prediction probability vector. Categorical cross-entropy loss function is used to compute the loss of between predicted classes and true class labels. The forward pass of MRCNN network algorithm is illustrated in Figure 1.

Once the forward computation of the MRCNN is established, the only challenge is to link the two networks such that the gradients can passed on from the coarse level network to the fine level network during back-propagation. This link is essential for obtaining gradients for the weights of the fine level network. The final loss between the y_{pred} and y_{true} of the coarse level network is first computed using categorical cross-entropy loss. Back-propagating this loss through the coarse level network is trivial. Once we obtain the gradients for input coarse level voxel embedding, we compute the gradient of η_b and use that to backpropagate the same in the fine level voxel grid. Let the gradient of the loss with respect to coarse input be $d\theta_1$, using prefix sum, we track the gradients of the outputs of fine level network (η_b) and use it to back-propagate through the network.



(a) (b)

Figure 2: Multi-level voxelization of B-rep CAD models. The fine level voxelization is performed only near the boundaries of the coarse level voxelization. The final resolution is equivalent to having dense level voxels throughout the model.

It is also worthwhile to note that since the same *Fine-level CNN* is shared among all the boundary voxels, the gradients of θ_2 for *Fine-level CNN* are computed for all boundary voxels only once.

With the gradients linked, the network could be trained end-to-end to update its weights θ_1 and θ_2 in such a way that the loss L, of the final prediction is minimized. The network parameters' update could be performed using the *Adam* optimizer [6]. The complete operation of MRCNN is explained schematically in Figure 1.

4. Experimental Results & Discussion

In this section, we present the classification results of the proposed MRCNN framework on ModelNet10 and ModelNet40 datasets [18] that contain 3D geometric models of 10 and 40 different categories respectively. The 3D models are voxelized using the voxelization scheme mentioned in Section 2, yielding a set of coarse voxel grid and fine voxel grids with a single resolution of 8^3 and 32^3 respectively. Additionally, we also voxelized two sets of multi-resolution data to test the efficacy of MRCNN; a 8^3 coarse voxel grid with a 4^3 fine voxel grid giving an effective resolution of 32^3 resolution and a 32^3 coarse voxel grid with a 4^3 fine voxel grid, resulting in a effective resolution of 128^3. We conducted a set of experiments on the 4 different resolutions of data and compared the classification performance between a *Coarse-Level CNN* applied on the coarse and dense resolution data and MRCNN applied on the multi-resolution data. For the multi-resolution data, we applied our proposed MRCNN by randomly sampling 40% of the *coarse-level* boundary voxels, and used the fine resolution voxels of these coarse boundary voxels as input to the *Fine-level CNN*. We then selectively embed the output of *Fine-level CNN* in the coarse level boundary voxels and continue the forward pass. Empirically, we find that sampling 40% of boundary voxels gives a good classification performance without prolonging the training time excessively.

Figure 3 shows the mean test accuracy of object classification using MRCNN on ModelNet10 test dataset by running multiple inferences with various network hyperparameters. Variance in the classification accuracies are represented by the shaded region. We see that there is a clear trend showing better performance for higher effective resolution. Comparing the performance of a regular CNN on the coarse 8^3 resolution data with the performance of MRCNN on multi-resolution data, it is evident that MRCNN enables has better performance. Subsequently, a regular CNN applied on a dense voxel grid of 32^3 is able to achieve a slightly better classification accuracy than both. Due to memory constraints of GPUs, we are unable to demonstrate the performance of a *Coarse-level CNN* applied on dense

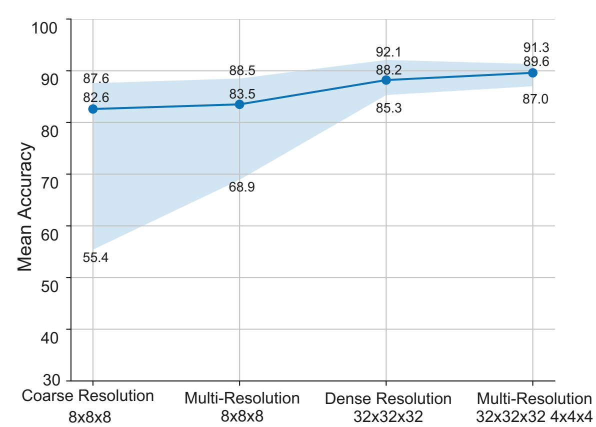


Figure 3: Mean classification accuracies with different input resolutions on ModelNet10 dataset. Coarse and dense resolutions are trained with a conventional 3DCNN while the multi-level voxel grids are trained with MRCNN.

Table 1: Comparison of deep learning frameworks with voxel based representation for ModelNet10 object recognition. * represents value interpreted from plot

Method	Data Representation	Accuracy %
MRCNN	**Multi-level voxels**	**91.3**
OctNet [13]	Octree Voxels	91.0*
3D Shapenets [18]	Voxels	83.5
VoxNet [9]	Voxels	92.0
Beam Search [19]	Voxels	88.0
3DGAN [17]	Voxels	91.0
binVoxNetPlus [8]	Voxels	92.3
LightNet [21]	Voxels	93.9

resolution data of 128^3. Nonetheless, using MRCNN, we are able to train and achieve the best classification performance using an effective resolution of 128^3 represented by a coarse resolution of 32^3 and a finer resolution of 4^3.

Comparisons of our object classification results with the performance of other spatial deep learning methods are tabulated in Tables 1 and 2 for ModelNet10 and ModelNet40 dataset respectively. We highlight the performance of MRCNN with respect to OctNet due to the similarities in data representation (high resolution voxel grid) and classification task that exploits the sparsity in spatial data in both the frameworks. In addition to that, we compare MRCNN performance with other voxel based methods employed on the ModelNet datasets. We can see that MRCNN (91.3%) outperforms some of the voxel based methods and is better at classification than OctNet (91.0%) for ModelNet10. A similar trend is seen in ModelNet40 classification accuracies.

An additional advantage of the MRCNN framework is lower GPU memory utilization during training of the network. In Figure 4, we show a comparison between the memory requirements of the GPU for training on four different resolutions of voxel data with constant batchsize. The memory required by a GPU scales polynomially (n^3) with the voxel grid resolution n, hence we were unable to train a dense-level network on 128^3 voxel resolution (shown as a blue hatched bar). We can see that MRCNN training with multi-level voxel grid representations utilizes considerably less memory than a dense CNN network training on the same effective resolution dense voxel grid.This highlights the effect of sparsity where the increase in classification performance scales non-linearly with data resolution.

Table 2: Comparison of deep learning frameworks with voxel based representation for ModelNet40 object recognition. * represents value interpreted from plot.

Method	Data Representation	Accuracy %
MRCNN	**Multi-level voxels**	**86.2**
OctNet	Octree Voxels	85.5*
3D Shapenets	Voxels	77.3
VoxNet	Voxels	83.0
Beam Search	Voxels	81.26
3DGAN	Voxels	83.3
binVoxNetPlus	Voxels	85.47
LightNet	Voxels	88.93

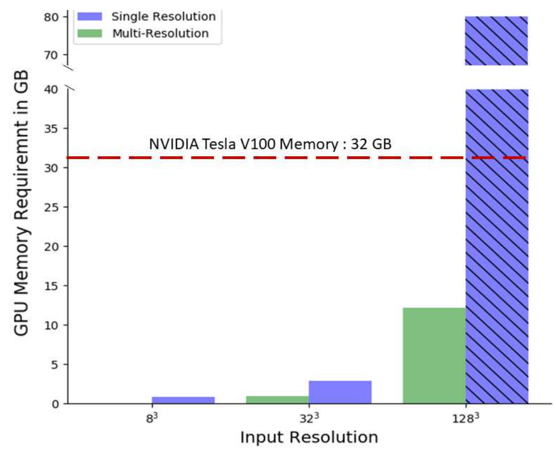


Figure 4: GPU memory usage of MRCNN training & equivalent CNN training on specified voxel grid resolutions. Red horizontal line shows the current prominent GPU capacity. Blue hatched bar depicts the anticipated memory usage while training a 128^3 dense voxel grid on CNN.

5. Conclusions

In this paper, we explore a novel deep learning architecture, MRCNN, to learn from 3D data in a hierarchical manner using multi-level voxel-based data structures. Our object recognition results show that MRCNN performance is significantly better and robust compared to that of the regular CNNs trained on coarse-resolution data while having similar memory requirements. MRCNN also performs almost as well as CNNs trained on dense data with equivalent resolution while keeping the memory requirements significantly lower. Future works will include exploring efficacies of MRCNN on various object recognition datasets as well as other relevant computer vision problems where extraction of multi-scale features is critically important.

Acknowledgements

This paper is part of the Deep Learning for Geometric Shape Understanding workshop [2] in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2019). This work is supported in part by NSF under Grant No. CMMI:1644441.

References

[1] A. Abdul-Rahman and M. Pilouk. *Spatial data modelling for 3D GIS*. Springer Science & Business Media, 2007.

[2] I. Demir, C. Hahn, K. Leonard, G. Morin, D. Rahbani, A. Panotopoulou, A. Fondevilla, E. Balashova, B. Durix, and A. Kortylewski. SkelNetOn 2019 Dataset and Challenge on Deep Learning for Geometric Shape Understanding. *arXiv e-prints*, 2019.

[3] S. Ghadai, A. Balu, S. Sarkar, and A. Krishnamurthy. Learning localized features in 3D CAD models for manufacturability analysis of drilled holes. *Computer Aided Geometric Design*, 62:263 – 275, 2018.

[4] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris. Deep learning advances in computer vision with 3D data: A survey. *ACM Computing Surveys (CSUR)*, 50(2):20, 2017.

[5] 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.

[6] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] J. Li, B. M. Chen, and G. H. Lee. So-net: Self-organizing network for point cloud analysis. *arXiv preprint arXiv:1803.04249*, 2018.

[8] C. Ma, Y. Guo, Y. Lei, and W. An. Binary volumetric convolutional neural networks for 3D object recognition. *IEEE Transactions on Instrumentation and Measurement*, pages 1–11, 2018.

[9] D. Maturana and S. Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, Sept 2015.

[10] D. Meagher. Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2):129–147, 1982.

[11] 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*, pages 5648–5656, 2016.

[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

[13] G. Riegler, A. O. Ulusoys, and A. Geiger. Octnet: Learning deep 3D representations at high resolutions. *arXiv preprint arXiv:1611.05009*, 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] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proc. ICCV*, 2015.

[16] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. *arXiv preprint arXiv:1703.09438*, 2017.

[17] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 82–90, USA, 2016. Curran Associates Inc.

[18] 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*, pages 1912–1920, 2015.

[19] X. Xu and S. Todorovic. Beam search for learning a deep convolutional neural network of 3D shapes. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3506–3511, Dec 2016.

[20] G. Young and A. Krishnamurthy. GPU-accelerated generation and rendering of multi-level voxel representations of solid models. *Computers & Graphics*, 75:11–24, October, 2018.

[21] S. Zhi, Y. Liu, X. Li, and Y. Guo. Toward real-time 3D object recognition: A lightweight volumetric CNN framework using multitask learning. *Computers & Graphics*, 71:199 – 207, 2018.