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
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, $\phi$, of the total boundary voxels, $\Phi$, in a 3D voxelized model. Each of these $\phi$ boundary voxels, with individual fine voxel grid $\phi_2$, 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 $\eta_b$ for each of the selected boundary voxels $\phi$. We replace the original coarse voxel grid values with $\eta_b$ 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 $\eta_b$, 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 $\eta_b$ 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 ($\eta_b$) and use it to back-propagate through the network.
It is also worthwhile to note that since the same Fine-level CNN is shared among all the boundary voxels, the gradients of $\theta_2$ 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 $\theta_1$ and $\theta_2$ 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 an 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 resolutions.
Table 1: Comparison of deep learning frameworks with voxel based representation for ModelNet10 object recognition. * represents value interpreted from plot.

<table>
<thead>
<tr>
<th>Method</th>
<th>Data Representation</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRCNN</td>
<td>Multi-level voxels</td>
<td>91.3</td>
</tr>
<tr>
<td>OctNet [13]</td>
<td>Octree Voxel</td>
<td>91.0*</td>
</tr>
<tr>
<td>3D Shapenets [18]</td>
<td>Voxel</td>
<td>83.5</td>
</tr>
<tr>
<td>VoxNet [9]</td>
<td>Voxel</td>
<td>92.0</td>
</tr>
<tr>
<td>Beam Search [19]</td>
<td>Voxel</td>
<td>88.0</td>
</tr>
<tr>
<td>3DGAN [17]</td>
<td>Voxel</td>
<td>91.0</td>
</tr>
<tr>
<td>binVoxNetPlus [8]</td>
<td>Voxel</td>
<td>92.3</td>
</tr>
<tr>
<td>LightNet [21]</td>
<td>Voxel</td>
<td>93.9</td>
</tr>
</tbody>
</table>

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.

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


