Eurographics Workshop on 3D Object Retrieval (2017)
I. Pratikakis, F. Dupont, and M. Ovsjanikov (Editors)

LightNet: A Lightweight 3D Convolutional Neural Network for Real-Time 3D Object Recognition

S. Zhi, Y. Liu, X. Li, and Y. Guo

College of Electronic Science and Engineering, National University of Defense Technology, China

Abstract

With the rapid growth of 3D data, accurate and efficient 3D object recognition becomes a major problem. Machine learning methods have achieved the state-of-the-art performance in the area, especially for deep convolutional neural networks. However, existing network models have high computational cost and are unsuitable for real-time 3D object recognition applications. In this paper, we propose LightNet, a lightweight 3D convolutional neural network for real-time 3D object recognition. It achieves comparable accuracy to the state-of-the-art methods with a single model and extremely low computational cost. Experiments have been conducted on the ModelNet and Sydney Urban Objects datasets. It is shown that our model improves the VoxNet model by relative 17.4% and 23.1% on the ModelNet10 and ModelNet40 benchmarks with less than 67% of training parameters. It is also demonstrated that the model can be applied in real-time scenarios.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations , I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene Analysis—Shape, I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene Analysis—Object recognition, I.5.1 [PATTERN RECOGNITION]: Models—Neural nets

1. Introduction

Real-time 3D object recognition is a significant and critical research topic in the computer graphics and computer vision areas for its numerous applications including human-machine interaction, self-driving cars and intelligent robots. With the rapid development of 3D sensors such as LIDARs and RGB-D cameras, 3D data become increasingly accessible. Currently, a growing number of large 3D object repositories are available [DDQHD13, WSK*15, CFG*15], making the development of deep learning based 3D object recognition algorithms possible. Due to their high-level representation through hierarchical non-linear transformations, Convolutional Neural Networks (CNNs) have been increasingly investigated for 3D object recognition systems.

Feature learning based 3D object recognition methods can broadly be classified into two categories according to data representation: multi-view based and volumetric representation based. Several multi-view methods [XSZ*16, SMKLM15, SBZB15, QSN*16] learn features from a collection of 2D projection images of 3D objects rendered in various view points. These methods can use existing 2D pretrained networks to achieve impressive performance. However, projections from a large number of view points have to be rendered, especially when the object is not uprightly oriented [SMKLM15, SBZB15, HLH*16]. Besides, multi-view methods just provide 2D contour representations of 3D objects [FXD*15] and do not include sufficient geometrical information as 3D representations because some detailed information (e.g., curvatures) are not encoded. Consequently, 3D volumetric networks become increasingly popular. Wu et al. [WSK*15] proposed a 3D generative volumetric network for 3D object recognition and established the 3D ModelNet dataset. Subsequently, an increasing number of 3D volumetric CNNs have been proposed to work on complete 3D shapes [GGGDGR*16, MS15]. Qi et al. [QSN*16] proposed a volumetric 3D CNN by subvolume supervision to address overfitting. Sedaghat et al. [SZB16] used the orientation information of objects to boost the category prediction accuracy. Brock et al. [BLRW16] fused several very deep 3D neural networks to obtain better performance than multi-view CNNs. Although good results have been achieved, the computational cost of these existing deep learning models are high and their frameworks are complex. The large number of parameters make the network hard to train and unsuitable for on-board real-time applications. Moreover, the large amount of parameters make the network prone to overfitting small datasets and can only work well given sufficient training samples. In this paper, we focus on real-time 3D object recognition using a single 3D CNN model without ensemble. Our model achieves comparable results to the state-of-the-art methods with reduced complexity of the network model.

We propose a lightweight convolutional neural network (namely, LightNet) based on VoxNet [MS15] by leveraging multitask learning to improve both the category prediction accuracy and efficiency.

DOI: 10.2312/3dor.20171046

Multitask learning can be considered as a regularization term applied to neural networks to exploit the similarities among different tasks and avoid overfitting. Besides, prior information or domain knowledge can be used in the training process to obtain better accuracy and learning efficiency via multitask learning as compared to single task-specific learning models [Car98]. Inspired by previous work [QSN*16, SZB16] and the fact that human can perceive both the category and orientation information of 3D objects only through partial shape of objects, we improve the recognition performance of 3D CNN by forcing our model to predict category labels and orientation information through entire or partial shapes. In real-world applications, 3D objects are usually occluded in cluttered environment and orientation is a significant feature for object recognition. It is therefore, reasonable to combine these two auxiliary learning tasks. Furthermore, we simplify the framework of our proposed model to make it more computationally efficient and easier to train. Subsequently, the proposed lightweight CNN model can work well on large datasets but also scale well to small datasets without overfitting. The proposed model has been tested on synthesized 3D CAD models and real-world LIDAR point clouds. Experimental results show that our LightNet model can learn robust features by multitask learning and achieve promising category prediction accuracy in real-time on benchmark datasets. It improves the original model [MS15] by relative 17.4% and 23.1% on two popular 3D category prediction benchmarks ModelNet10 and ModelNet40 [WSK*15] with less than 67% training parameters. In addition, our model is more efficient than most existing 3D CNNs for 3D object recognition. It takes 3-5 ms to classify an object.

The major contributions of this work are summarized as follows. **First,** we propose a lightweight volumetric 3D CNN for 3D object recognition. It has less training parameters as compared to existing models including VoxNet [MS15], FusionNet [HZ16] and VRN Ensemble [BLRW16]. **Second,** we combine different kinds of auxiliary learning tasks into a network framework to handle both small and large datasets without obvious overfitting. **Third,** comparative experiments have been conducted on the ModelNet dataset and the Sydney Urban Objects dataset. It is shown that the proposed model provides a basic structure for on-board real-time recognition tasks for its small storage requirement and low computational cost.

This paper is structured as follows. Section 2 gives a literature review of 3D object recognition methods, with a focus on CNN based approaches. Section 3 presents our model and implementation details. Section 4 evaluates our lightweight 3D CNN framework. Section 5 finally concludes the paper.

2. Related Work

The core of 3D object recognition is to extract discriminative, concise, and low-dimensional 3D shape features. Classical methods aim to design features according to specific tasks and domain knowledge. On the contrary, recent deep learning (especially CNN) based approaches automatically learn powerful 3D features in an end-to-end manner with promising generalization performance.

2.1. Hand-Crafted 3D Shape Features

Existing hand-crafted 3D shape features can be broadly divided into two major categories: global features and local features [GBS*14, BA10, KPVG10]. Global shape features process a 3D shape as a whole but are unsuitable for recognizing occluded objects in cluttered scenes. Examples of global shape features include viewpoint histogram [RBTH10] and shape distributions [OFCD02]. In contrast, local shape features outperform their global counterparts in cluttered scenes. Representative 3D local shape features include spin image [JH98], rotational projection statistics (RoPS) [GSB*13], heat kernel signatures (HKS) [SOG09] and fast point feature histogram (FPFH) [RBB09]. Besides, several 3D local features are extended from 2D image features, e.g., 3D SURF [KPW*10] and 2.5D SIFT [LS09]. These methods have been successfully applied in various areas including 3D shape matching, object recognition and 3D shape retrieval. However, they highly rely on human design and domain knowledge. Consequently, it is challenging for those shape features to work on large 3D repositories consisted of various objects from different domains or tasks.

2.2. 3D CNN based Methods

CNNs have been successfully used for detection, segmentation and recognition of objects in images [LBH15]. Later, 2.5D CNNs are extended to RGB-D data for 3D object category prediction by considering depth channel as an additional channel [SHB*12, Ale16]. Therefore, 3D geometric information is not fully utilized.

3D CNN is first used in video analysis by considering time as the third dimension [KTS*14, YWZ*15]. Wu et al. [WSK*15] designed a convolutional deep belief network to reconstruct 3D shapes from 2.5D RGB-D images and represented voxelized geometrical shapes using a binary probabilistic distribution for the recognition task. Su et al. [SMKLM15] used view-based representations for 3D shapes to take full advantage of established 2D CNN frameworks (e.g., VGG-M) pretrained on ImageNet. They achieved good recognition accuracy through view-pooling layer and noticed that there was a large gap between the performance of 3D volumetric CNN and multi-view CNN. VoxNet [MS15] was then proposed for real-time 3D object recognition to mitigate the gap. It provides a concise network structure with reasonable performance for real-time applications. Qi et al. [QSN*16] introduced two new volumetric 3D CNNs to use auxiliary training, anisotropic probing and Network In Network (NIN) structure [LCY13], with a comparable performance to multi-view CNNs being achieved. FusionNet [HZ16] was proposed to fuse two volumetric CNNs and one multi-view CNN in the output layer. It combines 3D and 2D features to boost the performance. FusionNet outperforms previous work in recognition performance on the ModelNet dataset. In addition, Sedaghat et al. [SZB16] improved the performance based on VoxNet using orientation estimation. So far, the state-of-the-art recognition performance on ModelNet is achieved by [BLRW16] via an ensemble of 6 deep networks. Voxception-ResNet has 45 layers and 18M parameters, and takes 6 days for training. Besides, several multi-view approaches [SBZB15, JLD16] and volumetric methods [GGGDGR*16, WZX*16] can also be found in literature.

Although promising 3D object recognition performance has been achieved by existing network models, they either lack of generality in training scheme or sacrifice computational efficiency. Our model is different from those approaches in two aspects. **First,** our lightweight 3D CNN considers both recognition accuracy and efficiency. Therefore, we can achieve satisfying results in real time with a small number of training parameters. The compact structure also allows our model to be converged in a short time during training. **Second,** we leverage on different types of auxiliary tasks to mitigate the limitation of 3D feature learning caused by the relatively shallow network structure. Our model can learn robust features not only on synthetic CAD models but also on occluded objects in cluttered scenes. Therefore, our model has the potential for applications in real scenarios.

3. The Proposed LightNet Model

In this section, our model for 3D object recognition is introduced.

3.1. Binary Volumetric Occupancy Grid

Occupancy grids [Thr03] can be used to represent a 3D scene as discrete grids. Each unit in this lattice cell is called a voxel, which is described by a random variable. Similar to most 3D CNNs, the input to our model should be voxels, i.e., occupancy grids. Consequently, 3D data represented by point clouds (Fig. 1a) or meshes (Fig. 1b) should be converted to volumetric data (Fig. 1c) before being recognized.

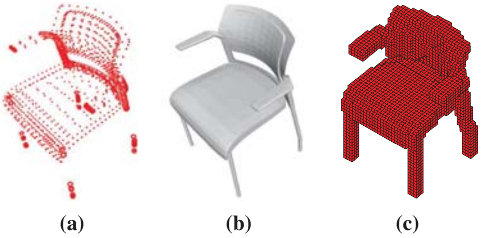


Figure 1: *3D shape representations. (a) Point cloud, (b) Mesh, (c) Volumetric representation.*

Specifically, a shape is represented as a binary 3D tensor, where a binary '1' indicates that the voxel is inside the 3D surface or intersects with the 3D surface, '0' indicates the voxel is outside the 3D surface or dose not intersect with the 3D surface. The grid resolution has influences on the performance of CNNs, specifically, a larger grid size reserves more shape details but is also more computationally expensive, while a smaller grid size reduces the computational cost but makes the representation less discriminative. In this paper, the grid resolution is set to $24 \times 24 \times 24$ to achieve a compromise between computational efficiency and discriminativeness. The voxel grid of interest is further padded with four '0' voxels in all directions to reduce the boarder artifacts during convolution. Consequently, the final grid size is $32 \times 32 \times 32$.

3.2. Network Framework

Our model is inspired by the fact that, human can sense both the category and the orientation of an object simply from partial 3D data, and also know which part acts as an important feature for a 3D object. In this paper, our model facilitates 3D object recognition by multitask learning. The framework of our model is highly compact, as illustrated in Fig. 2. The first three layers (including the input layer) follow the same pattern as VoxNet [MS15]. However, our model is deeper and integrates both subvolume supervision task (for object category prediction) [QSN*16] and orientation estimation task [SZB16] to avoid overfitting and improve discriminativeness.

The input layer has a size of $32 \times 32 \times 32$ to accept binary occupancy grids. The output of each 3D convolutional layer is activated by a leaky rectified nonlinearity unit (LReL) with its parameter equal to 0.1, which allows for a small, non-zero gradient when the unit is saturated and inactive [MHN13]. We also add an additional conlolutional layer and a max-pooling layer to improve the discriminativeness of the learned feature and to reduce the number of parameters, since most of the parameters in VoxNet lie in the first fully connected layer.

In the fifth layer, the network is divided into two branches: the main branch and the auxiliary branch. The main branch consists of a fully connected layer and a softmax output layer. For the auxiliary branch, we slice the neurons in the same spatial position along channels from the output of last max-pooling layer (size of $2 \times 2 \times 2 \times 128$ corresponding to x, y, z axes and channels) and reshape them to 8 one-dimensional vectors with a length of 128. After appending the new reshaped fully connected layer with a softmax layer, we assign a category prediction task to each sliced vector to improve recognition performance from partial shape. The receptive field of each neuron in the sliced vector of the auxiliary branch is $23 \times 23 \times 23$, which occupies 72% voxels of the input space.

Besides, we also combine the orientation estimation task with our model to boost recognition performance. We add a parallel output layer to predict the orientation of an object. Here, nine orientation estimation layers are extended in a form of softmax layers. Although orientation is a continuous variable and should be better estimated by regression, we consider orientation estimation as a category prediction task to make the training more convenient [SZB16]. Moreover, from a data augmentation perspective (Sec. 3.3), it is also reasonable to sample the orientation using discrete labels. Finally, average pooling is imposed on the output layers for all tasks to make the final decision. Consequently, we can fully use the information from each path in our model and probability from softmax function. The overall framework of our model is more compact than [SZB16] because of the final average pooling. We found in our experiments that max-pooling achieves worse performance than average pooling, that is perhaps because a lot of useful information for prediction is abandoned, especially in the auxiliary branch where the information is very limited.

For these reasons, the cross-entropy losses (Eq. 1) for both category prediction and orientation estimation tasks are added to obtain the final total loss (Eq. 2):

$$L_{Cross-entropy} = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\}\log \hat{y}^{(i)}\right] \quad (1)$$

$$L_{Total} = \alpha L_{Classification} + \beta L_{Orientation}, \quad (2)$$

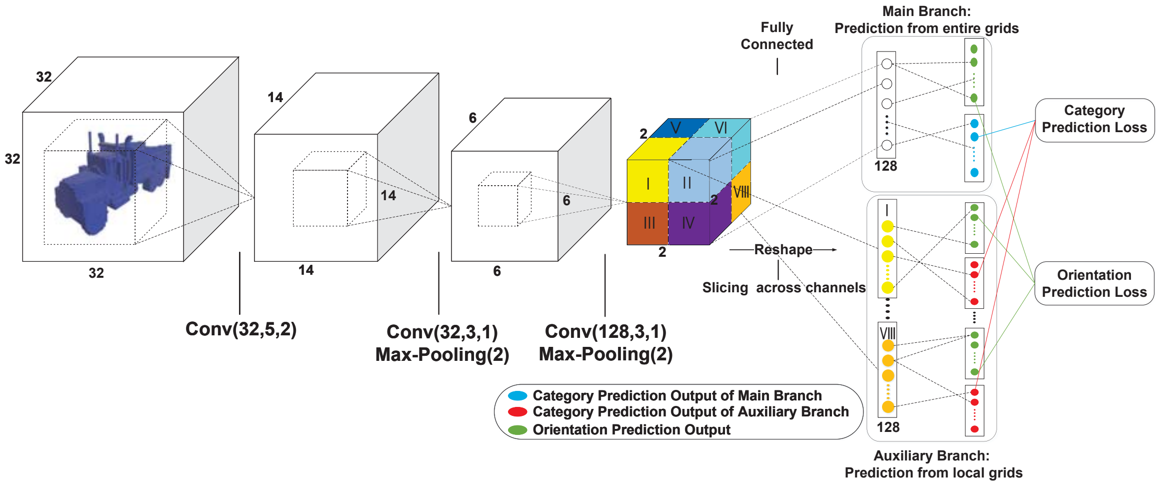

32
32
32
14
14
14
6
6
6
2
2
2
V
VI
I
II
III
IV
VIII
Fully Connected
Main Branch: Prediction from entire grids
128
Category Prediction Loss
Orientation Prediction Loss
Reshape
Slicing across channels
I
VIII
128
Conv(32,5,2)
Conv(32,3,1) Max-Pooling(2)
Conv(128,3,1) Max-Pooling(2)
● Category Prediction Output of Main Branch
● Category Prediction Output of Auxiliary Branch
● Orientation Prediction Output
Auxiliary Branch: Prediction from local grids

Figure 2: *The LightNet framework with auxiliary tasks. Conv(m,k,s) represents m convolution kernels of size k × k × k with a stride of s steps. Max-Pooling(p) means that the maximum value in a volume of p × p × p is used for pooling. Each of the nine fully connected layers have a length of 128, including the one in the main branch and the eight ones in the auxiliary branch. The number of output nodes for orientation estimation task is 12, while the number of category prediction output neurons depends on the dataset.*

where in Eq. 1, $1\{\cdot\}$ is the indicator function, m denotes the total number of training samples, k is the number of output neurons in the final softmax layer (e.g., $k = 10$ for the category prediction of ModelNet10, $k = 12$ for the orientation estimation), $y^{(i)}$ and $\hat{y}^{(i)}$ stands for the true label and its corresponding prediction for the i^{th} output neuron, respectively; in Eq. 2, L_{Total}, $L_{Classification}$ and $L_{Orientation}$ represent the final loss of our model, the loss of object category prediction, and the loss of orientation estimation, respectively. Since our final task is object category prediction rather than orientation estimation, and the auxiliary orientation estimation task is expected to guide our model to learn more powerful and robust feature representations, here we set $\alpha = \frac{2}{3}$ and $\beta = \frac{1}{3}$ to give more importance to category prediction task.

We also use a dropout layer [SHK*14] after each convolutional layer and fully connected layer to achieve better generalization. Besides, we adopt the weight initialization approach given in [HZRS15], and initialize the two output layers in the main branch by a zero-mean Gaussian distribution with a variance of 0.01. The overall network contains only about 300,000 parameters, with the majority of parameters being given in fully connected layers. Therefore, our lightweight model can be easily trained to obtain a satisfying generalization results without a complicated multistage training process. Furthermore, overfitting can also be reduced by our compact framework and we show that the lightweight 3D CNN can achieve promising results on both small and large 3D data repositories.

3.3. Data Augmentation

As discussed in Sec. 1, orientation information is important for 3D object recognition and it is required to obtain a rotational invariant representation for 3D objects. In order to make our model more robust to orientation variations, we augment the training samples by rotating each 3D shape along the z-axis for 12 times with a step of 30 degrees. Accordingly, our auxiliary orientation estimation task has 12 output neurons.

During training, we considered augmented copies of training data as separate samples, and use all the category prediction results from the 12 augmented copies to vote for the final result. In addition, we randomly mirror and translate each object to augment the data during training [MS15]. The augmented data are used to learn invariance to transformations such as rotation and translation. The mirror operation along the x and y axes is conducted with a probability of 0.2 and the translation is within the range from -2 to 2 voxels in all three directions. We also converted the voxel values from $\{0,1\}$ to $\{-1,1\}$ to make the mean of the training data be 0. Besides, we keep the input size to 32 × 32 × 32 in all experiments to make our training scheme concise and efficient.

3.4. Training

Our proposed model was implemented in Python using the Keras deep learning library on top of Tensorflow. Experiments were performed on a single NVIDIA Geforce GTX 1080 GPU enabled by CUDA 8.0, cuDNN 5.1, an Intel Core i7-6700K CPU and 32G RAM.

Network training is achieved by Stochastic Gradient Descent (SGD) with Nesterov momentum [SMDH13], the momentum value is set 0.9 and the batch size is set 32. The objective loss C_{opt} for optimization is given in Eq. 3, which is composed of a cross-entropy loss defined in Eq. 2 and an L_2 weight regularization.

$$C_{opt} = L_{Total} + \frac{\lambda}{2n}\sum_{w} w^2, \tag{3}$$

where w stands for all the training parametes (excluding bias terms) in our model and n denotes the total number of these parameters; the weighting parameter λ is set 0.001. We apply a stepwise anneal-

Table 1: *Recognition accuracy on ModelNet40 with different training methods.*

Training Method	Accuracy on ModelNet40
Train from scratch	82.9%
Fine-tuning **(Pretrained on ModelNet10)**	**86.9%**

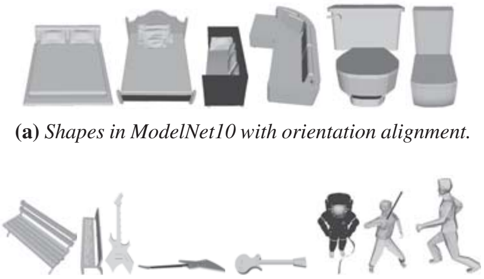
ing schedule for the learning rate and set the learning rate ranging from 0.001 to 0.00001.

4. Experiments and Results

We tested our lightweight 3D CNN model and the state-of-the-art methods on two popular 3D object recognition datasets, i.e., the Princeton ModelNet dataset [WSK*15] and the Sydney Urban Objects dataset [DDQHD13].

4.1. Experiments on the ModelNet Dataset

ModelNet is a large 3D repository of clean CAD models (shapes), the ModelNet10 and ModelNet40 subsets are commonly used and consist of 10 and 40 categories, respectively. There are 4,899 CAD shapes in the ModelNet10 dataset, the orientation of each object in ModelNet10 has been manually aligned by the authors. Therefore, it is a proper benchmark to test our model with orientation estimation and subvolume category prediction tasks. ModelNet40 has 12,311 CAD shapes. Although ModelNet40 dataset contains 3D shapes with various orientations, our model still achieves good performance on this dataset through fine-tuning. Several example models in the ModelNet dataset are shown in Fig. 3.



(a) *Shapes in ModelNet10 with orientation alignment.*

(b) *Shapes in ModelNet40 without orientation alignment.*

Figure 3: *Example 3D shapes in the Princeton ModelNet dataset.*

The voxelized versions augmented by 12 copies are provided in the dataset by scaling each shape to a occupancy grid of size $30 \times 30 \times 30$. Therefore, we pad each 3D voxelized shape by zero voxels to the size of $32 \times 32 \times 32$ for our model. We used the typical train/test split [SBZB15, MS15, QSN*16] originally included in the dataset.

To test our model on ModelNet40 without orientation alignment, we used fine-tuning to show the significance of multitask learning framework. Since ModelNet10 is a subset of ModelNet40, there should be commonalities in representations between these two datasets. Therefore, we first removed the orientation estimation output layers and initialized our model with the corresponding weights learned on the ModelNet10 dataset, an then tuned the network on ModelNet40 with a low learning rate to take advantage of both the orientation estimation and subvolume category prediction tasks. Compared to the one trained on ModelNet40 from scratch, experimental result shows higher recognition accuracy (as shown in Table 1) and faster convergence speed can be achieved by the fine-tuning method. This indicates that our model can learn powerful 3D representations. We then compared our model with the state-of-the-art 3D object recognition models on the ModelNet10 and ModelNet40 datasets, as shown in Table 2.

4.1.1. Results on ModelNet10

As shown in Table 2, our model outperforms most of the existing 3D volumetric networks and all of the multi-view networks on the ModelNet10 dataset, with an accuracy of 93.39% being achieved.

We first compare our LightNet with multi-view based network models. Although these models benefit from network ensemble and 2D network structure pretrained on the ImageNet 1K and ModelNet40 datasets, our single LightNet model still achieves higher recognition accuracy on the ModeNet10 dataset. It clearly shows the ability of our model to directly learn discriminative 3D features from 3D shapes.

We then compare our LightNet model with all the single volumetric network frameworks. It can be seen that our 3D CNN achieves comparable accuracy to the state-of-the-art algorithms, with a slight drop (i.e., of 0.41%) of accuracy as compared to ORION F2C. Note that, a basic and simple structure is used in our model to introduce orientation prediction task while a fine-to-coarse structure is used in ORION to achieve improved performance [HZ16]. Compared to the ORION Basic model, our proposed LightNet achieves almost the same recognition performance with much less parameters and higher computational efficiency. It can also be observed that our model improves VoxNet by relative 17.4% with less than 67% training parameters. Compared to Voxception and VRN (which has 45 layers), our model obtains comparable or slightly better accuracy. It also shows that our LightNet achieves a large accuracy improvement as compared to PointNet.

We further compare our LightNet model with ensemble volumetric network frameworks. Compared to FusionNet, our model achieves a slightly higher accuracy (i.e., about 0.28%) and significantly reduces the number of parameters by two orders of magnitude. Specifically, FusionNet has about 118M parameters while our model has only about 0.3M parameters. Although VRN ensemble achieves the best accuracy result, it has a very deep and relatively complex structure (e.g., stochastic depth), making the network very slow to converge (about 6 days) [BLRW16] and difficult to train. In addition, its accuracy on the ModelNet10 dataset was achieved by the VRN ensemble framework pretrained on the ModelNet40 dataset. It is reported that the accuracy is 94.71% when the network is trained and tested on the ModelNet10 dataset, which is almost the same as ours. In contrast, our model has a more compact and shallower structure than VRN ensemble. Our model is a single network with 6 layers and about 0.3M parameters, while VRN en-

Table 2: *Category prediction results on the ModelNet dataset. Az stands for azimuth rotation and El stands for elevation rotation. '-' means that information is not provided for the corresponding item in the related paper.*

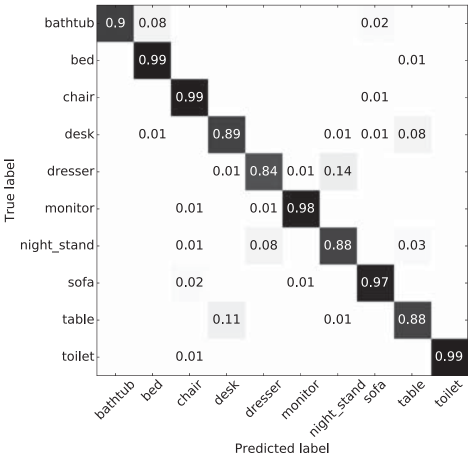
Type of Framework	Method	Pretrain	Size	Augmentation	Category Prediction Accuracy	
					ModelNet40	ModelNet10
Single, Volumetric	3DShapeNets [WSK*15]	ModelNet40	∼38M	Az×12	77.32%	83.54%
Ensemble, Volumetric	VRN Ensemble [BLRW16]	ModelNet40	∼90M	Az×24	**95.54%**	**97.14%**
Single, Volumetric	VRN [BLRW16]	ModelNet40	∼18M	Az×24	91.33%	93.61%
Single, Volumetric	Voxception [BLRW16]	ModelNet40	-	Az×24	90.56%	93.28%
Single, Volumetric	VoxNet [MS15]	-	∼0.92M	Az×12	83%	92%
Single, Volumetric	ORION Basic [SZB16]	-	VoxNet Based	Az×12	-	93.40%
Single, Volumetric	ORION F2C [SZB16]	-	VoxNet Based	Az×12	-	93.80%
Single, Volumetric	PointNet [GGGDGR*16]	-	∼80M	-	-	77.60%
Single, Volumetric	3D-NIN [QSN*16]	-	-	(Az, El)×60	86.10%	-
Single, Volumetric	Subvolume Sup. [QSN*16]	-	∼16M	(Az, El)×60	87.20%	-
Single, Volumetric	AniProbing [QSN*16]	-	-	(Az, El)×60	85.90%	-
Single, Volumetric	LightNet	ModelNet10	**∼0.30M**	Az×12	**86.90%**	**93.39%**
Ensemble, Vol.+Mul.	FusionNet [HZ16]	ImageNet 1K ModelNet40	∼118M	(Az, El)×60	90.80%	93.11%
Single, Multi-view	DeepPano [SBZB15]	-	-	-	82.54%	88.66%
Ensemble, Multi-view	MVCNN [SMKLM15]	ImageNet 1K ModelNet40	VGG-M Based	80 Views	90.10%	-
Ensemble, Multi-view	Pairwise Decomp. [JLD16]	ImageNet 1K ModelNet40	VGG-M Based	12 Views	90.70%	92.80%

semble is an ensemble of networks with 45 layers and 18M parameters. Consequently, our model is easier to train and more efficient for computing. All these results clearly demonstrates the effectiveness of our LightNet model. Furthermore, the convergence time for training our model is also significantly shorter then these models, as discussed in Sec. 4.3.

To further demonstrate the effectiveness of our model, a confusion matrix is shown in Fig. 4. It is clear that most objects can be correctly recognized. Note that, top false positives occur at 1) *dresser* → *nightstand* (14%), 2) *table* → *desk* (11%), 3) *nightstand* → *dresser* (8%), 4) *desk* → *table* (8%). That is mainly because these objects are highly similar in shapes (see Fig. 5) and even cannot be visually distinguished by humans.

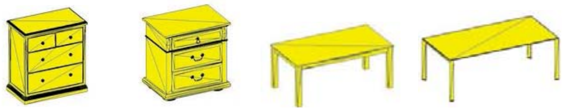
4.1.2. Results on ModelNet40

It can be observed from Table 2 that our model obtains a compelling accuracy of 86.93%, which improves VoxNet by relative 23.1%. This result also shows that our model is able to learn effective representations for 3D objects, which can be well generalized to 3D objects of other categories. Compared to the Subvolume Supervision model (with an accuracy of 87.2%), we obtain almost the same category prediction accuracy but considerably reduce structure complexity. That is because the Subvolume Supervision model include several additional time-consuming 3D convolutional layers and operations. Further, we observed in our experiments that 3D NIN and Subvolume Supervision models severely overfit the ModelNet10 dataset during training and work poorly on the ModelNet10 test dataset. This indicates that those models have too many parameters to train and cannot handle small scale datasets. However, our model can maintain good performance on both ModelNet10 and ModelNet40 datasets.



True label \ Predicted label	bathtub	bed	chair	desk	dresser	monitor	night_stand	sofa	table	toilet
bathtub	0.9	0.08						0.02		
bed		0.99							0.01	
chair			0.99					0.01		
desk		0.01		0.89			0.01	0.01	0.08	
dresser				0.01	0.84	0.01	0.14			
monitor			0.01		0.01	0.98				
night_stand			0.01		0.08		0.88		0.03	
sofa			0.02			0.01		0.97		
table				0.11			0.01		0.88	
toilet			0.01							0.99

Figure 4: *A confusion matrix of category prediction achieved on the ModelNet 10 dataset.*



(a) *Dresser* **(b)** *Night stand* **(c)** *Table* **(d)** *Desk*

Figure 5: *Examples of false category prediction results. In our experiments, 12 dressers were classified as night stands, and 11 tables were classified as desks. Each pair of objects are highly similar in shapes.*

4.2. Experiments on the Sydney Urban Objects Dataset

The Sydney Urban Objects dataset contains 631 scans of common urban road objects in 26 categories acquired by a Velodyne LIDAR [DDQHD13]. The commonly used data for object recognition contains 588 objects from 14 categories, including four-wheel drive (21), wall (20), wall or building (20), bus (16), car (88), person (152), pillar (20), pole (21), traffic lights (47), traffic sign (51), tree (34), truck (12), trunk (55), ute (16), and van (35) [DDQHD13]. Since point clouds in this dataset were collected in real-world urban scenes with significant variations in viewpoints and occlusion, recognizing these severely occluded objects is a highly challenging task, as illustrated in Fig. 6.

Figure 6: *Examples of point clouds in the Sydney Urban Objects dataset.*

Since our model works on volumetric data, we first transformed the point clouds to voxels. The resolution of the object occupancy grid was set to $24 \times 24 \times 24$ and then padded to the size of $32 \times 32 \times 32$. We also augmented the volumetric data by rotating each point cloud by 12 times along the z-axis. In contrast, 18 rotations along the z-axis are used in [MS15] and various number of rotations are used for different categories in [SZB16].

We divided the Sydney Urban Objects dataset into 4 folds for training and testing, the same as in [DDQHD13]. During training, we used three folds of the dataset to train our model and the rest data to test the recognition performance. Same as [MS15, SZB16, DDQHD13], we used the F_1 score weighted by class support as our performance metric to consider the unbalanced data distribution of the Sydney Urban Objects dataset. The average weighted F_1 score over the four folds is shown in Table 3.

Table 3 shows that our model outperforms VoxNet and the methods proposed in [DDQHD13, CDLS14]. In [DDQHD13, CDLS14], efficient 3D shape features are first learned and then a SVM is used to classify objects. Compared to these two methods, our model can scale well to large datasets with better real-time performance. As shown in Table 3, the ORION Fusion structure achieves the best result for this task. However, our training scheme and network structure is concise and easy to train for its smaller number of weights, achieving comparable performance to the ORION Basic model.

4.3. Computational Time

The training and recognition time of our model is shown in Table 4. In our experiments, our model can classify a 3D voxelized shape within 5ms. This clearly shows that our model is suitable for real-time object recognition applications for its simple framework and small number of parameters. The training time is also shorter than other models. Note that, the timing results are related to hardwares. It is also observed that the recognition time of our model on ModelNet40 is less than that on ModelNet10, that is because our model used for ModelNet40 does not contain the orientation prediction task, while our model for ModelNet10 is complete and more time-consuming.

Table 3: *Comparison of category prediction performance on the Sydney Urban Objects dataset.*

Method	Average F_1 score
UFL+SVM [DDQHD13]	0.67
GFH+SVM [CDLS14]	0.71
VoxNet [MS15]	0.72
ORION Basic [SZB16]	0.767
ORION Fusion [SZB16]	**0.778**
LightNet	**0.76**

5. Conclusion

In this paper, we proposed a lightweight 3D CNN for real-time 3D object recognition with small number of training parameters. We effectively learn 3D representations using multitask learning, including category and orientation prediction from both entire and partial shapes. Our LightNet model achieves nearly the state-of-the-art recognition accuracy on the ModelNet and Sydney Urban Objects datasets. Extensive experiments have been conducted to show the superior recognition accuracy and computation efficiency of our LightNet model. In the future, we plan to integrate our LightNet model into a real robot vision system, and extend it to other real-time tasks such as 3D object detection and segmentation.

References

[Ale16] Alexandre L. A.: 3D Object Recognition Using Convolutional Neural Networks with Transfer Learning Between Input Channels. In *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 889–898. 2

[BA10] Bayramoglu N., Alatan A. A.: Shape Index SIFT: Range Image Recognition Using Local Features. 352–355. 2

[BLRW16] Brock A., Lim T., Ritchie J., Weston N.: Generative and Discriminative Voxel Modeling with Convolutional Neural Networks. *arXiv preprint arXiv:1608.04236* (2016). 1, 2, 5, 6, 8

[Car98] Caruana R.: Multitask Learning. In *Learning to learn*. Springer, 1998, pp. 95–133. 2

[CDLS14] Chen T., Dai B., Liu D., Song J.: Performance of Global Descriptors for Velodyne-Based Urban Object Recognition. In *IEEE Intelligent Vehicles Symposium Proceedings* (2014), IEEE, pp. 667–673. 7

[CFG*15] Chang A. X., Funkhouser T., Guibas L., Hanrahan P., Huang Q., Li Z., Savarese S., Savva M., Song S., Su H., et al.: Shapenet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012* (2015). 1

[DDQHD13] De Deuge M., Quadros A., Hung C., Douillard B.: Unsupervised Feature Learning for Classification of Outdoor 3D Scans. In *Australasian Conference on Robitics and Automation* (2013), vol. 2. 1, 5, 7

[FXD*15] Fang Y., Xie J., Dai G., Wang M., Zhu F., Xu T., Wong E. K.: 3D Deep Shape Descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition* (2015). 1

[GBS*14] Guo Y., Bennamoun M., Sohel F., Lu M., Wan J.: 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 11 (2014), 2270–2287. 2

Table 4: *A comparison of computational time on the ModelNet dataset.*

Methods	Training Time	Recognition Time	GPU Device
VoxNet [MS15]	6-12 hours	6 ms	K40
VRN Ensemble [BLRW16]	6 days	-	Titan X
LightNet	ModelNet10: **3.4 hours** ModelNet40: **6.7 hours**	**4.8 ms** **3.5 ms**	GTX 1080

[GGGDGR*16] GARCIA-GARCIA A., GOMEZ-DONOSO F., GARCIA-RODRIGUEZ J., ORTS-ESCOLANO S., CAZORLA M., AZORIN-LOPEZ J.: PointNet: A 3D Convolutional Neural Network for Real-Time Object Class Recognition. In *International Joint Conference on Neural Networks* (2016), IEEE, pp. 1578–1584. 1, 2, 6

[GSB*13] GUO Y., SOHEL F., BENNAMOUN M., LU M., WAN J.: Rotational Projection Statistics for 3D Local Surface Description and Object Recognition. *International Journal of Computer Vision 105*, 1 (2013), 63–86. 2

[HLH*16] HAN Z., LIU Z., HAN J., VONG C., BU S., LI X.: Unsupervised 3D Local Feature Learning by Circle Convolutional Restricted Boltzmann Machine. *IEEE Transactions on Image Processing* (2016). 1

[HZ16] HEGDE V., ZADEH R.: FusionNet: 3D Object Classification Using Multiple Data Representations. *arXiv preprint arXiv:1607.05695* (2016). 2, 5, 6

[HZRS15] HE K., ZHANG X., REN S., SUN J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *The IEEE International Conference on Computer Vision* (2015). 4

[JH98] JOHNSON A. E., HEBERT M.: Surface Matching for Object Recognition in Complex 3-Dimensional Scenes. *Image and Vision Computing 16*, 9 (1998), 635–651. 2

[JLD16] JOHNS E., LEUTENEGGER S., DAVISON A. J.: Pairwise Decomposition of Image Sequences for Active Multi-View Recognition. *arXiv preprint arXiv:1605.08359* (2016). 2, 6

[KPVG10] KNOPP J., PRASAD M., VAN GOOL L.: Orientation Invariant 3D Object Classification Using Hough Transform Based Methods. In *Proceedings of the ACM workshop on 3D object retrieval* (2010), ACM, pp. 15–20. 2

[KPW*10] KNOPP J., PRASAD M., WILLEMS G., TIMOFTE R., VAN GOOL L.: Hough Transform and 3D SURF for Robust Three Dimensional Classification. In *European Conference on Computer Vision* (2010), Springer, pp. 589–602. 2

[KTS*14] KARPATHY A., TODERICI G., SHETTY S., LEUNG T., SUKTHANKAR R., FEI-FEI L.: Large-Scale Video Classification with Convolutional Neural Networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2014), pp. 1725–1732. 2

[LBH15] LECUN Y., BENGIO Y., HINTON G.: Deep Learning. *Nature 521*, 7553 (2015), 436–444. 2

[LCY13] LIN M., CHEN Q., YAN S.: Network In Network. *arXiv preprint arXiv:1312.4400* (2013). 2

[LS09] LO T.-W. R., SIEBERT J. P.: Local Feature Extraction and Matching on Range Images: 2.5D SIFT. *Computer Vision and Image Understanding 113*, 12 (2009), 1235–1250. 2

[MHN13] MAAS A. L., HANNUN A. Y., NG A. Y.: Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing* (2013), vol. 30. 3

[MS15] MATURANA D., SCHERER S.: VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2015), IEEE, pp. 922–928. 1, 2, 3, 4, 5, 6, 7, 8

[OFCD02] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D. P.: Shape Distributions. *ACM Transactions on Graphics 21*, 4 (2002), 807. 2

[QSN*16] QI C. R., SU H., NIESSNER M., DAI A., YAN M., GUIBAS L. J.: Volumetric and Multi-View CNNs for Object Classification on 3D Data. *arXiv preprint arXiv:1604.03265* (2016). 1, 2, 3, 5, 6

[RBB09] RUSU R. B., BLODOW N., BEETZ M.: Fast Point Feature Histograms (FPFH) for 3D Registration. In *IEEE International Conference on Robotics and Automation* (2009), pp. 1848–1853. 2

[RBTH10] RUSU R. B., BRADSKI G., THIBAUX R., HSU J.: Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), pp. 2155–2162. 2

[SBZB15] SHI B., BAI S., ZHOU Z., BAI X.: DeepPano: Deep Panoramic Representation for 3-D Shape Recognition. *IEEE Signal Processing Letters 22*, 12 (2015), 2339–2343. 1, 2, 5, 6

[SHB*12] SOCHER R., HUVAL B., BATH B., MANNING C. D., NG A. Y.: Convolutional-Recursive Deep Learning for 3D Object Classification. In *Advances in Neural Information Processing Systems* (2012), pp. 665–673. 2

[SHK*14] SRIVASTAVA N., HINTON G. E., KRIZHEVSKY A., SUTSKEVER I., SALAKHUTDINOV R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research 15*, 1 (2014), 1929–1958. 4

[SMDH13] SUTSKEVER I., MARTENS J., DAHL G., HINTON G.: On the Importance of Initialization and Momentum in Deep Learning. In *Proceedings of The 30th International Conference on Machine Learning* (2013), pp. 1139–1147. 4

[SMKLM15] 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* (2015), pp. 945–953. 1, 2, 6

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L. J.: A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Computer Graphics Forum 28*, 5 (2009), 1383–1392. 2

[SZB16] SEDAGHAT N., ZOLFAGHARI M., BROX T.: Orientation-Boosted Voxel Nets for 3D Object Recognition. *arXiv preprint arXiv:1604.03351* (2016). 1, 2, 3, 6, 7

[Thr03] THRUN S.: Learning Occupancy Grid Maps with Forward Sensor Models. *Autonomous Robots 15*, 2 (2003), 111–127. 3

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1912–1920. 1, 2, 5, 6

[WZX*16] WU J., ZHANG C., XUE T., FREEMAN B., TENENBAUM J.: Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *Advances in Neural Information Processing Systems* (2016), pp. 82–90. 2

[XSZ*16] XU K., SHI Y., ZHENG L., ZHANG J., LIU M., HUANG H., SU H., COHEN-OR D., CHEN B.: 3D Attention-Driven Depth Acquisition for Object Identification. *ACM Transactions on Graphics 35*, 6 (2016), 238. 1

[YWZ*15] YE H., WU Z., ZHAO R.-W., WANG X., JIANG Y.-G., XUE X.: Evaluating Two-Stream CNN for Video Classification. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (2015), ACM, pp. 435–442. 2