A NEW ROTATION-INVARIANT DEEP NETWORK FOR 3D OBJECT RECOGNITION

Yachi Zhang[1], *Zongqing Lu*[1,*], *Jing-Hao Xue*[2], *Qingmin Liao*[1]

[1]Graduate School at Shenzhen, Tsinghua University
[2]Department of Statistical Science, University College London
luzq@sz.tsinghua.edu.cn

ABSTRACT

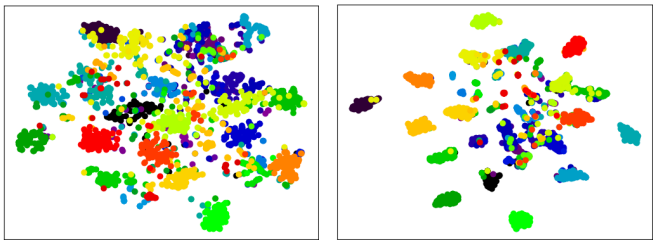
When inputs are rotated, most 3D convolutional neural networks (CNNs) will have their performance much dropped, especially for those models with voxelized input of 3D objects. The newly proposed Spherical CNNS, with the concept of the rotation-equivariant spherical correlation, aims to achieve rotation invariance. Inspired by this, we propose a new rotation-invariant deep network to recognize rotated 3D objects. Specifically, we adopt the spherical representation and the spherical correlation S^2 layer of Spherical CNNs, for their capacity of representing 3D objects and rotation equivariance. In the meantime, we improve the computational efficiency and expressiveness of Spherical CNNs, by replacing its time-consuming and depth-limited SO(3) layer with a PointNet-style network architecture. Hence our proposed network can maintain the equivariance as the network grows deeper while substantially reducing its runtime, leading to a much better efficiency and expressiveness of rotation-invariant representation. Experimental results show that our network performs better than or comparable to the state-of-the-art methods in the ModelNet40 classification challenge.

Index Terms— Deep learning, Object recognition, Rotation invariant, 3D representation

1. INTRODUCTION

Expressive descriptors of 3D objects are important to many valuable practical problems in computer vision and graphics, such as scene understanding and surface reconstruction. With the rise of commodity depth sensors and the rapid improvements in 3D modeling tools, we can now readily capture and model a huge amount of 3D data. The increasing availability of 3D data makes it possible to learn expressive descriptors of 3D objects by using deep networks such as convolutional neural networks (CNNs).

There are many types of 3D data available, such as volumetric data, views of 3D objects, and point clouds. For some specific types of 3D data, recently many inspiring deep network-based methods have been proposed. For instances, methods in [1–5] address the volumetric data; multi-view CNNs [6, 7] are popular for their direct use of 2D CNNs pre-trained by large 2D image datasets; also, some methods [8, 9] work on the point clouds directly. These methods usually perform well on aligned 3D objects. However, when it comes to rotated data, their performance drops substantially. Most CNNs are known sensitive to rotation, sometimes would make the outputs of a 3D object with different orientations look unrelated.



(a) Spherical CNNs (b) Our network

Fig. 1. The t-SNE embedding features [10] of models in ModelNet40 based on their features generated by (a) Spherical CNNs and (b) our network. We can observe that our network is more expressive: the intra-class features are more compact and the inter-class distances are larger.

This paper aims to learn rotation-invariant representations of 3D rotated objects, by proposing a new network architecture to better handle rotations. Our work is inspired by, and aims to improve, the attractive work (Spherical CNNs) in [11] of the spherical correlation, which is equivariant to rotations. More specifically, in [11], there are two types of correlation, the spherical correlation and the rotation group correlation, implemented in the so-called S^2 layer and the higher SO(3) layer, respectively. However, there remain two issues with Spherical CNNs: One is that its SO(3) layer is time-consuming and depth-limited, and the other is that, more undesirable, its feature descriptor of 3D objects is suboptimal in terms of expressiveness (see Fig. 1(a)), as the equivariance cannot be held as the number of SO(3) layers increases.

Hence we propose a new network by preserving the strength of Spherical CNNs while replace its SO(3) layers with another more expressive network architecture. Specifically, we adopt the spherical representation and the S^2 layer of Spherical CNNs, for their capacity of representing 3D ob-

jects and rotation equivariance. In the meantime, we improve the computational efficiency and expressiveness of Spherical CNNs (see Fig. 1(b)), by replacing its SO(3) layer with a network architecture of PointNet-style [8,9]. PointNet and PointNet++ in [8, 9] have proved able to express well aligned 3D objects, and can be deeper and more expressive than Spherical CNNs.

Our network performs like a multi-view network for a large number of views of 3D objects, and thus intuitively can be more expressive and invariant to rotations than the multi-view CNNs of [6, 7]. To show that the descriptors learned by our network are nearly invariant to the arbitrary rotations of the input, as well as to compare our network with other methods, we conduct experiments on the ModelNet40 shape classification challenge. Our network performs better than or on a par with the state of the art.

To summarize, the contribution of this paper is that we propose a novel rotation-invariant deep network for 3D object classification. Our network can improve the state-of-the-art Spherical CNNs, through preserving its strength in equivariance to rotations and enhance its efficiency and expressiveness by incorporating the strength of PointNets.

The rest of the paper is organized as follows: We start with a review of related work in Section 2, then present details of the proposed network architecture in Section 3. Last, we perform and analyze experiments on the ModelNet40 classification challenge in Section 4 and draw conclusions and suggest future work in Section 5.

2. RELATED WORK

Due to the successes of CNNs on 2D images, researchers try to use CNNs to address 3D images. Recently, many inspiring and novel CNN-based approaches have been proposed. These methods can be roughly categorized into three groups, according to the types of their inputs: volumetric data, 2D views of 3D objects, or directly 3D point clouds.

An early work developed on the basis of volumetric data is the 3D ShapeNets [1]. 3D ShapeNets directly extends the 2D convolution to 3D, which needs to transform the point cloud to a uniform voxel grid. However, the volumetric representation is in low resolution, which makes the computation expensive to achieve better, higher-resolution performance. Many other voxel-based methods [2–5] attempted to overcome the voxel grid resolution issue to improve performance.

Two-dimensional view-based approaches utilize the 2D convolution to solve 3D problems. These methods tried to benefit from the success of 2D CNNs on 2D images. For 2D images, large image datasets such as ImageNet [12] are available, which allow CNNs to learn features that are general for different tasks. Hence these methods can pre-train their networks by using those large datasets, and many of them actually perform better than the voxel-based methods. For example, in multi-view CNN (MVCNN) [6], multi-view images of a 3D shape are captured with different virtual cameras from fixed view points. Each view are fed into weight-shared CNNs and a view pooling layer is applied. However, to get a better performance, MVCNN needs to render more views of a 3D object, which inevitably incurs more computation. For a comprehensive comparison between the voxel-based CNNs and muti-view CNNs for 3D object classification, see Qi *et al*. [13]. Another work based on 2D images [14] learned representations from the depth map of a 3D shape.

However, the above-mentioned voxel-based and 2D view-based approaches generally fail to address the rotation of 3D objects. Therefore, our work aims to develop a method to extract the rotation-invariant representations of 3D objects and perform better.

Unlike the above-mentioned methods which convert the point cloud to a voxel grid or project a 3D object into different views, some researches directly process unordered point sets. These methods can achieve better performance in object classification, compared with those methods which address volumetric data. A typical method in this category is PointNet [8] and PointNet++ [9]. PointNet [8] is designed for deep learning on raw 3D point sets. This network can address the invariance to permutations and transformations of the input points. It is then extended to PointNet++ [9], which improves the PointNet by handling the variations in point density. PointNet and PointNet++ can achieve good performance on the aligned version of the ModelNet40 benchmark. However, these methods suffer a sharp drop in performance when arbitrary rotations are present, because the input is not aligned to a canonical space. To tackle this issue, we may consider applying a PointNet-style architecture, but using a different representation of data as input such that no alignment is needed to be done in advance.

CNNs work well on translation but not on rotation, hence some new methods are proposed to process rotations. Group equivariant convolutional networks (G-CNNs) was proposed in [15], which proved the equivariance of group-convolutions and preservation of rotational equivariance, and achieved good performance on the CIFAR10 dataset and a rotated MNIST dataset. Inspired by G-CNNs, Worral *et al*. [16] learned interpretable transformations with encoder-decoder networks. Their latest work CubeNet [17] used a 3D rotation equivariant CNN for voxel representations. They introduced a group convolutional network with linear equivariance to translations and right angle rotations in three dimensions. Their work achieved the state-of-the-art performance on the ModelNet10 classification challenge. Recently, Cohen *et al.* [11] proposed an attractive novel network, called Spherical CNNs, which is to address spherical signals. They proposed a concept of the rotation-equivariant spherical correlation. They used the fast Fourier transform (FFT) to compute the spherical correlation efficiently.

3. PROPOSED NETWORK

The diagram of the proposed network is shown in Fig. 2; we present each of its constituents in detail in following sections.

3.1. Spherical Representation

To exploit the rotation equivariance that the spherical correlation [11] brings, we utilize the S^2 layer of Spherical CNNs as our first layer. The S^2 layer needs spherical images as input, so we first transform 3D images into spherical representations. To achieve this, we project the 3D images onto an enclosing unit sphere in the way that [11] does. The ray is cast from the surface towards the origin, and the value of the signal is decided by the first intersection of the ray with the model; Fig. 3 illustrates the transform. We collect 6 channels and set bandwidth to 64. For more details, please refer to [11].

3.2. Equivariant Layer

Many researches attempted to exploit the equivariance of convolutional networks [15]. The equivariance can be simply defined as

$$\Phi \circ L_R = T_R \circ \Phi, \tag{1}$$

where T_R and L_R are two operators not necessarily the same, and a layer Φ is said to be equivariant if some operator T_R satisfies the above equation.

CNNs are equivariant to translation, but fail in address rotation. Group equivariant convolutional networks [15] tried to deal with this problem, and CubeNet [17] used group convolution to address the equivariance on volumetric data. However, CubeNet is discretized and implemented with Z_4-convolution, which represents only 4 rotations. So CubeNet is less expressive. The Spherical CNNs [11] considers more rotations. It defines the spherical correlation and the rotation group correlation. We choose the spherical correlation layer (the so-called S^2 layer) as our equivariant layer, and the spherical correlation between two functions f and ψ is defined as

$$[\psi * f](R) = \int_{S^2} \sum_{k=1}^{K} \psi_k \left(R^{-1} x\right) f_k(x)\, dx, \tag{2}$$

where f and ψ are functions defined on sphere; R denotes the rotation; and K is the number of channels of functions and S^2 is the unit sphere, which can be defined as the set of points $x \in \mathbb{R}^3$ and parameterized by the spherical coordinates $\alpha \in [0, 2\pi]$ and $\beta \in [0, \pi]$.

With the reason that the output of spherical correlation is a signal on SO(3), [11] also proposed the SO(3) group correlation in the higher layers after the S^2 layer. As mentioned in [11], the equivariance error grows with the resolution and the number layers for the discretized version. This limits the depth and the expressiveness of the network. In addition, when using the SO(3) group correlation, the runtime increases greatly as the network becomes deeper; this again limits the depth of the network. Therefore, we propose to replace the SO(3) layers with PointNet-style feature extractor layers to be described in section 3.3, which can lead to a deeper, more expressive, and more efficient network with only 1/4 of the original runtime.

The output of the equivariant S^2 layer is then indexed by rotation on SO(3) and fed into a PointNet-style part for feature extraction.

3.3. Feature Extractor Layers

The output feature map of the S^2 layer is fed into the feature extractor layers, for which we choose to develop a PointNet-style network.

As discussed before, the PointNet directly uses point clouds as input, and add two joint alignment networks to align all input sets and features to a canonical space before feature extraction. They believe that these joint alignment networks can make the learned representation invariant to certain geometric transformations. These networks work well on the point clouds rotated around the vertical axis, but when point clouds are rotated randomly in every direction, their performances drop.

We investigate further the joint alignment networks, and find that the joint alignment networks mainly project the point cloud to a canonical space to increase the association of points globally, which is beneficial even though we are using the feature maps indexed by rotations. Therefore, we adopt a joint alignment net to link between two MLP networks (see Fig. 2). This joint net itself contains a shared MLP(64,128,1024) for each rotation, a max pooling layer, and a fully connected MLP(512,256); batchnorm is used for all of its layers with ReLU except for its last layer. In addition, we add a regularization term to the softmax training loss, to ensure the feature alignment matrix close to be orthogonal:

$$L_{reg} \quad = \quad \|I - AA^T\|_F^2, \tag{3}$$

where A denotes the feature alignment matrix.

4. EXPERIMENTAL RESULTS

4.1. Data, Data Augmentation and Training Settings

We evaluate our model one the ModelNet40 benchmark for 3D shape classification [1]. The ModelNet40 dataset contains 12311 CAD models from 40 man-made object categories, which are split into 9843 for training and 2468 for testing.

We augment the training and testing models in the dataset by rotating the point clouds randomly in every direction.

During training, we use the stochastic gradient descent optimizer with an initial learning rate of 0.01 for 300 epochs.

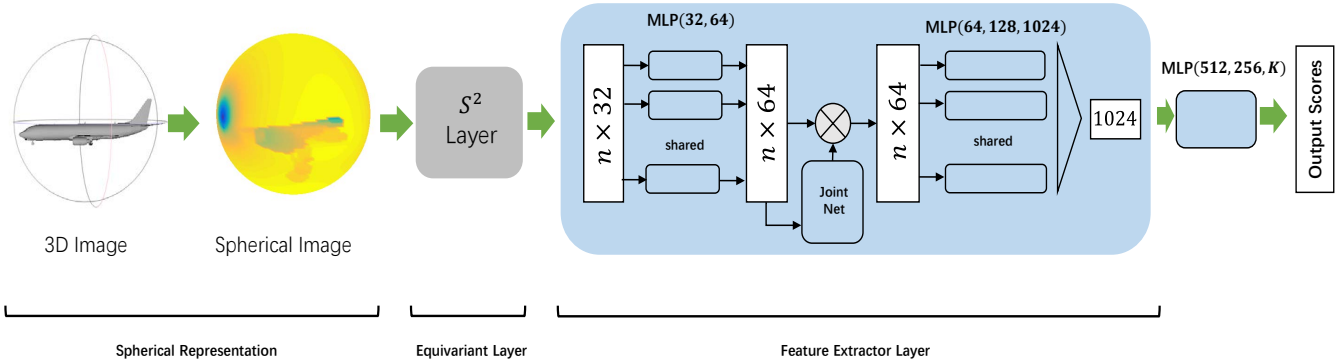


Fig. 2. Diagram of the proposed network. Firstly, we transform 3D images into spherical images (Section 3.1). Then we feed the spherical images into the S^2 layer (Section 3.2), the output of which then enters into two MLP networks aggregated through a feature joint alignment net (Section 3.3), in which n is the number of rotations. Finally after max pooling, another MLP of three fully connected layers are to produce the output.

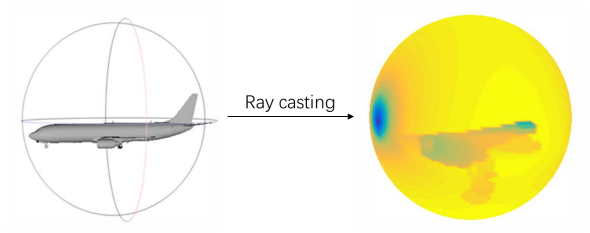


Fig. 3. Generation of spherical images: The ray is cast from the surface of the unit sphere towards the origin. The value of the signal is decided by the first intersection with the model.

We set the momentum to 0.9 and the batchsize to 64. The learning rate is divided by 10 for every 100 epochs. Batchnorm is used for all layers with ReLU except for the last layer. Dropout layers with probability of 0.5 are used in the fully connected layers. The weight of the regularization term in the softmax training loss is set to 0.01.

We implement our training on Geforce GTX1080 Ti, and the forward-pass and backward-pass take around 0.2 seconds, while the Spherical CNN with the SO(3) layers takes about 0.8 seconds.

4.2. 3D Objects Classification

We choose some state-of-the-art methods for the evaluation and comparison of our proposed method. These competing methods are PointNet [8], PointNet++ [9], Spherical CNNs [11], Voxnet [2], RotationNet 20X [7], Esteves *et al*. [18], MVCNN 12X [6] and MVCNN 80X [6]. They are tested by using the default settings of their published code. For Spherical CNNs, it consists of an initial S^2 conv-BN-ReLU block followed by two SO(3) conv-BN-ReLU blocks. For all methods, the training and test 3D objects are augmented in the way mentioned above.

Table 1 lists the classification accuracy of these methods on the ModelNet40 benchmark dataset.

Table 1. Classification accuracy per instance for the ModelNet40 benchmark. The results of VoxNet, RotationNet, MVCNN 12X, MVCNN 80X and Esteves *et al*. are taken from [18]. Top two results are highlighted in bold.

Method	Accuracy
VoxNet [2]	0.745
RotationNet 20X [7]	0.802
MVCNN 12X [6]	0.776
MVCNN 80X [6]	0.861
PointNet [8]	0.841
PointNet++ [9]	0.852
Spherical CNNs [11]	0.843
Esteves *et al*. [18]	**0.869**
Ours	0.867

From Table 1, we can observe the followings.

Firstly, our method performs better than most of the compared methods, and is very comparable to the best performer (Esteves *et al*. [18]). Nevertheless, we note that our network is twice faster than the method of Esteves *et al*.

Secondly, the method VoxNet based on volumetric data [2] performs the worst.

Thirdly, the multi-view methods RotationNet 20X [7] and MVCNN 12X [6] perform better than VoxNet, and they may get an even better performance by increasing the number of views (MVCNN 80X), because different views could be regarded as a rotation group, which means more views would represent more rotations. Our proposed network could be re-

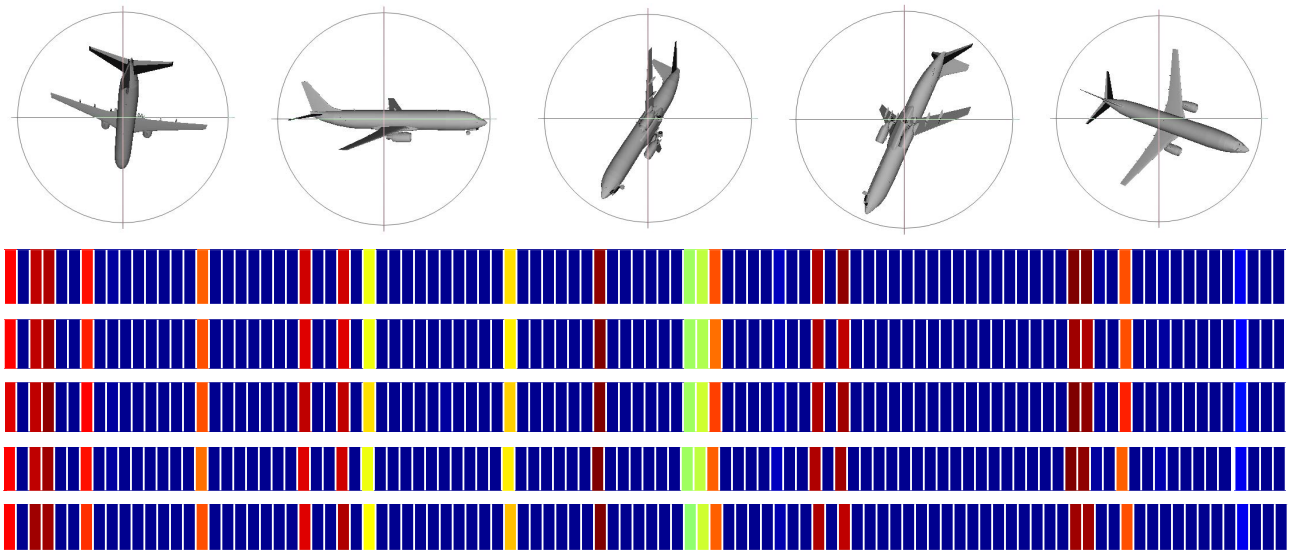


Fig. 4. Top row: five arbitrary rotations of the 3D objects. Five lower rows: descriptors corresponding to the five rotations in the top row. The descriptors learned by our network are nearly invariant to these rotations.

garded as a kind of muti-views network with a large number of views, so our network is more expressive than RotationNet and MVCNN, as the results indicate.

Finally, as discussed before, our method is actually a hybrid of PointNet [8] and Spherical CNNs [11] by combining their strengths, therefore we expect our method to outperforms the PointNet and the Spherical CNNs, which is indeed the case as shown in the table. The PointNet attempts to align the point set, but still uses CNNs to generate the feature transform. Because of the weakness of CNNs for rotations, it cannot work well for rotated objects (e.g. no better than MVCNN 80X here). The Spherical CNNs are equivariant to rotation, but its higher SO(3) layers are limited by the bandwidth. Without the ability to build a deeper architecture, it could not generate highly expressive representations.

In addition, for illustrative purposed, Fig. 4 shows some arbitrary rotations and their corresponding representations generated by our proposed network. It can be clearly observed that our representations are nearly invariant to those input rotations.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a novel rotation-invariant neural network architecture for 3D object classification, by exploiting the best of two recent and attractive networks, Spherical CNNs and PointNet. On the one hand, we adopt the spherical representation and the spherical correlation S^2 layer of Spherical CNNs, for their capacity of representing 3D objects and rotation equivariance. On the other hand, we improve the computational efficiency and expressiveness of Spherical CNNs, by replacing its time-consuming and depth-limited SO(3) layer with a PointNet-style network architecture. Hence our proposed network can maintain the rotation equivariance as the network grows deeper while run fast. Experimental results have shown that our proposed network can perform better than or comparable to the state-of-the-art methods.

In the future, because transforming 3D images to spherical images remains time-consuming in our implementation, it would be our future work to improve on this.

6. REFERENCES

[1] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

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

[3] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston, "Generative and discriminative voxel

modeling with convolutional neural networks," *arXiv preprint arXiv:1608.04236*, 2016.

[4] Yangyan Li, Soeren Pirk, Hao Su, Charles R Qi, and Leonidas J Guibas, "FPNN: Field probing neural networks for 3D data," in *Advances in Neural Information Processing Systems*, 2016, pp. 307–315.

[5] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Advances in Neural Information Processing Systems*, 2016, pp. 82–90.

[6] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[7] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida, "RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, vol. 1, no. 2, pp. 4, 2017.

[9] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[10] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[11] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling, "Spherical CNNs," *arXiv preprint arXiv:1801.10130*, 2018.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[13] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas 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*, 2016, pp. 5648–5656.

[14] Gil Elbaz, Tamar Avraham, and Anath Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 2472–2481.

[15] Taco Cohen and Max Welling, "Group equivariant convolutional networks," in *International conference on machine learning*, 2016, pp. 2990–2999.

[16] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow, "Interpretable transformations with encoder-decoder networks," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017, vol. 4.

[17] Daniel Worrall and Gabriel Brostow, "CubeNet: Equivariance to 3D rotation and translation," *arXiv preprint arXiv:1804.04458*, 2018.

[18] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis, "Learning SO(3) equivariant representations with spherical CNNs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 52–68.