3D Convolutional Neural Networks for Object Recognition

Saumya Didwania
Center for Data Science
New York University
sd4469@nyu.edu

Mars Wei-Lun Huang
Center for Data Science
New York University
wh2103@nyu.edu

Stephen Spivack
Center for Data Science
New York University
ss7726@nyu.edu

Abstract

Recently, there have been many advances within the computer vision community in the use of 3D convolutional neural networks (CNNs) for object recognition tasks, mainly focused around point-based, view-based, and volumetric-based methods. In this set of experiments, we selected two benchmark datasets–ModelNet40 and MosMedData–to compare the training times and accuracy using a selection of the most highly-cited models. We found that the view-based method performed best for ModelNet40 in terms of test accuracy, but that training time was slower and more memory-intensive compared to the point-based and volumetric-based methods.

1. Introduction

Since the inaugural ImageNet Challenge [7] in 2012, convolutional neural networks have been one of the most popular methods to solve computer vision tasks. While recently the focus has been to make 2D CNNs more resource efficient [2], there has been growing work in the 3D CNN community as well. Although applications such as facial recognition and augmented reality can be done with 2D images, video-based tasks are increasingly reliant on 3D CNNs to achieve better performance [1]. In this paper, we will implement and explore point-based, view-based, and volumetric-based methods using the ModelNet40 dataset [10], as well as view-based and volumetric-based CNNs using a subset of MosMedData [4].

2. Related Work

Currently, there are three common methods for 3D object recognition: point-based, view-based, and volumetric-based. Point-based methods directly model the cloud of points; view-based methods project a 3D object into multiple 2D views and then model each view to understand view or patch features; and volumetric-based methods quantize the 3D image into voxels and then implement 3D convolutions directly on those voxels.

2.1. Point Based

For point-based methods, we examined works such as PointNet [5] and PointNet++ [6]. The basic idea of PointNet is to learn spatial encodings of each point using non-linear mappings and then aggregating the point features to a global point cloud. While this model is robust to deformation and noise, as well as invariant to the order in which points are processed, it is not as proficient at detecting small, local image details.

PointNet++ extends the original version by combining features from multiple scales. Each level of each scale consists of sampling and grouping the inputs to partition the point cloud into overlapping local regions and then extracts features by learning a higher dimensional representation of the input region.

2.2. View Based

Multi-view CNNs were one of the earlier attempts at modeling multiple views using CNNs [8] while more recently, View-GCN's have become popular and achieve higher performance than MVCNNs [9]. These models represent a 3D object through multiple 2D images, allowing for feature learning via existing 2D architectures. The surface information of a 3D object can provide high-level features that are generated with multiple 2D views.

2.3. Vox Based

3D ShapeNets [10] were an early attempt for 3D shape analysis using 3D CAD models and set a benchmark for object classification. Depending on the representation, the structural and geometric properties can change, and VoxNet [3] was proposed as a way to exploit those properties for 3D CAD models and LiDAR point clouds. VoxNet uses a shallow 5-layer network inputting 32x32x32 voxel data.

3. Experimental Details

Our approach for this paper was to replicate the various models using benchmark datasets for the multiple methods and better understand the accuracy, training time, and resource efficiency for these models.

3.1. Datasets

For our experiments we focused on two datasets: ModelNet40 and MosMedData. ModelNet40 consists of 3D CAD models across 40 image classes of everyday objects such as airplanes, sofas, plants, etc. MosMedData consists of human lung computed tomography (CT) scans for patients both with and without COVID-19 related findings. We used a subset (n = 200) of MosMedData [11], with an equal partition (n = 100) per class.

3.2. Models

For ModelNet40, we trained the following models:

- Point-based: PointNet.
- View-based: MVCNN [8], View-GCN [9].
- Volumetric-based: VoxNet [3].

For MosMedData, we considered the following models:

- Volumetric-based: 3D CNN [11]. It is a simplified version of the 3D CNN model in [12].
- View-based: 2D CNN. We modified 3D CNN by considering image depth as the number of filters (Appendix A for details).

3.3. Experiments

We ran all models across multiple systems depending on the size of both the dataset and 3D CNN method. For the most part we used Google Colab's GPU instance to train the models due to processing efficiency and compatibility with CUDA for multiprocessing capabilities.

4. Results

As shown in Table 1, view-based models outperformed point-based and volumetric-based models in terms of test instance accuracy on ModelNet40. This may be because there are an abundance of publicly available 2D images and pre-trained models that can extract representative information as the backbone of view-based models. In the worst case, their performance may just reduce to the 2D case. However, view-based models suffered from longer training and inference time proportional to the number of views. For example, MVCNN was trained with 12 or 60 views, and View-GCN was configured with 12 or 20 views. A model with more views usually can attain higher accuracy since it can learn more shape information. Point-based and volumetric-based models trained much faster than the view-based ones (at least 5x per epoch on Google Colab) but had higher losses and lower accuracy on the test sets.

Method	Data	Model	Test Acc
Point-Based	ModelNet40	PointNet	88.00%
View-Based	ModelNet40	MVCNN	92.53%
View-Based	ModelNet40	View-GCN	98.30%
Volumetric-Based	ModelNet40	VoxNet	86.22%
Volumetric-Based	MosMedData	3DCNN	68.33%
View-Based	MosMedData	2DCNN	76.67%

Table 1. Summary of results.

For MosMedData, the view-based 2D CNN has higher accuracy than the 3D CNN.As shown in Figure 1, the training accuracy of the 3D CNN started fluctuating around 0.7 and the validation accuracy stalled at 50% during the first 7 epochs. By contrast, the line plots observed for the 2D CNN are quite different between training and validation. We can observe that the 2D CNN can lead to 100% training accuracy and its validation accuracy does not stall in the beginning. Since the 2D CNN does not model the hierarchical structure on the depth dimension and has fewer parameters, it may be easier to train. On the contrary, the 3D CNN contains too many parameters compared to the given amount of limited training data, and it is less likely to be well-trained.

5. Conclusion and Future Work

In this paper, we show that while the view-based methods had the lowest test loss and highest accuracy, they also took the longest to train. Depending on the domain and the type of input, each of the methods present their own benefits and use cases. The point-based methods have been found to train quickly and perform with moderate accuracy compared to the other two methods. The volumetric-based models such as VoxNet are more commonly used in medical imaging and can be directly applied to CT scans.

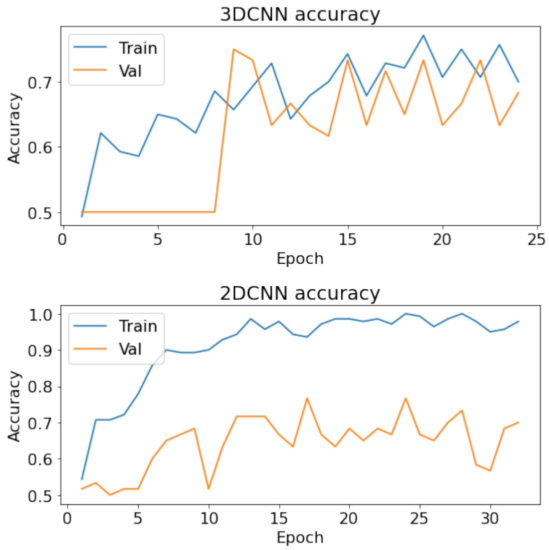

Figure 1. Training and validation losses/accuracies of 3DCNN on MosMedData with batch size 2.

Further research in 3D CNNs could explore training on larger datasets, specifically more medical imaging datasets, given their real-world importance. Since deep learning networks generally take a long time to train and perform better with more data, one could extend the medical imaging dataset–as we were only able to access a small subset–to a large enough sample size to notice finer patterns and possibly aid in prescribing treatment approaches. Given that 2D CNNs may outperform 3D CNNs with less data, one might also want to combine both architectures to extract meaningful features using 2D models so that we can train 3D CNNs more stably. Additionally, one could extend the findings of this paper to use video data to better predict how medical diagnostic scans change over time. With the vast amount of video data recorded in the medical fields, one could apply the model to show how scans change over time depending on the use of certain medications and see whether certain 3D CNNs have better performance on specific types of diagnoses.

References

[1] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1

[2] Okan Köpüklü, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. Resource efficient 3d convolutional neural networks. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1910–1919, 2019. 1

[3] Daniel Maturana and Sebastian A. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 1

[4] Sergey P. Morozov, Anna E. Andreychenko, Ivan A. Blokhin, Pavel B. Gelezhe, A. P. Gonchar, Alexander Nikolaev, Nikolay A. Pavlov, Valeria Yu. Chernina, and Victor A. Gombolevskiy. Mosmeddata: data set of 1110 chest ct scans performed during the covid-19 epidemic. 2020. 1

[5] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 1

[6] C. Qi, L. Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 1

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015. 1

[8] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, 2015. 1

[9] Xin Wei, Ruixuan Yu, and Jian Sun. View-gcn: View-based graph convolutional network for 3d shape analysis. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1847–1856, 2020. 1

[10] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. 1

[11] Hasib Zunair. 3d image classification. http://keras.io/examples/vision/3D_image_classification/, 2020. 1, 2

[12] Hasib Zunair, Aimon Rahman, Nabeel Mohammed, and Joseph Paul Cohen. Uniformizing techniques to process ct scans with 3d cnns for tuberculosis prediction. In *PRIME@MICCAI*, 2020. 2

A. 2DCNN Network Architecture

The 2D CNN takes an input with size 128x128x64. It consists of four convolutions with kernel size 3. The numbers of filters is 64, 128, 64, 32, respectively. Each convolution is followed by a max pooling with size 2 and batch normalization. The last batch normalization is followed by a flatten layer, a fully-connected layer with hidden size 512, a dropout layer with 0.3 dropout rate, and a fully-connected layer with hidden size 1.