CS596 - Project 6: 3D object recognition Report

Feng Zhang
dept. Computer Science
Bishop's University
Sherbrooke,Canada
FZHANG182@UBishops.ca

Tianye Zhao
dept. Computer Science
Bishop's University
Sherbrooke,Canada
TZHAO18@UBishops.ca

***Abstract*—The 3D recognition technology increasingly play a necessary role in our life. It has been a state-of-the-art hi-tech the scientists and technical staffs try to research and develop. In addition, the application of depth sensors, people can scan 3D models from the real world. Under this way, the demand for high quality 3D models will increase. Today, there are still rarely 3D models offered in public. In this project, we choose ShapeNet dataset to be trained, validated and tested. This track aims to provide a benchmark to evaluate large-scale shape retrieval based on the ShapeNet dataset. Meanwhile, we utilize algorithm of deep learning to calculate and validate the dataset with python3 . At last, we can recognize most of models utilized these algorithms and the dataset. In this way, we hope we can obtain much more satisfied result in the near future.**

Keywords—3D recognition technology,3D model, deep learning, ShapeNet

INTRODUCTION

The increasing availability of 3D models requires scalable and efficient algorithms to manage and analyze them, to facilitate applications such as virtual reality, 3D printing and manufacturing, and robotics among many others. The ShapeNet 3D dataset has provided a large of data for developers and researchers' studying and researching. We explore the dataset to train. In addition to category recognition, another natural sum. The challenging recognition task is shape completion: given a 2.5D depth map of an object from one point of view, what is it What's the possible 3D structure behind it? For example, humans do You don't need to see the legs of a table to know they're there And what they look like behind the visible On the surface. Similarly, even if we see a coffee cup From its side, we know it's empty inside In the middle, there's a handle on the side. And, the algorithm we use help us get much useful outcome. However, the part of our work still need to be improved upon later. We researched many materials from websites of stanford and MIT, which gave us lots of surprise ideas.

I. DATASET

We use the ShapeNetCore subset of ShapeNet which contains about 51,300 3D models over 55 common categories, each subdivided into several subcategories. Models are provided in OBJ format and two dataset versions are available: consistently aligned (regular dataset), and a more challenging dataset where models are perturbed by random rotations. Category and subcategory labels are provided for training and validation models as comma-separated files with a header row specifying the meaning of each column: modelId, synsetId (category label) and subsynetId (subcategory label).

II. PREPROCESSING

We use the a few methods to preprocess raw data for further deep learning.

A. Voxelization

In previous works for this dataset, researchers tend to generate 2D image sequences by taking projection of 3D objects. Instead of that, we decide to try voxelization which was brought up by professor. First read all the vertices in *.obj* file. Find out axis-aligned bounding box and the longest edge, try to represent the object in 32x32x32 cube by voxels. Save the voxels in 3D binary matrix. Each cell represents if there is a vertex located in this cell, which calculated by Euclidean distance. In the result matrix, we preserved the scale of the object.

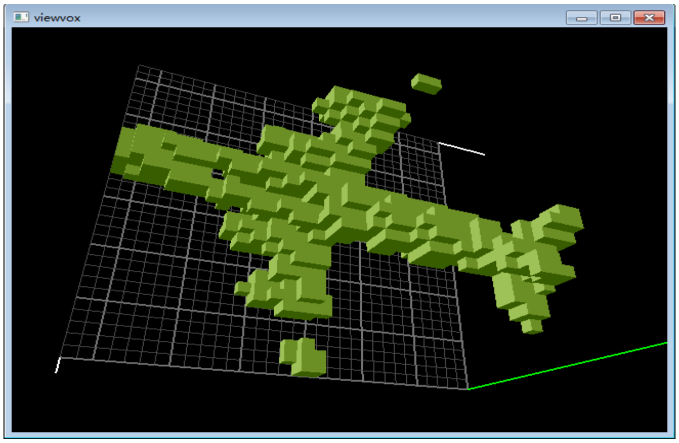

Figure 1: Airplane model_000003.obj

value=1 if E(voxel center, nearest vertex) < threshold

B. Numpy data

After we get 32x32x32 binary matrix for each object, we add one more dimension as channel before save the whole matrices sequences in *.npy* file which is efficient for further computation. Thus, the data in *X_train.npy* for feeding to our architecture is 5D with number of instances, number of channels (1 in this case), and 32x32x32 (User defined bin size for object voxelization). Moreover, we have to append the respective object number which is the folder in this case to each instance and use label encoder to transform the one-hot vector to indices. Save all the data in *y_train.npy. X_val.npy* and *y_val.npy* are generated in same way.

III. 3D CNN Architecture

The reason we choose bin size as 32 mostly because cifar10 dataset (32x32) has been used for algorithm test intensively. Thus, it would be much easier to modify existing 2D CNN architecture to be adapted for this project.

A. VGG3D

We adapt VGG architecture for this project as it is easy-understanding and providing pretty good results comparing to LeNet. In fact after several attempts, we can simple change all the 2D related layers to 3D and adjust a few parts, then it can work for our preprocessed dateset.

As we have noticed that this ShapeNet dataset is extremely imbalanced. It's better to consider to penalize class with more instances to make the architecture robust for prediction.

B. Summary

Due to time constraint, we couldn't work on the whole training dataset. Only around 25% training data were preprocessed. We noticed when we train for about 50 epoches, our model has achieved quite high accuracy. However, it performs around 50% accuracy in validation data. The reason caused this low accuracy score may be not sufficient training data for making prediction. Moreover, as we are using perturbed dataset which means all objects are in random orientation. Perhaps a better preprocessing method should be considered like using rotated-bounding-box to generate less sparse matrices or apply some distortion to generate result matrices.

Further Work

There are lots of work can be done in future. First, program GPU code for fast voxelization. Due to time and space constraint, we couldn't work the whole training dataset, and we cannot make it successful to write algorithm using GPU. In fact, we have tested Nearest Neighbor algorithm with Pytorch, once we want to produce result matrix at one time, it beyonds the GPU capacity what Google Colab can provide. If work for one voxel at one time, it costs more time comparing to current CPU code which utilize fast algorithm to search for nearest vertex. Secondly, as we have reduced bin size to 32 for efficiency. Researchers may increase this value to keep more features from objects. Last, there are more CNN architecture to be test.

Acknowledgment

First of all, I am very grateful to my mentor, PHD. Mohammed Ayoub Alaoui Mhamdi. He can help me to improve my understanding of deep learning. Under his careful guidance from this project, we can use some tools to design a simple neural network and implement verification. Thanks to my alma mater, who can give us an opportunity to follow Professors to enjoy a more advanced lifestyle, given me a rational look at the world, and more confidence to learn top tech around the world.

References

[1].HE K., ZHANG X., REN S., SUN J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR abs/1502.01852 (2015). 5

[2].IOFFE S., SZEGEDY C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR abs/1502.03167 (2015). 5

[3].KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25. Curran Associates, Inc., 2012, pp. 1097–1105. 5

[4].KUHN H. W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly 2 (1955), 83–97. 7

[5].LECUN Y., BOSER B., DENKER J. S., HENDERSON D., HOWARD R. E., HUBBARD W., JACKEL L. D.: Backpropagation applied to handwritten zip code recognition. Neural Computation 1, 4 (Dec. 1989), 541–551. 6

[6].LI B., LU Y., LI C., GODIL A., SCHRECK T., AONO M., BURTSCHER M., CHEN Q., CHOWDHURY N. K., FANG B., FU H., FURUYA T., LI H., LIU J., JOHAN H., KOSAKA R., KOYANAGI H., OHBUCHI R., TATSUMA A., WAN Y., ZHANG C., ZOU C.: A comparison of 3d shape retrieval methods based on a large-scale benchmark supporting multimodal queries. Computer Vision and Image Understanding 131 (2015), 1–27. 7

[7] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. Neural Computation, 14(8):1711–1800, 2002.

[8] T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In ICML. ACM, 2008.

[9] L. Younes. On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates, March 17 2000.

[10] Mark J. Huiskes and Michael S. Lew. The MIR Flickr retrieval evaluation. In MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval, New York, NY, USA, 2008. ACM.

[11] Zhirong Wu, et al. 3D ShapeNets: A Deep Representation for Volumetric Shapes. Retrieved from https://www.csail.mit.edu/.

[12] Angel X. Chang, et al. ShapeNet: An Information-Rich 3D Model Repository. arXiv:1512.03012, 2015.

[13] M. Savva, et al.(2016) SHREC'16 Track Large-Scale 3D Shape Retrieval from ShapeNet Core55. Retrieved from https://cs.stanford.edu/.