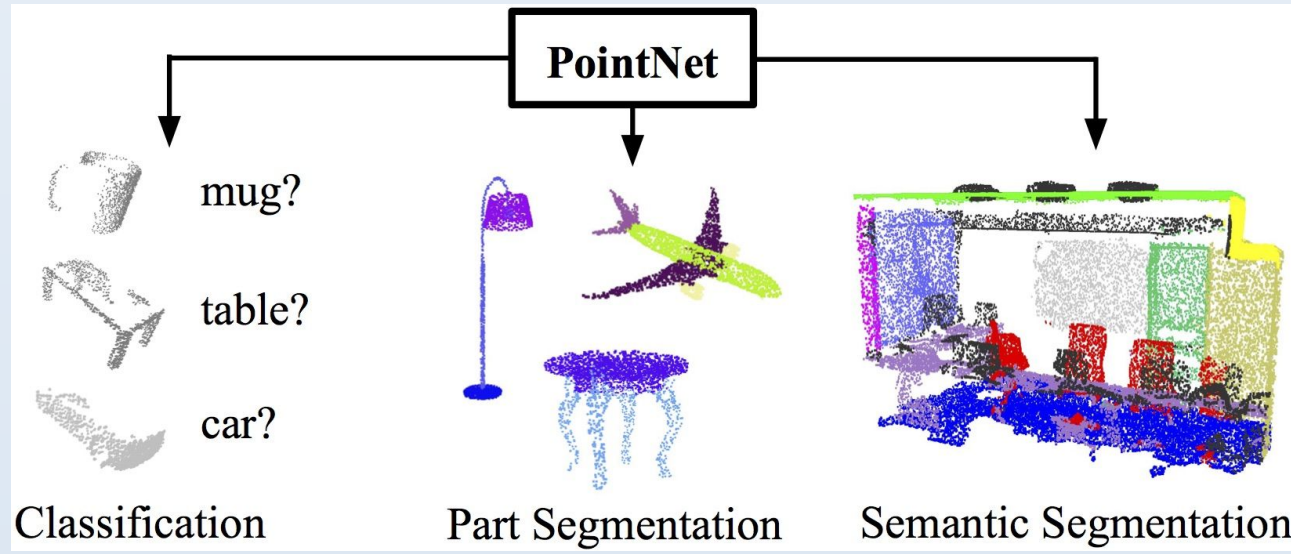


Predicting



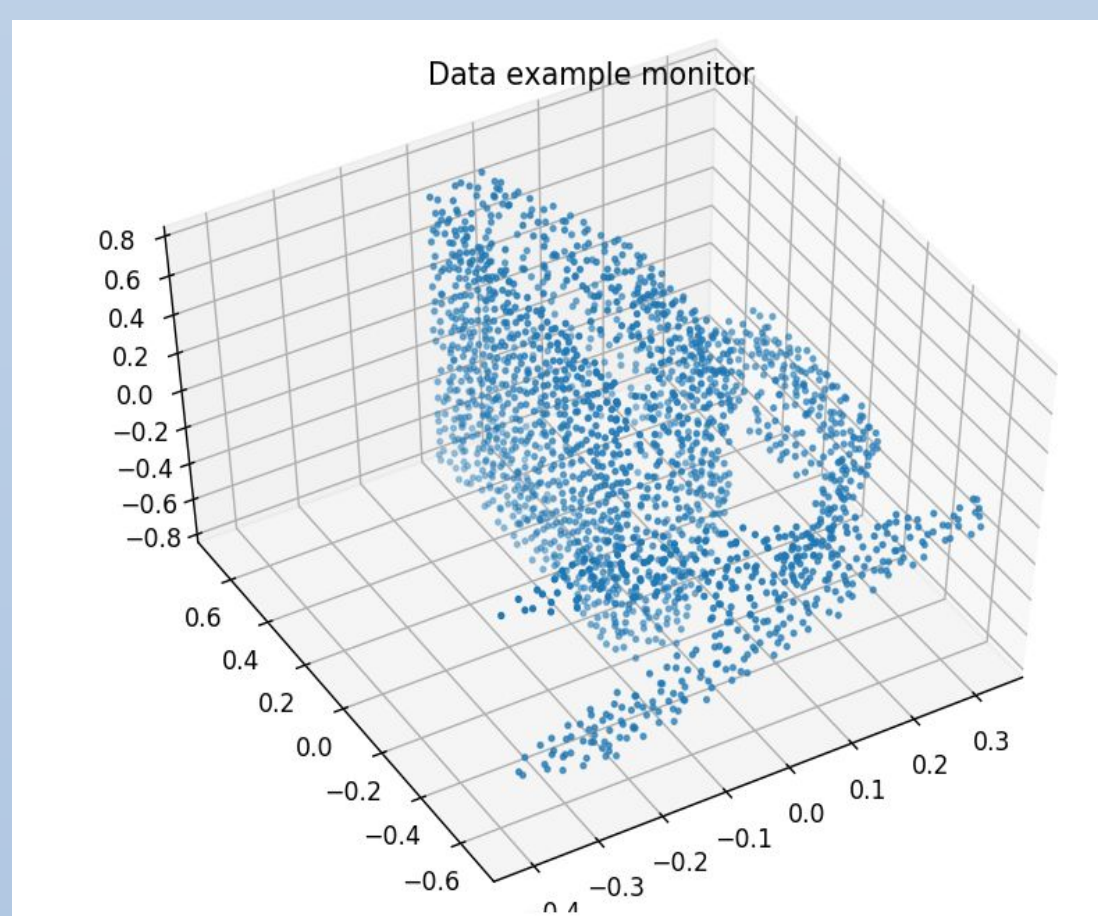
3D point cloud is a very important and common 3D data type. It contains a series of points which consists of the coordinates and some labels.

Traditional method to classify the point cloud is rendering the whole data into 3D object, while this would cost a lot.

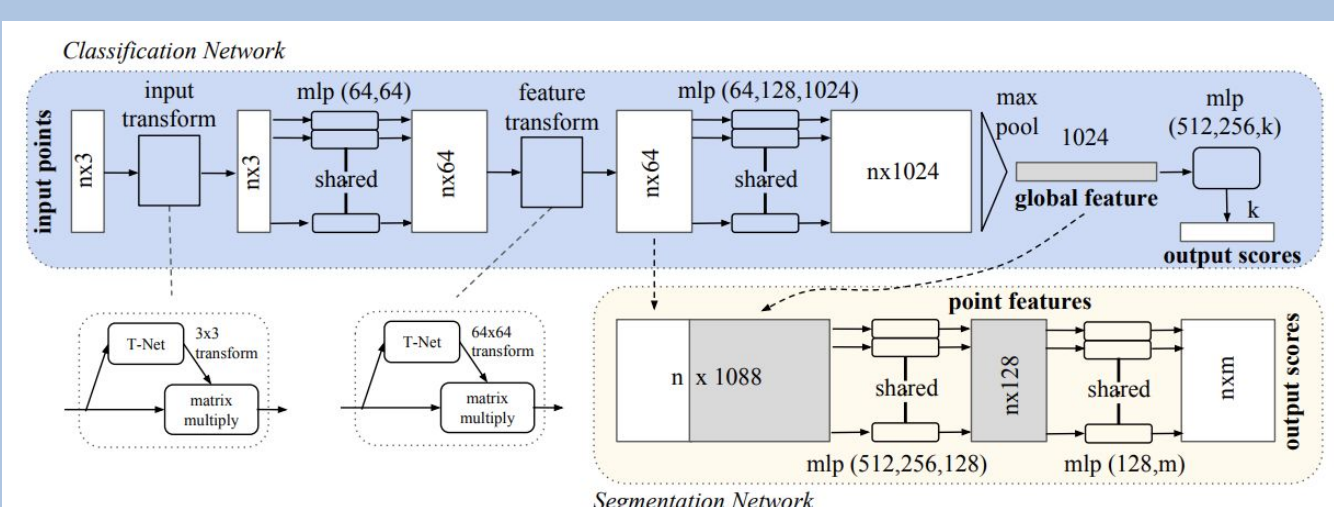
Our goal is to predict which category the points belong to directly from the sequence of points instead of visualize it first.

Data

The dataset used is from Modelnet40, preprocessed into .h5 files to be easily read into numpy arrays. The data is described by labeled point clouds which are unordered points in 3D space. They are loaded in by (batch, number of points, coordinates) with a label among 40 classes.



Models

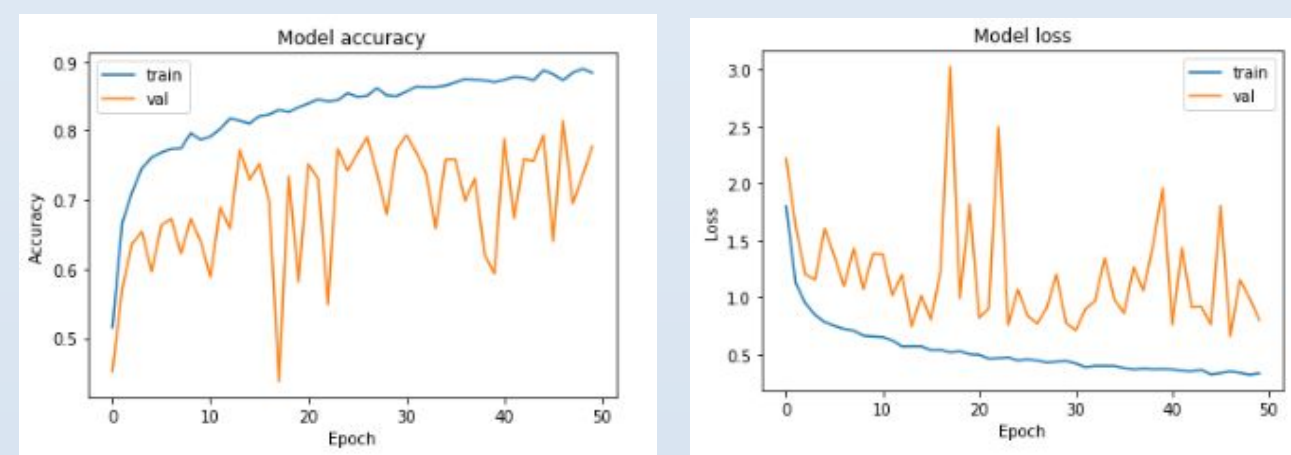


We employed the PointNet package which was originally developed by Prof. Leonidas J. Guibas and his group. The package is written in python based on keras, consists of a series of 1D convolution to extract features and max pooling part to pick the most important features. MLP here stands for multi-layer perceptron and represent a symmetrical function regardless the data's input order..

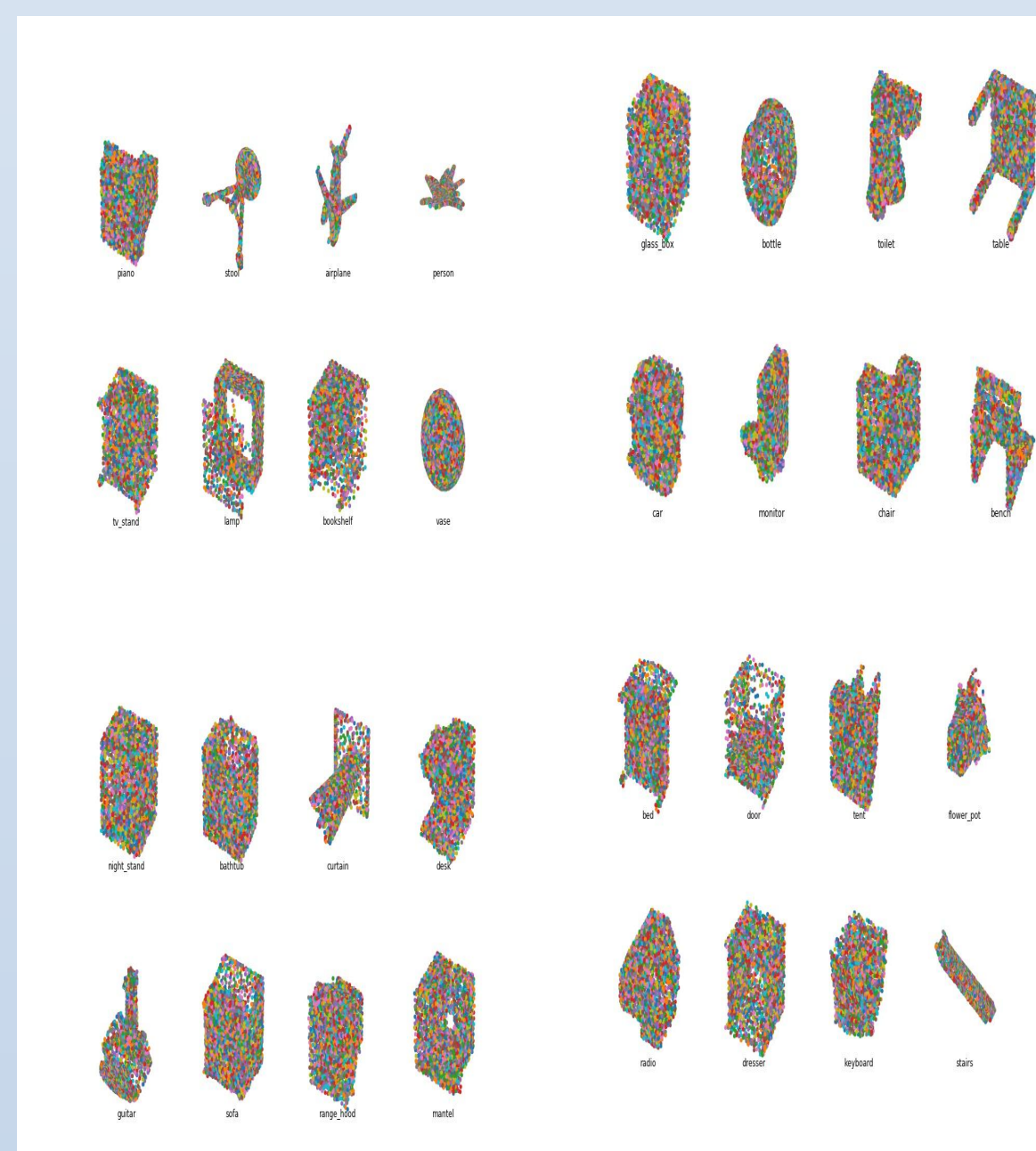
Results

We are able to predict 40 different shape with the input as point cloud.

Below are point cloud examples and the associated predicted label from our model



Dropout rate at 0.2.



Final testing accuracy: ~80.36%

Discussion

T-net here is a mini-transformation function while not demonstrated clearly in this paper. After looking through the code, the T-net in fact represents a series of 1D convolution to extract the features pixel-wisely, which would be used by max pooling and fully connected network for classification. While convolution could be a good choice here, picking another function could be better. The intrinsic property of T-net is more like a N to 1 mapping and it should be sensitive enough to the minor changes of points' position, we guess that designing a more computationally efficient function would accelerate the whole frame work.

It appears that to take care of the issue of the point clouds being unordered and therefore not applicable to certain convolution methods involving stride, the model uses a series of 1D convolutional layers which does not share the issue. Dropout layers with high dropout rates are used to prevent overfitting of the training data as some point clouds may have many redundant points which results in higher performance on unseen data.

Features

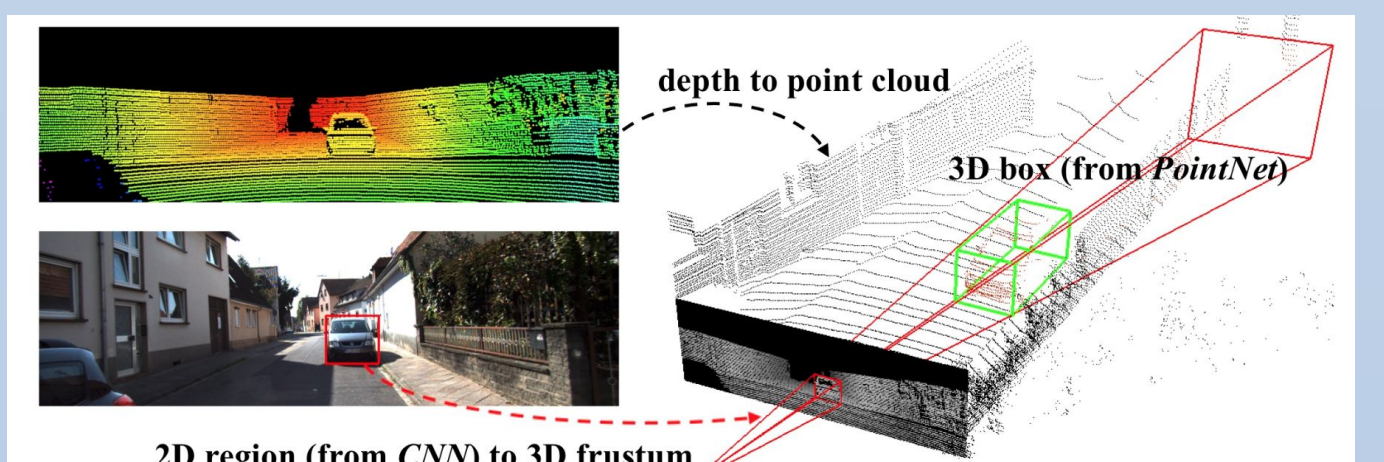
The features are the just input points cloud:

Training sets consist of 9840 samples of 2048 of 3 dimensionals points

Since we are using of Deep Learning, we are able to use the raw data points without any features extraction. The pointnet model is able to extract the features and learn by itself.

Future

If there was more time, an interesting application is to use object detection using the same Pointnet model base(Frustum Pointnet).



Further possible directions include point cloud segmentation as well 3D point cloud reconstruction using additional data.

References

- [1] L. J. Guibas, K. Mo, C. R. Qi, and H. Su, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," arXiv:1612.00593[cs.CV], April 10, 2017.
- [2] Y. Li, "Keras implementation for Pointnet," (2018), GitHub repository, Jan. 2018. [Online]. Available: <https://github.com/garyli1019/pointnet-keras>. [Accessed: Jun. 3, 2019].
- [3] A. Khosla, S. Song, X. Tang, Z. Wu, J. Xiao, F. Yu, and L. Zhang, "3D ShapeNets: A Deep Representation for Volumetric Shapes," in *Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, 7-12 Jun. 2015, Boston, MA, USA [Online]. Available: <https://modelnet.cs.princeton.edu/>. [Accessed: Jun. 3, 2019]