Semantic Labelling of 3D Point

[[1]](#footnote-2)

Akhil Krishna, Jyoti Kumari, PriyanshiChirania, Tanmay Pachpande

Dr.Suneet Kumar Gupta, Mr. Sahibnoor Singh Chahal, Mr. Yash Raj Saxena

**Abstract: --We deploy an algorithm for semantic labeling of each 3D point in order to make the robot understand the nature of the environment. We implement a methodology to make use of the geometric features along with the intensity from segments of point cloud. Also, a classifier corresponding to it is built for the prediction of associated objects in the surrounding. The network we used is PointNet which provides a unified architecture for the application of object classification. This classifies a given point cloud into 40 main classes mentioned in the dataset ModelNet40.For instance, if a room is messed up with clothes and we want to get it cleaned by the robot. So, the robot will first identify the boundary points of the object and will assign it the semantic label i.e. a proper meaning that it is a cloth. So, this forms the basic building block of 3d object detection.**

# **INTRODUCTION**

2D image classification has become very familiar and reached to its best extent. The next is 3D image classification. It is a new technology which is growing rapidly at the present time. There was no data for the 3D image classification a few years ago, now with the increase in demand, the data is being generated by using sensors like LiDAR and Depth Sensors. Now we have the necessary amount of data available. The most commonly used approach for this problem is transforming the point cloud data into 3D voxel grids. This renders data unnecessarily voluminous and causes issues. Our network, PointNet directly consumes point clouds. PointNet is highly effective and efficient.

.

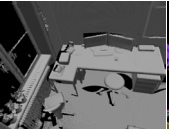


Figure 1 : Input Point cloud



Figure 2 : Semantic Labelling

# **Related work**

In previous works, the point cloud is converted to other representations before it’s fed to a deep neural network. When voxelization is done, the deep learning that can be used will be 3D CNN.Likewise, for rendering/projection, 2D CNN will be used. For feature extraction, a fully connected CNN will be used. All these ways of working always transform the point cloud. This issue of transformation is solved by our model PointNet. It uses point cloud directly, instead of converting it into other representations. But voxelization is the most accurate approach so far for classifying the 3D point cloud.

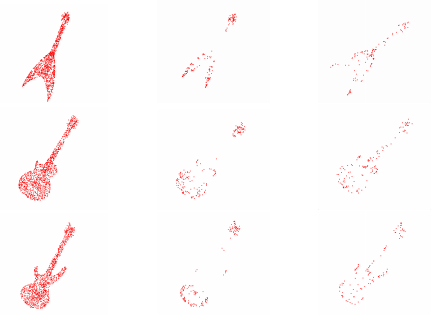
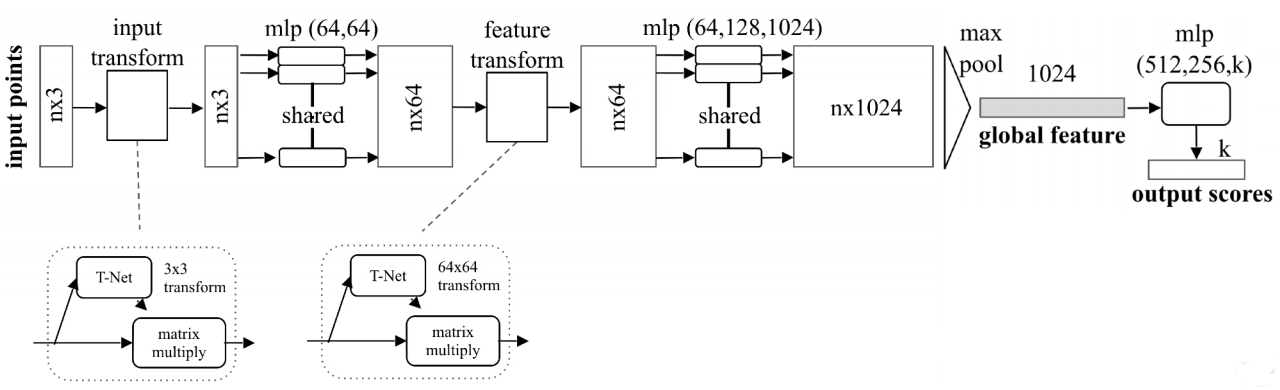


Figure 3: After using max pooling the points occurred for all different layers; Left: Normal point cloud; Middle: active points in layer one; Right: active points in layer two.

Those approaches involved a lot of complications during pre-processing and it was time consuming. This is quite contrasting to our approach which, for classification scores an accuracy of nearly 88 percent.

# **METHODOLOGY**

Dealing with unordered dataset requires an approach that uses a single symmetric function such as pooling (max pooling). The network effectively learns a set of optimisation criteria which learns only those points in the point cloud which are informative, and it encodes the reason for choosing those points. The fully connected layer that is obtained after this filtering process aggregates the optimal values that have been learnt into the global descriptor to obtain the original shape (reconstruction may not be the exact replica of the practical world object). Each point of the point cloud goes through a transformation process that is independent of the other point clouds. Thus, a spatial transformer that is data dependent can be added so that it can convert the data to canonical form before they go through the PointNet for processing.

The network that is responsible for classifying takes n points in a point cloud as input and applies input and feature transformations successively and draws out the point features by max-pooling. The output obtained is the classification score for m classes. The extension of the classification net is the segmentation net. The segmentation net concatenates the local features with the global ones and its output is per point score. MLP (x, y) stands for the multi-layered perceptron, the number in the bracket (x, y) stands for the size of the layer. Batch norm is applied to those layers which use ReLu as

the activation functions. Dropout layers are used for the ultimate(last) MLP in the classification net. [5]

Construction of symmetric functions using neural networks:

f (x1, x2 ,…, xn) = γ ! g(h(x1),…,h(xn))is symmetric given that g is symmetric. Here f, g and h are functions.

A Hausdorff continuous symmetric functionf:2y→**R** (Real Numbers) can be approximated byusing PointNet.

Figure 4: PointNet Architecture

**;**

**is PointNet**

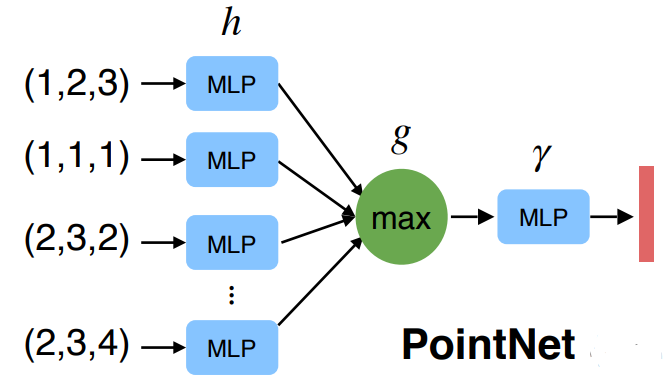


Figure 5: Architecture of PointNet Basic model

Input Alignment is done by Transformer network:

Transformation of the network that is data-dependent [4]. Matrix multiplication can be used for transformation function. The transform parameter matrix which is used for matrix multiplication will be close to orthogonal which means:

This is called a Regularisation. Here I is the identity matrix and B is the input matrix after transformation.

And the transform is as in figure

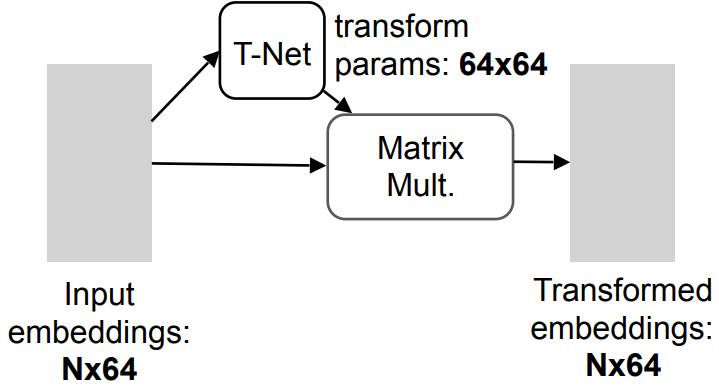


Figure 6 : Input transformation of an N\*64 to 64\*N matrix

# **Experimental Results**

We evaluate our algorithm on ModelNet40 dataset for 3D Object Classification in which there are 12,311 CAD models of 40 different categories. Which we have divided into 9,843 for training and 2,468 for testing. The data is in the format of the h5 file format. The best accuracy is obtained once the model is trained until 40 epochs. We get the accuracy of every class in the dataset ModelNet40. The test loss, when compared to GCN, is very less. The graph representing the same is attached here.

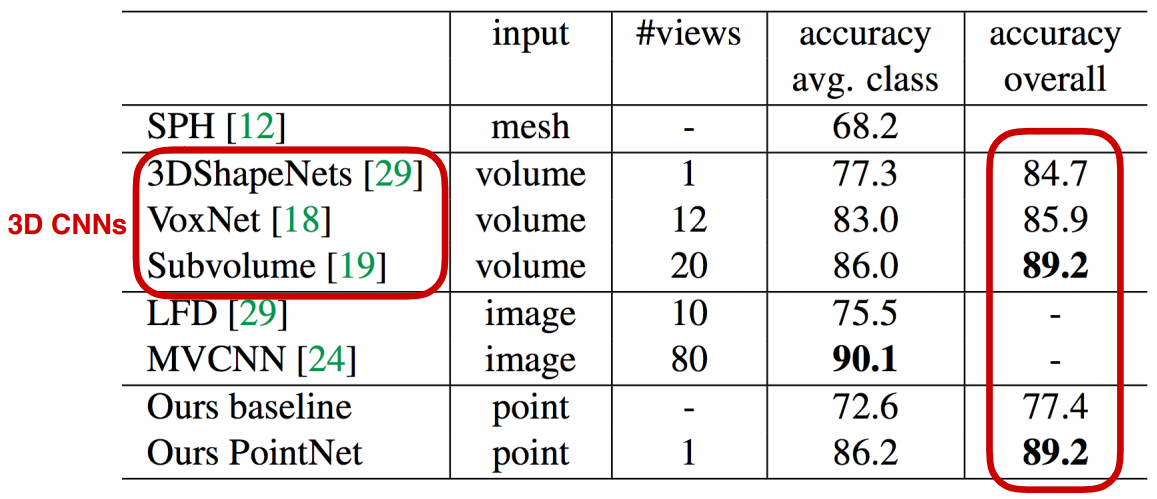


Figure 7: The accuracies of different approaches

A screenshot of a cell phone

Description automatically generated

Figure 8 : Point GCN and PointNet Test set loss comparison vs the number of epochs.

The overall accuracy of the model is 88.11%. The stability using accuracy (mean instance) is compared with PointGCN and the graph for the same is here.

A close up of a piece of paper

Description automatically generated

Figure 9 : PointNet and PointGCN stability of the model comparison for 50 trials.

# **CONCLUSION**

The idea of 3D object classification or processing can be used to solve many real-life problems due to which this project has some future work.This can be done by increasing the number of classes and converting a real-life image and into a 3D Point Cloud which can actually solve many problems like object detection. 3D point cloud classification is a paramount job with applications in robotics, AR (augmented reality) and urban planning.

Due to this project we have got a chance to explore many models which can be used to classify 3D Objects (volumetric 3D images, 3D Mesh images, 3D Point Cloud) like graph CNN, 3D CNN etc.

The limitations of the project are it is limited to 40 classes so if a point cloud isn’t part of those 40 classes and is passed through the model it will show the class which is close to the original one. The dataset contains only 13,000 images.

# **References**

[1] **Classification Project Using Graph CNN**

<https://github.com/maggie0106/Graph-CNN-in-3D-Point-Cloud-Classification>

[2] **Classification And Segmentation Using PointNet**

<https://github.com/charlesq34/pointnet> -

[3] ***Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas***; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 652-660

[4] ***Yingxue Zhang and Michael Rabbat***, “A Graph-CNN for 3D Point Cloud Classification” in Conf. on Acoustics, Speech and Signal Processing (ICASSP), Alberta, Canada, 2018.

[5] ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume I-3, 2012 XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia.

1. [↑](#footnote-ref-2)