VoxelNet for Point Cloud Based 3D Object Detection (Review)

Zhou, Y., & Tuzel, O. (2017). VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, https://arxiv.org/abs/1711.06396.

Anudeep Reddy Puthalapattu
Department of Computer Science
Houghton, United States
aputhala@mtu.edu

Ravi Teja Inala
Department of Computer Science
Houghton, United States
rinala@mtu.edu

Sai Teja Mummadi
Department of Computer Science
Houghton, United States
mummadi@mtu.edu

***Abstract*— With the rapid increase in the use of autonomous cars, augmented reality, and robots, the main problem is the detection of 3D point clouds. Lidar point clouds by nature are highly sparse. For interfacing with a region proposal network (RPN), a lot of focus is on hand-crafted feature representations. In this paper, the need of removing manual feature engineering is addressed, and proposed VoxelNet which unifies feature extraction and 3D bounding box prediction into one stage. According to this paper, VoxelNet divides lidar point cloud into equally spaced 3D voxels which then transforms a group of points within each voxel into a unified feature representation through the voxel feature encoding (VFE) layer. Detections are generated by connecting a unified layer formed with RPN. It is observed that VoxelNet outperforms other methods by a large margin. This network also gives good results especially for pedestrians and cyclists, just only by using Lidar.**

Keywords—autonomous, VoxelNet, rpn, lidar, point cloud.

I. Introduction

Image-based detection has provided good results in terms of object detection, but depth information is missed when dealing with 2D objects. Lidar sensors provide reliable depth information which can be used to localize and characterize various shapes easily. Lidar sensors generate a hundred thousand points which are stored in the form of point clouds. Figure 1 represents some of the point clouds formed by the lidar sensor for cars, pedestrians, and bicycles. In some of the cases, it would be very difficult to distinguish whether the object is a car, pedestrian, or bicycle.

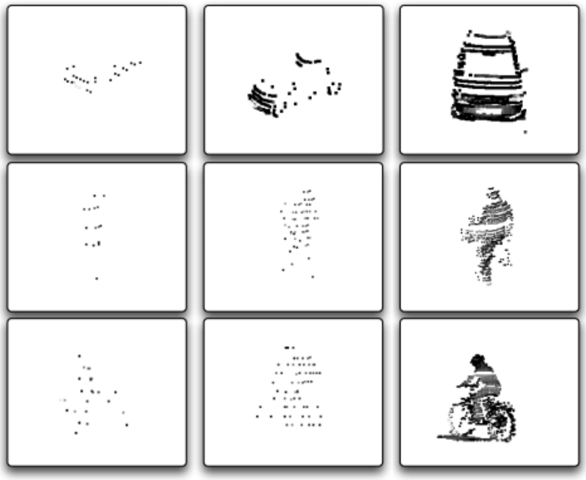

Figure 1: Point clouds for car, pedestrian, cyclist

Voxel Net can be used for learning discriminative features from point clouds and predict accurate 3D bounding boxes.

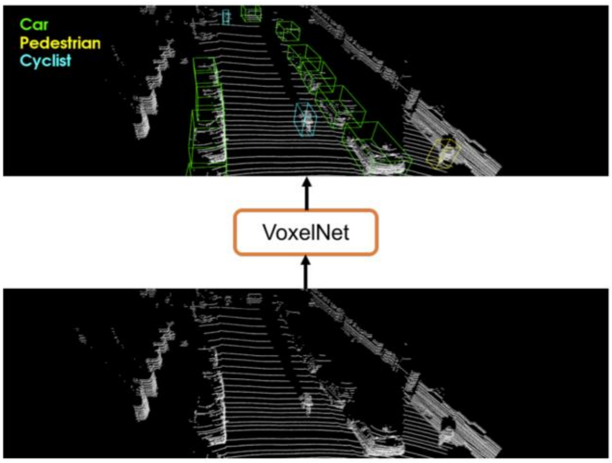

Figure 2: VoxelNet produces 3D detection results from point clouds

II. Background and Motivation

Lidar sensors provide reliable depth information which can be used to localize and characterize various shapes easily. The points clouds formed by lidar are usually sparse. It is mainly because of non-uniform sampling of 3d space, effective range of sensors. There are several approaches to deal this problem. They mainly focus on rasterizing (converting from vector to pixels) point clouds and then encoding each of them with handcrafted features. A major breakthrough was moving from handcrafted features to machine-learned features.

PointNet learns pointwise features directly from point clouds. The improved version of it enables the network to learn local structures at different scales. In these two approaches, feature transformer networks are being trained on 1k points. But typical point clouds that are obtained from Lidar contain 100k points training on above mentioned methods demands high computational power and more memory is required. These are the main challenges that are addressed in this paper.

Regional proposal network (RPN) is another algorithm for efficient object detection. This approach requires data to be dense which is not the case with lidar generated output. This paper tries to address this shortcoming.

There is a rapid development of 3D sensor technology these days by which many cars are using them. Many algorithms inferred 3D bounding boxes from 2D images. But the results obtained by this are bound by depth information. There are also muti-modal fusion algorithms that combine both images and lidar to improve the performance compared to that obtained by lidar-only particularly for pedestrians and bicycles. But, in this case synchronization between sensors is required and it may be completely not functioning when one of the sensors fails.

III. ARCHITECTURE SUMMARY

The architecture of the VoxelNet has mainly three different blocks which are Feature Learning network, Convolutional layers followed by the Region proposal block.

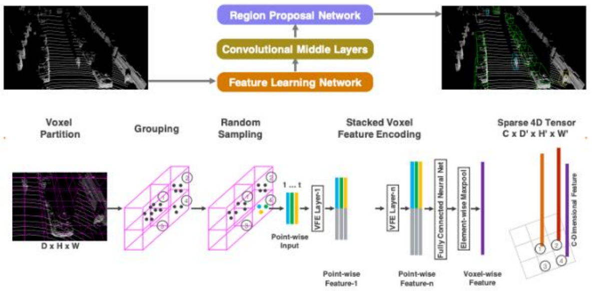

Figure 3: VoxelNet Architecture

As we can see from the picture, the given input, which is the 3D point cloud, first passes through the feature learning network and then to the convolutional layers and further to the region proposal network.

Feature Learning Network: The 3D point cloud which is given as an input is first divided into equal spaced voxels. Each voxel is block which consists of certain number of points in the point cloud. As the data from the point cloud is sparse, sometimes the voxels might be empty and other voxels have more density of points. After the voxels are created, the points are then grouped together within the voxels. Due to numerous reasons such as the relative direction of the object, different texture of the objects, the sparse LiDAR data. After the grouping of certain points, each voxel contains varied number of points. From the Figure, we can see that voxel has a greater number of points than the voxel two. Whereas the voxel 3 has no points in it. After the grouping of the points, the random sampling of the data is performed. Usually, each LiDAR point cloud consists of 100,000 points which cannot be processed effectively on the computers as there are memory issues. Therefore, random sampling of the points would reduce the computational burden and decreases the imbalance in the points. It also adds certain variation to the data for the training of the network. The next step in the algorithm is the Voxel Feature Encoding layers.

Stacked Voxel Feature Encoding: Each voxel contains different set of points; therefore, the voxel feature encoding is explained for one voxel. A non-empty voxel which contains of n different points is passed into the layer. Firstly, the local centroid of the voxel is calculated, then the offset of each point is calculated with respect to the centroid and augmented with the data. Now the data points are passed to the fully connected network. This fully connected network transforms the datapoints into the feature space.

This feature space is aggregated to form the shape of the object. This feature depicts the objects in the voxel. The Fully connected layer is composed of linear layer followed by the Batch normalization and the Rectified linear unit activations. Furthermore, elementwise max pooling is applied on the layer. Then the non-empty voxels are augmented with the features, all the voxels are encoded similarly.

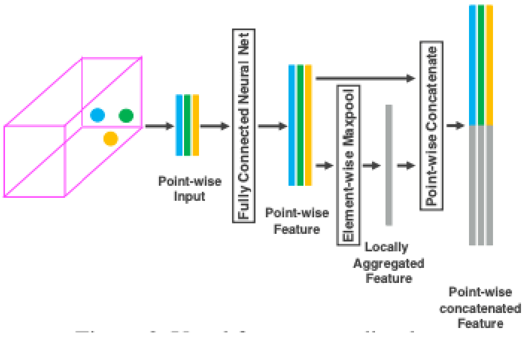
Point-wise Input
Fully Connected Neural Net
Point-wise Feature
Element-wise Maxpool
Locally Aggregated Feature
Point-wise Concatenate
Point-wise concatenated Feature

Figure 4: Voxel feature encoding layer

As the output feature contains both pointwise and locally aggregated features, combining the VFE layers would give us a clear representation of the shape of the object. The main reason for using the sparse non-empty voxels is getting the shape of the model without really spending a large amount of energy.

Convolutional Middle Layers: The convolutional layers are added to the output which was generated from the VFE layers. Each convolutional layer applies 3D convolution on the input in addition to the batch normalization and the ReLU layers. The convolutional layers add up the voxel features with a large receptive field. This helps the model to learn more about the current dimensions and shapes.

Region Proposal Network: The Region proposal networks are building blocks of the object detection models. In this model as well, the RPN architecture is used in addition to the VFE and CNN layers. This resulted in an end-to-end trainable model.

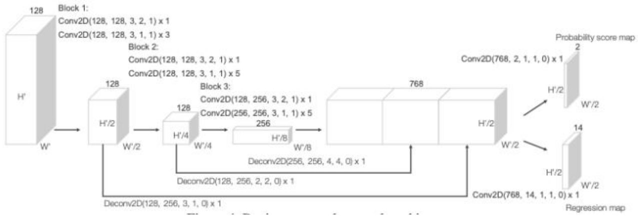
128
Block 1:
Conv2D(128, 128, 3, 2, 1) x 1
Conv2D(128, 128, 3, 1, 1) x 3
Block 2:
Conv2D(128, 128, 3, 2, 1) x 1
Conv2D(128, 128, 3, 1, 1) x 5
Block 3:
Conv2D(128, 256, 3, 2, 1) x 1
Conv2D(256, 256, 3, 1, 1) x 5
H'
W'
128
H'/2
W'/2
128
H'/4
W'/4
256
H'/8
W'/8
768
H'/2
W'/2
Deconv2D(256, 256, 4, 4, 0) x 1
Deconv2D(128, 256, 2, 2, 0) x 1
Deconv2D(128, 256, 3, 1, 0) x 1
Probability score map
2
Conv2D(768, 2, 1, 1, 0) x 1
H'/2
W'/2
14
H'/2
W'/2
Conv2D(768, 14, 1, 1, 0) x 1
Regression map

Figure 5: Region Proposal network

The RPN has mainly 3 different blocks of the CNNs, each of these layers down sample the feature maps. This is achieved through using a convolutional layer with a stride of 2. After the CNN layer, the batch normalization and ReLU operations are performed. After down sampling the blocks, the outputs are then up sampled into certain sizes. These sizes are fixed and concatenated with each other. This results in a high-resolution feature map. The feature map is them mapped to the outputs which are probability scores and the regressions. The regression map gives us the length, width, height of the bounding boxes.

Loss Function: The loss function of the model is designed in order to incorporate the 3D ground truth box which has the values (xgc, ygc, zgc, lg, wg, hg, θg). Here the x, y, z are the center location of the object, l, w, h are the length, width, height of the bounding box. θ g represents the rotation of the ground truth in the Z-Axis. Both cross entropy and regression loss are used in the calculation of loss.

Multiple parameters have been setup to train on different datasets, in the car detection dataset the point cloud is in the range of [−3, 1] × [−40, 40] × [0, 70.4] meters. This is in the axis of Z, Y, X. The voxel size is also set differently for each different dataset. This improves the prediction accuracies over the datasets. The data is further augmented to improve the training accuracies in the model.

IV. Main Contribution

To identify the objects and their shapes, LiDAR is used but there are challenges such as occlusion, sampling of 3D space which is non-uniform, and relative pose. To mitigate these problems, point clouds are crafted manually for 3D object detection. There are other approaches where the encoding of a voxel of 3D Voxel grid with hand-crafted features. Region Proposal Network for efficient object detection requires data to be dense and ordered structure.

The author proposed voxel nets, a new architecture for point cloud-based 3D detection which operates directly on sparse 3D points. It predicts 3D bounding boxes through learning feature representation simultaneously from point clouds. It avoids manual feature extraction, unlike traditional methods.

The proposed architecture was evaluated to detect 3D objects using the framework provided by the KITTI benchmark. The experimental results show that VoxelNet outperforms the traditional 3D detection methods like the state-of-the-art LiDAR method by a huge margin. Also, from the view of the LiDAR point cloud, the method was achieved in detecting the cyclists and pedestrians as well.

The devised method processed stacked VFE operations in parallel across voxels and points of dense tensor structure which is converted from the point cloud.

The experimental results evaluated on the KITTI test set are shown in Table 1.

Benchmark	Easy	Moderate	Hard
Car (3D Detection) Car (Bird's Eye View)	77.47 89.35	65.11 79.26	57.73 77.39
Pedestrian (3D detection) Pedestrian (Bird's Eye View)	39.48 46.13	33.69 40.74	31.51 38.11
Cyclist (3D Detection) Cyclist (Bird's Eye View)	61.22 66.70	48.36 54.76	44.37 50.55

Table 1: Performance results provided by author in the paper

The author introduced a three-step procedure for data augmentation and generated without the requirement to be stored on disk which involves LiDAR points and ground truth of 3D bounding box through applying global scaling and global rotation to all the ground truth.

The new proposed model does parallel processing efficiently on a voxel grid and also takes an advantage of sparse point structure.

The proposed model VoxelNet uses only LiDAR for detection unlike the KITTI benchmark which uses LiDAR point clouds and both RGB images.

V. Strengths

The authors proposed VoxelNet architecture has the following strengths:

- The proposed architecture of VoxelNet has a feature encoding layer which is the key design in VFE layers.
- The modified RPN architecture was combined with the CNN middle layers and feature learning network to form a trainable which form end-to-end.
- The dataset is divided into three parts which included a validation dataset of 3769 samples and avoid the split having the same sequence in both validation and training sets.
- The metrics are evaluated on the KITTI benchmark framework with IoU being considered for assessing the performance.
- The proposed model of VoxleNet has an inference of 33ms only (Refer Table 1).

VI. Drawbacks

The authors proposed VoxelNet architecture has the following drawbacks:

- There is a double the difference in detection from cars to pedestrians.
- Access to the test server is limited as the ground truth of the test set is not available.
- The evaluation process is extensive due to the unavailability of ground truth for the test set.
- The three different forms of data augmentation increase the time complexity.

VII. CONCLUSION

The contribution of the author in developing new architecture VoxelNet for 3d object detection and differentiating its performance from the traditional approach, main contribution of author's proposed model and their motivation to develop this model, value is added, strengths and drawbacks of the proposed architecture are discussed.

VIII. ACKNOWLEDGMENT

Sincere thanks to Dr. Xiaoyong Yuan for providing frequent feedback.

IX. References

[1] Zhou, Y., & Tuzel, O. (2017). VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, https://arxiv.org/abs/1711.06396.