EasyChair Preprint

№ 3665

Deep Learning Methods on 3D-Data for Autonomous Driving

Ahmed Elkhateeb

June 22, 2020

Deep Learning Methods on 3D-Data for Autonomous Driving

Ahmed Elkhateeb

Intelligent Embedded Systems, University of Kassel, Germany

Abstract. Computer vision tasks as semantic Instance segmentation play a key role in the most recent technological applications such as autonomous driving, robotics and augmented/virtual reality. With the aid of artificial intelligence, object classification and instance segmentation became more approachable tasks in comparison to the former classical methods. Over the past decade different Deep Learning (DL) architectures such as R-CNN family and RPN were introduced to address such tasks on 2D data representations. Recently after the availability of sensors that can capture 3D information. New DL architectures with a backbone of the RPN and R-CNN were developed to work on the different 3D data representations and address their challenges. Yet the challenges are mostly set by the nature of the 3D data obtained by different kinds of sensors such as LiDAR and stereo cameras. Which were mostly deployed in the Autonomous Driving field for acquiring 3D information. Respectively point clouds and RGB-D are the 3D data representations produced by these kinds of sensors. This paper contains a survey on the state-of-art DL approaches that directly process 3D data representations and preform object and instance segmentation tasks. The DL architectures discussed in this work are designed to process point cloud data directly. As Autonomous Driving rely mostly on LiDAR scanners for 3D data representation.

1 Introduction

Humans perform the vision analysis of the surrounding environment quite effortlessly with the aid of one vision sensor the eye. Multiple kinds of vision sensors i.e: stereo camera, radar and LiDAR are now used in the autonomous systems. Each of which runs on different technology. Which changes the nature of the data acquired by the sensor.

This work is motivated by the trend and the goal to develop autonomous tools that can conduct knowledge extraction with no to least human intervention possible. Instance segmentation, object classification and localization are the key factors to achieve this goal. Lately the performance of the developed DL architectures and getting more accurate and robust. Shedding the lights on these methods and assessing them will pave the way to understand, apply and improve such methods.

Object classification and instance segmentation play a key role in autonomous application. As the ability of extraction of useful information from the surrounding environment is crucial in the decision process of autonomous driving. Over the last decade many recent milestones where achieved in the field of computer vision tasks with the aid of DL architectures. Most of which started as a 2D feature extraction algorithms [4,3,6,11]. Which can successfully apply object classification and instance segmentation tasks on 2D images. The extension of existing 2D DL architectures to be able to perform on 3D data is not often a straightforward task. The different representations of 3D data as point clouds or RGB-D imposes the challenges in the development process of 3D DL architectures.

In this work, an overview of the recent DL architectures on 3D data is presented. Looking closely into their architectures paradigms and their main elements. PoinNet [9] and its family proposed by Qi et al. is a pioneer in this field. Due to the ability to work directly on the raw point clouds. A detailed description of the PointNet[9] architecture is shown as well as the VoxelNet [12]. Both reported performances in object classification and semantic segmentation are reviewed according to the KITTI [7]. VoxelNet architecture is discussed in detail with the key differences in the approach taken by both architectures to handle the raw nature of the point clouds. The different nature of the 3D data representation is overviewed. In addition to the most used sensor technologies in the autonomous driving to acquire the 3D data. A comparison between the sensors and their ability will be also shown.
This work was to conduct a comparison of the state of art deep learning approaches that are developed for the multi dimensional data that are used mostly in the autonomous driving application, defining and classification of the multi dimensional data is crucial to develop and understand the models and their design paradigm. The contribution of this paper is summarized as follows:

First An overview on different sensor technologies used in the autonomous driving field.
Second we ought to have a grasp on the different classifications of 3D data according to their nature and structure.
Third Secondly discussing the state of art approaches that can handle these kind of data while understanding the design paradigm of each of them.
Forth showing the reported accuracy results by both approaches in the KITTI benchmark [7].

2 Vision Sensors in Autonomous Driving

Vision is vital in the decision process in the autonomous driving systems. Most of the driving related decisions whether breaking, throttling or steering. Rely on the acquired vision information of the 3D space around the vehicle. Vision Sensors used in Autonomous driving required to capture the information of the shapes, colors and distances. A lot of information can be derived from such form these main features i.e. Ego-localization/mapping, speeds of moving objects, relative distances objects and the vehicle as well as the textures. Unfortunately

not all the information can be provided by one vision sensor. Sensors like stereo cameras, LiDAR and Radars are mostly mounted on the vehicles to acquire the surrounding vision information. A comparison between the abilities of the cameras and LiDAR is shown in following table. Since Radar and LiDAR share most of the features. One of the biggest advantages of the Radar on LiDAR is the robustness to bad weather conditions i.e. snow and rain.

Features	LiDAR	Stereo Camera
Distance Measurement	2 m - 200 m	$\propto Resolution$
Ambient Light Susceptibility	sensitive, High night-vision-performance	sensitive
Weather Conditions Susceptibility	sensitive	sensitive
Cost	high	low

The depth image values in stereo vision is calculated by the triangulation method. Which is shown in the following equation.

$$d = f * b/x \tag{1}$$

where

d calculated depth value
f focal length
b base distance (distance between the cameras)
x pixel size in mm

As shown in the above equation the size of the pixel is inversely proportional to the calculated distance. With high resolution the number of pixels that describes space is high. Which results in small pixel sizes.

3 Data

Representation of the same type of information varies depending the technology of the sensor that captures these raw 3D data. The ideal 3D data should grasp all different kinds of information of the 3D objects in the scene e.g. structure, colour and texture of each object as well as the distribution across the 3D space. Since the ideal does not exist, a trade-off between the emphasis of these features is expected. There are two major classifications that all the representation abide to: Euclidean-structured and non-Euclidean data. In this work not all types and variation of the two classes of data are going to be extensively presented. Since the focus is on the types of sensors that are reliable enough to be used in the Automotive field e.g.: LiDAR sensors and Stereo cameras.[2]

3.1 Euclidean

As the name states, this kind of data are governed by a common system of coordinates in a grid-like structure. Typically, Euclidean spaces are defined by $\mathcal{R}^N$ (n: number of dimensions)-often in XYZ coordinates. The main types falling under this class are: RGB-D, Volumetric, Multi-view data, descriptors, and projections. In this work we will only focus on the RGB-D data.[2]

RGB-D As the price of stereo cameras dropped and due to the their adequate representation of the 3D information by this technology. It has been widely applied in various technological fields. The depiction of 3D information is performed in a 2.5D [1] representation manner. This technique of the 3D information representation provides two images for every frame, the first is the conventional RGB 2D representation, the second is the relevant depth (D) map of the captured RGB frame (hence the 0.5 D). The availability of the data in this form also plays an important role in the wide usage in various Deep Learning architectures.[2]

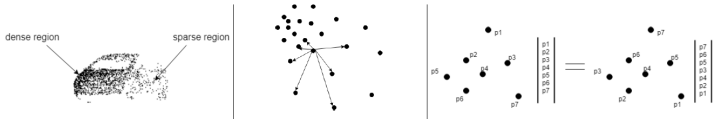
Since this data are parametrized, the upgrade of existing deep learning architectures from 2D to 3D space is possible. Nevertheless the transformation of the RGB-D to another space representation like the 3D Point cloud is also possible since the RGB-D is considered to be higher level of 3D information depiction that of the Point cloud data.[8]

3.2 Non-Euclidean

Non-Euclidean Data differ from their counterparts that they are not govern by a global coordinate system. Upgrading the existing 2D deep learning paradigms to be able to comprehend this kind of data is not easy as the Euclidean data, since they are not presented through vectors. The main types of this class of data are point clouds, 3D meshes and graphs. We will discuss point clouds exclusively in this work.[1]

Point Clouds 3D Point clouds are represented as a sparse matrix containing clustered clouds which represent the objects in the space. Each cluster on its own can be realized with a common parametrized coordinate space which is immune to rotation and translation. One can see it as a set of small Euclidean coordinate systems.[1]

There are multiple challenges facing the researchers in designing a deep learning paradigm that can deal with point clouds. The raw features of the point clouds enforce the approaches to be able to deal with the irregular structure of the point clouds. Some of these challenges are; the sparse and the uneven distribution of the information across clusters. Nevertheless, these clusters have different densities and distribution across the space. In addition, the information is stored as sets in lists thus changing the order of the data points in the sets does not alter the representation of the scene. Recent works [10,9,8,12] overcome the need of altering the format of the point clouds. Thus, being able to directly apply deep learning architectures on raw point clouds.



(a) Irregular. Sparse and dense regions

(b) Unstructured. No grid, each point is independent and distance between neighboring points is not fixed

(c) Unordered. As a set, point cloud is invariant to permutation

Fig. 1: Figure 1.Point Cloud Challenges [2]

4 Deep Learning Architectures

The deep learning paradigms here to discuss have the ability to work directly on the raw point clouds irregular structure. Both of the following architectures are applied for object classification, instance segmentation and object localization.

4.1 PointNet

Qi et al. proposed a model architecture that deals with the nature of the point clouds directly. *PointNet* addresses every challenge stated by the nature of the point clouds in a different manner. The idea is to transform the raw input feature to a more global feature representation which is immune to the data perturbations and permutations. Needless to say this transformation function has to be symmetric and this is achieved by deploying an MLP (Multi-Layer Perceptron) network followed by single variable function and as a symmetric function max pooling function is used. In order to apply a successful classification paradigm both local and global features and required. This is done by a Segmentation Network which concatenates the both the aggregated global features and the point feature (as shown in figure 1) to produce a new feature which is appreciative to both the global and the local information. [reference]. For improving robustness to permutation, an affine transformation is applied on the input features and the feature vector after the first MLP network [10,5] as shown in Figure 2.

Pointnet by design does not apprehend the metric structure and its different scales. Thus resulting a limitation to its ability to recognise fine-grained patterns. A hierarchical scaling paradigm was necessary to capture and generalize over the metric scaling structure. Which was introduce by Qi et al and is followingly reviewed. [5,9]

PointNets ++ was introduced to achieve two goals first is to partition the point set with respect to different scales, secondly is to globalize and abstract the learned weights across the each feature learner model. PointNet is applied in this design as a feature learner benefiting from its robustness to the unordered structure as well as the permutations and perturbations of the data.

Although PointNet and PointNet++ were pioneers in the object classification and semantic segmentation. The object detection and localization in the 3D space was an issue needed to be addressed in their following work.

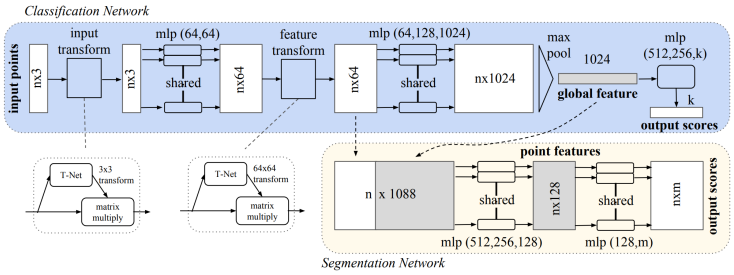


Fig. 2: PointNet Architecture. The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.[10]

Frustum PointNet was introduced to address the localization process of each of the classified objects in the scene. The 3D bounding box in this proposal is a frustum, which is generated with the aid of the RGB-D frame of the respective scene. Taken into consideration that the RGB-D data are higher level 3D data forms and can be converted to point clouds with the aid of the camera projection matrix. Aware of the perspective scales of the objects with respect to the sensors. With the aid of the previously mentioned givens; the Frustums are generated by lifting the 2D bounding boxes on the 2D images. These 2D regions of interest are proposed by a classical 2D region proposal network FPN [8]; see Figure 3.

4.2 VoxelNet

VoxelNet [12] proposed another approach for handling the sparsity in the LiDAR point clouds as well as the highly variable point density distribution. This is a generic 3D detection framework can object classification and localizing the reespective 3D bounding boxes on point clouds in an end to end fashion. VoxelNet splits the point cloud into equivalent 3D voxels, then encodes each of the voxel via stacked voxel feature extraction (VFE) layers. The convolutional layers afterwards aggregate the features of each voxel. Finally these volumetric representations are fed into a region proposal network [4,3] that produces the detection results. VoxelNet architecture consists of three main blocks as shown in Figure 4. Firstly, the Feature learning network (VFE) which is discussed followingly and considered as the key innovation in this architecture.

The point clouds are subdivided into equally sized 3D voxels[12]. This will result into varying densities of points between the voxels due to the sparsity of the point clouds [2]. This is handled by a sampling technique which draws equal number of samples (T) [12] from the non-empty voxels. This produces voxels with a uniform number of points[2]. In addition to, that the number of samples

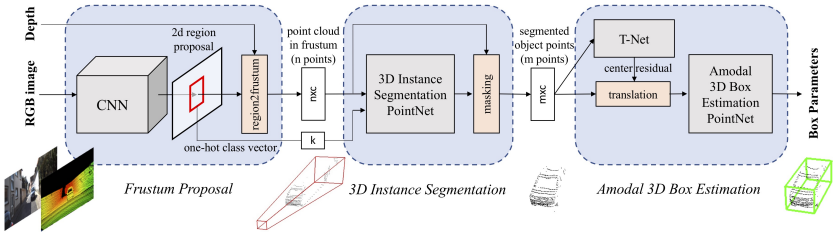


Fig. 3: Frustum PointNets for 3D object detection. We first leverage a 2D CNN object detector to propose 2D regions and classify their content. 2D regions are then lifted to 3D and thus become frustum proposals. Given a point cloud in a frustum ($n \times c$ with n points and c channels of XYZ, intensity etc. for each point), the object instance is segmented by binary classification of each point. Based on the segmented object point cloud ($m \times c$), a light-weight regression PointNet (T-Net) tries to align points by translation such that their centroid is close to amodal box center. At last the box estimation net estimates the amodal 3D bounding box for the object.[8]

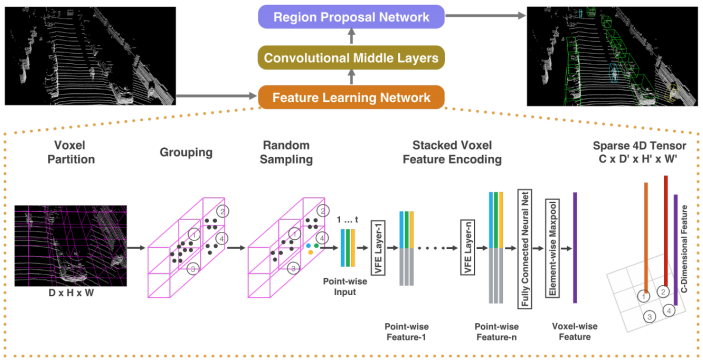


Fig. 4: VoxelNet architecture. The feature learning network takes a raw point cloud as input, partitions the space into voxels, and transforms points within each voxel to a vector representation characterizing the shape information. The space is represented as a sparse 4D tensor. The convolutional middle layers processes the 4D tensor to aggregate spatial context. Finally, a RPN generates the 3D detection.

has to be less than the number of points contained by each voxel [12]. So by portioning and grouping of the raw point clouds the non-uniform and sparsity nature of the point clouds is handled. Nevertheless, the mentioned sampling technique addresses the varying densities of the distribution of the point clouds in the 3D space.

A point-wise-global representation for each voxel is acquired by calculating the local mean between the points of each voxel. The point-wise-local information is then obtained by calculating the relative distance of each local point to the the respective centroids [12]. Both information are fed into a fully connected layer (FCN) to produce a point-wise the feature map. Then these point-wise feature are then stacked and fed into a MaxPooling layer to obtain an element-wise-global feature [2] as shown in Figure 5.

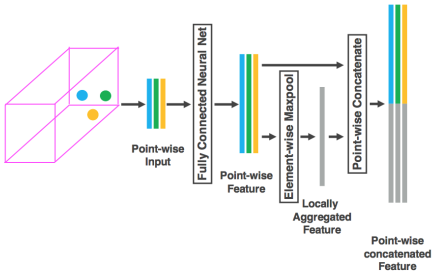


Fig. 5: Voxel feature encoding layer.

5 Results

KITTI [7] dataset was developed by Karlsruhe Institute of Technology et al. and is one of the best known datasets in autonomous driving. It can be used for tasks such as: stereo, optical flow, visual odometry, 3D object detection and 3D tracking. For each task they provide benchmarks as well an evaluation metric. KITTI provides 5 classes; Road, City, Residential, Campus and Person for their raw data. The dataset comprise 389 stereo and optical flow image pairs, stereo visual odometry sequences of 39.2 km length, and more than 200k 3D object annotations captured in cluttered scenarios (up to 15 cars and 30 pedestrians are visible per image).

The performance scores shown in the next section are the reported scores of both DL architectures formerly introduced.

Parameter	F-PontNet[8]			VoxelNet[12]		
Data-type	RGB-D and Point Cloud			Point Cloud		
mAP (Moderate)	**65.39**			58.25		
class	Easy	Mod.	Hard	Easy	Mod.	Hard
Car	88.7	**84**	75.33	**89.35**	79.26	**77.39**
Pedestrian	**58.09**	**50.22**	**47.2**	46.13	40.74	38.11
Cyclist	**75.38**	**61.96**	**54.68**	66.7	54.76	50.55

Table 1: Performance on the KITTI Birds Eye View detection benchmark

Parameter	F-PontNet[ref]			VoxelNet[ref]		
Data-type	RGB-D and Point Cloud			Point Cloud		
mAP (Moderate)	**57.35**			49.05		
class	Easy	Mod.	Hard	Easy	Mod.	Hard
Car	**81.2**	**70.39**	**62.19**	77.47	65.11	57.73
Pedestrian	**51.21**	**44.89**	**40.23**	39.48	33.69	31.5
Cyclist	**71.69**	**56.77**	**50.39**	61.22	48.36	44.37

Table 2: Performance on the KITTI 3D object detection benchmark

5.1 Accuracy Scores

6 Summary and Conclusion

The availability of 3D vision technology charged the progress in the autonomous driving field. Deep Learning architectures are focused now more than ever on the challenging Computer vision tasks. As overviewed in this paper, PointNet and VoxelNet can preform directly raw point clouds and achieve a state-of-the-art performance results in Object/instance segmentation. Although they share their common features in their architecture as having the similar backbones of RPN/R-CNN 2D architectures in object detection. They differ in approaching the point clouds for the application of the feature extraction step. While Qi et al.[9] choice was by performing the affine linear transformation and the symmetrical function MaxPooling directly on the point clouds. VoxelNet[12] chose to group the point clouds in equally sized voxels and then applying sampling for computation overhead reduction followed by non-linear feature transformation and the symmetrical function MaxPooling. Both are considered successful approaches in performing state-of-art performance on object classification and semantic segmentation.

6.1 Future of Computer Vision in Autonomous Driving

The progress achieved in the introduced DL techniques in this work is undeniably very promising and has a lot of potential in the AD field. But the reliance only on the LiDAR as a vision sensor does not deliver all the requirements

needed for a fully independent Autonomous Driving systems i.e. level 4 or 5. The texture information is vital in a fully reliable object detection technique. Differentiation between a rubber tire or a plastic bag cannot be inferred from the information contained in point clouds. Adding to that the weather elements as snow or rain can obscure the emitted/received photons from the LiDAR sensors. Radars in contrast posses the robustness against such weather conditions. Large scale neural network models are vital in the design of a reliable driving system. Since they are trained of data in the number of billions. As well as they have over hundred features. In other words, they are able to process and preform on much more information and preform the imposed CV tasks. Many approaches like sensor fusion and domain adaptation are employed to address the augmentation of information and the generalization of the learned features. To provide the DL models some immunity against the changing weather conditions and the lack of visual information provided by a single vision technology.

References

1. Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Bjorn Ottersten. A survey on deep learning advances on different 3d data representations. 2019.
2. Saifullahi Aminu Bello, Shangshu Yu, and Cheng Wang. Review: deep learning on 3d point clouds. 2020.
3. Ross Girshick. Fast r-cnn. 2015.
4. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014.
5. David Griffiths and Jan Boehm. A review on deep learning techniques for 3d sensed data classification. *Remote Sensing*, 11(12):1499, 2019.
6. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. 2018.
7. Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
8. Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. 2018.
9. Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. 2017.
10. Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. 2017.
11. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. 2016.
12. Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. 2017.