Applied Intelligence (2023) 53:6844–6855
https://doi.org/10.1007/s10489-022-03930-5

Semantic segmentation of 3D LiDAR data using deep learning: a review of projection-based methods

Alok Jhaldiyal [1] · Navendu Chaudhary [1,2]

Accepted: 23 June 2022 / Published online: 11 July 2022

Abstract
LiDAR sensor is an active remote sensing sensor that is increasingly used to capture 3D information of real-world objects. Real-time decision-making applications such as autonomous driving heavily rely on 3D information to navigate an urban environment. LiDAR data processing is, however, very complex and resource-intensive. Deep learning on point cloud is a recent advancement that is aimed to extract 3D information. Deep learning implementations include procedures where raw points are fed to neural networks and converted to 3D voxels. Individual voxels are fed to 3D convolutional layers and techniques that transform the 3D points into 2D images and utilize the well-established 2D CNNs. Of these, the two former methods are majorly reviewed, while the projection-based methods are less reviewed although the technique is widely used in numerous applications. To fill the gap, this paper examines the existing literature on projection-based methods by detailing the recent progress made. Identifying the state-of-the-art methodology and summarizing the important interventions are among the significant tasks covered in this paper.

Keywords LiDAR point cloud · Deep learning · Semantic segmentation · Projection-based methods

1 Introduction

A classified environment is a critical element for several decision-making systems. LiDAR (Light Detection and Ranging) has been extensively used to classify the objects in a scene. How the points are considered drives the various techniques for point classification. These techniques fall into three domains, namely treating points as 3D points, treating points as voxels, and projecting the point cloud and its information on a 2D plane as images. The former two have very large computational overheads that require numerous resources, especially for real-time applications.

Whereas, the projection-based methods serve a dual purpose. Firstly, they reduce the computation overhead. Secondly, the properties of LiDAR, such as intensity, are used as information in feature classification. Many researchers have reported this method to be highly accurate although slightly less than the point- or voxel-based approaches, which is a trade-off. The projection-based classification research considers a single or fusion of various projection approaches that drive the information extraction by well-established machine learning models. Even the less reviewed projection-based techniques have segmentation accuracies that are at par with the existing techniques. The lower computation overhead and memory requirements enhance the adaptability and applicability of these techniques to various applications, including real-time applications. Motivated by these, this review aims to present a detailed analysis of projection-based techniques and strives to make the following contributions:

1) Present a detailed and comprehensive review of projection-based methods, and compare them with other existing approaches.
2) Identify the existing state-of-the-art methodology adopted in the past and present studies.
3) Track the evolution of methods by evaluating the deep learning architecture and interventions incorporated.

✉ Alok Jhaldiyal
ajhaldiyal@ddn.upes.ac.in

Navendu Chaudhary
navendu@sig.ac.in

[1] School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India

[2] Symbiosis Institute of Geoinformatics, Symbiosis International (Deemed University), Pune, India

2 Background

2.1 LiDAR point cloud

3D LiDAR is a prominently used sensor that captures the distance information from the surrounding environment. LiDAR instrument comprises a laser setup, scanner, and specialized Global Positioning System (GPS). The common platforms on which LiDAR instruments are mounted include airplane-, car-, and ground-based setups referred to as Airborne Laser Scanner (ALS), Mobile Laser Scanner (MLS), and Terrestrial Laser Scanner (TLS), respectively. LiDAR uses a pulsed laser to target an object and measures the time when the reflected light is received back. The points where the LiDAR pulsed laser hits are captured and a cumulative set of such captured points is called a point cloud. The set of captured points has a respective measure of laser scan, i.e., distance information and associated 3D coordinate values (x,y,z). Qi, Su, et al. [1] defined point clouds as "Simple and unified structures that avoid meshes' combinatorial irregularities and complexities." Point cloud data finds applicability in near real-time applications, including autonomous driving, remote sensing, and augmented reality; thus, the processing needs to be fast and efficient.

The processing of LiDAR point cloud is quite challenging. This is due to the sheer size of points, the dataset's randomness and lack of structure, and the occlusion that infests most LiDAR scans. Currently, point cloud utilization for 3D information extraction is a hot area of research, such that all the progress accomplished thus far is not a decade old.

2.2 Deep learning of point clouds

Earlier methods for working with point cloud required highly trained engineers to tediously retrieve information using handcrafted methods. These handcrafted techniques required well-optimized approaches considering the massive size of data and presence of noise in the data. Having processed the point cloud data with handcrafted methods, the processed data often needed to be optimized before it could be used by the target application [2, 3]. Handcrafted methods were tedious and dependent on the engineer's intelligence and expertise. This was unsuited for most modern applications that benefit from 3D data and have near real-time information requirements with high precision.

Deep learning has attained notable success in object classification, semantic segmentation, and object detection with 2D images (Yang et al., 2021; Liu et al., 2022; Huang et al., 2021; Sun et al., 2021). Ninety percent of the advances in computer vision and machine learning apply to 2D images. Significant advances made by deep learning in 2D images have led to the application of deep learning to 3D LiDAR data, which has been attracting more and more attention [4, 5]. The past five to six years have seen unprecedented growth in the use of deep learning in LiDAR point cloud data, and this has mainly been due to the firm representative learning that deep learning models are capable of doing. The availability of open benchmark datasets of point cloud data such as ScanObjectNN [6], ModelNet [7], ShapeNet [8], PartNet [9], ScanNet[10], Semantic3D [11], and Kitti vision benchmark dataset [12, 13] have played a key role in driving the research that has happened in recent years. Table 1 summarizes the available open 3D LiDAR datasets for semantic segmentation by giving details such as the number of captured points, classes present, size of data, and category of scanning sensors. Deep learning models train and exploit the semantic and deep feature representations directly from the data, thus requiring no handcrafted methods unlike the earlier methods of point cloud processing. However, compared to 2D images, the application of deep learning on 3D point cloud data is quite challenging since this data is highly unstructured, heterogeneous, and high dimensionality [4, 22–24]. Deep learning methods mainly attempt to solve the following problems: identifying the representation of a high-density point cloud from a sparse point cloud; making a size and permutation invariant network; and optimizing the processing of a large amount of data for time and resource usage [25]. Occlusion due to cluttered scenes, blind spots, unintended points, and point misalignment are the other issues that need to be addressed [26].

The workflow of 3D point cloud processing includes the pre-processing and registration of raw point data, classification, object detection, tracking, and point cloud segmentation. 3D shape classification understands the point and then assigns a global descriptor shape to the point. 3D object detection utilizes a 3D detector to detect an object in the point cloud and orient a bounding box around it. 3D object tracking estimate the position of the object in subsequent frames by knowing the position in the current point cloud frame. Semantic segmentation separates the points from a point cloud to different subsets or classes. It mainly identifies the parent class of a point. It is technically defined as an approach to attain a fine-grained understanding of the captured 3D frames and annotate a precise class label for each point. Semantic segmentation has considerably impacted the success of deep learning, as the labelled data to train a deep learning model is retrieved from the semantic segmentation of point cloud. Currently, much research on the semantic segmentation of LiDAR data is ongoing and still full of challenges (Sun et al., 2022; Duan et al., 2021; Bai et al., 2022). Semantic segmentation has a few challenges: size of the data and large-scale segmentation are significantly challenging; it is a resource-intensive task that requires immense memory and computation; and the heterogeneity of the point cloud is a challenge, as it has no structure and is entirely disordered. State-of-the-art deep learning methods on point cloud can be categorized as point- [1, 27–34], voxels- [35–39], and projection-based methods (Wang, 2018c; [40–44]).

Table 1 Available datasets for 2D LiDAR semantic segmentation. The number of classes in the bracket demote the annotated classes, while the rest are classes used for evaluation. '-' implies that the data was unavailable

S. No.	Dataset Name	Year	Scanned Points Count	Scanned Classes Count	No. of Scans	Spatial Size	Sensors
1	Oakland [14]	2009	1.6 M	5 (44)	17	–	MLS
2	ISPRS [15]	2012	1.2 M	9		–	ALS
3	Paris-rue-Madame [16]	2014	20 M	17	2	–	MLS
4	IQmulus [17]	2015	300 M	8 (22)	10	–	MLS
5	ScanNet [10]	2017		20 (20)	1513	8×4×4	RGB-D
6	S3DIS [18]	2017	273 M	13 (13)	272	10×5×5	Matterport
7	Semantic3D [11]	2017	4000 M	8 (9)	15/15	250×260×80	TLS
8	Paris-Lille-3D [19]	2018	143 M	9 (50)	3	200×280×30	MLS
9	SematicKITTI [12]	2019	4549 M	25 (28)	23,201/20351	150×100×10	MLS
10	Toronto-3D [20]	2020	78.3 M	8 (9)	4	260×350×40	MLS
11	DALES [21]	2020	505 M	8 (9)	40	500×500×65	ALS

2.2.1 Point-based methods

Point-based methods directly work on the unstructured and irregular point clouds. These methods directly interact with the points and take individual points as input and output a labelled point or label the entire point cloud. Point-based methods can be categorised as PointNet- [1, 30], deep convolutional neural network- (CovNets-based) [31, 45], and autoencoder-based (AE-based) methods [46–48]. PointNet-based methods are widely used and have acted as a benchmark for the studies that followed. PointNet [1] is a pioneering work and breakthrough that opened deep learning for direct work with points without rendering them to voxels or 2D images. PointNet used the max-pooling function with each layer in the network by learning an optimization function and aggregating the optimized values to a global descriptor. PointNet was followed by PointNet++ [30], an improved version of PointNet that included the ability to capture local structures, which was lacking in the previous version. However, point-based methods fail to consider the pre-existing correlation between the local points. Working with raw point cloud data is highly challenging, as the point cloud is irregular, unstructured, and disordered, thus having high processing and time complexities, such that their use in near real-time applications is limited.

2.2.2 Voxel-based methods

Voxels-based methods involve the breakdown of the entire point cloud into 3D regular cubes called voxels or volume elements and the application of a 3D Convolution Neural Network (CNN) on each voxel. The workflow of voxel-based methods comprises two phases, namely offline and online [22]. The offline phase marks the conversion of point cloud data into fixed-size 3D voxels, and a normal vector is added to discriminate the individual voxels. The online phase marks the start of learning by involving a CNN that is designed with a set of convolutional, max-pooling, and other fully connected layers. Although voxel-based methods are efficient, they are enormously computation-intensive and have high memory requirements. Further, voxel-based methods fail to capture the inherent natural structure of the raw 3D point cloud in the voxels that it creates [49].

2.2.3 Projection-based methods

Projection-based methods benefit from the well-established and matured 2D CNNs. These methods project the 3D point cloud to 2D images and then identify the class labels by running projected custom images through existing 2D CNNs. Having labelled each point, the 2D images are projected back to the respective point cloud. The projection of the point cloud is crucial, as it decides the fate of the final segmentation result. Methodologies exist where the 3D point cloud is projected to 2D images from multiple virtual camera views, referred to as Multi View-Based methods, or the 3D point data is projected to 2D images based on different projection techniques, referred to as projection-based methods. The 3D point cloud projection to the 2D plane is based on some established projection techniques, i.e., Spherical projection [42] and Birds Eye View (BEV) projection [50].

Projection-based methods rely on compact representation, i.e., the 2D grid, and can capture local geometry as well [22]. Considering real-time applications, the high computational cost incurred by transforming data to voxels or working with raw point data renders these methods unfeasible [40, 51]. However, the projection-based techniques utilize the matured 2D deep learning approaches that are fast enough to cater to the requirements of the current applications with near real-time processing needs. These techniques are less

computation-intensive, and, thus, do not heavily rely on Graphical Processing Units (GPU) or other high-end machines.

2.3 Evaluation metrics

Several evaluation metrics have been proposed to evaluate the accuracy of segmentation and other processing tasks. The most commonly used metrics for segmentation includes the Intersection of Union (IoU), Overall Accuracy (OA), and mean IoU (mIoU). IoU is a metrics to evaluate the accuracy of the object detector and define the percentage overlap between the mask and prediction output.

$$IoU_n = \frac{TP_n}{TP_n + FP_n + FN_n} \tag{1}$$

$$OA = \frac{\sum TP_n}{Total\ number\ of\ points} \tag{2}$$

$$mIoU = \frac{\sum IoU_n}{N} \tag{3}$$

Here, N is the total number of labels, TP is the total number of points for true positives, FP is the number of points for false positives, and FN is the number of points for false negatives, where n is the nth label. IoU is used to estimate the segmentation accuracy of each class, whereas OA and mIoU are used to evaluate the overall accuracy of segmentation.

3 Evaluation of point cloud segmentation techniques

Semantic segmentation of 3D point cloud data has evolved from handcrafted methods to deep learning-based methods, thus resulting in significant improvement of the segmentation accuracies. Deep learning on point cloud is difficult and complex. The methods that exist, i.e., point-, voxel-, and projection-based methods, have tried to ease out the complexity of processing. The large size of the unstructured data, large memory and high computation needs, and most importantly, the area-specific requirements of the application immensely impact the segmentation process. When we talk of application-specific requirements, we refer to applications that might need real-time segmentation results, such as autonomous driving. When dealing with real-time applications, the processing time is pivotal. Point-based methods consume the raw point cloud data and learn the feature class of each point independently. Thus, it has significantly higher memory and GPU requirements. Voxel-based methods break the dataset into smaller 3D voxels and segment them voxel by voxel to reduce the computation requirements; however, the memory requirements are very high. Furthermore, when the data is split into smaller voxels, the geometrical relationship among points is compromised. Finally, projection-based methods project point clouds in a 2D plane using a suitable projection technique. The lack of labelled training data essentially restricts deep learning on point cloud, which is what projection-based methods try to work out. Projection-based techniques exploit the well-established capability of 2D CNNs. The projected images are passed through CNN to segment, and once the respective labels are identified, the images are projected back. Table 2 compares the different deep learning-based point cloud segmentation techniques by analysing some state-of-the-art methods.

3.1 Algorithmic approaches to point cloud segmentation methods

This section discusses the different deep learning architectures used to segment a point cloud by referring to some benchmark studies. To elaborate on these methods, we refer to PointNet [1] architecture for project-based methods, Voxnet [52] for voxel-based methods, and FPS-Net [53] model architecture for projection-based techniques.

3.1.1 Point-based methods

PointNet is the pioneering work that segments point cloud without transforming the raw data to any intermediate form like voxels or images. PointNet takes n points as input where each point vector P_i is a vector of x, y, and z coordinates. Model outputs n × m scores where m represents the different classes. The complete architecture of PointNet can be classified into three major components: a max-pooling layer, which aggregates information of all the input points; a structure to combine local and global information; and two alignment networks to combine input points and features, as seen in Fig. 1. The max-pooling layer is a symmetric function to aggregate features. The function handles key objectives including sorting the input in canonical order, training RNN, and augmenting all the permutations in the training data. MLP uses the global signature that is generated by the max-pooling layer for classification, and the global signature can be used to combine local and global information for segmentation as well. The alignment network handles any variance in the predictions by aligning the input that is set to a canonical form before feature extraction.

3.1.2 Voxel-based methods

Figure 2 shows the broad architecture of VoxNet. It has two components, namely a volumetric grid that identifies the spatial occupancy, and 3D CNN to predict class labels from the occupancy grid. The occupancy grid is a lattice of voxels with an estimate of spatial occupancy. VoxNet uses dense arrays in small volumes of 32 × 32 × 32 voxels as an input to CNNs.

Table 2 Comparisons of OA (Overall Accuracy) and mean IoU (mIoU) of some point-, voxel-, and [1] projection-based state-of-the-art methods

Segmentation Method	Model Name	Input	Overall Accuracy	mean IoU
Point-based	PointNet	50 K Points	89.2	14.6
	PointNet++	50 K Points	90.7	20.1
	PointCNN	1024 points	92.2	–
Voxel-based	3D Shapenet	voxel	84.7	–
	VoxNet	voxel	85.9	–
	VoxelNet	voxel	–	12
	OctNet	voxel	86.5	–
Projection-based	Rotation Net	12 views	97.37	–
	MVCNN	12 views	90.1	–
	SqueezeSegV2	64X2048 pixels	–	39.7
	RangeNet++	64X2048 pixels	–	52.2
	SqueezeSeg	64X2048 pixels	–	29.5
	SalsaNext	64X2048 pixels	–	59.5
	SalsaNet	64X2048 pixels	–	45.4

After finalizing the voxel reference frame and resolution, the occupancy of the grid is computed. VoxNet uses three approaches, namely binary occupancy grid, density grid, and hit grid, to estimate the occupancy. The occupancy grid is passed to a feed-forward CNN that comprises the following: an input layer where each value of the grid cell of the occupancy model is updated so that it is in range; convolutional layers that take four dimensions as input for which three dimensions are spatial, and one is the feature map, which generates f feature maps where f is the number of learned filters; pooling layers that does the downsampling of data; and fully connected layers that has n neurons, where the output of each neuron is a learned combination. ReLU's are used to output the class labels directly.

3.1.3 Projection-based methods

In FPS-Net, three projected images, i.e., point coordinates, point depths, and point intensities, are generated using the spherical projection method. The intervention in FPS-Net is that it separately considers the modality of each image and runs them, first through a fusion network and then through a 2D convolutional network. A multi-receptive field residual dense block (MRF-DB) is a block of features extractor through which each modality image is processed and aggregated to a dense contiguous structure. The structure is passed through a 1×1 convolution to reduce the dimensionality. A network of encoders and decoders process the fused feature. While the encoder comprises MRF-DB and downsampling of layers, the decoder block has recurrent convolutional block (RCB) and upsampling layers, as shown in Fig. 3. Lastly, the 2D information is mapped to a 3D point, and labels are assigned to each class.

4 Deep learning on projected datasets

Studies that apply deep learning on projected point cloud data for segmentation have predominantly used the methodology with multiple phases: projecting 3D point cloud to synthetic 2D images; training deep learning model using the generated 2D images; pixel-wise segmentation and labelling of classes;

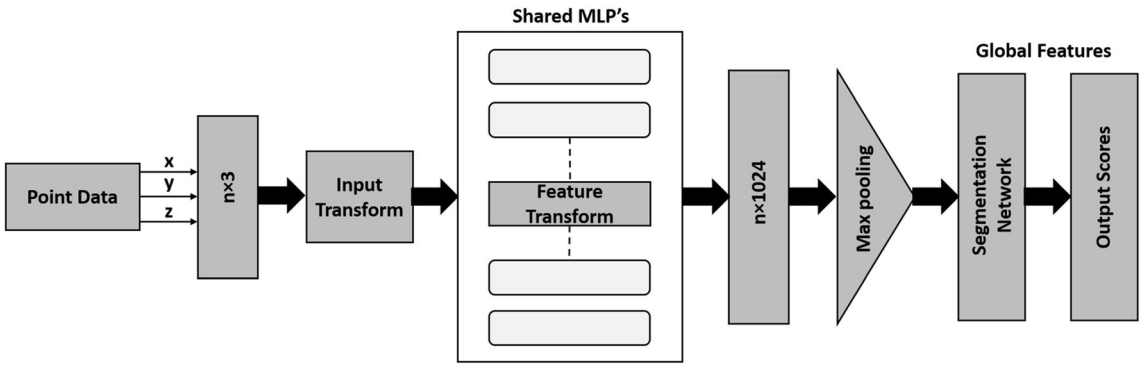
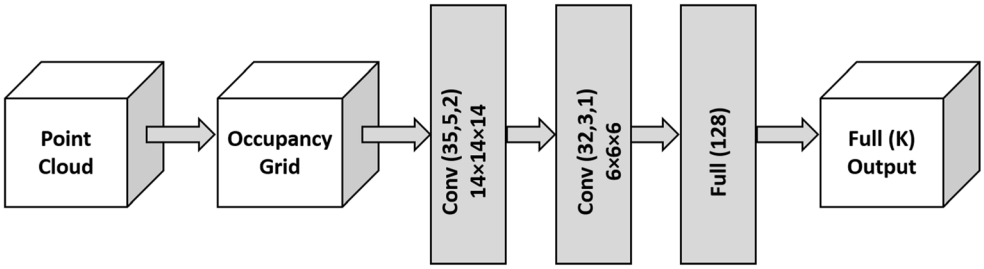


Fig. 1 PointNet Architecture

Fig. 2 VoxNet Architecture



and projecting the segmented data back to 3D point cloud data. Projection of the 3D point cloud a crucial step that determines the overall performance of the complete segmentation process. As the projection-based methods benefit from established deep learning architectures of 2D structured data, the rest of the stages of model training and segmentation phases are not that challenging. At the core of projection-based techniques lie the 3D to 2D projection techniques. The performance of the whole model depends on the efficient and effective projection of 3D point cloud into 2D image space. Any loss of structure and intrusion by noise will adversely affect the training of the model and the identification of the existing local patterns.

4.1 2D projection techniques

Image projection involves the mapping of the spherical surface view of the scanned environment to a 2D image by projecting the spherical coordinates (θ, ϕ, r) to 2D image coordinates (x,y). Predominant projection techniques used to project 3D point cloud to 2D images include Spherical projection [42], Bird's eye view projection [50], and Projection fusion-based hybrid techniques.

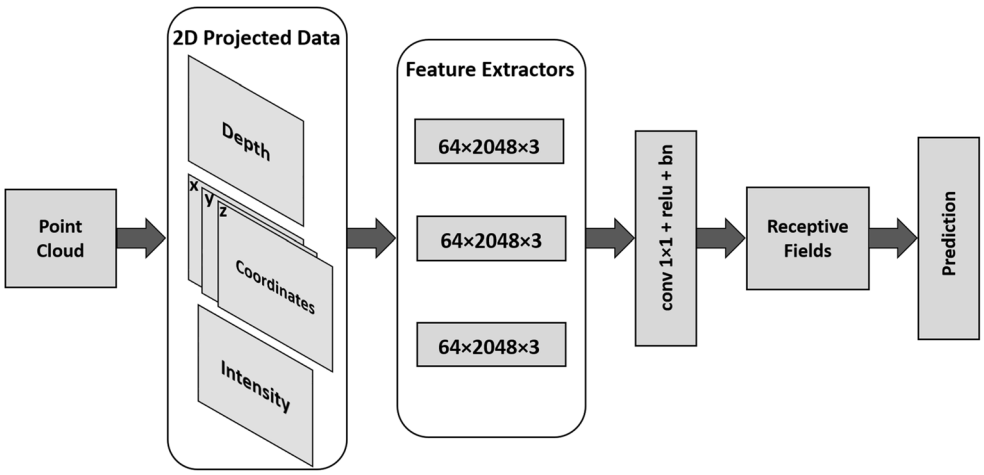
4.1.1 Spherical projection

Spherical projection, also called front view projection, is widely used in deep learning solutions to reduce the dimensionality of 3D LiDAR data. Lasers scan the environment at 360° that leads to a geometric shape of a hollow cylinder with LiDAR at its centre and captured points on the curved surface. When the curved surface of this hollow cylinder is projected perpendicular to the principal axis on a plane, it results in a spherical projection image. The translation to pixel coordinates is a multi-step process; the pixel coordinates are computed by applying mathematical procedures to spherical coordinates in the first step. The next step involves translating the origin, which currently is at the centre of the image. The origin is moved from the centre of the 2D projected image to the top left corner of the image, and as in computer vision, the norm is to have the *origin* there. Further, this step involves normalizing and scaling, thus ensuring that the image captures most of the relevant points and is usable to the deep learning model, and as with the convolutional network, image size is preferred to be in powers of two. Having computed the pixel coordinates for each point, the point information is encoded into it. The encoded information includes the point's x, y, and z coordinates. The point intensity and distance or range of points from LiDAR are added to the pixel, thus resulting in the final image that is fed to the CNN.

4.1.2 Bird's eye view projection

Bird's Eye View (BEV) projection was first introduced in 2018 by Beltrán et al. [50]. Novel BEV projection was part of a perception system called BirdNet [50], a 3D detection and

Fig. 3 FPS-Net Architecture



classification system that displays the BirdNet's framework. The LiDAR point cloud is transformed into a BEV image that is N × N meters, has a cell size of δ, and is encoded as a 3-Channel image that includes height, intensity, and density. Usually, for each vertical level of LiDAR, a height map is generated that results in multiple slices of the height map. The intensity refers to the reflectance value of the point. Density refers to the number of points per cell, and it needs normalization, as it is directly proportional to the number of laser beams fired by the LiDAR. This is a significant limitation: it renders BEV projection useful only for training some neural networks, and not for any other use of LiDAR sensors. The normalization of density is done by performing an intersection analysis of an image with same resolution as that of the BEV image, considering the maximum possible points in each cell and the actual number of points captured in the field of view of the laser beam.

4.1.3 Fusion of projected datasets as training data

The fusing of multiple projections aims to recover the data loss that is incured due to the projection of 3D data to 2D images. This strategy is a recent advancement that renders the projection-based methods more effective by countering the information loss incurred due to the projection technique applied. Alnaggar et al. [51] proposed a Multi-Projection Fusion (MPF) framework that analyses the spherical and BEV projection on the KITTI dataset. CNNs are separately trained using the spherical and BEV projected datasets, and later, the segmentation results of both are combined. This approach achieved higher accuracy than the state-of-the-art methods such as RangeNet++ [40] and PolarNet [44].

4.2 Projection-based point cloud segmentation models

Various studies have utilized projected LiDAR data for training deep learning models to segment the point cloud data. The subsequent sections discuss the different state-of-the-art methodologies adopted and the interventions introduced to enhance the segmentation and runtime accuracy. Table 2 compares the studies on critical parameters including the projection techniques, dataset used, and methodological intervention.

4.2.1 *SqueezeSeg* [42]

Wu et al. (2018) implemented an end-to-end pipeline named SqeezeSeg based on CNN, used a 3D point cloud projected as 2D images, and outputted the labelled point-wise data. Moreover, in its pipeline, it used a Conditional Random Field (CRF) to refine the segmented points further. The LiDAR scanned points were sparse and irregular, so voxelization was avoided, as it would have led to many empty voxels. Spherical projection was used to project 3D points to compact 2D images. A tensor of 64*512*5 was used as input, with 64 LiDAR data vertical channels, 512 front view grids, and 5 feature classes. Fire modules [54] were used to reduce the parameter size, and CRF was used to refine the labelled points. Datasets used were KITTI dataset, and the training samples were collected from GTA V using a virtual LiDAR mounted on a car. The class-level accuracy attained is mentioned in Table 3.

4.2.2 *SqueezeSegV2* [43]

Wu et al. (2018) implemented an end-to-end pipeline named SqeezeSeg based on CNN, used a 3D point cloud projected as 2D images, and outputted the labelled point-wise data. Moreover, in its pipeline, it used the CRF to refine the segmented points further. The LiDAR scanned points were sparse and irregular, so voxelization was avoided, as it would have led to many empty voxels. Spherical projection was used to project 3D points to compact 2D images. A tensor of 64*512*5 was used as input, with 64 LiDAR data vertical channels, 512 front view grids, and 5 feature classes. Fire modules [54] were used to reduce the parameter size, and CRF was used to refine the labelled points. Datasets used were KITTI dataset, and training samples were collected from GTA V using a virtual LiDAR mounted on a car. The class-level accuracy attained is mentioned in Table 3.

4.2.3 *RangeNet++*[40]

RangeNet++ is a projection-based approach proposed to segment point cloud data, and the projected 2D images, referred to as range images, are generated by the spherical projection method. RangeNet++ improvises from the previous techniques by considering the skewness in the scanned data due to the lack of coordination between the moving vehicle and rotating LiDAR sensor. Segmentation in RangeNet++ relies on CNN and utilizes the encoder-decoder hourglass-shaped architecture that is similar to that of Wu et al.'s (2018). The decoder uses modified DarkNet [58] backbone architecture, thus allowing the usability of aspect ratios other than square aspect ratios. When reconstructing point cloud back from the rendered images, all the original points obtained during the rendering are considered as sampled points that are substantially lower than the actual scanned points. Post segmentation, it is normal to get blurry images as an output of inference that requires intensity rendering. RangeNet++ replaces CRF used in SqeezeSeg [42] and SqueezeSeg V2 [43] by GPU-based calculations for the nearest neighbour search on the complete point cloud. RangeNet++ outperforms most previous models in both accuracy and runtime.

Table 3 Projection-based methods to date, including the dataset used, 3D point to 2D image projection method, and major interventions that stepped up the efficiency

S. No.	Network Name	Year	Dataset Used	Projection Method	Major Intervention
1	Squeeze Seg (B. [42])	2018	KITTI (Andreas [13]) Self-Recorded (Grand Theft Auto V Video Game)	Spherical	Used CRF as a post-processing method to refine results
2	PointSeg (Wang, 2018)	2018	KITTI (A [55])	Spherical	Used RANSAC to remove outliers and improve results
3	Squeeze Seg V2 (B. [43])	2019	KITTI (Andreas [13]) Self-Recorded (Grand Theft Auto V Video Game)	Spherical	Context Aggregation Module(CAM) is proposed to immunize from dropout noise
4	RangeNet++ [40]	2019	KITTI Odometry Benchmark [12, 13]	Spherical	Replaces CRF with GPU based calculations
5	VolMap [41]	2019	Scala Valeo Laser scan KITTI (Andreas [13])	BEV	Combined BEV projection methods with voxelization to create VolMap using which the model was trained.
6	PolarNet [44]	2020	SemanticKITTI [12] A2D2 [56] Paris-Lille [19]	Modified BEV	Introduced Polar Bird's Eye View Projection to handle uneven distribution of points by BEV Projection
7	SalsaNet [57]	2020	KITTI road benchmark dataset (Andreas [13])	BEV Spherical	Consider classes with more instances in the KITTI dataset and use an auto annotating module.
8	MPF [51]	2021	Semantic KITTI [12]	Fusion of BEV and Spherical	Combined the segmentation results of CNN trained by BEV projected dataset and spherical projected dataset.
9	FPS-Net [53]	2021	Semantic KITTI (Andreas [13])	Spherical	Trained the network, considering the modality gaps in projected images.

4.2.4 *VolMap* [41]

VolMap, proposed by Radi and Ali [41], attempts to overcome the occlusion that is introduced by spherical projection. In spherical projection, when a single point of reference is considered, the points of one object occlude those of another. VolMap is inspired by VoxelNet [39] and uses bird-eye-view projection and the concept of voxelization to create a volumetric bird-eye-shaped view where the height represents the different LiDAR layers. This volumetric map is used as input to train the model and further segment the point cloud. Comparative analysis of VolMap revealed 20 times higher accuracies, compared to PointNet++ [30], which uses raw point cloud data that directly comes from the sensor. There was a 60% improvement in the accuracies in the segmentation of classes and a 50% higher run time, compared to pure, spherical projection-based methods.

4.2.5 PolarNet [44]

PolarNet, proposed by Zhang et al. [44], introduces a more convenient LiDAR representation, i.e., the polar coordinate system that considers points irrespective of the irregular spatial distribution. The motivation behind PolarNet was the flaws that existed in Bird's Eye View projection where the projection points are organized as rings of varying radius, and when a standard Cartesian coordinate system is applied, the points get irregularly distributed. A grid close to the sensor has a dense accumulation of points hiding the fine details, and a grid far from the sensor has less sparse points. This difference in the spatial distribution of points hindered the effectiveness of segmentation. Instead of assigning points based on the Cartesian coordinate system, the proposed polar bird's eye view projection calculates the azimuth and radius of each point by keeping the sensor position as the origin and assigns points to the grids. Polar bird's eye view projection evenly distributes the scanned point, and the payload on the deep learning model is lessened by the even distribution of points. PolarNet reported higher segmentation accuracies, compared to other state-of-the-art methods, as mentioned in Table 1. Moreover, significant improvements were reported in per-class segmentation accuracies.

4.2.6 PointSeg [59]

PointSeg, proposed by Wang [59], used SqueezeNet [54] as a base used in SqueezeSeg [42] and combines RGB segmentation methods to implement fast semantic segmentation. The 3D LiDAR dataset used was the benchmark KITTI dataset (Andreas [13]), and the images were generated using spherical projection methods. The PointSeg utilized Random Sample Consensus (RANSAC) during the projection of spherical images back to point cloud data to remove outliers. During post-processing, RANSAC enhanced the segmentation efficiency from that of state-of-the-art methods, mainly SqueezeSeg [42].

4.2.7 *SalsaNet* [57]

Aksoy et al. [57] proposed "SemAntic Lidar Data Segmentation Network," i.e., SalsaNet, an encoder-decoder architecture having consecutive ResNet [60] blocks those segments of BEV projected point cloud data. The segmentation process only emphasizes on-road and vehicle segments and excludes pedestrians and cyclists due to fewer instances in the KITTI dataset. SalsaNet introduces auto-labelling to increase the annotated dataset using MultiNet [61], a network already trained on the KITTI road benchmark dataset (Andreas [13]). SalsaNet is trained separately on images projected using BEV projection and spherical projection to analyse how the projection results vary. The results achieved are projection invariant and have higher accuracy with both projection methods. The comparison of SalsaNet is made separately in Table 3, as the segmented classes vary from those considered in other studies.

4.2.8 *FPS-net* [53]

FPS-Net leverages from the fact that the images generated after projection, i.e., an image of point coordinates (x,y,z), point intensity, and point depth, are processed without understanding their inherent characteristics or modality. However, each image channel has gaps in modalities that lead to less efficient segmentation, as the CNN processes them with the same kernels. FPS-Net treats each image separately, learns from each image modality, and in the final stages, fuses the modality learned features. FPS-Net achieved higher segmentation accuracies and had lesser run times than the state-of-the-art methods (Tables 4 and 5).

5 Results

Table 3 shows the results achieved by the existing methods on the available 3D LiDAR KITTI dataset, a benchmark dataset for evaluating the quality of 3D point cloud segmentation models. For the ease of feasible comparative analysis, we have considered the same dataset and similar classes for all the identified methods. The key highlights of the investigation in this detailed review are as follows:

- Deep learning on point cloud is challenging, and the methods thus far either directly apply deep learning on 3D points or focus on converting the unstructured 3D data to structured data and then use deep learning methods. This review considered projection-based methods that project 3D points to 2D images before running deep learning architectures. Projection-based methods are preferred, as they are fast, less computation-intensive, and suitable for real-time result applications such as autonomous driving.

Table 4 Comparative 3D segmentation results on 3D benchmark KITTI dataset

S. No.	Network Name	Size	Per Classe IoU (%)			Processor	Runtime	Scan/ Sec
			Car	Pedestrian	Cyclist			
1	Squeeze Seg (B. [42])	64×2048 px	60.9	22.8	26.4	Titan X	8.7MS	66
2	Squeeze Seg-CRF (B. [42])	64×2048 px	64.6	21.8	25.1		13.6MS	55
3	PointSeg (Wang, 2018)	64×2048 px	67.4	19.2	32.7	1080Ti GPU	14MS	–
	PointSeg with RANSAC (Wang, 2018)	64×2048 px	67.3	23.9	38.7		12MS	–
4	Squeeze Seg V2 (B. [42])[BN+M+FL+ CAM]	64×2048 px	73.2	27.8	33.6	–	–	50
5	RangeNet++ 53 [40]	64×2048 px	86.4	36.2	33.6	Quadro P6000	64×2048 px: 26MS	13
		64×1024 px	84.6	27.5	27.7		64×1024 px: 47MS	25
		64×512 px	81	16.8	25.8		64×512 px: 82MS	52
6	RangeNet++ (KNN)[40]	64×2048 px	91.4	38.3	38.8			12
		64×1024 px	90.3	29.6	34.2			21
		64×512 px	87.4	18.2	29.5			38
7	VolMap [41]	64×2048 px	79.71	33.62	53.28	–	25.7MS	–
8	PolarNet [44]	64×2048 px	93.8	43.2	40.2	–	–	–
9	FPS-Net [53]	64×2048 px	98.7	61.5	49.4	GeForce RTX 2080 TI Graphics Card	20.8 FPS	

Comparative analysis is based on the achieved segmentation accuracy that is measured through mean intersection over-Union (IoU) and model running time in milliseconds. The modalities are Conditional Random Field (CRF), Random Sample Consensus (RANSAC), Batch Normalization (BN), LiDAR mask (M), Focal Loss (FL), Context Aggregation Module (CAM), Frame Per Second (FPS), and K-Nearest Neighbor

Table 5 Comparative 3D segmentation results on 3D benchmark KITTI dataset of SalsaNet with SqueezeSeg V1 and V2. Comparative analysis is based on the achieved segmentation accuracy that is measured through mean intersection over-Union (IoU) and model running time in milliseconds

S. No.	Network Name	Size	Per Class IoU			Processor	Runtime
			Background	Road	Vehicle		
1	Squeeze Seg	64×2048 px	97.42	66.86	45.13	NVIDIA Tesla V100-DGXS-32GB GPU	6.77MS
2	Squeeze Seg V2	64×2048 px	97.99	69.47	61.93		10.24MS
3	SalsaNet BEV	64×256 px	98.19	71.67	69.19		6.26MS

- Projection-based methods are new, and all the work has been done recently in the last five years. The availability of open LiDAR datasets in recent years is one of the primary reasons for this. Further, benefiting from the applicability of 3D LiDAR data has driven the research in new application areas including autonomous driving, augmented reality, and urban mobility.
- In most cases, almost all of the reviewed projection-based studies work on outdoor/urban scene classification and use MLS data, which is the outdoor KITTI dataset. Analysing the segmentation accuracies, it is visible that most models have high accuracies in labelling big objects such as cars and substantially lower accuracies in small size objects.
- All the research and innovations on projection-based involves improvisations in the deep learning architectures and working out new or customized projection techniques. The inclusion of MPF, RANSAC, ResNet, and MultiNet, and consideration of the modalities are notable improvements to the deep learning architecture. Using polar coordinates, the fusing of voxelization with projection technique and the fusing of projections before model training are major advances made to improve the projected LiDAR data.
- Most proposed methods have considered small-sized data. The larger LiDAR scene needs consideration, and the techniques need validation on the same. The unavailability of LiDAR data for a larger area and the considerable cost incured in the new survey could be why studies have restricted themselves to small-sized datasets.
- Wide differences in the accuracies can be seen between the different classes of segmented points. This difference can form the basis of identifying the techniques for segmenting the classes of interest if required by any future study.

6 Conclusion

This survey is one of the most updated reviews of deep learning on projected LiDAR data. The focus of this survey was to present the current status quo of the existing framework and identify the state-of-the-art methods. In the discussion, a detailed background of all relevant theories applicable to the technique were included and the different studies conducted to date were individually detailed. We successfully identified the improvisations made in the projection techniques and deep learning architecture. The survey includes a comparative analysis of all the proposed models based on data size, segmentation accuracies, and runtime. Future research will benefit from this study, as it is one of its kind and reviews the most recent works. The study will aid in identifying a suitable and feasible approach to work with projected LiDAR data, and the gaps identified will guide any new studies.

References

1. Qi CR, Su H, Mo K, Guibas LJ (2017) PointNet: deep learning on point sets for 3D classification and segmentation. Proceedings - 30th IEEE conference on computer vision and pattern recognition, CVPR 2017 2017-Janua: 77–85. https://doi.org/10.1109/CVPR.2017.16
2. Hänsch R, Weber T, Hellwich O (2014) comparison of 3D interest point detectors and descriptors for point cloud fusion. ISPRS annals of the photogrammetry, remote sensing and spatial Information Sciences II–3 (September): 57–64. https://doi.org/10.5194/isprsannals-ii-3-57-2014
3. Liu W, Sun J, Li W, Hu T, Wang P (2019) Deep learning on point clouds and its application: a survey. Sensors (Switzerland) 19(19): 1–22. https://doi.org/10.3390/s19194188
4. Guo Y, Wang H, Hu Q, Liu H, Liu L, Bennamoun M (2019) Deep learning for 3D point clouds: a survey. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 43(12):4338–4364. https://doi.org/10.1109/tpami.2020.3005434
5. Jiang D, Li G, Tan C, Huang L, Sun Y, Kong J (2021) Semantic segmentation for multiscale target based on object recognition using the improved faster-RCNN model. Futur Gener Comput Syst 123:94–104. https://doi.org/10.1016/j.future.2021.04.019
6. Uy MA, Pham QH, Hua BS, Nguyen T, Yeung SK (2019) Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. Proceedings of the IEEE International Conference on Computer Vision 2019-Octob: 1588–97. https://doi.org/10.1109/ICCV.2019.00167
7. Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J (2015) 3D ShapeNets: a deep representation for volumetric shapes. Proceedings of the IEEE computer society conference on computer

vision and pattern recognition 07-12-June: 1912–20. https://doi.org/10.1109/CVPR.2015.7298801
8. Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S et al (2015) ShapeNet: An Information-Rich 3D Model Repository. http://arxiv.org/abs/1512.03012. Accessed 18 Aug 2021
9. Mo K, Zhu S, Chang AX, Yi L, Tripathi S, Guibas LJ, Su H. (2019) Partnet: a large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. Proceedings of the IEEE computer society conference on computer vision and pattern recognition 2019June: 909–18. https://doi.org/10.1109/CVPR.2019.00100
10. Dai A, Chang AX, Savva M, Halber M, Funkhouser T, Nießner M (2017) ScanNet: richly-annotated 3D reconstructions of indoor scenes. Proceedings - 30th IEEE conference on computer vision and pattern recognition, CVPR 2017 2017-Janua: 2432–43. https://doi.org/10.1109/CVPR.2017.261
11. Hackel T, Savinov N, Ladicky L, Wegner JD, Schindler K, Pollefeys M (2017) Semantic3D.Net: a new large-scale point cloud classification benchmark. ISPRS annals of the photogrammetry, remote sensing and spatial information sciences 4 (1W1): 91–98. https://doi.org/10.5194/isprs-annals-IV-1W1-91-2017
12. Behley J, Garbade M (n.d.) SemanticKITTI : a dataset for semantic scene understanding of LiDAR sequences. no. iii. Accessed 18 Aug 2021
13. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 3354–3361. https://doi.org/10.1109/CVPR.2012.6248074
14. Munoz D, Bagnell JA, Vandapel N, Hebert M (2009) Contextual Classification with Functional Max-Margin Markov Networks. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009 2009 IEEE: 975–82. https://doi.org/10.1109/CVPRW.2009.5206590
15. Rottensteiner F, Sohn G, Jung J, Gerke M, Baillard C, Benitez S, Breitkopf U (2012) THE ISPRS BENCHMARK on URBAN OBJECT CLASSIFICATION and 3D BUILDING RECONSTRUCTION. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 1 (September): 293–98. https://doi.org/10.5194/isprsannals-I-3-293-2012
16. Serna A, Marcotegui B, Goulette F, Deschaud JE (2014) Paris-Rue-Madame Database: A 3D Mobile Laser Scanner Dataset for Benchmarking Urban Detection, Segmentation and Classification Methods. ICPRAM 2014 - Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods, 819–24. https://doi.org/10.5220/0004934808190824
17. Vallet B, Brédif M, Serna A, Marcotegui B, Vallet B, Brédif M, Serna A, Marcotegui B, Terramo NP (2015) TerraMobilita / IQmulus Urban Point Cloud Analysis Benchmark To Cite This Version : HAL Id : Hal-01167995 TerraMobilita / IQmulus Urban Point Cloud Analysis
18. Armeni I, Sener O, Zamir AR, Jiang H, Brilakis I, Fischer M, Savarese S (2016) 3D semantic parsing of large-scale indoor spaces. Proceedings of the IEEE computer society conference on computer vision and pattern recognition 2016-Decem: 1534–43. https://doi.org/10.1109/CVPR.2016.170
19. Roynard X, Deschaud J-E, Goulette F, Roynard X (2018) Paris-Lille-3D : a large and HighQuality ground truth urban point cloud dataset for automatic segmentation and classification to cite this version : HAL id : Hal-01695873 Paris-Lille-3D : a large and high-quality ground truth urban point cloud dataset Fo
20. Tan W, Qin N, Ma L, Li Y, Du J, Cai G, Yang K, Li J (2020) Toronto-3D: a large-scale Mobile LiDAR dataset for semantic segmentation of urban Roadways2211. IEEE computer society conference on computer vision and pattern recognition workshops 2020-June: 797–806. https://doi.org/10.1109/CVPRW50498.2020.00109
21. Varney N, Asari VK, Graehling Q (2020) DALES: a large-scale aerial LiDAR data set for semantic segmentation. IEEE computer society conference on computer vision and pattern recognition workshops 2020-June: 717–26. https://doi.org/10.1109/CVPRW50498.2020.00101
22. Bello SA, Yu S, Cheng W, Adam JM, Li J (2020) Review: deep learning on 3D point clouds. Remote Sens 12(11):1–34. https://doi.org/10.3390/rs12111729
23. Yan X, Zheng C, Li Z, Wang S, Cui S (2020) PointasNL: robust point clouds processing using nonlocal neural networks with adaptive sampling. Proceedings of the IEEE computer society conference on computer vision and pattern recognition, 5588–97. https://doi.org/10.1109/CVPR42600.2020.00563
24. Yang Z, Sun Y, Liu S, Jia J (2020) 3DSSD: point-based 3d single stage object detector. Proceedings of the IEEE computer society conference on computer vision and pattern recognition, 11037–45. https://doi.org/10.1109/CVPR42600.2020.01105
25. Haoming Lu, Rey HS (2020) Deep learning for 3D point cloud understanding: a survey. ArXiv Preprint ArXiv
26. Bello SA, Yu S, Wang C (2020) Review: deep learning on 3D point clouds. ArXiv
27. Huang Q, Wang W, Neumann U (2018) Recurrent slice networks for 3D segmentation of point clouds. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, no 1: 2626–35. https://doi.org/10.1109/CVPR.2018.00278
28. Li Y, Bu R, Di X (2018) PointCNN : convolution on X transformed points. no. NeurIPS
29. Liu J, Ni B, Li C, Yang J, Tian Q (2019) Dynamic points agglomeration for hierarchical point sets learning. Proceedings of the IEEE international conference on computer vision 2019-Octob: 7545–54. https://doi.org/10.1109/ICCV.2019.00764
30. Qi CR, Yi L, Su H, Guibas LJ (2017) PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Advances in Neural Information Processing Systems 2017-Decem: 5100–5109
31. Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, Solomon JM (2018) Dynamic graph CNN for learning on point clouds," January. http://arxiv.org/abs/1801.07829
32. Xie S, Liu S, Chen Z, Tu Z (2018) Attentional ShapeContextNet for Point Cloud Recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 4606–15. https://doi.org/10.1109/CVPR.2018.00484
33. Zhao H, Jiang L, Fu CW, Jia J. (2019) Pointweb: Enhancing Local Neighborhood Features for Point Cloud Processing. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2019-June: 5560–68. https://doi.org/10.1109/CVPR.2019.00571
34. Jiang M, Wu Y, Zhao T, Zhao Z, Lu C (2018) Pointsift: A sift-like network module for 3d point cloud semantic segmentation. arXiv preprint arXiv:1807.00652. http://arxiv.org/abs/1807.00652
35. Choy C, Gwak J, Savarese S (2019) 4D Spatio-temporal Convnets: Minkowski convolutional neural networks. Proceedings of the IEEE computer society conference on computer vision and pattern recognition 2019 June: 3070–79. https://doi.org/10.1109/CVPR.2019.00319
36. Graham B, Engelcke M, Van Der Maaten L (2018) 3D semantic segmentation with submanifold sparse convolutional networks. Proceedings of the IEEE computer society conference on computer vision and pattern recognition, 9224–32. https://doi.org/10.1109/CVPR.2018.00961
37. Meng HY, Lin G, Lai YK, Manocha D (2019) VVNet: Voxel vae net with group convolutions for point cloud segmentation. In: Proceedings of the IEEE/CVF international conference on

computer vision, pp 8500–8508. https://doi.org/10.1109/ICCV.2019.00859
38. Tchapmi L, Choy C, Armeni I, Gwak J, Savarese S (2018) SEGCloud: semantic segmentation of 3D point clouds. proceedings - 2017 international conference on 3D vision, 3DV 2017, 537–47. https://doi.org/10.1109/3DV.2017.00067
39. Zhou Y, Tuzel O (2018) VoxelNet: end-to-end learning for point cloud based 3D object detection. Proceedings of the IEEE computer society conference on computer vision and pattern recognition, 4490–99. https://doi.org/10.1109/CVPR.2018.00472
40. Milioto A, Vizzo I, Behley J, Stachniss C (2019) RangeNet ++: fast and accurate LiDAR semantic segmentation. IEEE international conference on intelligent robots and systems, no. i: 4213–20. https://doi.org/10.1109/IROS40897.2019.8967762
41. Radi H, Ali W (2019) VolMap: A Real-Time Model for Semantic Segmentation of a LiDAR Surrounding View. http://arxiv.org/abs/1906.11873
42. Wu B, Wan A, Yue X, Keutzer K (2018) SqueezeSeg: convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. proceedings - IEEE international conference on robotics and automation, 1887–93. https://doi.org/10.1109/ICRA.2018.8462926
43. Wu B, Zhou X, Zhao S, Yue X, Keutzer K (2018) SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud. 4376–82. http://arxiv.org/abs/1809.08495
44. Zhang Y, Zhou Z, David P, Yue X, Xi Z, Gong B (2020) PolarNet : an improved grid representation for online LiDAR point clouds semantic segmentation
45. Kuffer M, Pfeffer K, Sliuzas R, Baud I (2016) Extraction of slum areas from VHR imagery using GLCM variance. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 9(5):1830–1840. https://doi.org/10.1109/JSTARS.2016.2538563
46. He T, Huang H, Yi L, Zhou Y, Wu C, Wang J, Soatto S (2019) Geonet: Deep geodesic networks for point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6888–6897. https://doi.org/10.1109/CVPR.2019.00705
47. Zamorski M, Zięba M, Klukowski P, Nowak R, Kurach K, Stokowiec W, Trzciński T (2020) Adversarial autoencoders for compact representations of 3D point clouds. Comput Vis Image Underst 193:102921. https://doi.org/10.1016/j.cviu.2020.102921
48. Remelli E, Baque P, Fua P (2019) NeuralSampler: Euclidean Point Cloud Auto-Encoder and Sampler. http://arxiv.org/abs/1901.09394
49. Ku T, Veltkamp RC, Boom B, Duque-Arias D, VelascoForero S, Deschaud JE, Goulette F et al (2020) SHREC 2020: 3D point cloud semantic segmentation for street scenes. Computers and Graphics (Pergamon) 93:13–24. https://doi.org/10.1016/j.cag.2020.09.006
50. Beltrán J, Guindel C, Moreno FM, Cruzado D, García F, De La Escalera A (2018) BirdNet: A 3D Object Detection Framework from LiDAR Information. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC 2018-Novem: 3517–23. https://doi.org/10.1109/ITSC.2018.8569311
51. Alnaggar A, Yara MA, Amer K, ElHelw M (2021) Multi projection fusion for real-time semantic segmentation of 3d lidar point clouds. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 1800–1809. https://doi.org/10.1109/wacv48630.2021.00184
52. Maturana D, Scherer S (2015) VoxNet: a 3D convolutional neural network for real-time object recognition. IEEE international conference on intelligent robots and systems 2015-Decem: 922–28. https://doi.org/10.1109/IROS.2015.7353481
53. Xiao A, Yang X, Lu S, Guan D, Huang J (2021) ISPRS journal of photogrammetry and remote sensing FPS-net : a convolutional fusion network for large-scale LiDAR point cloud segmentation. ISPRS J Photogramm Remote Sens 176 (September 2020):237–249. https://doi.org/10.1016/j.isprsjprs.2021.04.011
54. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2017) 50 X FEWER PARAMETERS AND < 0. 5MB MODEL SIZE. 1–13
55. Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision Meets Robotics: The KITTI Dataset. The International Journal of Robotics Research, no. October: 1–6 32:1231–1237
56. Geyer J, Ricou X, Chung AS, Maximilian M, Dorn S, Martin J, Sturm M, Oelker M, Ag A (n.d.) A2D2 : Audi Autonomous Driving Dataset
57. Aksoy EE, Baci S, Cavdar S (2020) SalsaNet: Fast Road and Vehicle Segmentation in LiDAR Point Clouds for Autonomous Driving. IEEE Intelligent Vehicles Symposium, Proceedings, no. Iv: 926–32. https://doi.org/10.1109/IV47402.2020.9304694
58. Redmon J, Farhadi A (2018) YOLOv3: An Incremental Improvement. http://arxiv.org/abs/1804.02767
59. Wang Y (n.d.) PointSeg : real-time semantic segmentation based on 3D LiDAR point cloud
60. He K, Sun J (2016) Deep residual learning for image recognition. https://doi.org/10.1109/CVPR.2016.90
61. Teichmann M, Weber M, Marius Z, Cipolla R, Urtasun R. (n.d.) MultiNet : Real-Time Joint Semantic Reasoning for Autonomous Driving

Alok Jhaldiyal received the M.Tech. degree in remote sensing and GIS from the Indian Institute of Remote Sensing (IIRS), Dehradun, India, in 2015 and is a PhD student at Symbiosis International University, Pune, India. He currently works as an assistant professor at the School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. His current research interests include GeoAI, urban analytics and satellite image processing.