Semantic Segmentation of Point Clouds and Images for Autonomous Vehicles: Trends, Challenges, and Opportunities

Afsana A. Munia, Reshma Kunjumon, Abbas Khosravi, Ibrahim Hossain, Ashikur Rahman

***Abstract*—Although research on self-driving vehicles (SDV) started long ago, developing a robust perception system is still challenging for researchers in academia and industry. Object classification, object detection and tracking, road detection, semantic segmentation, and instance segmentation are all introduced as the perception problems of the autonomous driving platform. Tons of methods have already been proposed to solve these perception problems. Some methods use a single modality (image only or Lidar only), and others use multi-modality. This review paper categorizes all semantic segmentation methods used for self-driving vehicles into three categories depending on their modality. We compare these methods according to their performance and mention the datasets used for evaluation. Some of the most common datasets in this field are also discussed in this paper. We highlight the challenges and scope of future research and discuss the background of this research field so that this paper can be used as a base for a new researcher to start research on semantic segmentation.**

***Impact Statement*—Semantic segmentation is a very useful perception tool. It can be used to get per-pixel semantic labels of the scenarios captured by the sensors of autonomous vehicles, which can then be used to identify potential obstacles, based on which planning and control decisions are made. Although many works have discussed semantic segmentation models, there are not many review papers that consolidate these works from a sensory perspective. In this paper, we analyze single and multimodal model architectures and compare their performances to guide researchers in deciding which direction (Lidar or camera, Lidar and camera) may be beneficial for them. The most frequently used datasets in this arena and the main features of the data are all elaborately discussed. Furthermore, the challenges that researchers may encounter, and the scope of future research are also included. This paper could provide a one-stop source for new researchers working in this area.**

***Index Terms*—Autonomous driving, Semantic segmentation, Multi-modal datasets, Deep-learning, Point clouds.**

I. Introduction

Over the last few decades, enhancing driving safety by decreasing human intervention and increasing autonomous driving technology has been one of the most challenging tasks for researchers. However, after embracing deep learning technologies in the perception phase of autonomous driving, a remarkable change has occurred in the perception algorithm's overall performance and robustness. Autonomous driving perception problem includes object detection and tracking, semantic segmentation, instance segmentation, panoptic segmentation, road detection, etc. (see Fig. 1). Furthermore, the performance of convolutional neural networks has been proven to be state-of-the-art compared to all other deep learning techniques for developing the perception algorithms [1].

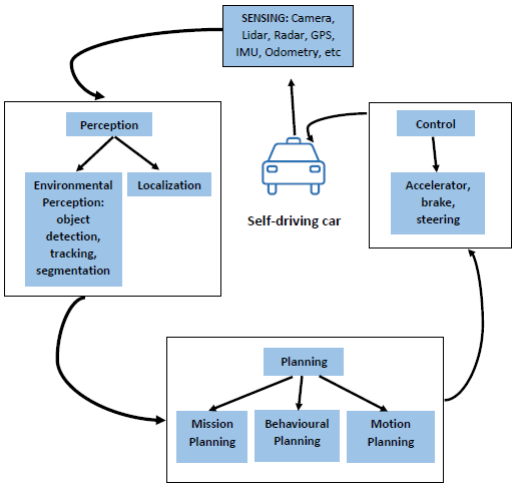

Fig. 1. Different computing modules of self-driving cars.

Identifying and localizing multiple objects in a scene is done by an object detection mechanism, which is one of the significant perception tasks for the autonomous vehicle. Most state-of-the-art methods published for object detection follow either the two-stage or the one-stage approach. The first stage is used for region proposal generation by extracting a scene in the two-stage approach. Then, the second stage is used to classify and further localize the object. On the other hand, in the one-stage approach, no proposal is generated, but instead, a feature map is created to predict bounding boxes. Object detection methods can be classified as 2D object detection methods and 3D object detection methods. However, depth information is required for the various driving task, but the 2D method does not provide depth information. In 2019, Arnold et al. [2] wrote a survey on 3D object detection methods. A one-stage object detector named BANet is proposed by Wang et al. [3] for improving small and multiple object detection in traffic scenes. The BANet combines multichannel attention (MCA) blocks, alpha-effective intersection-over-union losses, and a multiple attention fusion (MAF) module which provides not only improved average precision (AP) for small and multi-object detection but also improved frames per second (FPS).

Afsana A. Munia, Reshma Kunjumon, A. Khosravi, and I. Hossain, are with the Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Waurn Ponds, VIC 3216, Australia (e-mail: amunia@deakin.edu.au) and A. Rahman is from Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh.

We only focus on semantic segmentation in this research, so we skip other perception tasks. Semantic segmentation, also known as dense prediction, labels each pixel of an image with a corresponding class of what is being represented [4]. Long et al. [5] have built a fully convolutional network (FCN) for image segmentation, which could be considered the pioneering work of image segmentation because this is the first work that trained an FCNs end-to-end pixel-wise prediction. Ronneberger et al. [6] proposed a convolutional network named U-Net that performs best on the ISBI challenge to segment neuronal structures. Though U-Net is not trained with self-driving datasets in the original paper, we referred U-Net here to understand the concept of image segmentation using a deep convolution network. However, U-Net works on 2D image data, following the same idea Cieck et al. [7] introduced 3D U-net, replacing all 2D operations with their 3D counterparts. The input and output of this network is a 3D voxel. The proposed method performs well on the Xenopus kidney, which is a complex, highly variable 3D structure. Using U-Net Shojaiee et el. [8] proposed a multispectral Encoder Fused Atrous Spatial Pyramid Pooling (EFASPP) deep network, which consists of two encoders for visible and thermal images and a decoder combining information from the two spectra. The significant contributions of this work include a low-volume, highly-performant multispectral semantic segmentation network for smart vehicles and a new multispectral dataset for night-time traffic scenes since sufficient public datasets are lacking.

Recently, point-based segmentation methods are gaining popularity because it provides depth information. The concept of semantic segmentation is also the same for the point cloud. It separates the points from the point cloud into several subsets according to their semantic meanings [9]. All point-based semantic segmentation methods can be categorized as one of these four: projection-based, discretization-based, point-based, and hybrid methods. [10] and [11] are two examples of projection-based methods. Huang et al. [12] proposed a 3D point cloud labeling scheme using discretization-based methods. PointNet [13] is the first point-based method which directly works on point clouds, [14], and [15] are also examples of point-based methods.

An enormous amount of research has already been done on the perception system of autonomous driving ([16], [17], [18]), but there is still a lot to discover and update. Moreover, solving different types of perception tasks require different kinds of methods. This review paper elaborately discusses one perception task, semantic segmentation, especially in autonomous driving. This paper aims to give a complete overview of semantic segmentation so that new researchers can easily understand ”what has already been done?”, ”where the improvement may be possible?” and ”what are the future research scopes?”.

The significant contributions of this paper are listed below:

1) Providing a comprehensive review of existing datasets and deep learning techniques for semantic segmentation for autonomous vehicles.
2) Discuss the most frequently used datasets for designing and testing algorithms for self-driving vehicles.

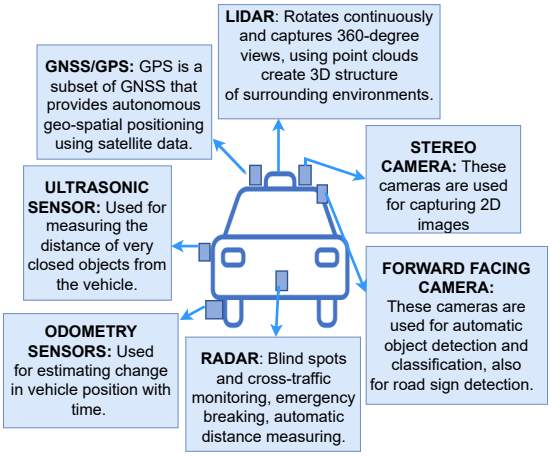

Fig. 2. Measurement and sensing components of a self-driving car.

3) Reviewing and comparing semantic segmentation methods for point clouds and camera images.
4) Analyzing the performance of different algorithms for semantic segmentation.
5) Providing guidelines for future research.

We organize this paper in such a way that a reader can get a complete overview (from sensor selection to segmentation results) of semantic segmentation in autonomous driving. Hence, we include the most common sensors used by self-driving vehicles, their properties and self-driving datasets in Sec. II. We listed a few commonly used datasets in Sec. III to know more about the self-driving dataset. Semantic segmentation, which is the main focus of this research, is discussed in Sec. IV and some comparison of these approaches are shown in Sec. V. Finally, challenges and Future Work is presented in Sec. VI, and a conclusion is given in Sec. VII.

II. Background

This section analyzes different sensing modalities and briefly describes some commonly used publicly available self-driving datasets.

Most state-of-the-art methods have used multiple sensing modalities simultaneously in recent times to accomplish the perception task of the autonomous vehicle. In contrast, most earlier models only used a single sensing modality (mainly depending on camera images). Camera (visual, thermal), LIDAR, radar, sonar, IMU, and GPS are the sensors on which self-driven vehicles rely to understand the surroundings to navigate safely (Fig. 2).

The intelligent autonomous driving system tries to mimic human vision by implementing cameras in various positions, e.g., front, up, back, left, or right, to collect images from different views. Schneider et al. [19] proposed a multi-modal CNN architecture where they used RGB-D images from the Cityscapes dataset [20]. In addition, they evaluate the proposed model for semantic segmentation and object detection. Road marking is an essential requirement of the self-driving car; in

[21], Jang et al. built a lane-level road marking segmentation using a monocular camera. Though the camera is the most commonly used sensor for collecting high-resolution visual data for the perception of autonomous vehicles, it has some limitations. For example, its performance deteriorates during adverse weather conditions and only camera images are not always sufficient for accurate distance measurement of the object.

In recent times, the LIDAR has become one of the most reliable sensors for researchers due to its powerful ability to measure distances or estimate depth. LIDAR follows the same working principle as light, spreads thousands of infrared light beams upon its surroundings, and calculates the reflection time. Using these signals, LIDAR depicts a 3D structure of the surrounding environment for an autonomous vehicle. This 3D structure is nothing but point clouds, which is basically a collection of 3D points in irregular format. PointNet, a highly efficient and effective 3D classification and segmentation network, was proposed in 2017 by Qi et al. [13]. PointNet is the first network that can consume point clouds directly. LIDAR has the quality to measure accurate depth information and is immune to illumination; for this reason, Wang et al. [22] used the LIDAR sensor to provide an effective and fast solution for road boundary detection. Although LIDAR performs far better in low-light conditions than the camera, its implementation cost is ten times higher than the camera.

Radar's working principle is like LIDAR, but radar depends on radio waves rather than light waves. The elapsed time between the emitted wave from the radar to the obstacles is used by the self-driving car for measuring the surrounding object's distance, velocity, and angle. However, they are robust against different lighting conditions and adverse weather, but using only radar for object classification is challenging due to their low-resolution property. Generally, two types of radars are used in autonomous vehicles: Impulse radar and frequency-modulated continuous wave (FMCW) radar [23].

Several other sensors are used for specific functions of self-driving cars, including an IMU (inertial measurement unit) sensor for detecting a vehicle's location and orientation, and a sonar sensor for detecting an automated parking system. In addition, GPS (global positioning system) technology is used for the navigation purpose of the self-driving car, and real-time geographical data received from several GPS satellites are used to calculate longitude, latitude, and speed [24].

A self-driving car attempts to make a robust decision based on real-time data taken from multiple sensors as soon as it understands the surrounding environment. Sensor fusion technology continuously feeds information from multiple sensors to the self-driving vehicle to provide reliable results. However, different companies' test vehicles' sensor setups could differ depending on their requirements and targeted tasks. For example, Waymo, BMW, and Uber do not use the same sensor setup.

Different perception algorithms are built to solve various perception tasks of self-driving cars, such as object detection, semantic segmentation, instant segmentation, etc. These algorithms need some data for training, testing, and evaluation purposes. There are a lot of publicly available datasets from which researchers can choose one or more most appropriate datasets to align with their research area. However, some datasets only provide raw sensor data. Also, few provide annotations and labels, benchmark suites, open-source code and coding tools, stereo, optical flow, SLAM, calibration data, and many more. To know more about "which datasets are suitable for what self-driving algorithms?" we suggest the interested reader go through the paper by Yin et al. [25]. Real driving datasets and synthetic datasets could be used to train and test self-driving vehicles, but we exclude synthetic datasets in this paper. The Dataset section (III) contains further details regarding the publicly accessible datasets.

III. Benchmarks and Datasets

A large number of self-driving datasets with different data formats and sizes are already available, which researchers can use according to their specific research criteria related to autonomous driving. KITTI [26], SemanticKITTI [27], cityscapes [28], and nuscenes [29] are some most popular datasets used for training and testing self-driving algorithms. However, not all datasets used for self-driving cars provide the same data format or traffic conditions; even their sensing modalities may differ. For example, CamVid [30] dataset collects data from the monocular color camera, while [31] and [32] datasets use visual and thermal camera images. The camera is one of the most common sensors for self-driving data collection, and almost every dataset use camera images. At the time of this writing, the Stanford track collection [33] dataset is the one that does not use any camera sensor. Instead, it uses Velodyne 64 LIDAR and Applanix (GPS/IMU) for data collection. Although, most benchmark datasets collect data from multiple sensors. Data collection is typically done by a manually driven vehicle equipped with a set of sensors such as one or more cameras, LIDAR, radar, GPS, and IMU. Sensor placement may differ from one system to another depending on the purpose of the collected data [25].

Table IV depicts a sketch of how datasets can differ according to sensor setup, traffic condition, providing information, etc. We also attach all related research papers with the corresponding dataset in table IV, and these research papers together could draw a complete picture of how these datasets work.

The KITTI Vision Benchmark Suite [26] developed by the researchers of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago in 2012 is the most cited dataset to date. Providing the novel challenging benchmarks for the tasks of stereo, optical flow, visual odometry / SLAM, and 3D object detection, ground truth for individual benchmarks, and diverse set of evaluation matrices makes KITTI [34] popular in the research of autonomous driving. However, there are few relatively new datasets available [35], [29], [28], which consider diverse weather and light conditions as well as collect data from multiple cities. In contrast, KITTI collected data from a mid-sized city in Germany named Karlsruhe on a sunny day. The nuScenes dataset [29] uses the entire sensor suite of autonomous driving such as LIDAR, radar, camera, IMU, and GPS for data collection and supports lots

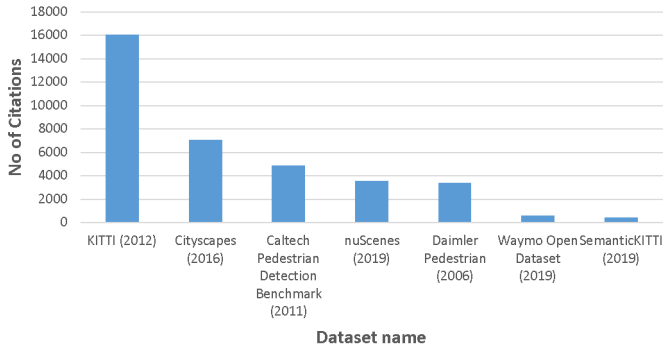

Fig. 3. Some highly cited datasets.

of perception tasks. This dataset was first published in 2019 and has already reached nearly 4000 citations because of its diverse set of locations, times, and weather conditions. They also consider the class imbalance problem and include more scenes with rare classes, such as bicycles. Another popular dataset of 2019 is Waymo open dataset [35], composed of two datasets: the perception dataset and the motion dataset. The perception dataset contains not only the projections but also independently generated labels, 3D bounding box labels in LIDAR data and 2D bounding box labels in camera images. They also provide a collection of 14 key points from across the human body. The expanded version of the perception dataset has released in March 2022. The motion dataset contains 103354 segments, where each segment includes 20 seconds of object tracks at 10Hz and is provided as a shared TFRRecord format file.

In this paper, we only provide the details of six popular datasets, and we mean by the word 'popular' is the highest number of citations achieved by the dataset. To calculate the number of citations, we searched the related research papers on the dataset web page, checked the individual citation from Google Scholar, and summed up the total citation. Fig. 3 shows the total number of citations for specific datasets. From 2012 to the present, the KITTI dataset achieved the highest number of citations.

IV. Semantic Segmentation

In this section, we provide an extensive survey on a variety of semantic segmentation methods used in the perception system of self-driving vehicles. According to the type of data used for segmentation, we organize semantic segmentation into three subsections: Semantic Segmentation from Images, Semantic Segmentation from Point Clouds, and Semantic Segmentation from Multi-modal Data.

A. Semantic Segmentation from Images

For image segmentation, convolutional neural networks are used as a powerful visual model, which could be self-trained end-to-end, pixels-to-pixels; in [5], Long et al. showed this. They have built the pioneer fully convolutional networks that can take any arbitrary input and produce correspondingly-sized output, and their method also exceeds the state-of-the-art methods. According to them, their model is the first that can train FCNs end-to-end for pixel-wise prediction and from supervised pre-training. They have adapted contemporary classification networks such as AlexNet [36], VGG net [37], and GoogLeNet [38] into fully convolutional networks. Then transfer the learned representations by fine-tuning all layers by backpropagation to the segmentation task. The proposed fully convolutional network solves the inborn strain of semantic segmentation: Semantics (global information resolves "what") and location (local information resolves "where") by combining what and where.

SegNet [39] is an efficient architecture for semantic segmentation that focuses simultaneously on memory and computational time during inference. SegNet is a deep fully convolutional neural network consisting of encoder and decoder networks. The encoder part exactly follows the layers in VGG16 [37] except for the fully connected layers. This is why SegNet is smaller in size and easier to train than other networks. The architecture of SegNet is quite simple, 13 convolutional layers of the encoder network are connected to 13 corresponding decoder layers, followed by a soft-max classifier for pixel-wise classification. Fig. 4 represents the complete architecture.

However, providing a probabilistic segmentation output or, in other words, predicting model uncertainty makes Bayesian SegNet unique from other segmentation methods. When it comes to self-driving applications and the medical sector, a meaningful measure of uncertainty is equally crucial as high segmentation accuracy. In order to obtain the posterior distribution of softmax class probabilities, the Bayesian SegNet model is trained with dropout, and the posterior distribution of weights is sampled at test time with dropout. Dropout is a simple technique that randomly drops units inside the neural network layer during training, which prevents data overfitting [40]. Using the bayesian interpretation of dropout (1), Gal and Ghahramani propose the MC dropout technique.

$$L_{dropout} = \frac{1}{N}\sum_{i=1}^{N} E(y, \hat{y_i}) + \lambda \sum_{i=1}^{L} (||W_i||_2^2 + ||b_i||_2^2) \quad (1)$$

Bayesian SegNet follows the same technique for achieving probabilistic property; hence the aim is to find the posterior distribution over the convolutional weights upon the given dataset.

$$p(\omega|X, Y), \omega = (W_i)_{i=1}^{L}, \quad (2)$$

L is the number of layers of a NN

Although for the performance evaluation of Bayesian SegNet, authors use CamVid [41], SUN RGB-D [42] and Pascal VOC 2012 [43] datasets, here we discuss results related to CamVid dataset because it aligns with this research. Fig. 5 shows the qualitative results of Bayesian SegNet on CamVid dataset, where the first row is the input image and the second row is the corresponding ground truth. The third and fourth row shows predicted segmentation and model uncertainty, respectively. The quantitative performance of Bayesian SegNet is compared with thirteen previously developed segmentation methods and achieves the highest class average and mean IoU.

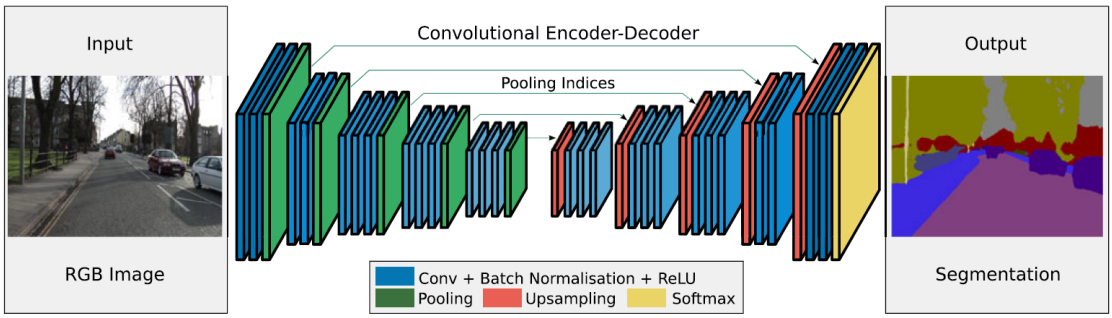

Fig. 4. A complete overview of SegNet architecture [39].

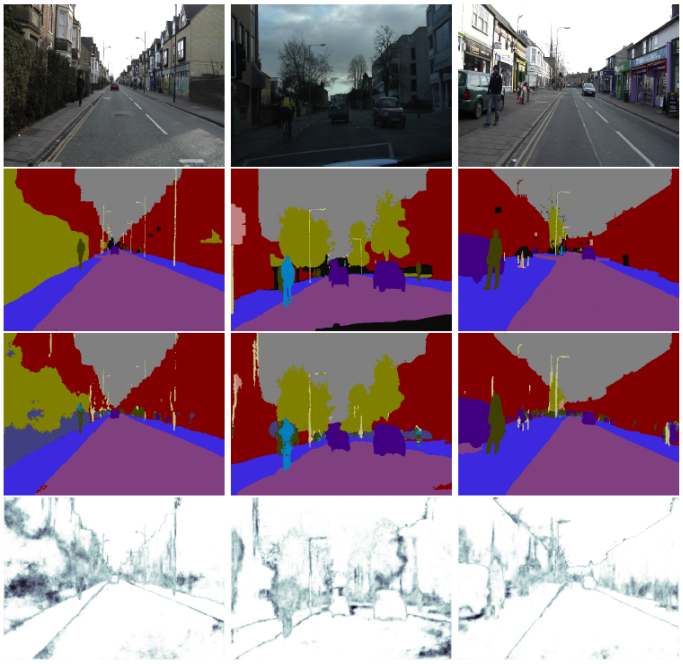

Fig. 5. Bayesian SegNet results on CamVid dataset for semantic segmentation and model uncertainty [44].

TABLE I
QUANTITATIVE RESULTS OF DIFFERENT MODELS ON CAMVID DATASET.

Methods	Dataset	Class Avg.	Global Avg.	Mean IoU
FCN 8 [5]	CamVid	64.2	83.1	52.0
SegNet-Basic [39]	CamVid	62.3	82.8	46.3
SegNet [39]	CamVid	71.20	90.40	60.10
Bayesian SegNet-Basic [44]	CamVid	70.5	81.6	55.8
Bayesian SegNet [44]	CamVid	76.3	86.9	63.1

It is also noted that among these thirteen methods, no one deals with uncertainty estimation of the predicted output except the Bayesian SegNet.

In this section, we have discussed three semantic image segmentation models: the first one is Fully Convolutional Networks (FCN)[5], which is the first model that trains end-to-end for pixel-level classification. The second one is another core segmentation engine known as SegNet [39]. The third one is based on SegNet but introduces model uncertainty in results output, which is crucial in such applications (self-driving cars, medical sector). Table I compares the results of these models on the CamVid dataset where SegNet-Basic and Bayesian SegNet-Basic are smaller versions (consisting of a smaller number of layers) of the original SegNet and Bayesian SegNet architecture, respectively. From the overall observation of Table I, it can be agreed that Bayesian SegNet [44] performs best on the CamVid dataset. However, the global average value is higher in SegNet (90.40%), where this value for Bayesian SegNet is 86.9%. Still, due to adding model uncertainty to the predicted result, Bayesian SegNet could be considered more trusted compared to those which do not consider model uncertainty.

B. Semantic Segmentation from Point Clouds

A point cloud is 3-dimensional data acquired from sensors such as LiDAR and RGB-D cameras. Each point has at least three coordinates: x, y, z and some additional feature channels, such as light intensity, colour, normal, etc., may be added according to the targeted output. High-resolution point cloud data can provide rich 3D geometric and depth information about the surrounding environment. However, it also possesses several challenges for semantic segmentation. The distribution of points in a point cloud is plagued with issues such as unordered nature, sparsity, non-uniform density, occlusions, etc. The lack of a structured nature in the distribution makes it difficult to apply deep learning tools such as CNNs which rely on grid-based patterns for repetitive processing, directly on points. The deep learning techniques used for semantic segmentation in point clouds can be segregated into point-based, grid-based and projection-based methods based on the input data representation. The following sections discuss each of the categories in detail.

1) Point-based methods: Point-based methods operate directly on 3D points to learn the per-point features. Due to the large number of data points available in a point cloud which makes it computationally infeasible, point-based methods resort to downsampling of points to reduce the number of data points. Many types of sampling can be used such as random sampling, farthest point sampling, segmentation-based sampling, etc. The type of sampling chosen must be carefully considered because the distribution of points in the point cloud is non-uniform and irregular. Sampling methods such as random sampling can easily drive a sparse point set into oversampling or a dense point set into undersampling. Farthest point sampling is by far the most balanced form of

sampling and hence, widely used. Point-based methods utilize operators such as Multi-layer Perceptrons (MLP), point-wise convolutions and graph operators to extract the local features from points.

Point-based methods using MLP: PointNet [13] is the first work that directly consumes point clouds and uses MLP for extracting per-point features giving state-of-the-art performance for object classification, object part segmentation, and semantic segmentation. The architecture of PointNet is simple because it processes each point identically and independently at the early stage. The complete architecture consists of one classification network followed by a segmentation network (Fig. 6). Many subsequent works have modelled their architectures based on PointNet. PointNet addresses the 3 major issues usually found in point-based data representations. One, they use a symmetric function such as max-pooling to isolate the output from input order variations. Two, Spatial Transformer Networks (STN) have been used to make the network invariant to object rotations and translations. Third, PointNet uses a shared MLP architecture to extract the per-point features from point-to-point interactions at multiple scales which eludes the problem of sparsity. Fully Connected Networks are used to achieve Semantic Segmentation from the aggregation of multi-scale features.

PointNet performs well for classification and segmentation tasks, but we bypass the classification part as it does not go with this research aims. For 3D object part segmentation, they have benchmarked the ShapeNet part dataset [45], where the total number of classes is 16. They have selected mean per-class intersection over union (mIoU) as the performance metric, compared these values with two other traditional methods, and found that PointNet gives the state-of-the-art score. After extending the network from part segmentation to semantic scene segmentation, they used the Stanford 3D semantic parsing dataset [46] for evaluation and got 78.62 overall accuracy. However, the datasets they used for benchmarking their model both contained indoor scenes and indoor objects only. Also, the max-pooling function aggregates the local features to a global vector, preventing fine-grained geometric and texture records which contribute to low performance in the case of complex, outdoor datasets such as KITTI.

These issues are resolved to an extent by the same authors in PointNet++ [47] which uses a Hierarchical Feature Learning strategy to partition the point cloud into overlapping sections using kNN, application of PointNet to individual sections and aggregating features from lower layers to abstract the global features. They use interpolation to propagate higher-level features to lower layers to extract per-point Segmentation labels, thereby utilizing the inherent distance metric available in the point cloud. They also use density-adaptive feature extraction modules to compensate for the non-uniformity of point clouds. Although PointNet++ achieved tremendous improvement over PointNet, it was also demonstrated on ScanNet dataset [48] containing indoor scenes and objects only. In PointSIFT [49], the authors replace the STN in PointNet by using direction-encoding units to learn features along different orientations and apply the PointSIFT modules at multiple scales. In Deep Fusion Network (DFNet) [50], authors have extracted local and global features in separate backbones using ViewNet for the former and PointNet for the latter and fused them together using FusionNet for 3D shape classification. Authors of global relation-aware attentional network (GRANET) [51] have developed separate attention modules for extracting local and global features and the features are finally integrated at multiple scales by the network. PointMLP [52] is a lightweight network that achieved state-of-the-art results at high inference speeds using residual MLP network.

Authors of Point Attention Transformer (PAT) [53] have developed a novel permutation-invariant Group Shuffle Attention (GSA) module instead of the traditional multi-head attention module used in [54] to learn the local interactions between neighbourhood points in a cluster. They have also created a sampling operation called Gumbel Subset Sampling (GSS) to dynamically select continuous clusters in the training phase as opposed to discrete, fixed clusters during inference. The learnable sampling operation gives more flexibility to the network to learn intricate feature representations within the selected cluster. Authors of [55] developed Point Transformer layer which adapts the self-attention mechanism in [54] to capture interactions between points within the local neighbour of a data point. The point transformer layer is sandwiched between 2 linear layers forming the Point Transformer block. They also observe that the position coordinates of the points can be naturally used for positional encoding, but they also develop a learnable position encoding system using an MLP.

Point-based methods using convolutions: These methods apply customised convolution operations to a selected neighbourhood of points making sure to keep the learned features invariant to the order and sparsity of input data. Most of them use random sampling to downsample the points and the k-Nearest Neighbour (kNN) Algorithm to select the local neighbourhood. PointCNN [56] uses an X-Conv operator while InterpCNN [57] has developed Interpolated Convolution, to extract and arrange the features from an unordered group of local points. SpiderCNN [58] uses a SpiderConv operator which combines a regular step function with a Taylor Polynomial to enhance the expressiveness of the convolution filters that capture local information. A-CNN [59] utilizes annular convolutions which capture geometric information in a ring-shaped manner to better incorporate geometric scalability for large-scale segmentation scenes. PointConv [60] extends the conventional 2D convolution operator to 3D using Monte Carlo approximation and training a Multi-Layer Perceptron to estimate the weights. Authors of FSS-Net [61] have proposed a few ground-breaking search techniques and up- and down-sampling methods that make the computations more efficient and less time and memory-consuming.

Point-based methods using graph operators: Graph-based methods consider the point cloud as a graphical network structure. The points are taken as vertices and the distance metric between them is considered as the edges of a graph, which can then be worked upon by graph convolutional networks that can map the dependencies by capturing the node-to-node interactions. Landrieu et al. [62] focus on large-scale point

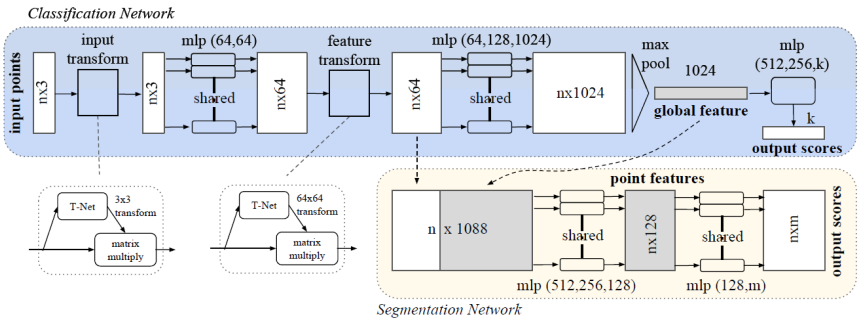

Fig. 6. The complete PointNet architecture consists of a classification network and segmentation network [13].

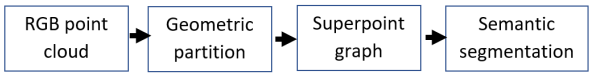

Fig. 7. Visualization pipeline steps in SPG [62].

clouds and proposed a deep-learning-based framework that not only performs well in indoor scene semantic segmentation but also shows significant improvement in outdoor scene semantic segmentation. They have introduced Super Point Graphs (SPG) from the organization of 3D point clouds. SPGs offer three compelling advantages, such as SPG considers entire object parts as a whole rather than classifying individual points or voxels, which makes identification easier. The second advantage is that they have analyzed the dependency of one object on another. For example, a footpath is beside the road, and vehicles are above the road. These relations are important for contextual classification. Third, the size of the SPG is smaller as they do not consider the total number of points, only consider the number of simple structures in a scene. They have used PointNets [13] and graph convolution for Super Point embedding and contextual segmentation in their network architecture. The visualization pipeline can be represented in four steps [Fig. 7]. Step 1 illustrates an input point cloud. These points are divided into SuperPoints in step 2, which are basic geometric shapes. The SuperPoint graph created by linking nearby SuperPoints is seen in step 3. Finally, step 4 shows the semantic segmentation by transforming the SuperPoints into compact embeddings, and graph convolution is used to process contextual information.

SPG model set state-of-the-art on two different types of publicly available datasets: Semantic3D [63] dataset(specialized for urban and rural scenes contains over 3 billion LIDAR points) and Stanford Large-Scale 3D Indoor Spaces (S3DIS) [46] dataset(specialized for 3D RGB indoor point clouds). For measuring the model performance, they have used three evaluation metrics, per-class intersection over union (IoU), per-class accuracy (Acc), and overall accuracy (OA). SPG achieved the highest performance measure for both datasets compared to other state-of-the-art methods. Dynamic GCNN (DGCNN) [64] is very similar to the PointNet architecture, replacing the regular MLP with MLP stacked with EdgeConv features. EdgeConv module searches the k-Nearest Neighbourhood and extracts the edge vectors of the points, capturing their geometric interactions. It also dynamically updates the graph after each layer to adapt to the point cloud structure. The authors of Regularized Graph Convolutional Neural Network (RGCNN) [65] have also attempted to dynamically adapt the graph network to the point cloud structure to capture maximum 3D interactions by updating the graph Laplacian matrix at each layer. GAPNet [66] adds GAPLayer modules within the PointNet [13] architecture to capture the local geometric interactions among neighbourhood points using the attention mechanism. Although Point-based methods improve the quality of segmentation results owing to the extraction of per-point features, these methods are not real-time feasible as they have significant memory and computational requirements. As the number of data points increases, the complexity further increases. Although downsampling helps to overcome this limitation to an extent, due to the non-uniform and sparse nature of point clouds, usual sampling methods tend to oversample or undersample certain regions of the point cloud. Although learnable sampling methods have been proposed in this regard, the additional networks contribute to the computational complexity of the model.

2) Grid-based methods: In grid-based methods, unordered points in the point cloud are converted to a uniform structured representation in the form of 3D grids or 3D volumetric representations called voxels. A voxel is analogous to a pixel in 2D. Just like a uniform set of pixels represents a 2D image, a set of 3D voxels can be made to represent the entire point cloud. The process of voxelising a point cloud involves discretising the points into regular 3D grids which may or may not have uniform sizes. Hence, some amount of geometric accuracy is lost in this process. However, the advantage of this process is that the voxelised point cloud can now be processed using Convolutional Neural Networks for feature extraction, making the process much more computationally efficient. The voxel-based semantic segmentation methods can be further classified based on whether the model relies on fixed-sized voxels or non-uniform, variable voxel sizes.

Voxel-based methods using fixed voxel sizes: These methods employ fixed-size voxels to quantize the entire point cloud. VoxNet [67] is one of the earliest papers to suggest applying 3D CNN on uniform-sized point grid voxels to

extract 3D features. However, since the point cloud can be sparse in many areas, especially for far-away objects, most of the voxels will be empty. Hence, applying 3D convolutions uniformly on all voxels is not computationally efficient. In 2014, the sparse convolutional operator [68] was developed, which made the application of CNNs on sparse data points such as in a point cloud much more efficient. Authors of [69] have applied the sparse convolutional operator on a voxelised point cloud for the semantic segmentation task. To obtain fine-grained 3D segmentation, SEGCloud [70] combines the advantages of 3D-FCNN, trilinear interpolation [71], and fully connected Conditional Random Fields (FC-CRF). First, 3D raw point clouds are converted into voxelized point clouds through some pre-processing. Next, a 3D fully convolutional neural network is used, which takes voxelized point clouds as input and provides voxel predictions as output. In order to return to the original 3D Points representation, a trilinear interpolation layer is then applied. In the last step, the raw point clouds and the obtained 3D point scores are passed through the FCCRF to produce the final results. Fig. 8 shows this end-to-end training process. Two indoor and two outdoor 3D datasets (NYU V2, S3DIS, KITTI, Semantic3D.net) were used to evaluate SEGCloud, where it shows comparable or better performance across all datasets. VVNet [72] uses a specialized segmentation technique for the voxelization of the point cloud and proceeds to extract features from the voxels using a vibrational autoencoder. In 2020, VoxSegNet [73] improved the feature extraction and aggregation processes by developing a spatially aware extraction module that extracts fine-grained details from each voxel and an attention-based feature aggregation module that attentively selects features from multiple scales.

TABLE II
QUANTITATIVE RESULTS OF DIFFERENT MODELS ON STANFORD 3D (INDOOR SCENE) [46] AND SEMANTIC3D (OUTDOOR SCENE) [63] DATASET.

Methods	Dataset	Global Acc.	Mean IoU
PointNet	Stanford 3D (indoor scene) [46]	78.62	47.71
PointNet++	Stanford 3D (indoor scene) [46]	81.0	54.5
PAT	Stanford 3D (indoor scene) [46]	70.83	60.07
Point Transformer	Stanford 3D (indoor scene) [46]	90.8	70.4
PointSIFT	Stanford 3D (indoor scene) [46]	88.72	70.23
PointCNN	Stanford 3D (indoor scene) [46]	88.1	65.4
SPG	Stanford 3D (indoor scene) [46]	85.5	62.1
SPG	Semantic3D (outdoor scene) [63]	94.0	73.2
DGCNN	Stanford 3D (indoor scene) [46]	84.1	56.1
SEGCloud	Stanford 3D (indoor scene) [46]	—	48.92
SEGCloud	Semantic3D (outdoor scene) [63]	88.1	61.3

Voxel-based methods using variable voxel sizes: As the name suggests, these methods utilize voxels which are variable in size depending on the density of the points in the different areas of the point cloud. This adaptive quantizing mechanism aligns well with the non-uniform and sparse nature of the point cloud. These methods rely on clustering algorithms such as KD-tree or Octree or even density-aware neural networks to generate variable-sized voxels for each section of the point cloud. For instance, OctNet[74] uses octree and KD-Net [75] uses Kd-tree for this purpose. PointGrid [76] learns per-point features and transformations from the point cloud and assigns them to the generated 3D grid. Authors of SpSequenceNet [77] apply 3D sparse convolutions for feature extraction and use an attention-based module for global feature generation and an interpolation-based module for feature aggregation to higher layers.

Although voxels-based methods are computationally more efficient than any other approaches, the discretization loses valuable 3D structural and geometric information. Also, the 3D volumetric representations require resource-intensive 3D convolutions to extract features from them.

3) Projection-based methods: Under this category are methods that project the 3D points to any of the perspective planes like front view or Bird's Eye View or even multiple views based on camera perspectives and apply 2D CNNs to extract features from the projected image. The accuracy of semantic segmentation depends on the accuracy of projection. Usually, the camera's projection matrix is used to project the 3D points onto the camera plane. Since the processing is done by 2D convolutional networks, the computational burden is much simpler than the point and grid-based methods and so is the inference speed. BEV projections also help in identifying occluded objects. Since the projection-based networks can achieve high inference speeds, most of them are also real-time implementable. BirdNet [78] is one of the earliest networks that convert 3D point clouds to 2D BEV projection views. BirdNet proposes a novel density-based encoding for BEV projection. polarNet [79] improves upon the shortcomings of the traditional BEV projection by considering a polar coordinate system for the BEV projection using the sensor position of the ego vehicle as the origin. SalsaNet [80] projected the 3D LiDAR points onto the BEV view and used a ResNet-based encoder-decoder model to realise the segmentation network.

Since projection from 3D to 2D results in an inherent loss of some 3D geometric information, these methods could suffer from not-so-accurate segmentation results. To overcome this, many works have focused on projecting the points from multiple perspectives to get an overall understanding of the scene. SnapNet [81] is one such network that employs multi-view projection for semantic segmentation.

C. Semantic Segmentation from Multi-modal Data (images and point clouds)

There is no doubt that environmental perception has a massive impact on the performance of a self-driving vehicle. Many perception methods have already been published to build a robust, accurate, and real-time perception system. Among them, a few methods use data from one type of sensor (LIDAR or images), and others fuse data from different sensors. However, methods that use multi-modal data outperform those that only use single-sensor data. This section will discuss the papers that used fusion technology to build their model.

In 2022, Ye et al. [82] proposed a CNN-based model for 3D semantic segmentation by fusing camera images and LIDAR data. To reduce the complexity of CNN, they merge point cloud and image data through the spherical projection technique. CNN takes this spherical image as input and provides a tensor as output. At the final stage, back projection is done, which gives a 3D semantic segmentation result. The overall process is shown in Fig. 9.

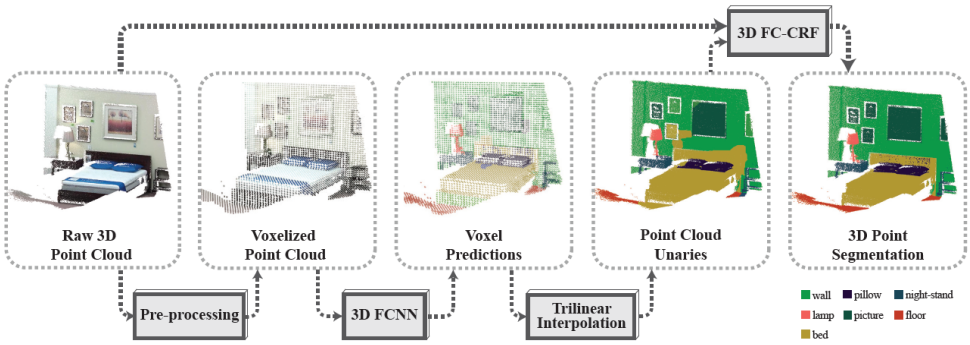

Fig. 8. A complete overview of the SEGCloud architecture [70].

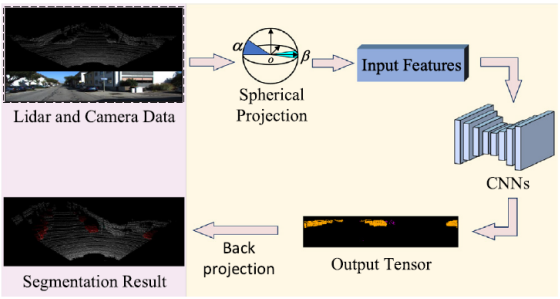

Fig. 9. A complete overview of 3D semantic segmentation system [82].

The proposed semantic segmentation network architecture is divided into the baseline module and the spatial module Fig. The baseline module is the modified version (use fewer parameters) of SqueezeSeg [83] and consists of two modules, fire and fireDeconv. The pooling operation in the baseline module is the reason behind spatial feature loss. To minimize this problem, they have introduced a spatial module. Here, the input tensor value could be up to 9 dimensional (X, Y, Z, I, D, R, G, B, M), where X, Y, Z represents 3D points, D is the distance between LIDAR sensor and 3D point, I is the reflection intensity, and M is the binary mask. In general, the input tensor shape is 64 x 512 x n. For the baseline module, n is 9, and for the spatial module, n is a variable and depends on the type of experiment.

For the experiment, they used the semanticKITTI [27] dataset, which is based on the KITTI dataset but contains more data. They have compared the performance (mIoU) of baseline architecture and spatial architecture with SqueezeSeg [83] and Dual SqueezeSeg methods. Experimental result shows that the mIoU value of the baseline module is slightly greater than SqueezeSeg (2.7%) and Dual SqueezeSeg (5.8%) methods. However, the spatial module's performance increases significantly when the input channel provides these "DRGBM" features.

Road detection is another challenging perception task, which is basically pixel-wise semantic segmentation between roads and other objects. Chen et al.[84] proposed a road detection method based on image and LIDAR point cloud named PLARD, Progressive LIDAR adaptation for road detection. Fusing LIDAR data with image data is quite challenging because raw LIDAR data and its features are not in the same formats as the visual data and visual features. For example, LIDAR data is a collection of 3D points in the 3D real-world space, whereas image data contains RGB pixel values on a 2D image plane. To reduce this gap, PLARD is introduced, which consists of two modules: data space adaptation and feature space adaptation.

An Altitude Difference-based transformation method (ATD) is used in the data space adaptation module, which takes raw LIDAR data as input and provides an altitude difference image as output. The feature space adaptation module adapts LIDAR features and transforms them into visual features through a cascaded fusion structure. Inside this module, the Feature Space Transformation(FST) module is used to transform the LIDAR feature into improved visual features. In each FST module, there is a corresponding Transformation Network(TN) for parameter learning. The cascaded fusion fuses the visual features and the adapted LIDAR features at the last four convolution stages. The final stage of PLARD is the parsing stage. The classification task is done in this stage by the PLARD from the integrated features and provides robust road detection results. The overall process is shown in Fig. 11.

For the experiment of their model, they used the KITTI road detection [34] benchmark, and for performance comparison, they used those evaluation matrices which are used in KITTI. The result shows that PLARD outperforms the 12 other state-of-the-art road detection models for all available KITTI road scene categories: urban marked roads (UM), urban multiple marked lanes (UMM), and urban unmarked roads (UU).

Xiao et al. [85] proposed a conditional random field (CRF) based hybrid model for road detection. This model effectively utilizes both the camera and the LIDAR by integrating the data in a probabilistic manner. In order to determine the road areas, the HybridCRF model can be effectively optimized via graph cuts. The camera and LIDAR are cross-calibrated in the HybridCRF model so that the point cloud can be aligned with the image by projecting the LIDAR points onto the image plane. $p = [x, y, z, 1]^T$ denotes a 3D point in the LIDAR coordinate, which is then converted to the camera coordinate by the following (3):

$$P_c = \mathbf{R}_{rect}\mathbf{T}_{velo}^{cam}P \tag{3}$$

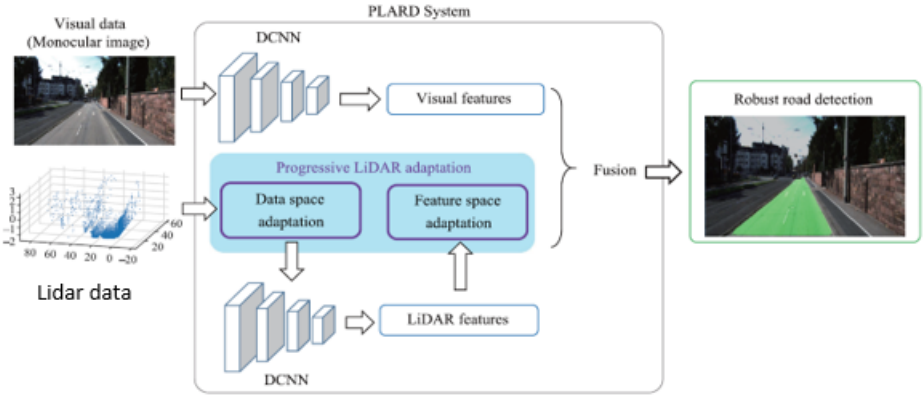

Fig. 10. An overview of the PLARD system [84].

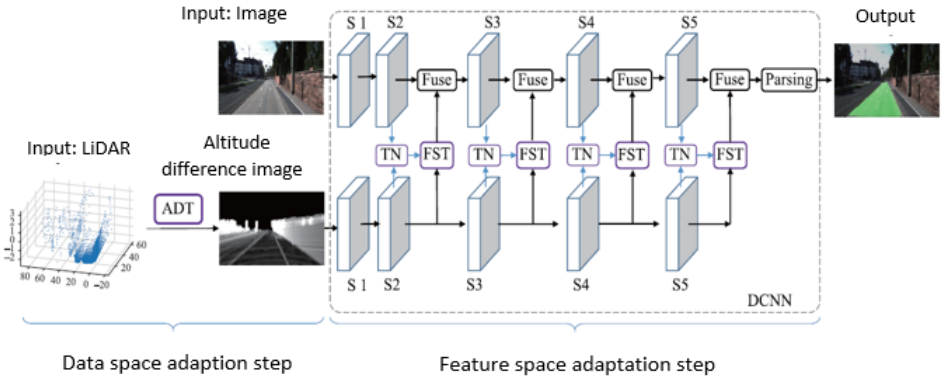

Fig. 11. A detailed working process of PLARD system [84].

Here, $\mathbf{R}_{rect}$ is the rectifying rotation matrix, while $\mathbf{T}_{velo}^{cam}$ is the transformation matrix used to convert LIDAR coordinates to camera coordinates. Points with a negative Z value are then eliminated after this phase. After that, using the projection matrix $\mathbf{P}_{rect}$, it is possible to project the remaining points onto the image plane using the following (4):

$$[u', v', W]^T = \mathbf{P}_{rect}[x_c, y_c, z_c, 1]^T \tag{4}$$

The LIDAR point p's projected pixel coordinates could then be found by using the formula $[u, v] = [\frac{u'}{w}, \frac{v'}{w}]$. It is also necessary to discard the points that project outside of the image's field of view (FOV). For a typical road scene, how a camera image and LIDAR point clouds are used is shown in Fig. 12.

Labels of the image pixels (P) and the LIDAR points (L) are projected onto the field of view of the image after the image and LIDAR point clouds have been aligned. Each random variable can have a value of either 0 or 1 because the road detection problem is defined as a two-class labeling problem. Three different types of edges are accounted for the nearby relationship: (i) edges between adjacent pixels in the image plane (pixel to pixel or E_{pp}), (ii) edges between adjacent LIDAR points in 3D space (LIDAR point to LIDAR point or E_{LL}), and (iii) edges between aligned LIDAR points and their corresponding pixels (pixel to LIDAR point or E_{pL}).

Another deep learning based road detection method has been developed by Caltagirone et al. [86] by fusing point clouds and camera images. However, as the point cloud's nature is unstructured and sparse, it is first projected onto the camera image plane, then upsampled to generate from 3D point clouds to 2D dense images.

A fully convolutional(FCN) encoder-decoder with an intermediate context module is used as the base neural network of this work. The encoder consists of 5 convolutional layers, the decoder consists of 6 convolutional layers, and the context module consists of 9 convolutional layers. This model uses two-stride convolution with 4 × 4 kernels during downsampling, leading to fewer memory requirements. An exponential linear unit (ELU) layer is placed after each convolutional layer using (5).

$$f(x) = \begin{cases} x & if x \geq 0 \\ e^x - 1 & otherwise \end{cases} \tag{5}$$

The authors train several fully convolutional neural networks (FCNs) to detect roads using either one sensor or three different fusion strategies: early, late and their proposed cross-fusion. Early fusion methods simply concatenate the input LIDAR and camera tensors in the depth dimension to create a tensor with six channels (RGBZYX). Then, as the input of the base FCN model, this tensor is used, which must learn features

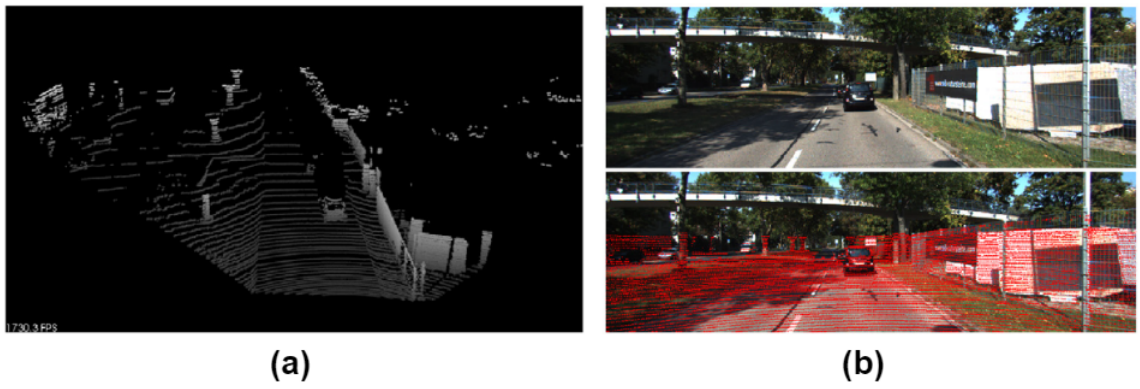

Fig. 12. (a) Point clouds and (b) corresponding camera image (upper right corner), a fusion of point clouds and image (bottom right corner) [85].

that incorporate both sensing modalities from scratch. Fusion takes place here at a fundamental abstraction level. Late fusion lies at the other end of the spectrum. Here, the data from the camera and LIDAR is integrated at the end of two separate processing branches.

Although these techniques are pretty simple, at which stage the fusion should be performed is manually assigned by the developers. The authors proposed a novel fusion strategy to overcome this dependency and named it cross-fusion. In every layer of the processing pipeline, trainable cross-connections are created between the LIDAR point clouds and the camera image processing branches. These connections start off at zero, which represents the absence of fusion, and are modified throughout training to achieve the proper level of integration using the following (6 and 7):

$$I_j^{Lid} = L_{j-1}^{Lid} + a_{j-1} L_{j-1}^{cam} \tag{6}$$

$$I_j^{cam} = L_{j-1}^{cam} + b_{j-1} L_{j-1}^{Lid} \tag{7}$$

Where I_j^{Lid} and I_j^{cam} represent the input tensors at depth j and $a_j, b_j \in \mathrm{R}$ with $j \in \{1, \ldots, 20\}$ are trainable cross-fusion parameters. Fig. 13 shows the graphical representations of cross-fusion strategy. Among the single modality and fusion networks taken into consideration in this experiment, the cross-fusion FCN performed the best. It also received good ratings for its performance on the KITTI road benchmarks.

TABLE III
QUANTITATIVE RESULTS OF DIFFERENT MODELS ON THE KITTI DATASET HERE ALL MODELS TAKE IMAGES AND POINT CLOUDS AS INPUT.

Methods	Dataset	MaxF	APre	PreRate	ReRate
PLARD [84]	KITTI	97.03	94.03	97.19	96.88
HybridCRF [85]	KITTI	90.81	86.01	91.05	90.57
LidCamNet [86]	KITTI	96.03	93.93	96.23	95.83

This section discusses four semantic segmentation methods that take images and point clouds as input. Table x shows their performance on the KITTI dataset using the maximum F1-measure (MaxF), average precision (APre), precision rate (PreRate) and recall rate (ReRate). We have not included the 3D segmentation method proposed by Ye et al. [82] in Table x because they use different datasets and performance metrics. The Table clearly shows that PLARD achieved the highest performance compared to the other two multimodal models.

V. COMPARISON OF THREE TYPES OF SEGMENTATION APPROACHES

In recent times, multimodal (collect data from different types of sensors) fusion models are getting more attention than unimodal (collect data from the same type of sensor) models due to their higher performance, [87]. Moreover, in a complex scenario (adverse weather or poor road conditions), relying on the data gathered from several sensors helps to reduce uncertainty. For example, camera is the most popular sensor in a self-driving vehicle. A self-driving vehicle uses a camera to understand surrounding environments and recognize objects like the human eye. But in adverse weather conditions like rain, fog, snow, and low light environments camera does not perform well. On the other hand, LIDAR is an active sensor that measures distances to objects precisely and operates independently in different lighting conditions because they perceive the surroundings using its own laser light pulses. However, they have a small coverage area, usually between 10 and 100 m, and no texture or color information is seen, instead providing sparse data [86]. Considering these advantages and disadvantages, it is simple to understand how using both sensor types could increase overall reliability.

VI. CHALLENGES AND FUTURE RESEARCH

A robust perception system is mandatory for self-driving vehicles to drive safely in different driving conditions and weather. However, achieving such a system is not easy because the model's performance not only depends on the higher accuracy, but it is also essential to know how confident the model is about the predicted results. For example, at any instant, the system may predict an object as a person; at that moment, it is necessary to know the model uncertainty with respect to other classes, such as street signs or cyclists. In this case, uncertainty is important because the system may have different behavioral decisions depending on different classes [88]. Therefore, uncertainty measurement should be considered a natural part of any predictive system's output, especially those related to real-life applications.

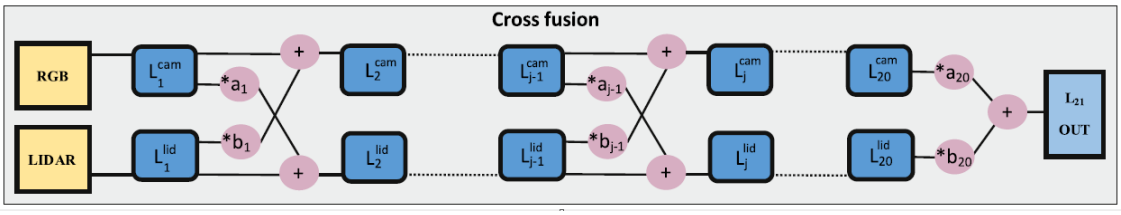

Fig. 13. The cross-fusion strategy for fusing LIDAR point clouds and camera images[86].

Generally, most deep learning algorithms focus on improving their accuracy without implementing solid mechanisms for quantifying predictive uncertainties (completely ignoring the importance of the element of doubts). However, there are a few methods that have already been developed that consider uncertainty estimation as a part of their segmentation, classification, or regression output ([89], [90], [91]). But it is rare to automatically reduce epistemic or model uncertainty after estimating it. Also, few methods calculate uncertainty accuracy for classification tasks, but this is also infrequent for segmentation tasks. While exploring the uncertainty quantification technique in different research papers, we found two research papers ([44] and [92]) that show two different types of uncertainty estimation for the exact same task. While there has not yet been an accepted or identified state-of-the-art method of estimating uncertainty, there is an excellent opportunity to conduct research in this field and contribute to uncertainty quantification.

There is a significant impact on the performance of the deep learning model on data quantity (how much data are available for training and testing) and data quality (precise information, less noise). Although many self-driving datasets are publicly available, researchers still face new challenges in ”collaborating data diversity” and ”aligning data received from multiple sensors”. First, however, it is necessary to focus on which perception task the collected data will be used during data collection. According to this, appropriate sensor selection and precise sensor alignment could be a research topic. Also, to ensure data quality, researchers might concentrate on reducing annotation errors and misalignment of different sensors.

Due to the superior performance over unimodal models, multimodal fusion models have drawn greater attention in recent years. However, data collected from different sensors are not aligned in the same spaces. For example, one common way of working with irregular point cloud data is to convert it into 3D voxel grids or images. However, this type of transformation may create unnecessarily voluminous data [13]. Hence, aligning data and features from different modalities without destroying their original properties is challenging and needs more attention in the future.

Although there is an enormous amount of multimodal methods have been proposed to solve different perception problems, there is still no fixed rule found about: ”what data need to fuse?”, ”what is the perfect fusing time (early, late, middle)?”, and ”which fusion operation needs to use?”. In most of the works, researchers only focus on fusing camera images with LiDAR data, but other autonomous vehicle sensors are

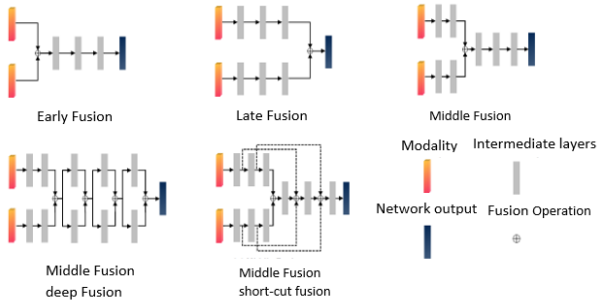

Fig. 14. An illustration of different fusion methods [97], for simplicity author restrict on two different modality but fusing between more than two modality is possible.

rarely fused. For example, a limited number of papers include Rader signals data for developing multimodal models for self-driving vehicles. For a specific perception task, fusing different types of data (one after another with varying combinations) and comparing the model’s performance may achieve a good perception model.

Deep neural networks allow feature fusion in various stages, such as early fusion, middle fusion, hybrid fusion, and late fusion. Fig. 14 shows some fusion methods. The deep multimodal fusion strategy was first introduced in 2013 by Couprie et al. [93]. They have proposed a segmentation network for indoor scene segmentation where they use the early fusion strategy and fuse RGB and depth channel. FuseNet [94], MVCNet [95] and MVCNet [96] are few other examples of early fusion. The late fusion strategy has higher flexibility and modularity than early fusion but requires high computation cost, and memory [97]. [98] and [99] uses late fusion strategy. It is a big challenge for the researchers to tell firmly which fusion scheme will give the best performance for a model. Before applying any fusion scheme to a model considering a few things may help: Stronger cross-modal information interaction is provided by early fusion strategy, flexible and scalable implementation is provided by late fusion, low-level and high-level both features are valuable for the final prediction and to learn representative features of segmentation model multi-level fusion may help [100].

The typical fusion operations: addition, concatenation, and ensemble, are not always sufficient for proper joint feature representations. For example, LIDAR point clouds provide more information than cameras in low-light conditions. Regular fusion operations do not consider this type of condition. The Mixture of Experts (MoE) approach may help in this situation. Furthermore, during fusion operations, always

consider redundancy, imbalance, and contradiction of data because these things may reduce the model's performance. Analyzing these issues is very challenging and needs more attention in the future. Overall, drawing a concrete conjecture on fusion technology could illuminate significant advances in autonomous driving technology.

VII. Conclusion

In this review paper, we discuss semantic segmentation from the perspective of autonomous vehicles. Depending on the data source, we segment semantic segmentation in three different approaches: Semantic Segmentation from Images, Semantic Segmentation from point clouds, and Semantic Segmentation from Multi-modal Data. Here, we discuss different semantic segmentation network architectures, training datasets, and performance. To the best of our knowledge, this review paper is unique because no review paper discusses others' work so deeply. Moreover, we draw a complete overview of autonomous driving perception systems such as "what sensors are used?", "what is the data collection procedure?" and "some publicly available datasets commonly used". "What are the main challenges?" and "where could the improvement be possible?" is also discussed in this review. In conclusion, this is a complete document for beginners who want to start researching semantic segmentation in autonomous driving.

VIII. Acknowledgement

This research was partially supported by the Australian Research Council's Discovery Projects funding scheme (projects DP190102181 and DP210101465).

References

[1] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 82–95, 2019.

[2] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE TITS*, vol. 20, no. 10, pp. 3782–3795, 2019.

[3] S.-y. Wang, Z. Qu, C.-j. Li, and L.-y. Gao, "Banet: Small and multi-object detection with a bidirectional attention network for traffic scenes," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105504, 2023.

[4] J. Jordan, "An overview of semantic image segmentation." 2018. [Online]. Available: https://www.jeremyjordan.me/semantic-segmentation/

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE CVPR*, 2015, pp. 3431–3440.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.

[7] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *MICCAI*. Springer, 2016, pp. 424–432.

[8] F. Shojaiee and Y. Baleghi, "Efaspp u-net for semantic segmentation of night traffic scenes using fusion of visible and thermal images," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105627, 2023.

[9] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE TPAMI*, vol. 43, no. 12, pp. 4338–4364, 2020.

[10] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," in *2018 IEEE ICRA*. IEEE, 2018, pp. 1887–1893.

[11] A. Boulch, B. Le Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks." *3DOR@ Eurographics*, vol. 3, 2017.

[12] J. Huang and S. You, "Point cloud labeling using 3d convolutional neural network," in *2016 ICPR*. IEEE, 2016, pp. 2670–2675.

[13] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE CVPR*, 2017, pp. 652–660.

[14] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[15] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *2020 IEEE/CVF CVPR*, 2020, pp. 11 108–11 117.

[16] G. Ros, S. Ramos, M. Granados, A. Bakhtiary, D. Vazquez, and A. M. Lopez, "Vision-based offline-online perception paradigm for autonomous driving," in *2015 IEEE WACV*. IEEE, 2015, pp. 231–238.

[17] S. Jarl, L. Aronsson, S. Rahrovani, and M. H. Chehreghani, "Active learning of driving scenario trajectories," *Engineering Applications of Artificial Intelligence*, vol. 113, p. 104972, 2022.

[18] L.-H. Wen and K.-H. Jo, "Deep learning-based perception systems for autonomous driving: A comprehensive survey," *Neurocomputing*, 2022.

[19] L. Schneider, M. Jasch, B. Fröhlich, T. Weber, U. Franke, M. Pollefeys, and M. Rätsch, "Multimodal neural networks: Rgb-d for semantic segmentation and object detection," in *Scandinavian conference on image analysis*. Springer, 2017, pp. 98–109.

[20] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *2016 IEEE CVPR*, 2016, pp. 3213–3223.

[21] W. Jang, J. Hyun, J. An, M. Cho, and E. Kim, "A lane-level road marking map using a monocular camera," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 187–204, 2021.

[22] G. Wang, J. Wu, R. He, and B. Tian, "Speed and accuracy tradeoff for lidar data based road boundary detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 6, pp. 1210–1220, 2020.

[23] Cadence. The use of radar technology in autonomous vehicles. [Online]. Available: https://resources.system-analysis.cadence.com/blog/msa2022-the-use-of-radar-technology-in-autonomous-vehicles

[24] W. Rahiman and Z. Zainal, "An overview of development gps navigation for autonomous car," in *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2013, pp. 1112–1118.

[25] H. Yin and C. Berger, "When to use what data set for your self-driving car algorithm: An overview of publicly available driving datasets," in *ITSC 2017*. IEEE, 2017, pp. 1–8.

[26] A. project of Karlsruhe Institute of Technology and T. T. I. at Chicago, "The kitti vision benchmark suite," 2012. [Online]. Available: http://www.cvlibs.net/datasets/kitti/

[27] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF ICCV*, 2019.

[28] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," in *CVPR Workshop on The Future of Datasets in Vision*, 2015.

[29] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[30] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[31] K. Takumi, K. Watanabe, Q. Ha, A. Tejero-De-Pablos, Y. Ushiku, and T. Harada, "Multispectral object detection for autonomous vehicles," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017, pp. 35–43.

[32] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *2015 IEEE CVPR*, 2015, pp. 1037–1045.

[33] A. Teichman, J. Levinson, and S. Thrun, "Towards 3d object recognition via classification of arbitrary object tracks," in *2011 ICRA*. IEEE, 2011, pp. 4034–4041.

[34] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE CVPR*. IEEE, 2012, pp. 3354–3361.

[35] Waymo, "Waymo open dataset," 2019. [Online]. Available: https://waymo.com/open/

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE CVPR*, 2015, pp. 1–9.

[39] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE TPAMI*, vol. 39, no. 12, pp. 2481–2495, 2017.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[41] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[42] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *2015 IEEE CVPR*, 2015, pp. 567–576.

[43] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[44] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[45] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3d shape collections," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.

[46] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *2016 IEEE CVPR*, 2016, pp. 1534–1543.

[47] C. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Neural Information Processing Systems*, 2017.

[48] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," *2017 IEEE CVPR*, pp. 2432–2443, 2017.

[49] M. Jiang, Y. Wu, Z. Zhao, and C. Lu, "Pointsift: A sift-like network module for 3d point cloud semantic segmentation," *ArXiv*, vol. abs/1807.00652, 2018.

[50] L. Qin, X. Xu, W. Che, Y. Zhang, and T. Liu, "Dynamic fusion network for multi-domain end-to-end task-oriented dialog," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 6344–6354.

[51] R. Huang, Y. Xu, and U. Stilla, "Granet: Global relation-aware attentional network for semantic segmentation of als point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, 2021.

[52] X. Ma, C. Qin, H. You, H. Ran, and Y. R. Fu, "Rethinking network design and local geometry in point cloud: A simple residual mlp framework," *ArXiv*, vol. abs/2202.07123, 2022.

[53] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian, "Modeling point clouds with self-attention and gumbel subset sampling," *2019 IEEE/CVF CVPR*, pp. 3318–3327, 2019.

[54] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017.

[55] L. Hengshuangzhao, J. Jiang, and P. Jia, "Point transformer," in *ICCV*, 2021.

[56] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Neural Information Processing Systems*, 2018.

[57] J. Mao, X. Wang, and H. Li, "Interpolated convolutional networks for 3d point cloud understanding," in *2019 IEEE/CVF ICCV*, 2019, pp. 1578–1587.

[58] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," *ArXiv*, vol. abs/1803.11527, 2018.

[59] A. Komarichev, Z. Zhong, and J. Hua, "A-cnn: Annularly convolutional neural networks on point clouds," *2019 IEEE/CVF CVPR*, pp. 7413–7422, 2019.

[60] W. Wu, Z. Qi, and F. Li, "Pointconv: Deep convolutional networks on 3d point clouds," *2019 IEEE/CVF CVPR*, pp. 9613–9622, 2018.

[61] J. Wang, Y. Zhuang, and Y. Liu, "Fss-net: A fast search structure for 3d point clouds in deep learning," *International Journal of Network Dynamics and Intelligence*, 2023.

[62] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *2018 IEEE CVPR*, 2018, pp. 4558–4567.

[63] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3d. net: A new large-scale point cloud classification benchmark," *arXiv preprint arXiv:1704.03847*, 2017.

[64] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *2019 IEEE/CVF CVPR*, 2019, pp. 10 288–10 297.

[65] G. Te, W. Hu, Z. Guo, and A. Zheng, "Rgcnn: Regularized graph cnn for point cloud segmentation," *Proceedings of the 26th ACM international conference on Multimedia*, 2018.

[66] C. Chen, L. Z. Fragonara, and A. Tsourdos, "Gapnet: Graph attention based point neural network for exploiting local feature of point cloud," *ArXiv*, vol. abs/1905.08705, 2019.

[67] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.

[68] B. Graham, "Spatially-sparse convolutional neural networks," *ArXiv*, vol. abs/1409.6070, 2014.

[69] F. Verdoja, D. G. F. Thomas, and A. Sugimoto, "Fast 3d point cloud segmentation using supervoxels with geometry and color for 3d scene understanding," *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1285–1290, 2017.

[70] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," in *2017 international conference on 3D vision (3DV)*. IEEE, 2017, pp. 537–547.

[71] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, 2002.

[72] H.-Y. Meng, L. Gao, Y.-K. Lai, and D. Manocha, "Vv-net: Voxel vae net with group convolutions for point cloud segmentation," in *2019 IEEE/CVF ICCV*, 2019, pp. 8499–8507.

[73] Z. Wang and F. Lu, "Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 9, pp. 2919–2930, 2020.

[74] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *2017 IEEE CVPR*, 2017, pp. 6620–6629.

[75] R. Klokov and V. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3d point cloud models," in *2017 IEEE ICCV*, 2017, pp. 863–872.

[76] T. Le and Y. Duan, "Pointgrid: A deep network for 3d shape understanding," in *2018 IEEE/CVF CVPR*, 2018, pp. 9204–9214.

[77] H. Shi, G. Lin, H. Wang, T.-Y. Hung, and Z. Wang, "Spsequencenet: Semantic segmentation network on 4d point clouds," in *2020 IEEE/CVF CVPR*, 2020, pp. 4573–4582.

[78] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "Birdnet: A 3d object detection framework from lidar information," in *ITSC 2018*, 2018, pp. 3517–3523.

[79] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, "Polarnet: An improved grid representation for online lidar point clouds semantic segmentation," in *2020 IEEE/CVF CVPR*, 2020, pp. 9598–9607.

[80] E. E. Aksoy, S. Baci, and S. Cavdar, "Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving," in *2020 IEEE IV*, 2020, pp. 926–932.

[81] A. Boulch, J. Guerry, B. L. Saux, and N. Audebert, "Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks," *Comput. Graph.*, vol. 71, pp. 189–198, 2017.

[82] C. Ye, H. Pan, X. Yu, and H. Gao, "A spatially enhanced network with camera-lidar fusion for 3d semantic segmentation," *Neurocomputing*, vol. 484, pp. 59–66, 2022.

[83] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," in *2018 IEEE ICRA*. IEEE, 2018, pp. 1887–1893.

[84] Z. Chen, J. Zhang, and D. Tao, "Progressive lidar adaptation for road detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 693–702, 2019.

[85] L. Xiao, R. Wang, B. Dai, Y. Fang, D. Liu, and T. Wu, "Hybrid conditional random field based camera-lidar fusion for road detection," *Information Sciences*, vol. 432, pp. 543–558, 2018.

[86] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "Lidar–camera fusion for road detection using fully convolutional neural networks," *Robotics and Autonomous Systems*, vol. 111, pp. 125–131, 2019.

[87] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Multimodal vehicle detection: fusing 3d-lidar and color camera data," *Pattern Recognition Letters*, vol. 115, pp. 20–29, 2018.

[88] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021.

[89] G. Carannante, D. Dera, N. C. Bouaynaya, R. Ghulam, and H. M. Fathallah-Shaykh, "Trustworthy medical segmentation with uncertainty estimation," *arXiv preprint arXiv:2111.05978*, 2021.

[90] R. Alizadehsani, D. Sharifrazi, N. H. Izadi, J. H. Joloudari, A. Shoeibi, J. M. Gorriz, S. Hussain, J. E. Arco, Z. A. Sani, F. Khozeimeh *et al.*, "Uncertainty-aware semi-supervised method using large unlabeled and limited labeled covid-19 data," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 3s, pp. 1–24, 2021.

[91] A. A. Munia, I. Hossain, S. M. Jalali, P. Tabarisaadi, A. Rahman, and S. Nahavandi, "Uncertainty-aware deep learning for segmenting ultrasound images of breast tumours," in *2023 IEEE SMC*, 2023, pp. 4228–4235.

[92] J. Postels, F. Ferroni, H. Coskun, N. Navab, and F. Tombari, "Sampling-free epistemic uncertainty estimation using approximated variance propagation," in *2019 IEEE/CVF ICCV*, 2019, pp. 2931–2940.

[93] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," *arXiv preprint arXiv:1301.3572*, 2013.

[94] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," in *ACCV*. Springer, 2016, pp. 213–228.

[95] L. Ma, J. Stückler, C. Kerl, and D. Cremers, "Multi-view deep learning for consistent semantic mapping with rgb-d cameras," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 598–605.

[96] S.-W. Hung, S.-Y. Lo, and H.-M. Hang, "Incorporating luminance, depth and color information by a fusion-based network for semantic segmentation," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2374–2378.

[97] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE TITS*, vol. 22, no. 3, pp. 1341–1360, 2020.

[98] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin, "Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling," in *ECCV*. Springer, 2016, pp. 541–557.

[99] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *2017 IEEE ICRA*. IEEE, 2017, pp. 4644–4651.

[100] Y. Zhang, D. Sidibé, O. Morel, and F. Mériaudeau, "Deep multimodal fusion for semantic image segmentation: A survey," *Image and Vision Computing*, vol. 105, p. 104042, 2021.

[101] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *2015 IEEE CVPR*, 2015, pp. 3061–3070.

[102] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[103] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *ITSC 2013*. IEEE, 2013, pp. 1693–1700.

[104] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss, "Towards 3d lidar-based semantic scene understanding of 3d point cloud sequences: The semantickitti dataset," *The International Journal of Robotics Research*, vol. 40, no. 8-9, pp. 959–967, 2021.

[105] N. Gählert, N. Jourdan, M. Cordts, U. Franke, and J. Denzler, "Cityscapes 3d: Dataset and benchmark for 9 dof vehicle detection," *arXiv preprint arXiv:2006.07864*, 2020.

[106] F. Flohr, D. Gavrila *et al.*, "Pedcut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues." in *BMVC*, 2013.

[107] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive bayesian filters: A comparative study," in *German Conference on Pattern Recognition*. Springer, 2013, pp. 174–183.

[108] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *ECCV*. Springer, 2014, pp. 618–633.

[109] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrila, "A new benchmark for vision-based cyclist detection," in *2016 IEEE IV*. IEEE, 2016, pp. 1028–1033.

[110] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *2020 IEEE/CVF CVPR*, 2020, pp. 2446–2454.

[111] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *2021 IEEE/CVF ICCV*, 2021, pp. 9710–9719.

[112] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," *arXiv preprint arXiv:2106.11810*, 2021.

[113] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF CVPR*, 2020, pp. 11 621–11 631.

[114] W. K. Fong, R. Mohan, J. V. Hurtado, L. Zhou, H. Caesar, O. Beijbom, and A. Valada, "Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3795–3802, 2022.

[115] T. Deruyttere, S. Vandenhende, D. Grujicic, L. Van Gool, and M.-F. Moens, "Talk2car: Taking control of your self-driving car," *arXiv preprint arXiv:1909.10838*, 2019.

[116] Z. Wang, B. Liu, S. Schulter, and M. Chandraker, "A parametric top-view representation of complex road scenes," in *2019 IEEE/CVF CVPR*, 2019, pp. 10 325–10 333.

TABLE IV: Summary of some popular self-driving datasets

Dataset	Sensor Used	Recording description	Provided Information	Perception Tasks
KITTI (2012) [34], [101], [102], [103]	Grayscale and color cameras, Velodyne 64 LIDAR, Inertial Navigation System (GPS/IMU), Varifocal lenses	RN: Karlsruhe, RT: Urban, rural, highways, WC: Sunny, T: Daylight	Raw and processed grayscale stereo sequences presented in png format, raw and processed color stereo sequences presented in png format, 3D Velodyne point clouds as binary float matrix, 3D GPS/IMU data as text file,calibration data as text file, 3D object tracklet labels as xml file.	Stereo, flow, sceneflow, odometry, multi-object tracking and segmentation, road segmentation, pixel-level and instance-level semantic segmentation.
Semantic-KITTI (2019) [27], [104]	Based on the odometry task of KITTI Vision Benchmark	Same as KITTI	3D point cloud as binary format for semantic segmentation, panoptic segmentation and semantic scene completion, also gives labels for each point in binary format.	Semantic segmentation, panoptic segmentation, 4D panoptic segmentation, moving object segmentation, semantic scene completion,
Cityscapes (2016)) [28], [20], [105]	Stereo vision, color cameras, GPS	RA: 50 cities, RT: Urban, WC: Good/ medium, T: Daylight	5000 annotated images with fine annotations, 20000 annotated images with coarse annotations, benchmark suites, evaluation server.	Pixel-level semantic labeling, instance-level semantic labeling, panoptic semantic labeling, 3D vehicle detection
Daimler Pedestrian (2006)) [106], [107], [108], [109]	Monocular and grayscale camera; stereo vision	RN: Beijing and others, RT: Urban, WC: N/A, T: N/A	raw / processed data (training / testing / validation sets); annotation;	Pedestrian segmentation, stereo-based pedestrian detection, pedestrian path prediction, occluded pedestrian classification, cyclist detection
Waymo Open Dataset (2019) [110], [111]	Mid-range LIDAR, short-range LIDAR, cameras (front and sides)	RN: 6 states of USA, RT: Downtown, suburban, WC: Regular and diverse, T: Daylight and nighttime	Motion dataset: Provides shared TFRecord format files containing protocol buffer data. Perception dataset: Contains independently-generated labels for LIDAR and camera data.	3D semantic segmentation, motion prediction, occupancy and flow prediction, 3D detection and tracking, real-time 3D detection, 2D detection and tracking, real-time 2D detection.
nuScenes (2019) [112], [113], [114], [115], [116]	LIDAR, RADAR, camera, IMU, GPS	RN: Boston and Singapore, RT: Urban, WC: Regular and diverse, T: Daylight and nighttime	All annotations, meta data: calibration, maps, vehicle coordinates etc. Ground truth labels for 23 object classes, 3D bounding box and attributes for each frame	Detection, tracking, prediction, LIDAR segmentation, panoptic segmentation.